DETEKSI KOMUNITAS OVERLAP MENGGUNAKAN LOCAL GREEDY EXTENDED DYNAMIC OVERLAPPING COMMUNITY DETECTION (GLOD) PADA PROTEIN KANKER PAYUDARA

TUGAS AKHIR

Disusun Oleh :
DEA REIGINA
123220020

PROGRAM STUDI INFORMATIKA
JURUSAN INFORMATIKA
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
YOGYAKARTA

2026

DETEKSI KOMUNITAS OVERLAP MENGGUNAKAN LOCAL GREEDY EXTENDED DYNAMIC OVERLAPPING COMMUNITY DETECTION (GLOD) PADA PROTEIN KANKER PAYUDARA

TUGAS AKHIR

Tugas Akhir ini sebagai salah satu syarat untuk memperoleh gelar sarjana Informatika
Universitas Pembangunan Nasional "Veteran" Yogyakarta

Disusun Oleh :

DEA REIGINA

123220020

PROGRAM STUDI INFORMATIKA

JURUSAN INFORMATIKA

FAKULTAS TEKNIK INDUSTRI

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"

YOGYAKARTA

2026

HALAMAN PENGESAHAN PEMBIMBING

DETEKSI KOMUNITAS OVERLAP MENGGUNAKAN LOCAL GREEDY EXTENDED DYNAMIC OVERLAPPING COMMUNITY DETECTION (GLOD) PADA PROTEIN KANKER PAYUDARA

Disusun Oleh:
Dea Reigina
123220020

Telah diuji dan dinyatakan lulus oleh pembimbing

pada tanggal: 17 April 2026

Menyetujui,

Pembimbing

Dr. Heru Cahya Rustamaji, S.Si., M.T.
NIDN. 0514067101

Mengetahui,

Koordinator Program Studi

Desyanto Bocdi Prasetyo, S.T., M.T.
NIDN. 0505127501

HALAMAN PENGESAHAN PENGUJI

DETEKSI KOMUNITAS OVERLAP MENGGUNAKAN LOCAL GREEDY EXTENDED DYNAMIC OVERLAPPING COMMUNITY DETECTION (GLOD) PADA PROTEIN KANKER PAYUDARA

Disusun Oleh:
Dea Reigina
123220020

Telah diuji dan dinyatakan lulus oleh penguji
pada tanggal: 17 April 2026

Menyetujui,

Penguji 1

Dr. Heru Cahya Rustamaji., S.Si., M.T.
NIDN. 0514067101

Penguji 2

Rifki Indra Perwira, S.Kom., M.Eng
NIDN. 0508078301

Penguji 3

Wilis Kaswidjanti, S.Si., M.Kom.
NIDN. 0513047601

Penguji 4

Dr. Awang Hendrianto Pratomo, S.T., M.T.
NIDN. 0025077701

SURAT PERNYATAAN
KARYA ASLI TUGAS AKHIR

Sebagai mahasiswa Program Studi Informatikai Fakultas Teknik Industri Universitas Pembangunan Nasional "Veteran" Yogyakarta, yang bertanda tangan di bawah ini, saya:

Nama : Dea Reigina
NIM : 123220020

Menyatakan bahwa karya ilmiah saya yang berjudul:
Deteksi Komunitas Overlap Menggunakan Local Greedy Extended Dynamic Overlapping Community Detection (Glod) pada Protein Kanker Payudara

Merupakan karya asli saya dan belum pernah dipublikasikan dimanapun. Apabila di kemudian hari, karya saya disinyalir bukan merupakan karya asli saya, maka saya bersedia menerima konsekuensi apa pun yang diberikan Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional "Veteran" Yogyakarta kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta
Pada Tanggal : 17 April 2026

Yang Menyatakan

Dea Reigina
123220020

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini:

Nama : Dea Reigina
NIM : 123220020
Prodi : Informatika

Dengan ini saya menyatakan bahwa judul Tugas Akhir
Deteksi Komunitas Overlap Menggunakan Local Greedy Extended Dynamic Overlapping Community Detection (Glod) pada Protein Kanker Payudara

adalah hasil kerja saya dan benar bebas dari plagiasi kecuali cuplikan serta ringkasan yang terdapat di dalamnya telah saya jelaskan sumbernya (Sitasi) dengan jelas. Apabila pernyataan ini terbuktu tidak benar maka saya bersedia menerima sanksi sesuai peraturan Mendiknas RI No 17 Tahun 2010 dan Peraturan Perundang-undangan yang berlaku.

Demikian surat pernyataan ini saya buat dengan penuh tanggung jawab.

Dibuat di : Yogyakarta
Pada Tanggal : 17 April 2026

Yang Menyatakan

Dea Reigina
123220020

ABSTRAK

Kanker payudara merupakan penyakit dengan kompleksitas molekuler tinggi karena melibatkan interaksi banyak protein dalam berbagai jalur biologis. Analisis jaringan interaksi protein memungkinkan pemahaman hubungan antar protein secara sistematis. Namun, jaringan biologis bersifat kompleks dan memungkinkan satu protein berada pada lebih dari satu kelompok fungsi, sehingga diperlukan metode deteksi komunitas yang mampu mengidentifikasi struktur overlapping. Penelitian ini bertujuan untuk menerapkan algoritma Local Greedy Extended Dynamic Overlapping Community Detection (GLOD) pada jaringan interaksi protein kanker payudara serta mengevaluasi kualitas komunitas yang terbentuk secara struktural dan fungsional.

Data protein diperoleh melalui API UniProt menggunakan kata kunci breast cancer dan menghasilkan 2.690 entri. Setelah seleksi atribut Gene Symbol dan proses deduplikasi, diperoleh 2.010 protein unik. Jaringan interaksi dibangun menggunakan STRING DB dengan confidence score 0,900 pada organisme Homo sapiens, menghasilkan 1.823 simpul dan 2.646 sisi. Setelah ekstraksi giant component, jaringan akhir terdiri dari 864 simpul dan 2.567 sisi. Algoritma GLOD diterapkan dengan parameter alpha 0,75 dan merging threshold 0,2 untuk mendeteksi komunitas overlapping. Sistem dikembangkan menggunakan Python dengan framework Django dan diuji menggunakan metode whitebox testing untuk memastikan setiap modul berjalan sesuai logika perancangan.

Hasil penelitian menunjukkan bahwa algoritma GLOD berhasil mengidentifikasi 16 komunitas overlapping dengan nilai rata-rata Normalized Node Cut (NNC) sebesar 0,3691. Berdasarkan distribusi kuartil, 4 komunitas termasuk kategori baik, 8 kategori cukup, dan 4 kategori rendah. Secara fungsional, enrichment analysis menunjukkan pengayaan signifikan pada beberapa jalur utama, antara lain Jalur dalam kanker pada Komunitas 1 (54,76%, -29,03), Kanker kolorektal pada Komunitas 2 (66,67%, -26,57), Proteoglikan dalam kanker pada Komunitas 6 (64,29%, -16,33), Proses siklus sel mitotik pada Komunitas 3 (66,67%, -28,23), serta Remodeling kromatin pada Komunitas 10 dan 14 (89,47%, -25,46). Nilai Log10(P) yang negatif menunjukkan signifikansi statistik yang kuat. Seluruh pengujian sistem menunjukkan hasil berhasil tanpa kesalahan fungsional. Penelitian ini menunjukkan bahwa pendekatan deteksi komunitas overlapping berbasis GLOD mampu menghasilkan struktur modular yang konsisten secara topologi dan relevan secara fungsional pada jaringan interaksi protein kanker payudara.

Kata Kunci: kanker payudara, GLOD, deteksi komunitas overlap, jaringan interaksi protein, enrichment analysis

ABSTRACT

Breast cancer is a complex disease that involves many proteins interacting in different biological pathways. Studying protein interaction networks helps to understand how these proteins are connected and work together. However, biological networks are complex, and one protein can belong to more than one functional group. Therefore, an overlapping community detection method is needed. This research aims to apply the Local Greedy Extended Dynamic Overlapping Community Detection (GLOD) algorithm to a breast cancer protein interaction network and to evaluate the structural and functional quality of the detected communities.

Protein data were collected from the UniProt API using the keyword breast cancer, resulting in 2,690 protein entries. After selecting the Gene Symbol attribute and removing duplicate data, 2,010 unique proteins were obtained. The protein interaction network was built using STRING DB with a confidence score of 0.900 for Homo sapiens, producing 1,823 nodes and 2,646 edges. After extracting the giant component, the final network contained 864 nodes and 2,567 edges. The GLOD algorithm was applied using alpha = 0.75 and merging threshold = 0.2 to detect overlapping communities. The system was developed using Python and the Django framework. System testing was conducted using the whitebox method to ensure that all modules worked according to the design.

The results show that the GLOD algorithm successfully identified 16 overlapping communities with an average Normalized Node Cut (NNC) value of 0.3691. Based on quartile distribution, 4 communities were categorized as good, 8 as moderate, and 4 as low quality. Functional enrichment analysis showed significant enrichment in several main pathways, such as Pathways in cancer in Community 1 (54.76%, -29.03), Colorectal cancer in Community 2 (66.67%, -26.57), Proteoglycans in cancer in Community 6 (64.29%, -16.33), Mitotic cell cycle process in Community 3 (66.67%, -28.23), and Chromatin remodeling in Communities 10 and 14 (89.47%, -25.46). The negative Log10(P) values indicate strong statistical significance. All system testing results were successful. This study shows that the GLOD-based overlapping community detection approach can identify modular and functionally consistent structures in a breast cancer protein interaction network.

Keywords: *breast cancer, GLOD, overlapping community detection, protein interaction network, enrichment analysis*

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat, karunia, serta penyertaan-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul *"Deteksi Komunitas Overlap Menggunakan Local Greedy Extended Dynamic Overlapping Community Detection (GLOD) pada Protein Kanker Payudara"* sebagai salah satu syarat untuk menyelesaikan program sarjana (S1) Program Studi Informatika, Jurusan Informatika, Fakultas Teknik Industri, Universitas Pembangunan Nasional "Veteran" Yogyakarta. Penyusunan Tugas Akhir ini tidak terlepas dari dukungan, doa, serta bimbingan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa yang senantiasa memberikan kesehatan, kekuatan, kemudahan, dan kelancaran kepada penulis selama proses pengerjaan Tugas Akhir ini.
2. Orang tua tercinta, Bapak Fersawan dan Ibu Wanti, yang selalu memberikan doa, dukungan, kasih sayang, serta semangat tanpa henti kepada penulis.
3. Adik-adik penulis, Stevany, Naira Az Zahra, dan Jihan Talitha, yang selalu memberikan dukungan dan menjadi penyemangat bagi penulis.
4. Bapak Dr. Heru Cahya Rustamaji, S.Si., M.T. selaku dosen pembimbing yang telah memberikan arahan, bimbingan, motivasi, serta meluangkan waktu dengan penuh kesabaran selama proses penyusunan Tugas Akhir ini.
5. Bapak Rifki Indra Perwira, S.Kom., M.Eng, Ibu Wilis Kaswidjanti, S.Si., M.Kom., dan Bapak Dr. Awang Hendrianto Pratomo, S.T., M.T. selaku dosen penguji yang telah memberikan kritik, saran, dan masukan yang membangun demi penyempurnaan Tugas Akhir ini.
6. Diri penulis sendiri, Dea Reigina, yang telah berjuang, bertahan, dan berusaha semaksimal mungkin dalam menyelesaikan Tugas Akhir ini di tengah berbagai tantangan yang dihadapi.
7. Naufal Rafid, yang selalu memberikan dukungan, semangat, serta menemani penulis dalam berbagai kondisi selama proses pengerjaan Tugas Akhir ini.
8. Mareen, Hikmah, Aulia, Audy, Wawa, dan Nadya, yang senantiasa memberikan dukungan dan semangat kepada penulis meskipun terpisah oleh jarak.
9. Teman-teman RK HIMATIF, yaitu Mbak Nisa, Naufal, Wijdan, Aira, Hafiz, Rifki, Amanda, dan Alma, atas kebersamaan, dukungan, serta pengalaman berharga yang telah diberikan.
10. Teman-teman penulis, yaitu Veyza, Salma, Aqsha, Lyta, Rani, Gita, Intan, Wijdan, dan Adit, yang telah membersamai penulis dalam proses pengerjaan Tugas Akhir ini.
11. Inez, Cece, Rahel, Ayas, Shinta, Syifa, serta seluruh teman-teman Informatika angkatan 2022, yang telah memberikan warna, cerita, dan kenangan selama masa perkuliahan penulis.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Kanker merupakan penyakit yang terjadi akibat gangguan regulasi sistem molekuler di dalam sel. Sel normal memiliki mekanisme pengendalian pertumbuhan, perbaikan DNA, serta kematian sel terprogram (apoptosis). Namun, ketika terjadi mutasi atau perubahan fungsi pada gen dan protein tertentu, sel dapat kehilangan kemampuan mengontrol pertumbuhannya dan berkembang menjadi sel kanker. Konsep ini dijelaskan dalam teori *hallmarks of cancer*, yang menyebutkan bahwa kanker berkembang karena adanya penumpukan gangguan pada sistem biologis dasar di dalam sel (Hanahan, 2022). Artinya, kanker tidak hanya dipandang sebagai pertumbuhan sel yang tidak normal, tetapi sebagai akibat dari kerusakan dan ketidakteraturan pada sistem molekuler yang kompleks di dalam tubuh.

Pada tingkat molekuler, protein berperan sebagai pelaksana utama hampir seluruh proses biologis dalam sel. Protein mengatur siklus sel, memperbaiki kerusakan DNA, mengontrol apoptosis, serta menjalankan berbagai jalur pensinyalan. Mutasi pada protein tertentu dapat mengganggu keseimbangan sistem ini. Sebagai contoh, mutasi pada TP53 merupakan salah satu mutasi yang paling sering ditemukan pada berbagai jenis kanker dan berkaitan dengan prognosis yang buruk (Chen et al., 2022). Selain itu, mutasi pada BRCA1 dapat mengganggu mekanisme perbaikan DNA dan secara signifikan meningkatkan risiko kanker payudara (Fu et al., 2022). Temuan tersebut menunjukkan bahwa perubahan pada protein memiliki peran krusial dalam perkembangan kanker. Namun, kanker tidak terjadi hanya karena gangguan pada satu protein saja. Protein di dalam sel bekerja secara terkoordinasi membentuk suatu sistem yang saling terhubung. Oleh karena itu, pemahaman kanker memerlukan pendekatan pada tingkat sistem biologis, bukan hanya pada gen atau protein tunggal.

Kanker payudara merupakan salah satu masalah kesehatan terbesar bagi perempuan, baik di tingkat global maupun nasional. Data GLOBOCAN menunjukkan bahwa penyakit ini terus menempati posisi teratas dalam jumlah kasus baru dan menjadi salah satu penyebab utama kematian pada wanita (Sung *et al.*, 2021). Kondisi tersebut menegaskan bahwa kanker payudara tidak hanya menjadi persoalan klinis, tetapi juga perlu dipahami melalui mekanisme molekuler yang mendasarinya. Secara molekuler, kanker payudara berkembang melalui berbagai proses yang saling berkaitan, di mana interaksi antarprotein menjadi komponen penting. Protein tidak hanya membentuk kompleks fisik, tetapi juga menjalin hubungan fungsional dalam jalur pensinyalan, regulasi aktivitas protein lain, serta pembentukan dan pemeliharaan struktur sel. Interaksi-interaksi tersebut membentuk jaringan biologis yang kompleks dan berperan besar dalam perkembangan kanker payudara (Szklarczyk *et al.*, 2023).

Interaksi tersebut dikenal sebagai jaringan interaksi protein-protein (PPI), yang merupakan representasi hubungan fisik antara protein-protein di dalam sel (Castellanos-Girouard, Serohijos and Michnick, 2024). Analisis jaringan PPI penting untuk memahami mekanisme penyakit, karena gen atau protein yang berhubungan dengan penyakit cenderung

tidak tersebar secara acak, melainkan membentuk kelompok atau komunitas tertentu dalam jaringan (Izudheen et al., 2020).

Untuk mempelajari jaringan PPI, digunakan pendekatan teori graf, di mana protein direpresentasikan sebagai simpul (*node*) dan interaksi antarprotein sebagai garis (*edge*) (Sri Suharini *et al.*, 2023). Melalui representasi graf ini, kita dapat mempelajari struktur hubungan antarprotein, termasuk kelompok protein yang membentuk komunitas. Komunitas tersebut merupakan sekumpulan protein yang saling terhubung secara padat dan bekerja sama menjalankan fungsi biologis tertentu, seperti jalur metabolisme atau pembentukan kompleks protein (Wang *et al.*, 2021). Dalam konteks kanker, gangguan tidak hanya terjadi pada satu protein, tetapi juga pada struktur komunitas yang mengatur fungsi biologis tertentu. Oleh karena itu, deteksi komunitas dalam jaringan PPI menjadi penting untuk mengidentifikasi modul biologis yang berperan dalam perkembangan kanker.

Salah satu tantangan dalam analisis jaringan PPI adalah adanya *overlapping communities*. Sebuah protein dapat terlibat dalam lebih dari satu fungsi biologis, sehingga dapat menjadi bagian dari lebih dari satu komunitas secara bersamaan (Wang et al., 2021). Kondisi ini membuat proses deteksi komunitas menjadi lebih kompleks. Berbagai metode telah dikembangkan untuk menangani komunitas yang saling tumpang tindih, tetapi masih memiliki sejumlah keterbatasan. Beberapa metode cenderung mengabaikan simpul pada area tumpang tindih sehingga hasilnya mendekati pembagian komunitas non-overlapping (Vieira, Xavier and Evsukoff, 2020). Metode berbasis *seed expansion* juga sering kali tidak stabil karena bergantung pada pemilihan benih awal yang acak dan sensitif terhadap parameter (Zhao *et al.*, 2023). Selain itu, banyak metode menghasilkan komunitas yang saling tumpang tindih secara berlebihan, sehingga muncul komunitas-komunitas yang terlalu mirip. Hal ini menimbulkan redundansi atau pengulangan dan membuat interpretasi biologis menjadi kurang jelas. Kekurangan-kekurangan tersebut menunjukkan adanya celah dalam penerapan metode deteksi komunitas, khususnya pada jaringan PPI yang besar dan kompleks (Song *et al.*, 2023).

Salah satu pendekatan yang dikembangkan untuk mengatasi permasalahan tersebut adalah algoritma GLOD (*Local Greedy Expansion Method for Overlapping Community Detection*). Algoritma ini dirancang untuk meningkatkan stabilitas pembentukan komunitas sekaligus mengurangi tumpang tindih yang berlebihan melalui tiga tahapan utama. Tahap pertama dalam GLOD adalah pembentukan benih komunitas (seeding phase), yang dilakukan melalui proses coarsening dengan menggabungkan simpul inti dan tetangga-tetangganya yang memiliki tingkat kesamaan tinggi berdasarkan common neighbor similarity. Pendekatan ini menghasilkan rough seeds yang terdiri dari beberapa simpul dengan keterkaitan lokal yang kuat, sehingga pemilihan benih menjadi lebih stabil dan representatif dibandingkan penggunaan satu simpul tunggal. Tahap kedua adalah perluasan komunitas (expansion phase), di mana komunitas dikembangkan secara iteratif dengan menambahkan simpul tetangga yang paling sesuai dengan memanfaatkan lebih dari satu ukuran (*fitness function*) untuk menilai kualitas komunitas. Ketiga, setelah semua komunitas ditemukan, ada tahap penggabungan (*merge phase*) yang menggunakan koefisien Jaccard yang dimodifikasi untuk mengurangi komunitas yang terlalu mirip (Song *et al.*, 2023).

Dengan karakteristik tersebut, algoritma GLOD dinilai relevan untuk diterapkan pada jaringan interaksi protein yang kompleks, termasuk dalam studi kanker payudara.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini:

1. Bagaiman mendeteksi komunitas overlap pada jaringan interaksi protein kanker payudara dengan algoritma *Local Greedy Extended Dynamic Overlapping Community Detection* (GLOD).
2. Bagaimana mengevaluasi kualitas struktur komunitas overlap yang dihasilkan oleh algoritma GLOD pada jaringan PPI kanker payudara berdasarkan metrik *Normalized Node Cut* dan pendekatan Enrichment Analysis

1.3 Batasan Masalah

Untuk menjaga ruang lingkup penelitian tetap terfokus, maka batasan masalah dalam penelitian ini ditetapkan sebagai berikut:

1. Dataset yang digunakan merupakan data sekunder yang diperoleh dari basis data publik *UniProt* dan *STRINGdb* dengan parameter required score highest confidence (0.900), sehingga penelitian ini tidak melakukan eksperimen laboratorium untuk validasi data secara *in vitro* maupun *in vivo*.
2. Objek penelitian dibatasi pada jaringan interaksi protein-protein (PPI) kanker payudara dan tidak mencakup jenis kanker lainnya.
3. Model jaringan protein yang digunakan dalam penelitian ini direpresentasikan dalam bentuk graf tidak berarah (*undirected*) dan tidak berbobot (*unweighted*).
4. Deteksi komunitas pada jaringan protein difokuskan pada komunitas tumpang tindih (*overlapping communities*) dengan menggunakan pendekatan *Local Community Detection* (LCD) berbasis algoritma GLOD (*Greedy Local Expansion for Overlapping Community Detection*).
5. Perbandingan hasil deteksi komunitas hanya difokuskan pada kualitas komunitas yang terbentuk, menggunakan metrik evaluasi normalized node cut dan analisis fungsional (*enrichment analysis*), tanpa membahas performa teknis lain seperti kompleksitas komputasi atau konsumsi memori.
6. Analisis fungsional (*enrichment analysis*) dilakukan menggunakan situs *Metascape* dengan cakupan *Gene Ontology* (GO) pada kategori *Biological Process*, *Molecular Function*, dan *Cellular Component*, serta analisis jalur *KEGG Pathway*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini:

1. Mengimplementasikan pendekatan Local Community Detection (LCD) dengan algoritma *algoritma Greedy Local Expansion for Overlapping Community Detection* (GLOD) untuk mendeteksi komunitas yang tumpang tindih (*overlap*) pada jaringan interaksi protein kanker payudara.
2. Mengevaluasi kualitas deteksi komunitas yang dihasilkan oleh algoritma GLOD pada jaringan PPI kanker payudara menggunakan metrik Normalized Node Cut serta

menginterpretasikan relevansi biologis komunitas tersebut melalui pendekatan enrichment analysis.

1.5 Manfaat Penelitian

Adapula manfaat dari penelitian ini:

1. implementasi algoritma deteksi komunitas lokal yang mampu menghasilkan struktur komunitas lebih stabil dan meminimalisir redundansi hasil pada jaringan interaksi protein kanker payudara.
2. Menyediakan referensi teknis berbasis algoritma GLOD untuk analisis jaringan PPI serta memberikan pemahaman mengenai peran biologis komunitas protein yang terdeteksi guna mendukung identifikasi biomarker atau target terapeutik potensial pada kanker payudara.

1.6 Tahapan Penelitian

Dalam bagian ini disampaikan tentang cara-cara yang digunakan dalam melakukan penelitian. Metode penelitian berisi :

1. Business Understanding

   Tahap *business understanding* bertujuan untuk merumuskan permasalahan dalam penelitian, yaitu keterbatasan metode deteksi komunitas konvensional dalam mengidentifikasi komunitas protein yang bersifat tumpang tindih. Oleh karena itu, penelitian ini bertujuan untuk menerapkan pendekatan *Local Community Detection* (LCD) menggunakan algoritma GLOD (*Greedy Local Expansion for Overlapping Community Detection*) guna mengidentifikasi struktur komunitas protein yang lebih representatif terhadap kompleksitas biologis kanker payudara.
2. Data Understanding

   Pada tahap *data understanding*, dilakukan pengumpulan dan pemahaman terhadap data yang digunakan dalam penelitian. Data yang digunakan merupakan data sekunder berupa informasi gen dan protein yang berasosiasi dengan kanker payudara. Data diperoleh dari basis data publik UniProt melalui layanan API, yang menyediakan informasi terkurasi mengenai protein dan fungsinya. Data yang dikumpulkan mencakup daftar protein yang telah dilaporkan memiliki keterkaitan dengan kanker payudara, yang selanjutnya digunakan sebagai dasar dalam pembentukan jaringan interaksi protein.
3. Data Preparation

   Tahap *data preparation* bertujuan untuk memastikan kualitas dan kesiapan data sebelum dilakukan analisis lebih lanjut. Proses yang dilakukan pada tahap ini meliputi:

   1. Seleksi Data (*Data Selection*): Memilih kolom data yang relevan, yaitu kolom yang berisi nama atau simbol protein.
   2. Penghapusan Duplikasi (*Data Deduplication*): Menghapus entri gen yang berulang untuk memastikan setiap gen bersifat unik.
   3. *Data mapping*: menyesuaikan simbol gen dengan identitas protein menggunakan *UniProt accession*.

   Setelah diperoleh daftar protein final, data tersebut dimasukkan ke dalam basis data STRING DB untuk membangun jaringan interaksi protein-protein (PPI). Jaringan PPI

yang terbentuk kemudian disaring dengan memilih giant component agar analisis difokuskan pada komponen jaringan terbesar dan paling terhubung. Selanjutnya, jaringan yang telah dipilih digunakan pada tahap pemodelan.

4. Modeling

   Pada tahap *modeling*, jaringan PPI dianalisis untuk mendeteksi komunitas protein yang saling tumpang tindih. Metode yang digunakan adalah *Local Community Detection* dengan algoritma GLOD, yang terdiri dari tiga fase utama, yaitu *seeding phase*, *expansion phase*, dan *merging phase*. Proses ini memungkinkan satu protein menjadi anggota lebih dari satu komunitas, sehingga struktur komunitas yang dihasilkan diharapkan mampu merepresentasikan keterkaitan biologis yang kompleks pada kanker payudara.

5. Evaluation

   Tahap *evaluation* dilakukan untuk menilai kualitas komunitas yang dihasilkan, baik secara struktural maupun biologis. Evaluasi struktural dilakukan menggunakan metrik *Normalized Node Cut* (NNC) untuk mengukur kualitas pemisahan komunitas dalam jaringan. Selanjutnya, evaluasi biologis dilakukan melalui *Enrichment Analysis*, dengan memasukkan daftar protein dari setiap komunitas ke platform bioinformatika yaitu Metascape. Analisis ini bertujuan untuk mengidentifikasi jalur biologis, fungsi molekuler, dan proses biologis yang signifikan, sehingga komunitas yang terbentuk dapat diinterpretasikan relevansinya terhadap mekanisme kanker payudara.

6. Deployment

   Tahap terakhir adalah *deployment*, yaitu penyajian dan visualisasi hasil penelitian dalam bentuk sistem interaktif. Pada tahap ini dikembangkan antarmuka pengguna (*User Interface*) untuk memudahkan pengguna dalam mengunggah data, menjalankan algoritma deteksi komunitas, serta melihat hasil analisis secara visual. Sistem ini dibangun menggunakan framework Django dan Bootstrap.

BAB II
TINJAUAN LITERATUR

2.1 Kanker Payudara

Kanker payudara adalah salah satu jenis kanker yang menjadi masalah kesehatan global, dan memiliki angka kematian yang tinggi di seluruh dunia (Sung et al., 2021). Menurut estimasi GLOBOCAN 2020, kanker payudara telah melampaui kanker paru-paru sebagai jenis kanker yang paling sering didiagnosis secara global, dengan perkiraan 2,3 juta kasus baru pada tahun 2020, yang menyumbang 11,7% dari seluruh kasus kanker (Sung et al., 2021). Di Indonesia, prevalensi kanker payudara juga sangat tinggi, menempati posisi pertama sebagai kanker dengan insiden tertinggi pada wanita (Ferlay *et al.*, 2021).

Penyakit ini ditandai oleh pertumbuhan sel-sel payudara yang tidak terkendali, seringkali membentuk tumor ganas yang dapat menyebar ke bagian tubuh lain jika tidak ditangani (Zhang and Liu, 2025a). Kanker payudara memiliki tingkat keragaman yang tinggi, baik antar pasien maupun di dalam satu tumor itu sendiri. Perbedaan karakteristik biologis ini membuat proses diagnosis dan penanganannya tidak bisa disamaratakan (Zhang and Liu, 2025). Oleh karena itu, deteksi sejak dini menjadi hal yang sangat penting. Pada tahap awal, peluang hidup pasien masih relatif besar, tetapi ketika kanker telah menyebar ke organ lain, risikonya meningkat secara signifikan dan dapat berujung fatal (Zhang and Liu, 2025).

Pada tingkat molekuler, kanker payudara dipicu oleh berbagai mutasi genetik dan perubahan pada protein yang berperan penting dalam mengendalikan aktivitas sel. Protein-protein seperti BRCA1 dan BRCA2 adalah contoh gen penekan tumor yang jika bermutasi dapat secara signifikan meningkatkan risiko kanker payudara (Betz *et al.*, 2025). Mutasi pada BRCA2, misalnya, bertanggung jawab atas lebih dari 40% kasus kanker payudara herediter dan berperan dalam transkripsi, perbaikan DNA, serta rekombinasi (Zhang and Liu, 2025). Selain itu, protein lain seperti TP53, PIK3CA, CDH1, AKT1, dan GATA3 juga sering mengalami mutasi dan dikenal sebagai gen pendorong utama dalam perkembangan kanker payudara (Betz *et al.*, 2025).

Pemahaman terhadap interaksi kompleks antar protein dalam sel kanker payudara sangat penting untuk menjelaskan mekanisme penyakit sekaligus menemukan target terapi baru. Jaringan interaksi protein-protein (PPI) digunakan untuk memetakan hubungan fisik antar protein, dengan protein sebagai simpul dan interaksinya sebagai penghubung (Zhang and Liu, 2025). Melalui analisis jaringan PPI, kanker payudara dapat dikaji secara lebih sistematis (Zhang and Liu, 2025). Pendekatan deteksi komunitas dalam jaringan PPI relevan karena dapat mengidentifikasi kelompok protein yang saling berinteraksi dan berperan dalam fungsi biologis ata u proses penyakit yang sama (Dilmaghani et al., 2022). Kelompok protein tersebut berpotensi menjadi titik penting secara klinis karena berkaitan dengan tekanan mutasi, tingkat kelangsungan hidup, dan respons terhadap terapi, sehingga dapat dimanfaatkan sebagai biomarker untuk diagnosis dan prognosis kanker payudara (Zhang and Liu, 2025)

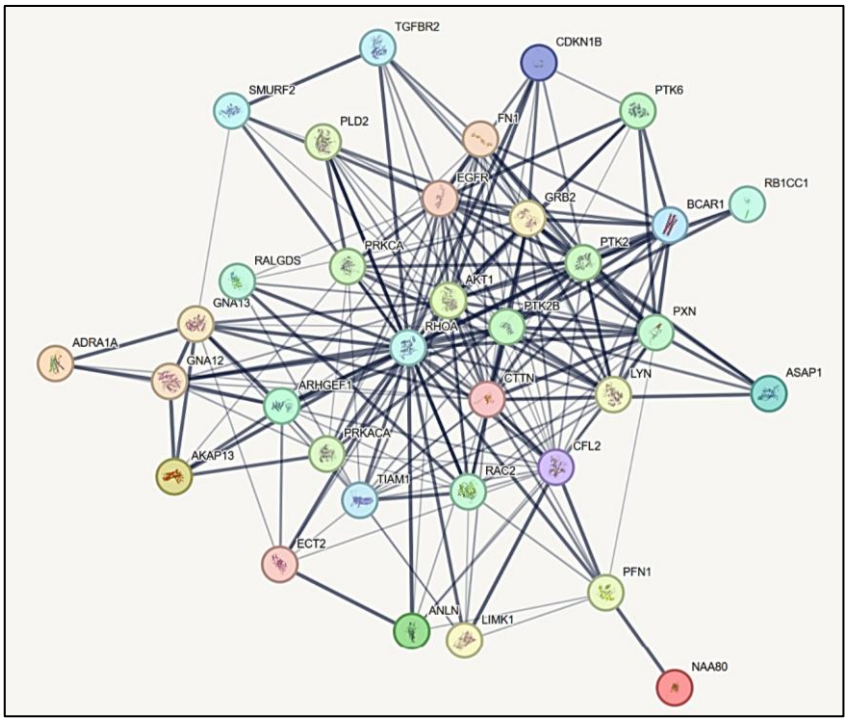
2.2 Protein-protein Interaction

Protein merupakan molekul kunci yang berperan dalam berbagai proses penting di dalam sel, mulai dari menjaga struktur sel hingga mempercepat reaksi biokimia. Dalam menjalankan fungsinya, protein tidak bekerja secara terpisah, melainkan saling berinteraksi dan membentuk jaringan komunikasi yang kompleks dan teratur, sebagaimana ditunjukkan pada Gambar 2.1 (Dilmaghani et al., 2022). Interaksi fisik antar protein ini dikenal sebagai Interaksi Protein-Protein/Protein-Protein Interaction (PPI). Jaringan PPI menyajikan gambaran menyeluruh mengenai hubungan fisik antar protein di dalam sel (Castellanos-Girouard, Serohijos and Michnick, 2024). Analisis jaringan ini berperan penting dalam memahami mekanisme kerja sel normal, sekaligus menjelaskan bagaimana gangguan pada interaksi protein dapat memicu terjadinya berbagai penyakit (Sri Suharini *et al.*, 2023).

Analisis jaringan PPI memegang peran penting dalam biologi sistem karena mampu mengungkap pola interaksi antar protein yang tidak terlihat secara langsung. Melalui pendekatan ini, fungsi protein yang belum diketahui dapat diperkirakan, mekanisme penyakit dapat dipahami, dan peluang penemuan obat baru dapat dibuka (Dilmaghani *et al.*, 2022). Jaringan PPI pada dasarnya menggambarkan komunikasi antar kelompok protein yang saling berinteraksi secara erat, sehingga analisisnya memberikan pemahaman yang lebih dalam mengenai prinsip organisasi sel (Platos, Id and Dra, 2024).

Dalam bidang onkologi, khususnya kanker payudara, analisis jaringan PPI menjadi alat yang efektif untuk mengkaji kompleksitas penyakit. Kanker payudara muncul akibat berbagai gangguan molekuler, termasuk mutasi genetik yang mengubah pola interaksi antar protein (Betz *et al.*, 2025). Dengan memetakan jaringan PPI pada sel kanker payudara, peneliti dapat mengidentifikasi komunitas atau modul protein yang berada di bawah tekanan mutasi dan memiliki pengaruh besar terhadap kelangsungan hidup pasien. Pendekatan ini membantu menjelaskan bagaimana perubahan pada tingkat gen secara bertahap memengaruhi sistem biologis pada tingkat protein (Zhang and Liu, 2025a).

Identifikasi modul protein yang berperan dalam perkembangan kanker payudara dapat memberikan pemahaman baru terkait mekanisme penyakit sekaligus membuka peluang penentuan target terapi. Beberapa penelitian menunjukkan bahwa gen penyebab kanker sering ditemukan pada protein yang terlibat dalam lebih dari satu komunitas. Protein tumpang tindih ini bersifat multifungsi dan berperan sebagai penghubung antar jalur sinyal, sehingga berpotensi menjadi target terapi yang lebih presisi dan efektif (Platos, Id and Dra, 2024).



Gambar 2. 1 Visualisasi jaringan PPI dengan STRING db

Gambar ini memperlihatkan bagaimana protein-protein (*node*) saling terhubung oleh interaksi (*edge*). Beberapa *node* memiliki banyak *edge*, menunjukkan derajat yang tinggi, sementara yang lain memiliki sedikit edge, mencerminkan perbedaan tingkat interaksi antar protein.

2.3 Teori Graf

Mengingat ukuran dan tingkat kompleksitasnya, jaringan interaksi protein umumnya dimodelkan menggunakan teori graf (Platos, Id and Dra, 2024). Dalam model ini, protein direpresentasikan sebagai simpul (node), sedangkan interaksi fisik yang telah terverifikasi antar protein digambarkan sebagai sisi (edge) yang menghubungkan dua simpul (Dilmaghani *et al.*, 2022). Pendekatan graf memungkinkan data biologis yang kompleks disusun ke dalam model matematis yang terstruktur, sehingga dapat dianalisis secara kuantitatif (Zhang and Liu, 2025a).

Secara formal, sebuah jaringan atau graf (G) dapat didefinisikan sebagai pasangan himpunan (V, E), yang dituliskan dalam:

$$G = (V, E) \tag{2.1}$$

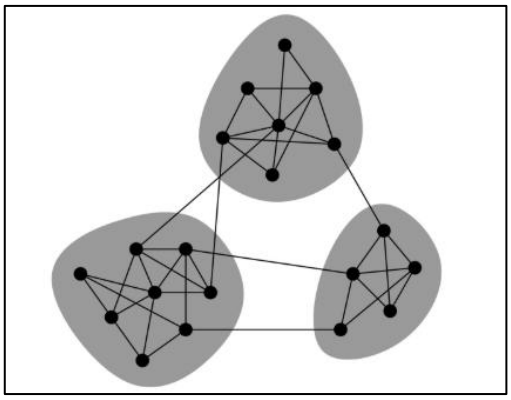
Keterangan:

G = Graf atau jaringan

V = Himpunan simpul (*nodes*)

E = Himpunan sisi (*edges*)

Ilustrasi representasi graf ditunjukkan pada Gambar 2.2, di mana titik merepresentasikan protein dan garis menunjukkan interaksi antar protein. Area yang diarsir menggambarkan kelompok atau komunitas protein yang saling terhubung secara padat, menandakan adanya struktur komunitas dalam jaringan.



Gambar 2. 2 Ilustrasi Graf, komunintas *non-overlapping* (a)

Setelah jaringan PPI direpresentasikan dalam bentuk graf, berbagai metode analisis jaringan dapat diterapkan untuk mengungkap pola yang tersembunyi. Salah satu fokus utama adalah deteksi komunitas, yaitu proses mengidentifikasi kelompok simpul yang memiliki kepadatan koneksi tinggi di dalam kelompok, tetapi relatif sedikit berinteraksi dengan simpul di luar kelompok tersebut (Zhao *et al.*, 2023). Komunitas semacam ini umumnya berkaitan dengan kompleks protein atau modul fungsional, di mana sejumlah protein bekerja bersama untuk menjalankan fungsi biologis tertentu (Platos, Id and Dra, 2024).

Representasi graf menjadi dasar yang penting karena mampu mengubah permasalahan biologis menjadi persoalan komputasi yang dapat dianalisis secara sistematis. Misalnya, proses menemukan keluarga protein baru dapat disederhanakan menjadi tugas mendeteksi subgraf atau komunitas dalam jaringan PPI menggunakan algoritma deteksi komunitas, seperti *Local Greedy Extended Dynamic Overlapping Community Detection* (GLOD) (Platos, Id and Dra, 2024). Dengan demikian, teori graf tidak hanya berfungsi sebagai alat visualisasi, tetapi juga sebagai kerangka kerja analitis yang kuat untuk memahami bagaimana komponen-komponen seluler diatur dan bagaimana gangguan pada jaringan ini dapat menyebabkan penyakit seperti kanker (Zhang and Liu, 2025a).

2.4 Deteksi Komunitas (*Community Detection*)

Deteksi komunitas (*community detection*) merupakan salah satu teknik penting dalam analisis jaringan kompleks yang bertujuan untuk mengidentifikasi kelompok simpul yang memiliki hubungan lebih kuat di dalam kelompoknya dibandingkan dengan simpul di luar kelompok tersebut (Ni *et al.*, 2019). Struktur komunitas ini biasanya muncul secara alami pada berbagai jenis jaringan dunia nyata, seperti jaringan sosial, jaringan informasi, maupun jaringan biologis, sehingga analisis komunitas dapat membantu memahami pola hubungan dan fungsi yang tersembunyi dalam suatu jaringan (Baltsou, Christopoulos and Tsichlas, 2022). Dalam konteks jaringan graf, komunitas umumnya didefinisikan sebagai subgraf yang memiliki kepadatan hubungan internal yang tinggi dan jumlah hubungan yang relatif lebih sedikit dengan bagian jaringan lainnya (Ni *et al.*, 2019). Dengan menemukan struktur komunitas tersebut, peneliti dapat memperoleh pemahaman yang lebih baik mengenai organisasi jaringan, interaksi antar entitas, serta pola penyebaran informasi atau fungsi biologis yang terjadi di dalam jaringan (Baltsou, Christopoulos and Tsichlas, 2022). Oleh karena itu, deteksi komunitas menjadi salah satu topik penelitian yang banyak

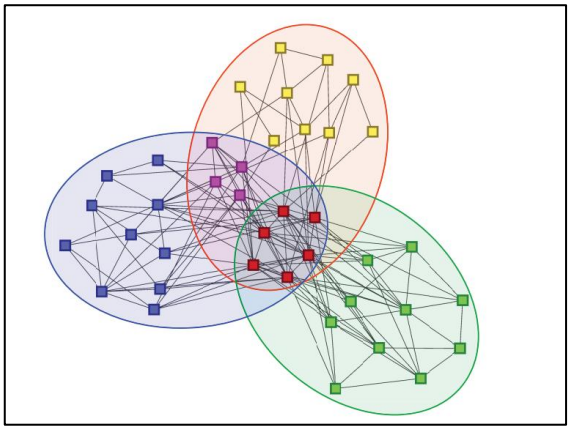
dikembangkan dalam bidang ilmu komputer dan analisis jaringan karena kemampuannya dalam mengungkap struktur modular dan hubungan tersembunyi yang terdapat pada berbagai sistem kompleks (Ni *et al.*, 2019).

2.4.1 Definisi dan Tujuan Deteksi Komunitas

Salah satu karakteristik utama dari jaringan kompleks adalah adanya struktur komunitas. Secara umum, komunitas didefinisikan sebagai sekumpulan simpul yang memiliki hubungan internal lebih rapat dibandingkan dengan hubungannya terhadap simpul di luar kelompok tersebut (Zhao *et al.*, 2023). Deteksi komunitas bertujuan untuk mengidentifikasi kelompok atau modul tersembunyi dalam jaringan, yang umumnya merepresentasikan unit fungsional atau kelompok simpul dengan peran dan karakteristik yang serupa (Baltsou, Christopoulos and Tsichlas, 2022).

2.4.2 Komunitas Non-Overlapping vs. Overlapping

Pendekatan deteksi komunitas konvensional umumnya membagi jaringan ke dalam kelompok-kelompok yang terpisah tanpa tumpang tindih, di mana setiap simpul hanya menjadi anggota satu komunitas (Ma, Liu and Tao, 2020). Namun, pada banyak jaringan nyata, misalnya jaringan sosial dan biologis, batas antar komunitas bersifat fleksibel dan tidak selalu terpisah secara tegas (Cheng *et al.*, 2021). Kondisi ini melahirkan konsep komunitas tumpang tindih, yaitu struktur komunitas yang memungkinkan satu simpul menjadi anggota dari lebih dari satu komunitas sekaligus, mencerminkan peran ganda simpul tersebut dalam jaringan. Perbedaan antara komunitas non-overlapping dan overlapping ditunjukkan pada Gambar 2.2 dan Gambar 2.3.



Gambar 2. 3 Ilustrasi Komunitas *overlapping* (b)

Gambar 2.2 dan Gambar 2.3 Perbedaan Struktur Komunitas *Non-Overlap* dan *Overlap*. Pada (a), setiap simpul hanya milik satu warna (komunitas). Pada (b), beberapa simpul menjadi bagian dari lebih dari satu warna, menunjukkan adanya tumpang tindih. Dalam jaringan interaksi protein (PPI), keberadaan komunitas tumpang tindih memiliki makna biologis yang penting. Struktur ini umumnya merepresentasikan protein multifungsi, yaitu protein yang terlibat dalam lebih dari satu proses atau fungsi seluler (Platos, Id and Dra, 2024). Protein tersebut berinteraksi dengan kelompok partner yang berbeda sesuai dengan fungsi yang dijalankannya. Oleh karena itu, pemodelan komunitas secara non-overlapping pada jaringan PPI berpotensi menghilangkan informasi penting dan tidak

mampu merepresentasikan kompleksitas biologis yang sebenarnya (Platos, Id and Dra, 2024).

Deteksi komunitas tumpang tindih menjadi krusial karena area tumpang tindih sering kali merepresentasikan titik penghubung antar jalur biologis yang berbeda. Berbagai studi menunjukkan bahwa protein yang berada pada persimpangan komunitas umumnya berperan sebagai regulator transkripsi atau protein pensinyalan yang terlibat dalam banyak proses seluler (Platos, Id and Dra, 2024). Hal ini mengindikasikan bahwa modul fungsional dalam sel saling terhubung melalui regulator bersama, sehingga analisis tumpang tindih komunitas dapat dimanfaatkan untuk mengkaji komunikasi silang antar proses biologis (Platos, Id and Dra, 2024).

Meskipun penting, deteksi komunitas tumpang tindih menghadapi sejumlah tantangan. Banyak algoritma yang ada cenderung menghasilkan komunitas yang terlalu besar dan tercampur, sehingga sulit untuk menginterpretasikan fungsi biologisnya secara spesifik. Selain itu, algoritma yang berbasis pada perluasan benih (*seed expansion*) seringkali tidak stabil dan sangat sensitif terhadap pemilihan benih awal, yang dapat menyebabkan hasil yang tidak konsisten (Vieira, Xavier and Evsukoff, 2020). Tantangan lainnya adalah bagaimana mengevaluasi kualitas partisi tumpang tindih secara akurat, karena metrik tradisional seperti modularitas seringkali dirancang untuk komunitas yang tidak tumpang tindih . Kesulitan-kesulitan ini menunjukkan perlunya pengembangan metode yang lebih kuat dan efisien, terutama yang mampu beroperasi dengan informasi lokal untuk mengatasi skalabilitas pada jaringan biologis yang besar dan kompleks (Platos, Id and Dra, 2024).

2.5 Local Greedy Extended Dynamic Overlapping Community Detection (GLOD)

Local Greedy Extended Dynamic Overlapping Community Detection (GLOD) merupakan metode deteksi komunitas tumpang tindih berbasis perluasan lokal secara serakah (*local greedy expansion*). Metode ini dikembangkan untuk mengatasi sejumlah kelemahan pada algoritma deteksi komunitas sebelumnya, terutama ketergantungan pada informasi global jaringan, sensitivitas tinggi terhadap pemilihan satu simpul awal *(single-seed bias*), serta kecenderungan menghasilkan komunitas yang redundan atau tidak stabil. GLOD memiliki tiga tahapan utama yang dilakukan secara berurutan, yaitu tahap inisialisasi benih (*seeding phase*), tahap perluasan komunitas (*expansion phase*), dan tahap penggabungan komunitas (*merge phase*) (Song *et al.*, 2023).

2.6.1 Tahap Inisialisasi Benih (*Seeding Phase*)

Pada tahap awal, algoritma GLOD membangun komunitas kasar (*rough communities*). Komunitas kasar ini berfungsi sebagai fondasi untuk *expansion phase* atau tahap perluasan. Pendekatan ini dirancang untuk mengurangi bias pemilihan satu simpul awal, yang sering menjadi permasalahan pada metode deteksi komunitas lokal konvensional. Rough community pada GLOD dibentuk dari wilayah jaringan yang memiliki kepadatan lokal tinggi, sehingga benih komunitas yang dihasilkan lebih representatif terhadap struktur komunitas yang sebenarnya (Song *et al.*, 2023).

Proses seeding dimulai dengan memilih sebuah simpul transisional v_i yang belum memiliki label komunitas. Simpul ini umumnya berada pada area jaringan yang aktif dan memiliki keterhubungan lokal relatif tinggi. Selanjutnya, tingkat kesamaan antara simpul v_i dan setiap tetangganya v_j dihitung menggunakan Common Neighbor Similarity (NC), yaitu persamaan 1 dibawah ini: (Song *et al.*, 2023).

$$NC(v_i, v_j) = \left|\mathrm{N}(v_i) \cap \mathrm{N}(v_j)\right| \quad (2.2)$$

Keterangan:

$\mathrm{NC}(v_i, v_j)$: nilai kesamaan tetangga umum antara simpul v_i dan v_j.

$N(v_i)$: himpunan semua tetangga langsung dari v_i.

$N(v_j)$: himpunan semua tetangga langsung dari v_j.

Nilai ini mengukur sejauh mana dua simpul berbagi tetangga yang sama. Semakin besar nilai NC, semakin besar kemungkinan kedua simpul berada dalam komunitas yang sama. Berdasarkan nilai kesamaan ini, simpul v_i digabungkan dengan tetangga-tetangganya yang memiliki kesamaan tertinggi untuk membentuk sebuah rough node V_i atau komunitas kasar (Song *et al.*, 2023). Setelah sejumlah rough community terbentuk, algoritma melakukan proses perankingan untuk memilih seed terbaik yang akan diperluas. Kriteria perankingan meliputi Jumlah simpul dalam rough community, Jumlah dan bobot sisi internal, Nilai sentralitas lokal rough community. Rough community dengan nilai sentralitas dan kepadatan tertinggi dipilih sebagai seed final. Strategi ini memastikan bahwa seed berasal dari inti komunitas lokal (*local dense region*), sehingga komunitas hasil perluasan memiliki struktur internal yang kuat dan mengurangi kemungkinan terbentuknya komunitas kosong (Song *et al.*, 2023). Untuk memperjelas proses pembentukan seed pada algoritma GLOD, disajikan dalam bentuk pseudocode pada Algoritma 1.

Algoritma 1: Seeding Phase

```
Input: Graph G = (V, E)
       NL = daftar node yang belum punya komunitas
Output: S = himpunan seed
S ← kosong
Selama NL tidak kosong:
    Untuk setiap node vi di NL:
        Pilih tetangga vj yang paling mirip dengan vi
        Lakukan ekspansi awal → hasilkan Vf
        Hitung similarity NC(vi)
        Ulangi sampai tidak ada tetangga yang lebih mirip
        Bentuk seed Vi = {vi} ∪ N(vi)
Pilih seed terbaik berdasarkan:
    - derajat node
    - jumlah node
    - jumlah edge

Masukkan ke S
Kembalikan S
```

Berdasarkan pseudocode pada Algoritma 1, proses pembentukan seed dilakukan dengan mempertimbangkan kesamaan tetangga dan kepadatan lokal sehingga menghasilkan kandidat komunitas awal yang representatif.

2.6.2 Tahap Perluasan Komunitas (*Expansion Phase*)

Tahap *expansion phase* bertujuan untuk mengembangkan seed terpilih menjadi komunitas yang stabil melalui proses iteratif. Pada tahap ini, simpul-simpul di sekitar komunitas dievaluasi untuk menentukan apakah simpul tersebut layak ditambahkan atau dihapus. Proses evaluasi dilakukan dengan mengombinasikan tiga fungsi kualitas, yaitu fitness function, fungsi $\omega(v_i)$, dan fungsi afiliasi $F(v, s)$ (Song *et al.*, 2023).

1. Fungsi Kebugaran (*Fitness Function*)

Fungsi kebugaran komunitas digunakan untuk mengukur kualitas struktur komunitas yang sedang dibangun. Secara matematis, fungsi ini didefinisikan pada persamaan 2 di bawah ini:

$$f(C) = \frac{d_{\text{in}}^{C}}{(d_{\text{in}}^{C} + d_{\text{out}}^{C})^{\alpha}} \tag{2.3}$$

Keterangan:

$f(C)$: nilai kelayakan (fitness) dari komunitas C,
d_{in}^{C} : jumlah derajat internal komunitas,
d_{out}^{C} : adalah jumlah derajat eksternal komunitas,
α : parameter resolusi yang mengontrol ukuran komunitas.

Berdasarkan fungsi tersebut, kebugaran sebuah node v dapat dihitung dengan:

$$f(v) = \begin{cases} f(C \cup \{v\}) - f(C), \forall v \in N(C); \\ f(C) - f(C - \{v\}), \forall v \in C. \end{cases} \tag{2.4}$$

Keterangan:

$f(v)$: nilai kelayakan simpul v.
C : komunitas saat ini.
v : Simpul yang sedang dievaluasi.
$N(C)$: himpunan simpul tetangga dari C.
$C \cup \{v\}$: komunitas setelah menambahkan simpul v.

Nilai α berperan penting dalam mengatur sensitivitas ukuran komunitas. Berdasarkan temuan Chen et al., nilai $\alpha = 0{,}8$ memberikan performa optimal dalam mendeteksi komunitas tumpang tindih. Kontribusi sebuah simpul terhadap komunitas dihitung berdasarkan perubahan nilai kebugaran ketika simpul tersebut ditambahkan atau dihapus dari komunitas (Song *et al.*, 2023).

2. Metode $\omega(v_i)$

Fungsi $\omega(v_i)$ mengevaluasi keterkaitan simpul kandidat dengan komunitas saat ini dengan mempertimbangkan tetangga tingkat pertama dan kedua. Fungsi ini membantu algoritma memilih simpul yang secara struktural paling relevan untuk memperkuat kepadatan komunitas (Song *et al.*, 2023). Dengan $N(v_i)$ sebagai himpunan tetangga dari node v_i, $N^2(v_i)$ sebagai tetangga tingkat dua, dan NC_i sebagai irisan antara tetangga node v_i dengan komunitas C, maka persamaan $\omega(v_i)$ didefinisikan pada persamaan 4 berikut:

$$\omega_{(v_i)} = \begin{cases} \max\limits_{v_j \in N_{Ci}} \left(\dfrac{\left|N(v_i) \cap N(v_j)\right| + 1}{\left|N(v_j)\right|} + 0.1 * \dfrac{\left|N_2(v_i) \cap N_2(v_j)\right| + 1}{\left|N_2(v_j)\right|} \right) & \Delta M \geq 0 \\ \dfrac{}{1.1} \\ 0, & \Delta M < 0 \end{cases} \quad (2.5)$$

Keterangan:

$\omega(v_i)$: nilai kelayakan simpul v_i,
v_j : simpul tetangga dari v_i yang sudah masuk komunitas C,
$N(v_i)$: tetangga langsung simpul v_i,
$N_2(v_i)$: tetangga dari tetangga v_i (himpunan tetangga tidak langsung),
NC_i : hasil irisan antara $N(v_i)$ dengan komunitas C.

3. Fungsi $F(\mathrm{v}, s)$

Fungsi $F(v, s)$ mengukur seberapa kuat keterhubungan sebuah simpul terhadap seed awal. Nilai fungsi ini berada pada rentang 0 sampai 1, di mana nilai tinggi menunjukkan bahwa simpul tersebut sangat dipengaruhi oleh struktur seed dan layak dimasukkan ke dalam komunitas (Song *et al.*, 2023). Misalkan v_k adalah node yang sedang dievaluasi dan $\mathrm{s} = \{\mathrm{v_i}, \mathrm{v_j}\}$ adalah pasangan *seed*, maka fungsi *F* didefinisikan sebaga persamaan 5:

$$F(v_k, s) = \frac{|(\mathrm{N}(v_k) \cap s)|}{|s|} \quad (2.6)$$

Keterangan:

$F(v_k, s)$: nilai keterikatan simpul v_k terhadap komunitas benih s,
v_k : simpul tetangga yang sedang dievaluasi,
s : himpunan simpul pada benih komunitas awal,
$N(v_k)$: himpunan tetangga dari v_k.

Untuk memperjelas proses perluasan komunitas pada algoritma GLOD, disajikan pseudocode pada Algoritma 2.

Algoritma 2: Expansion Phase

```
Input: Graph G = (V, E)
       S = himpunan seed
Output: C = komunitas
Untuk setiap seed s di S:
    Ambil semua tetangga s → Ne
    Untuk setiap node vi di Ne:
        Hitung:
            f(vi)   → fitness komunitas
            w(vi)   → nilai kepentingan node
            F(vi,s) → kedekatan dengan seed
    Pilih node terbaik (nilai maksimum dari f, w, atau F)
    Jika memenuhi syarat:
        Tambahkan vi ke komunitas C
Kembalikan C
```

Berdasarkan pseudocode pada Algoritma 2, proses perluasan komunitas dilakukan secara iteratif dengan memilih simpul terbaik berdasarkan kombinasi fungsi kualitas.

2.6.3 Tahap Penggabungan Komunitas (*Merge Phase*)

Tahap merge phase dalam algoritma GLOD menggunakan Koefisien Jaccard. Tujuan dari tahap ini adalah meningkatkan kualitas hasil deteksi dengan mengatasi masalah redundansi komunitas. Redundansi muncul akibat adanya tumpang tindih yang berlebihan (*excessive overlapping*), di mana beberapa komunitas berbeda memiliki anggota yang hampir sama. Kondisi ini dapat menyebabkan interpretasi hasil, khususnya dalam konteks biologis, menjadi kurang jelas (Song *et al.*, 2023).

Untuk mengatasi hal tersebut, GLOD mengidentifikasi komunitas-komunitas yang memiliki tingkat kesamaan tinggi, lalu menggabungkannya menjadi satu komunitas yang lebih besar. Metrik ini secara khusus dirancang untuk mengevaluasi kesamaan dengan mempertimbangkan overlapping nodes atau node-node yang dimiliki oleh lebih dari satu komunitas (Song *et al.*, 2023).

Secara umum, Koefisien Jaccard standar digunakan untuk menilai efektivitas deteksi komunitas dengan cara mengukur kesamaan antara komunitas hasil deteksi dan struktur komunitas sebenarnya (Song *et al.*, 2023). Rumus Koefisien Jaccard standar didefinisikan sebagai berikut:

$$J(C_{f,j}, C_{r,i}) = \frac{|C_{f,j} \cap C_{r,i}|}{|C_{f,j} \cup C_{r,i}|} \tag{2.7}$$

Dengan:

$J(C_{f,j}, C_{r,i})$: nilai kesamaan Jaccard antar pasangan komunitas.

$C_{f,j}$: komunitas deteksi ke- j.

$C_{r,i}$: komunitas ke- i.

Nilai J berkisar antara 0 sampai 1. Jika $J = 0$, maka dua komunitas sepenuhnya independen atau tidak terkait. Dalam metode GREESE, dua komunitas digabungkan jika nilai Jaccard melebihi sepertiga jumlah node dalam komunitas terkecil (Song *et al.*, 2023). Untuk memperjelas proses *merge phase*, pseudocode disajikan pada Algoritma 3.

Algoritma 3: Merge Phase

```
Input: OV = kumpulan node overlap
Output: komunitas hasil gabungan
Untuk setiap set overlap OVi:
    Untuk setiap node ovi di OVi:
        Hitung nilai kemiripan J(ovi)
    Hitung total J(OVi)
    Jika J(OVi) ≥ 1/3:
        Ambil komunitas terkait C = {c1, c2, ..., ck}
        Gabungkan semua komunitas dalam C
Kembalikan hasil komunitas
```

Berdasarkan pseudocode pada Algoritma 3, proses penggabungan komunitas dilakukan dengan mengukur tingkat kesamaan antar komunitas menggunakan koefisien Jaccard untuk mengurangi redundansi.

2.6 Evaluasi Komunitas

Evaluasi kualitas komunitas pada jaringan dengan struktur tumpang tindih memerlukan pendekatan yang berbeda dibandingkan dengan komunitas yang bersifat terpisah. Pada komunitas non-overlapping, batas komunitas umumnya ditentukan dengan memutus interaksi antar simpul. Sebaliknya, pada komunitas overlapping, batas justru sering terbentuk pada simpul yang menjadi anggota lebih dari satu komunitas, sehingga peran simpul menjadi lebih dominan dibandingkan sisi (Havemann *et al.*, 2012).

Untuk mengatasi tantangan evaluasi pada kasus ini, Havemann et al. (2012) memperkenalkan metrik *Normalized Node Cut*. Metrik ini mengukur konduktansi pada simpul-simpul perbatasan dalam komunitas tumpang tindih, yang merupakan adaptasi dari konsep konduktansi graf tradisional yang umumnya berbasis sisi. Secara matematis, Normalized Node Cut untuk suatu komunitas dengan himpunan simpul C didefinisikan sebagai total konduktansi ternormalisasi dari simpul-simpul perbatasan, dengan rumus:

$$\psi(C) = \frac{1}{k_{in}(C)} \sum_{i \in C} \frac{k_i^{in}(C) k_i^{out}(C)}{k_i} \tag{2.8}$$

Keterangan:

C	: komunitas yang sedang dievaluasi,
$i \in C$	: simpul anggota komunitas C,
$k_i^{in}(C)$	: Derajat internal simpul i,
$k_i^{in}(C)$	: Derajat eksternal simpul i,
k_i	: Total derajat simpul i.

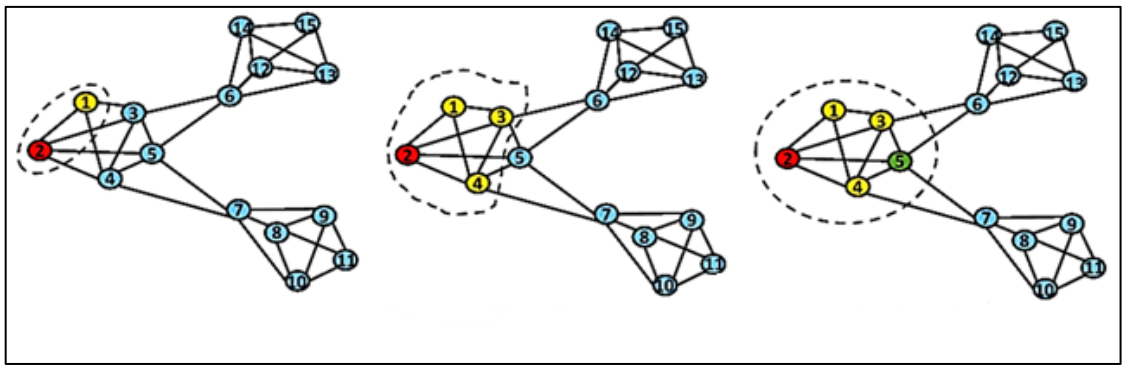
2.7 Local Community Detection (LCD)

Local Community Detection (LCD) merupakan pendekatan dalam deteksi komunitas pada jaringan kompleks yang berfokus pada identifikasi komunitas di sekitar simpul tertentu dengan memanfaatkan informasi lokal dari jaringan tanpa memerlukan pengetahuan terhadap keseluruhan struktur jaringan (Ni *et al.*, 2019). Pendekatan ini menjadi penting terutama pada jaringan berskala besar atau dinamis, di mana informasi global sering kali sulit diperoleh atau membutuhkan biaya komputasi yang tinggi (Baltsou, Christopoulos and Tsichlas, 2022). LCD bekerja dengan memulai proses dari satu atau beberapa *seed node*, kemudian memperluas komunitas tersebut secara bertahap dengan menambahkan simpul yang memiliki keterkaitan kuat berdasarkan struktur lokal jaringan (Ni *et al.*, 2019). Metode ini banyak digunakan dalam berbagai domain seperti jaringan sosial, jaringan web, dan jaringan biologis seperti *protein–protein interaction* (PPI), karena mampu mengidentifikasi kelompok simpul yang memiliki hubungan erat secara lebih efisien dibandingkan metode deteksi komunitas global (Baltsou, Christopoulos and Tsichlas, 2022). Selain itu, pendekatan lokal juga memungkinkan penemuan komunitas yang lebih relevan terhadap simpul tertentu serta lebih sesuai untuk jaringan yang memiliki struktur komunitas kompleks, termasuk komunitas yang saling tumpang tindih (*overlapping communities*) (Ni *et al.*, 2019).

2.5.1. Prinsip Dasar dan Keunggulan LCD

Deteksi komunitas lokal (LCD) merupakan pendekatan untuk mengidentifikasi struktur komunitas yang berfokus pada satu atau beberapa simpul awal (*seed*) tanpa memerlukan informasi mengenai keseluruhan topologi jaringan. Berbeda dengan metode global yang membagi seluruh jaringan menjadi beberapa komunitas, LCD hanya mengeksplorasi area di sekitar simpul awal hingga batas komunitas lokal dapat ditentukan (Baltsou, Christopoulos and Tsichlas, 2022).

Guo et al. (2021) mendefinisikan komunitas lokal sebagai subgraf yang secara struktural terbagi ke dalam tiga wilayah utama. *Core Area* (D) merupakan bagian inti komunitas dengan kepadatan koneksi sangat tinggi, di mana seluruh tetangga simpul di wilayah ini juga berada dalam komunitas yang sama. *Boundary Part* (B) adalah wilayah tepi komunitas yang berisi simpul-simpul dengan setidaknya satu tetangga di luar komunitas. Sementara itu, *Unknown Network* (U) mencakup bagian jaringan yang belum dieksplorasi oleh algoritma dan tidak dianggap relevan terhadap komunitas lokal yang sedang dianalisis (Guo *et al.*, 2022). Pembagian wilayah ini penting karena kualitas komunitas lokal ditentukan oleh kekuatan hubungan internal pada Core Area dibandingkan dengan koneksi yang mengarah ke Boundary. Semakin kuat keterikatan internal pada inti komunitas, semakin baik struktur komunitas lokal yang terbentuk (Guo *et al.*, 2022).



Gambar 2. 4 Skema Umum LCD

Secara umum, pendekatan deteksi komunitas terbagi menjadi dua jenis, yaitu metode global dan metode lokal. Metode global, seperti algoritma partisi graf konvensional, membutuhkan informasi lengkap mengenai seluruh struktur jaringan. Meskipun mampu menghasilkan pembagian komunitas yang menyeluruh, pendekatan ini kurang efisien ketika diterapkan pada jaringan dunia nyata yang berukuran sangat besar, bersifat dinamis, atau memiliki keterbatasan akses data, seperti jaringan web dan jaringan biologis yang kompleks (Baltsou, Christopoulos and Tsichlas, 2022). Sebaliknya, pendekatan lokal menawarkan cara kerja yang lebih fleksibel dengan memusatkan analisis pada bagian jaringan yang relevan terhadap simpul tertentu. Algoritma deteksi komunitas lokal (LCD) bekerja dengan mengeksplorasi lingkungan terdekat dari simpul yang diminati tanpa harus memproses seluruh jaringan. Dalam konteks jaringan interaksi protein (PPI), banyak fungsi biologis penting justru muncul dari pola interaksi lokal antar protein yang berdekatan. Pola semacam ini sering kali tidak tertangkap secara optimal oleh metrik global, sehingga pendekatan lokal dinilai lebih tepat dan kontekstual untuk analisis biologis (Dilmaghani *et al.*, 2022).

Kelebihan utama dari deteksi komunitas lokal terletak pada efisiensi dan skalabilitasnya. Karena hanya memerlukan sebagian kecil data jaringan, kebutuhan komputasi menjadi jauh lebih rendah dibandingkan metode global (Song *et al.*, 2023). Hal ini membuatnya sangat cocok untuk diterapkan pada jaringan berskala besar yang bisa memiliki jutaan atau bahkan miliaran simpul dan sisi. Selain itu, metode lokal tidak menuntut pengetahuan menyeluruh terhadap topologi jaringan, sehingga sesuai digunakan pada sistem dengan data yang bersifat terbatas, dinamis, atau terlalu besar untuk diproses secara terpusat (Ni *et al.*, 2019).

2.5.2. Evolusi Algoritma LCD: Dari Non-Overlap ke Overlap

Sejak diperkenalkan, algoritma deteksi komunitas lokal terus berkembang. Sebagian besar metode awal termasuk dalam pendekatan ekspansi benih secara serakah (*greedy seed expansion*), di mana komunitas dibangun secara bertahap dari satu atau beberapa simpul awal dengan menambahkan simpul tetangga yang mampu meningkatkan nilai fungsi kualitas local (Ni *et al.*, 2019). Beberapa metode yang banyak digunakan dalam pendekatan ini antara lain algoritma LFM yang mengandalkan fungsi fitness untuk mengendalikan proses ekspansi, serta metode berbasis ego-network yang memanfaatkan subgraf yang terdiri dari simpul pusat dan tetangga terdekatnya (Song *et al.*, 2023).

Meskipun cukup efektif, algoritma LCD generasi awal memiliki sejumlah keterbatasan. Salah satu kelemahan utama adalah ketergantungan yang tinggi pada pemilihan simpul awal, di mana hasil komunitas dapat berbeda secara signifikan jika benih berada di pusat atau di tepi komunitas (Ni *et al.*, 2019). Selain itu, banyak metode tersebut hanya mampu mendeteksi komunitas non-overlapping. Keterbatasan ini menjadi persoalan penting dalam jaringan biologis seperti jaringan interaksi protein, karena protein multifungsi secara alami terlibat dalam lebih dari satu proses biologis dan membentuk struktur komunitas yang saling tumpang tindih (Ni *et al.*, 2019).

2.8 Enrichment Analysis

Enrichment Analysis merupakan tahapan penting dalam interpretasi studi berbasis sistem, khususnya pada analisis OMICS. Tujuan utama analisis ini adalah mengidentifikasi apakah komunitas protein yang ditemukan menunjukkan keterwakilan berlebih pada fungsi biologis, jalur seluler, atau lokasi subseluler tertentu (Zhou *et al.*, 2019). Secara prinsip, enrichment analysis bekerja dengan cara membandingkan daftar protein dari setiap komunitas yang ditemukan dengan ribuan himpunan gen atau protein yang telah terdokumentasi dalam basis data biologis. Melalui perbandingan ini, dapat diidentifikasi apakah kelompok protein dalam suatu komunitas secara signifikan terkait dengan proses biologis tertentu yang sudah diketahui. Dengan demikian, analisis ini tidak hanya memberikan makna biologis terhadap struktur komunitas yang terdeteksi, tetapi juga membantu memvalidasi bahwa kelompok protein yang terhubung secara struktural dalam jaringan interaksi protein (PPI) memang bekerja sama dalam menjalankan fungsi biologsis yang koheren (Zhou *et al.*, 2019).

Proses analisis dimulai dengan mengambil daftar protein dari masing-masing komunitas yang dihasilkan oleh algoritma. Daftar tersebut kemudian dimasukkan ke dalam

platform bioinformatika berbasis web, seperti Metascape atau Enrichr, yang mengintegrasikan lebih dari 40 basis data biologis independen. Pada tahap ini, platform akan melakukan uji statistik menggunakan hypergeometric test dengan koreksi Benjamini-Hochberg terhadap nilai p-value untuk menentukan kategori fungsional mana yang signifikan diperkaya. Basis data yang sering digunakan dalam enrichment analysis antara lain Gene Ontology (GO) untuk mendeskripsikan proses biologis serta fungsi molekuler, dan KEGG maupun Reactome untuk jalur metabolisme serta pensinyalan seluler (Zhou *et al.*, 2019).

Salah satu tantangan yang sering muncul dalam enrichment analysis adalah adanya redundansi hasil, yaitu kondisi ketika banyak istilah biologis atau jalur yang mirip muncul secara bersamaan, sehingga menyulitkan interpretasi. Kondisi ini umumnya disebabkan oleh struktur hierarkis basis data, seperti GO, serta keterkaitan alami antar ontologi. Untuk mengatasinya, pendekatan modern biasanya mengelompokkan istilah-istilah yang mirip ke dalam klaster non-redundan berdasarkan kesamaan makna, sehingga hasil analisis menjadi lebih ringkas dan mudah dipahami (Zhou *et al.*, 2019). Hasil akhir dari enrichment analysis adalah profil fungsional untuk setiap komunitas protein yang terdeteksi. Profil ini berisi informasi mengenai jalur biologis, fungsi molekuler, maupun proses seluler yang signifikan terkait dengan komunitas tersebut. Sebagai contoh, komunitas yang diperkaya oleh istilah seperti regulasi siklus sel, perbaikan DNA, atau jalur pensinyalan MAPK dapat diinterpretasikan sebagai kelompok protein yang berperan penting dalam mekanisme biologis yang sering mengalami gangguan pada kanker payudara (Zhou *et al.*, 2019).

2.9 Penelitian Terdahulu

Penelitian mengenai deteksi komunitas pada jaringan protein-protein interaction (PPI) telah berkembang pesat dengan berbagai pendekatan komputasional. Secara umum, tujuan utama penelitian-penelitian tersebut adalah merancang algoritma yang tidak hanya efisien secara komputasi, tetapi juga mampu menghasilkan komunitas yang bermakna secara biologis. Hal ini menjadi penting karena dalam jaringan biologis, khususnya PPI, sering ditemukan struktur komunitas yang bersifat tumpang tindih (overlapping), di mana satu protein dapat terlibat dalam lebih dari satu proses biologis sekaligus.

Pendekatan awal dalam deteksi komunitas banyak berfokus pada eksplorasi struktur topologi jaringan melalui metode ekspansi lokal. Algoritma seperti LFM dan LOCD memanfaatkan fungsi fitness untuk memperluas komunitas dari simpul awal (seed). Meskipun cukup efektif, pendekatan ini memiliki keterbatasan karena sangat bergantung pada pemilihan benih awal dan parameter tertentu, sehingga hasil komunitas dapat menjadi tidak stabil dan sensitif terhadap perubahan konfigurasi awal. Untuk meningkatkan akurasi pemilihan seed, Wang et al. (2021) mengusulkan algoritma NLC (Neighbor Clustering Coefficient) yang memanfaatkan koefisien klasterisasi tetangga sebagai dasar ekspansi komunitas. Pendekatan ini menggabungkan informasi berbasis tepi dan struktur lokal guna mengurangi ekspansi berlebihan serta meningkatkan kualitas komunitas yang dihasilkan pada jaringan biologis.

Pengembangan selanjutnya dilakukan oleh Zhao et al. (2023) melalui algoritma OLCRE (Overlapping Community Detection Algorithm based on High-Quality Subgraph

Extension in Local Core Regions of Network). Berbeda dengan metode berbasis satu simpul, OLCRE menggunakan subgraf berkualitas tinggi sebagai benih komunitas, sehingga lebih stabil dan tidak terlalu bergantung pada parameter tertentu. Pendekatan ini menunjukkan peningkatan konsistensi dalam mendeteksi komunitas overlap. Lebih lanjut, Song et al. (2023) memperkenalkan algoritma GLOD (Local Greedy Extended Dynamic Overlapping Community Detection) sebagai penyempurnaan pendekatan lokal sebelumnya. GLOD menggunakan strategi coupled seeds untuk mengurangi bias satu simpul awal serta menerapkan merge phase berbasis koefisien Jaccard guna mengatasi redundansi komunitas akibat tumpang tindih berlebihan. Kombinasi mekanisme ini menjadikan GLOD lebih stabil dan adaptif dalam menangani struktur overlapping pada jaringan kompleks, termasuk jaringan PPI.

Selain pendekatan berbasis heuristik dan greedy expansion, metode berbasis deep learning juga mulai diterapkan. Zhang and Liu (2025b) mengembangkan model Graph Neural Network (GNN) untuk memetakan struktur komunitas protein kanker payudara secara hierarkis dengan memanfaatkan fitur sekuens protein, data interaksi PPI, serta anotasi Gene Ontology. Pendekatan ini memungkinkan identifikasi komunitas overlap yang kemudian dianalisis berdasarkan tekanan mutasi dan kelangsungan hidup pasien untuk menemukan biomarker potensial.

Dalam konteks jaringan kanker, Prayoga (2025) menerapkan algoritma Ant Colony Optimization (ACO) untuk mendeteksi komunitas pada jaringan interaksi protein kanker paru-paru. Penelitian tersebut menggunakan data dari IntOGen dan cBioPortal yang dikonstruksi menjadi jaringan PPI melalui STRING DB, kemudian mengevaluasi kualitas komunitas berdasarkan nilai modularity dan melakukan enrichment analysis menggunakan Metascape untuk menilai relevansi biologisnya. Hasilnya menunjukkan bahwa ACO mampu menghasilkan komunitas dengan nilai modularity yang stabil serta memiliki keterkaitan fungsional terhadap jalur biologis kanker paru-paru.

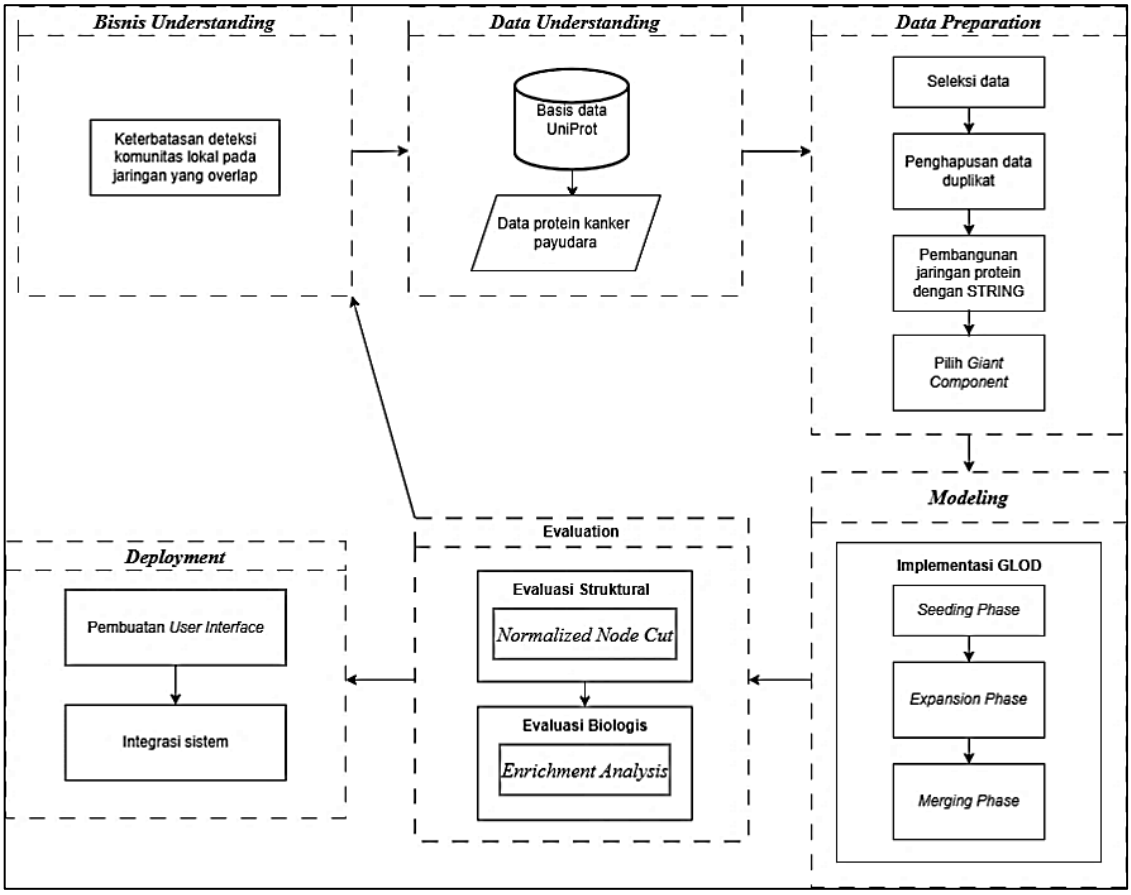
Berdasarkan berbagai penelitian tersebut, dapat disimpulkan bahwa meskipun banyak metode telah dikembangkan, tantangan utama dalam deteksi komunitas PPI masih berkaitan dengan kestabilan hasil dan pengelolaan tumpang tindih yang berlebihan. Oleh karena itu, penelitian ini berfokus pada penerapan algoritma GLOD pada jaringan interaksi protein kanker payudara. Pemilihan GLOD didasarkan pada kemampuannya dalam mengurangi bias pemilihan seed dan meminimalkan redundansi komunitas melalui mekanisme merge phase, sehingga diharapkan mampu merepresentasikan kompleksitas biologis kanker payudara secara lebih akurat.

Tabel 2. 1 State of The Art

No	Penulis	Metode/Nama	Tipe Jaringan	Domain Masalah	Dataset	Evaluasi
1	(Wang *et al.*, 2021)	*Neighbor Clustering Coefficient* (NLC)	Jaringan Kompleks	Jaringan Biologis	LFR Benchmark, PPI (*M. musculus*, *H. sapiens*, dll.)	EQ, NMI, CR, NNC, *Enrichment Analysis*
2	(Zhao *et al.*, 2023)	*Overlapping Community Detection Algorithm based on High-Quality Subgraph Extension in Local Core Regions of Network* (OLCRE)	Jaringan Kompleks	Jaringan Sosial	LFR Benchmark, Karate, Dolphins, Polbooks, dll.	NMI, EQ
3	(Song *et al.*, 2023)	*Local Greedy Extended Dynamic Overlapping Community Detection* (GLOD)	Jaringan Kompleks	Jaringan Dinamis	LFR Benchmark, Amazon, *Provenance Graph*	F-Score
4	(Ni *et al.*, 2019)	*Local Overlapping Community Detection* (LOCD)	Jaringan Kompleks	Jaringan Sosial	LFR Benchmark, Amazon	Recall, Precision, F-Score
5	(Dilmaghani *et al.*, 2022)	*Local Community Detection Algorithm for Protein Complexes with Gene Ontology* (LCDA-GO)	Jaringan Kompleks	Jaringan Biologis	Krogan PPI, Gene Ontology	Precision, Recall, F-Measure, Sensitivity, PPV, Accuracy, *Composite Score*
6	(Zhang and Liu, 2025b)	*Graph Neural Network untuk Pemetaan Hierarkis Komunitas Protein Kanker Payudara (GNN-BCPC)*	Jaringan Kompleks	Jaringan Biologis	PPI kanker payudara	Deteksi Komunitas Tumpang Tindih, Struktur Hierarkis, Analisis Enrichment
7	(Prayoga, 2025)	*Ant Colony Optimization (ACO)*	Jaringan Kompleks	Jaringan Biologis	PPI Kanker Paru-Paru	IntOGen, cBioPortal, STRING DB Modularity, Enrichment Analysis

BAB III
METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif non-eksperimental berbasis *in silico*, yaitu seluruh proses analisis dilakukan secara komputasional tanpa melibatkan eksperimen laboratorium. Objek yang dianalisis berupa jaringan interaksi protein yang tersedia dalam basis data publik dan bertujuan untuk mengidentifikasi kelompok protein yang saling berinteraksi secara padat di dalam jaringan kanker payudara. Kelompok tersebut disebut komunitas, komunitas ini dapat merepresentasikan modul fungsi atau jalur biologis tertentu. Algoritma yang digunakan adalah GLOD, yang memungkinkan satu protein menjadi anggota lebih dari satu komunitas (overlap), sehingga lebih sesuai untuk merepresentasikan kompleksitas sistem biologis. Alur penelitian mengikuti metodologi *CRISP-DM*, yang meliputi *problem understanding*, *data understanding, data preparation*, *modeling*, kemudian dievaluasi menggunakan *Normalized Node Cut* dan divalidasi melalui *enrichment analysis*, dan terakhir *deployment*, mencakup penyajian dan visualisasi hasil, sebagaimana ditunjukkan pada Gambar 3.1.



Gambar 3. 1 Alur penelitian

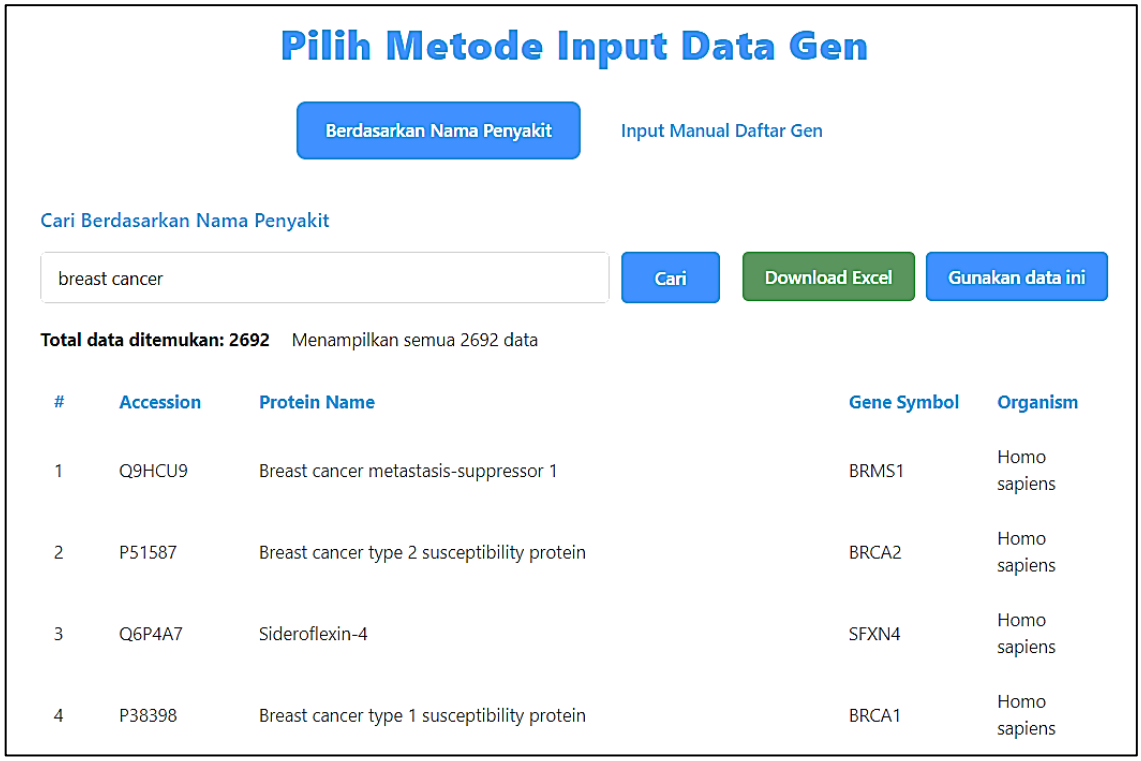
3.1 Business Understanding

Tahap Business Understanding bertujuan memahami permasalahan deteksi komunitas tumpang tindih pada jaringan interaksi protein kanker payudara. Algoritma deteksi komunitas lokal sering menghasilkan komunitas yang tidak stabil dan bersifat redundan sehingga sulit diinterpretasikan secara biologis untuk komunitas overlap. Oleh karena itu,

penelitian ini menerapkan algoritma GLOD untuk membentuk komunitas protein overlap yang lebih stabil dengan keberhasilan diukur melalui struktur jaringan dan validasi biologis menggunakan enrichment analysis.

3.2 Data Understanding

Tahap data understanding bertujuan untuk mengidentifikasi sumber data, struktur data, karakteristik atribut, serta kelayakan data sebelum dilakukan proses preprocessing dan pemodelan jaringan. Data yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh melalui UniProt REST API dengan kata kunci penyakit yang dimasukkan pengguna, misalnya *"breast cancer"*, dan dibatasi pada organisme Homo sapiens (organism_id:9606). Berdasarkan hasil pengambilan data menggunakan endpoint https://rest.uniprot.org/uniprotkb/search, diperoleh sebanyak 2.690 entri protein yang berkaitan dengan kanker payudara. Proses pengambilan dilakukan secara bertahap (pagination) hingga seluruh data berhasil dikumpulkan dalam format JSON, kemudian dikonversi ke dalam bentuk tabel (DataFrame) untuk keperluan analisis lebih lanjut. Tampilan hasil pencarian data melalui sistem ditunjukkan pada Gambar 3.2.



Gambar 3. 2 Pencarian data dari Uniprot

Gambar 3.2 memperlihatkan hasil pencarian dengan kata kunci *breast cancer* yang menghasilkan 2.690 entri protein Homo sapiens. Setiap entri merepresentasikan satu protein yang terindeks dalam basis data UniProt. Data yang diperoleh dari UniProt memuat beberapa atribut utama yang relevan untuk penelitian ini, yaitu:

1) Accession: ID unik protein pada UniProt (primary accession).
2) Protein Name: Nama lengkap protein.
3) Gene Symbol: Simbol gen utama yang merepresentasikan protein tersebut.
4) Organism: Nama organisme (Homo sapiens).

Untuk keperluan pembentukan jaringan interaksi protein, atribut yang digunakan sebagai identitas utama simpul adalah gene symbol, karena jaringan PPI pada STRING dibangun berbasis gen/protein name yang telah distandardisasi. Contoh struktur data hasil ekstraksi ditunjukkan pada Tabel 3.1.

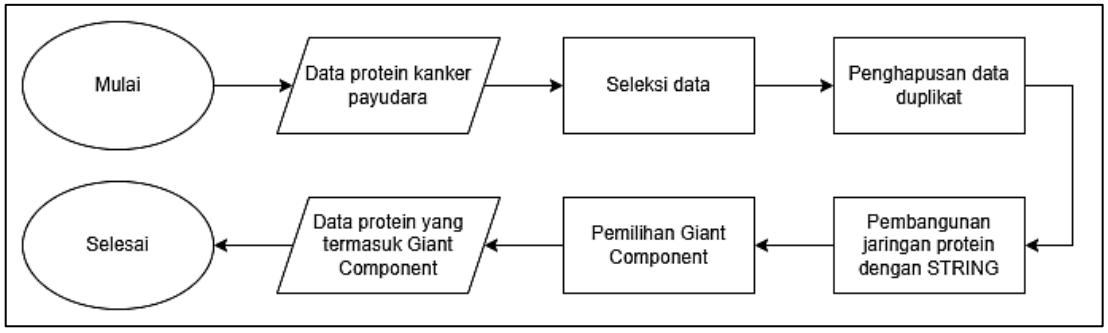
Tabel 3. 1 Contoh struktur data hasil ekstraksi

No	Accession	Protein Name	Gene Symbol	Organism
1	Q9HCU9	Breast cancer metastasis-suppressor 1	BRMS1	Homo sapiens
2	P51587	Breast cancer type 2 susceptibility protein	BRCA2	Homo sapiens
3	P38398	Breast cancer type 1 susceptibility protein	BRCA1	Homo sapiens
4	Q9UBW5	Bridging integrator 2	BIN2	Homo sapiens
5	Q4ZG55	Protein GREB1	GREB1	Homo sapiens

Tabel 3.1 menunjukkan bahwa setiap baris merepresentasikan satu entri protein unik berdasarkan accession. Namun, satu gene symbol dapat memiliki lebih dari satu accession akibat adanya isoform protein atau variasi anotasi dalam UniProt. Oleh karena itu, diperlukan tahap deduplikasi pada tahap berikutnya.

3.3 Data Preparation

Tahap *data preparation* bertujuan menyiapkan data agar layak digunakan pada proses pemodelan. Seluruh tahapan dilakukan menggunakan Python, meliputi seleksi data, penghapusan duplikasi, konstruksi jaringan, dan pemilihan *giant component*. Alur proses ditunjukkan pada Gambar 3.3.

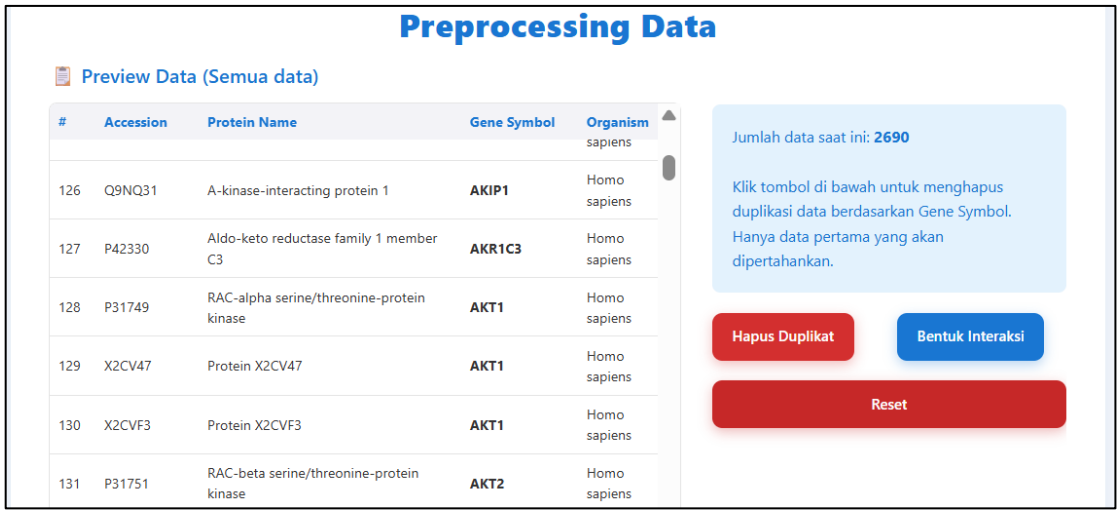


Gambar 3. 3 *Flowchart* data preparation

3.3.1. Data Selection

Tahap data selection dilakukan untuk memastikan bahwa hanya data protein yang relevan secara biologis dan konsisten secara struktural yang digunakan dalam pembentukan jaringan interaksi protein. Tahap *data selection* merupakan bagian dari proses *data preparation* sebagaimana ditunjukkan pada Gambar 3.3, dan dilakukan setelah pengambilan data dari UniProt (Gambar 3.2).

Tahap ini dilakukan untuk memastikan bahwa data yang digunakan relevan dengan konteks penelitian dan siap digunakan dalam pembentukan jaringan interaksi protein. Data diperoleh dari UniProt menggunakan kata kunci *"breast cancer"* dengan pembatasan organisme *Homo sapiens* (organism_id:9606), sehingga hanya protein manusia yang terkait kanker payudara yang diambil. Dari proses ini diperoleh 2.690 entri protein. Tampilan hasil data sebelum dilakukan proses lanjutan ditunjukkan pada Gambar 3.4. Gambar tersebut memperlihatkan struktur data yang terdiri atas kolom Accession, Protein Name, Gene Symbol, dan Organism, serta informasi jumlah data yang diperoleh.



Gambar 3. 4 Struktur data pada tahap data selection

Dalam penelitian ini, gene symbol dipilih sebagai identitas utama simpul jaringan, karena satu gen dapat memiliki lebih dari satu accession akibat isoform protein atau variasi anotasi UniProt. Jika accession digunakan sebagai ID utama, satu gen dapat direpresentasikan sebagai beberapa node sehingga berpotensi menimbulkan duplikasi biologis dan bias dalam struktur jaringan. Secara prosedural, tahap seleksi dilakukan dengan mengekstraksi atribut utama (*accession*, *protein name*, *gene symbol*, dan *organism*), kemudian memastikan setiap entri memiliki gene symbol yang valid. Setiap baris selanjutnya direpresentasikan sebagai satu entitas gen dengan gene symbol sebagai kunci utama dan accession sebagai informasi pendukung. Hasil seleksi ini menjadi input untuk tahap deduplikasi dan konstruksi jaringan berikutnya.

3.3.2. Data Deduplication

Meskipun setiap entri UniProt memiliki accession yang unik, satu gen dapat muncul lebih dari satu kali karena memiliki beberapa isoform protein atau variasi anotasi kurasi. Kondisi ini terlihat pada tahap data selection (Gambar 3.4), di mana terdapat gene symbol yang muncul lebih dari satu kali, seperti AKT1, sehingga berpotensi menyebabkan satu gen direpresentasikan sebagai beberapa simpul berbeda dalam jaringan.

Jika duplikasi tersebut tidak dihapus, maka satu gen dapat dihitung berkali-kali pada saat konstruksi jaringan. Hal ini dapat menyebabkan bias pada perhitungan derajat simpul, kepadatan komunitas, serta hasil evaluasi struktur komunitas. Oleh karena itu, proses deduplikasi dilakukan dengan menjadikan gene symbol sebagai kunci unik, sehingga setiap gen hanya direpresentasikan satu kali dalam jaringan. Contoh sederhana proses deduplikasi ditunjukkan pada Tabel 3.2 dan Tabel 3.3.

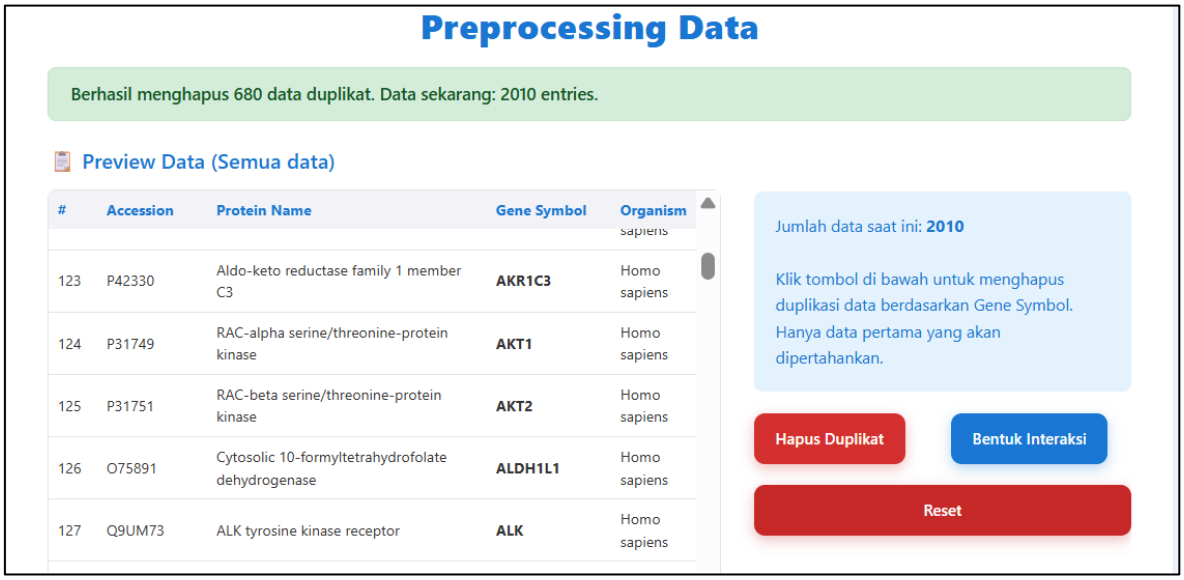
Tabel 3. 2 Contoh data sebelum proses deduplikasi

Accession	Gene Symbol	Protein Name
P31749	AKT1	RAC-alpha serine/threonine-protein kinase
Q9XYZ1	AKT1	AKT1 isoform 2
P31751	AKT2	RAC-beta serine/threonine-protein kinase

Tabel 3. 3 Contoh data setelah proses deduplikasi

Accession	Gene Symbol	Protein Name
P31749	AKT1	RAC-alpha serine/threonine-protein kinase
P31751	AKT2	RAC-beta serine/threonine-protein kinase

Jumlah data sebelum deduplikasi adalah 2.690 entri, pada Gambar 3.4 terlihat bahwa beberapa gene symbol muncul lebih dari satu kali, misalnya AKT1 yang tercatat sebanyak tiga entri berbeda akibat perbedaan accession atau isoform protein. Setelah proses deduplikasi berdasarkan *gene symbol*, setiap gen hanya direpresentasikan satu kali, sehingga tiga entri AKT1 tersebut direduksi menjadi satu entitas gen unik. Jumlah data setelah deduplikasi menjadi 2.010 gen unik, sehingga sebanyak 680 entri duplikat berhasil dihapus. Hasil proses ini ditunjukkan pada Gambar 3.5.



Gambar 3. 5 Hasil Proses Data Deduplication

3.3.3. Network Construction

Setelah diperoleh 2.010 gen unik hasil deduplikasi, tahap berikutnya adalah membangun jaringan interaksi protein menggunakan basis data STRING. Daftar *gene symbol* dikirim ke STRING dengan parameter organisme *Homo sapiens* (9606) dan batas *confidence score* sebesar 0,900. STRING mengembalikan data dalam bentuk *edge list*, yaitu tabel pasangan protein yang saling berinteraksi beserta nilai *confidence score*-nya.

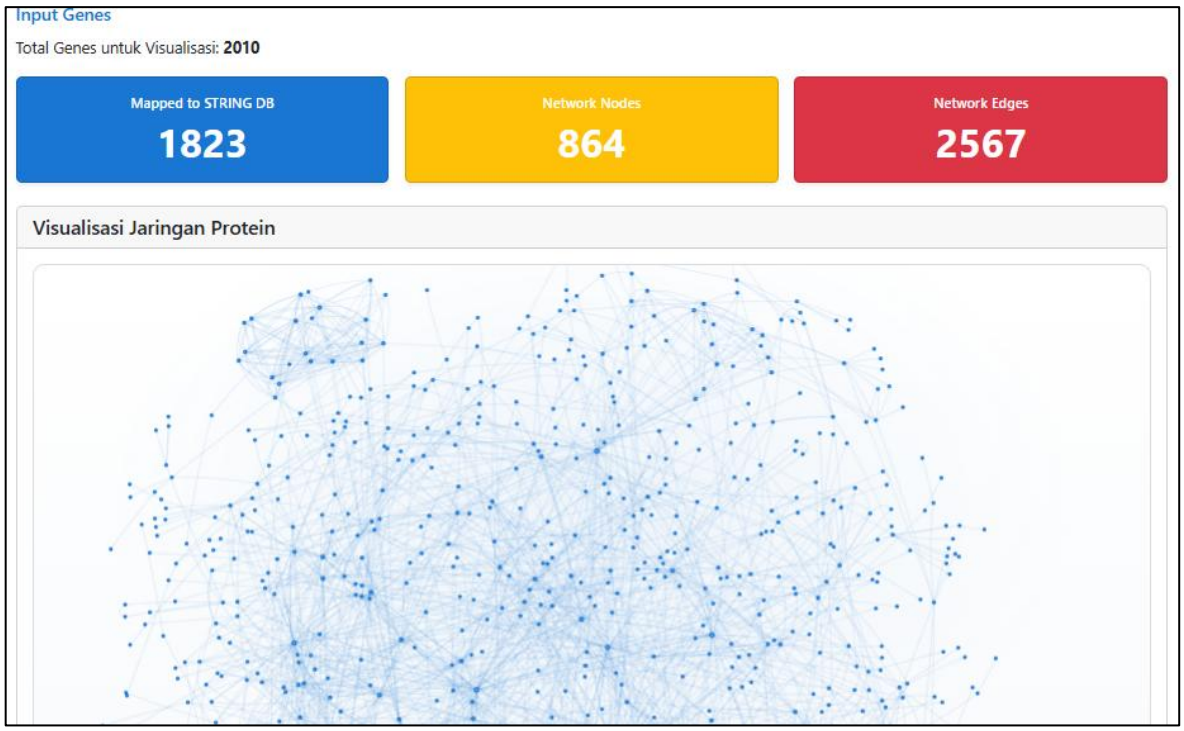
Hasil pemanggilan API STRING dikembalikan dalam bentuk edge list, yaitu tabel yang berisi pasangan protein yang saling berinteraksi. Struktur data ini terdiri dari dua atribut utama, yaitu *node1* dan *node2*, di mana setiap baris merepresentasikan satu hubungan interaksi antar protein. Contoh struktur data tersebut ditunjukkan pada Gambar 3.6. Setiap pasangan gen pada kolom *node1* dan *node2* merepresentasikan satu sisi (*edge*) dalam jaringan interaksi protein, sehingga tabel ini dapat langsung digunakan untuk membangun graf dengan gen sebagai simpul (*node*) dan interaksi sebagai sisi (*edge*). Setelah dilakukan pemfilteran berdasarkan *confidence score* $\geq$ 0,900, jaringan interaksi yang terbentuk terdiri dari 1.823 simpul (node) dan 2.646 sisi (edge). Jaringan tersebut kemudian direpresentasikan sebagai graf tidak berarah dan tidak berbobot (*unweighted graph*), sehingga seluruh interaksi yang memenuhi ambang batas diperlakukan setara dalam proses deteksi komunitas.

1	#node1	node2
2	CAMK1G	NOS3
3	RANBP9	WDR26
4	SNAI2	TP53
5	SNAI2	HDAC1
6	SNAI2	CDH1
7	SNAI2	TWIST1
8	RB1CC1	SQSTM1
9	RB1CC1	ATG5
10	RB1CC1	PTK2B
11	RB1CC1	TP53
12	RB1CC1	PIK3R4
13	RB1CC1	PTK2
14	RB1CC1	BECN1
15	HDAC7	EP300
16	HDAC7	FOXA1
17	ST6GALNAC1	ST3GAL1
18	ST6GALNAC1	GALNT5
19	EPHA8	EFNA1
20	TIGAR	PFKFB4

Gambar 3. 6 Struktur data dari *network construction*

3.3.4. Pemilihan Giant Component

Jaringan hasil tahap sebelumnya tidak seluruhnya terhubung, melainkan terdiri dari beberapa connected component. Untuk memastikan analisis dilakukan pada struktur yang representatif, dipilih komponen terbesar (giant component) sebagai jaringan utama, sementara komponen kecil yang terisolasi dihapus. Jaringan awal terdiri dari 1.823 node dan 2.646 edge, dan setelah proses ini diperoleh jaringan akhir dengan 864 node dan 2.567 edge. Hasil pemilihan giant component ditunjukkan pada Gambar 3.7, yang memperlihatkan struktur jaringan yang lebih terhubung secara global.

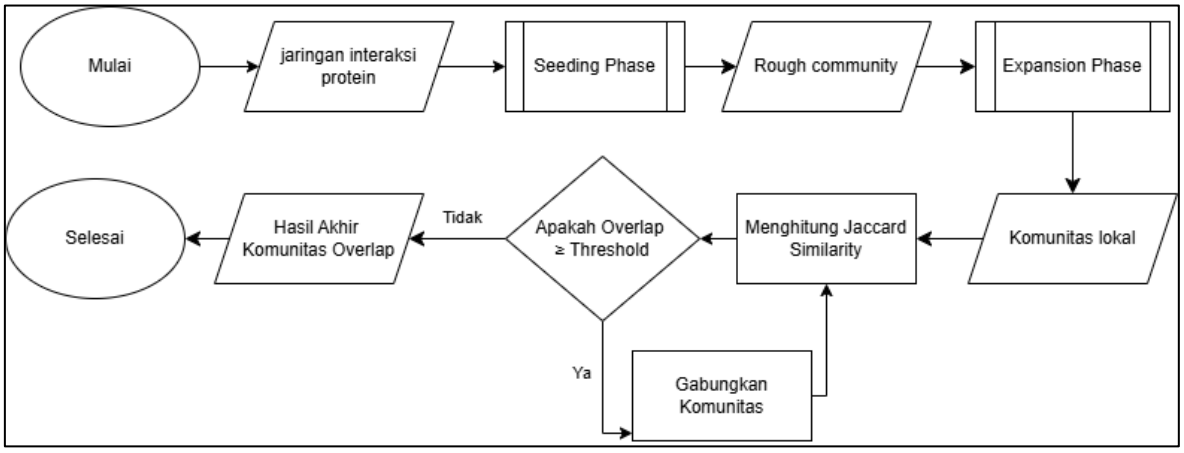


Gambar 3. 7 Pemilihan *giant component*

3.4 Modeling Local Community Detection berbasis GLOD

Tahap pemodelan merupakan inti penelitian ini karena pada tahap ini dilakukan proses deteksi komunitas secara komputasional menggunakan algoritma *Local Greedy Extended Dynamic Overlapping Community Detection* (GLOD). Jaringan interaksi protein kanker payudara terlebih dahulu direpresentasikan dalam bentuk graf tidak berarah (*undirected*) dan tidak berbobot (*unweighted*). Dalam representasi ini, setiap protein dinyatakan sebagai simpul (*node*), sedangkan interaksi antar protein dinyatakan sebagai sisi (*edge*) yang menghubungkan dua simpul. Graf disebut tidak berarah karena hubungan interaksi dianggap bersifat dua arah, dan tidak berbobot karena seluruh interaksi diperlakukan memiliki nilai yang sama. Representasi graf ini menjadi dasar dalam seluruh proses perhitungan algoritma, mulai dari pembentukan benih komunitas hingga tahap penggabungan komunitas.

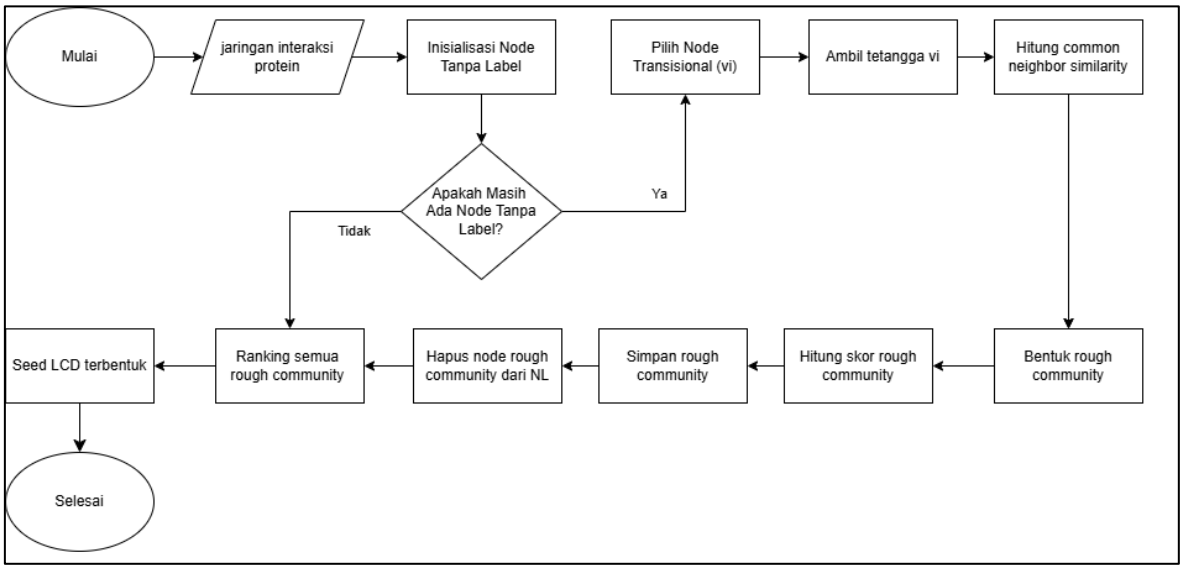
Setiap sisi menyatakan adanya hubungan interaksi fungsional antara dua protein, sedangkan derajat simpul dihitung sebagai jumlah sisi yang terhubung langsung dengan simpul tersebut. Representasi formal ini menjadi dasar dalam seluruh proses perhitungan pada tahapan *seeding*, *expansion*, dan *merging* dalam algoritma GLOD. Algoritma GLOD bekerja melalui tiga fase utama untuk menghasilkan komunitas *overlap* yang stabil dan akurat, yaitu pembentukan komunitas awal menggunakan strategi *seeding phase* berdasarkan keterhubungan dan koefisien pengelompokan tetangga yang tinggi, lalu perluasan komunitas secara rakus menggunakan beberapa fungsi *fitness* untuk mengoptimalkan struktur lokal, serta penggabungan komunitas yang memiliki tingkat tumpang tindih tinggi guna mengurangi redundansi hasil. Alur dari algoritma GLOD secara besar ditunjukkan pada Gambar 3.8.



Gambar 3. 8 Flowchart algoritma GLOD

Pada tahap pemodelan ini, fokus utama adalah implementasi algoritma GLOD untuk mendeteksi komunitas pada jaringan interaksi protein. Pengaturan parameter serta skenario eksperimen yang digunakan untuk mengevaluasi performa algoritma akan dijelaskan secara terpisah pada bagian evaluasi. *Local Community Detection* (LCD) diimplementasikan secara eksplisit pada dua tahap awal GLOD, yaitu *local seeding* dan *local expansion,* di mana pembentukan dan perluasan komunitas dilakukan sepenuhnya berdasarkan informasi ketetanggaan lokal simpul. Proses *seeding phase* berbasis LCD ditunjukkan pada Gambar 3.9. Tahap ini diawali dengan inisialisasi seluruh simpul sebagai node tanpa label komunitas.

Simpul-simpul tersebut disebut sebagai v_i dan dipilih sebagai pusat pembentukan komunitas lokal.



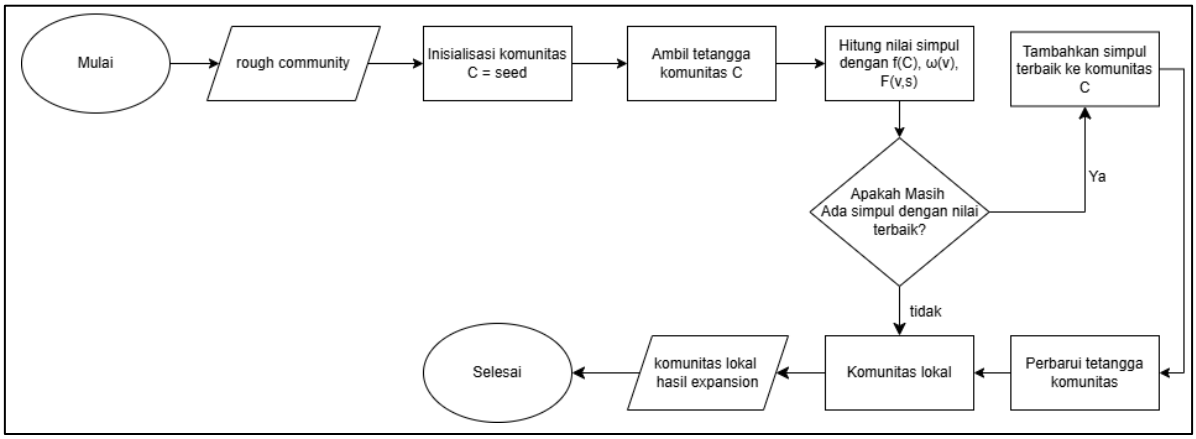
Gambar 3. 9 Flowchart Seeding Phase berbasis lokal

Pada tahap *seeding phase*, setiap simpul yang belum memiliki label komunitas dipilih secara bergantian sebagai pusat pembentukan komunitas lokal. Untuk setiap simpul pusat, seluruh tetangga langsungnya diidentifikasi dan dinotasikan sebagai $N(v)$, yaitu himpunan simpul yang terhubung langsung dengan simpul tersebut. Tingkat kedekatan antara simpul pusat dan tetangganya kemudian dihitung menggunakan Persamaan (2.2) pada Bab II, yaitu *Common Neighbor Similarity (CN)*. Secara intuitif, ukuran ini menghitung berapa banyak tetangga yang dimiliki bersama oleh dua simpul. Semakin banyak tetangga yang sama, maka semakin besar kemungkinan kedua simpul berada dalam komunitas yang sama. Berdasarkan nilai tersebut, simpul pusat digabungkan dengan tetangga yang memiliki nilai kesamaan tertinggi untuk membentuk *rough community* sebagai kandidat komunitas awal.

Dalam implementasinya, proses perhitungan dilakukan dengan terlebih dahulu menentukan himpunan tetangga masing-masing simpul, kemudian menghitung irisan kedua himpunan tersebut untuk memperoleh nilai CN. Nilai kesamaan ini digunakan sebagai dasar pembentukan rough community, di mana simpul pusat akan digabungkan dengan tetangga yang memiliki nilai kesamaan tertinggi. Hasil evaluasi rough community disimpan sebagai kandidat komunitas awal, kemudian seluruh simpul yang telah tergabung dihapus dari daftar node tanpa label (NL). Proses ini dilakukan secara iteratif hingga tidak terdapat lagi simpul tanpa label. Setelah seluruh rough community terkumpul, dilakukan proses *ranking* untuk menentukan rough community terbaik yang selanjutnya dipilih sebagai *seed LCD*. Tahap ini sepenuhnya merepresentasikan konsep LCD karena pembentukan seed dilakukan berdasarkan lingkungan tetangga terdekat dan metrik lokal, tanpa mempertimbangkan keseluruhan topologi jaringan.

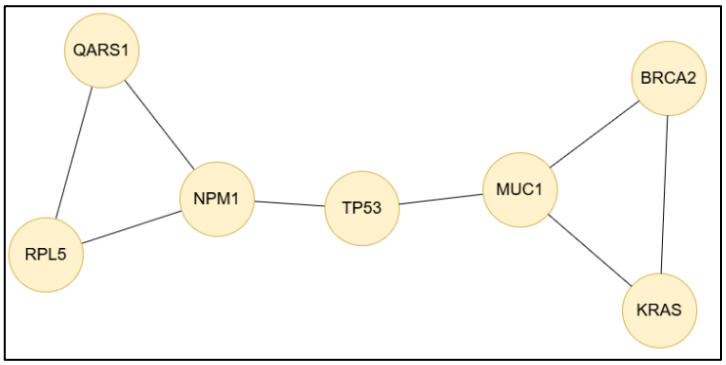
Setelah *seed LCD* terbentuk, proses dilanjutkan ke tahap *expansion phase* seperti ditunjukkan pada Gambar 3.10. Pada tahap ini, seed awal digunakan sebagai komunitas sementara C. Tetangga dari komunitas C diambil sebagai simpul kandidat, kemudian

masing-masing simpul dievaluasi menggunakan tiga fungsi lokal, yaitu fungsi kebugaran komunitas $f(C)$, nilai pengaruh simpul $\omega(v)$, dan derajat afiliasi simpul terhadap seed $F(v, s)$. Apabila terdapat simpul kandidat yang meningkatkan kualitas komunitas, simpul tersebut ditambahkan ke dalam komunitas C, kemudian daftar tetangga diperbarui dan proses evaluasi diulang. Mekanisme ini berlangsung secara iteratif hingga tidak ditemukan lagi simpul yang memenuhi kriteria penambahan. Tahap ekspansi ini memperlihatkan karakteristik utama LCD, di mana keputusan penambahan simpul sepenuhnya bergantung pada informasi lokal komunitas dan tetangganya, sehingga batas komunitas ditentukan melalui proses adaptasi lokal.



Gambar 3. 10 Flowchart Expansion Phase berbasis lokal

Sebagai ilustrasi proses analisis dan perhitungan algoritma deteksi komunitas overlapping, digunakan sebuah jaringan interaksi protein sederhana yang terdiri dari 7 node dan 8 edge. Jaringan ini digunakan sebagai data sampel untuk menjelaskan tahapan algoritma secara manual dan terstruktur. Ilustrasi jaringan interaksi protein data sampel ditunjukkan pada Gambar 3.11. Gambar tersebut memperlihatkan struktur hubungan antar protein dalam bentuk graf, di mana setiap simpul merepresentasikan protein dan setiap sisi menunjukkan adanya interaksi antar protein. Struktur graf ini digunakan sebagai dasar dalam menjelaskan proses pembentukan komunitas secara bertahap.



Gambar 3. 11 Ilustrasi jaringan interaksi protein

Detail relasi antar protein yang membentuk jaringan tersebut disajikan pada Tabel 3.4. Tabel ini menunjukkan pasangan protein yang saling berinteraksi, yang kemudian direpresentasikan sebagai sisi dalam graf pada Gambar 3.11.

Tabel 3. 4 Relasi antar protein

No	Node 1	Node 2
1	QARS1	NPM1
2	QARS1	RPL5
3	NPM1	RPL5
4	NPM1	TP53
5	TP53	MUC1
6	MUC1	BRCA2
7	MUC1	KRAS
8	BRCA2	KRAS

Jaringan interaksi protein ini terdiri dari tujuh node yang masing-masing merepresentasikan gen atau protein yang berperan dalam kanker payudara. Untuk memudahkan proses perhitungan dan penjelasan algoritma, setiap node diberikan simbol huruf. Daftar node beserta pemisalannya ditampilkan pada Tabel 3.5.

Tabel 3. 5 Daftar node dan pemisalan

Nama Protein	Simbol
QARS1	A
RPL5	B
NPM1	C
TP53	D
MUC1	E
BRCA2	F
KRAS	G

Pada tahap ini ditetapkan parameter dasar algoritma GLOD untuk mendukung proses pemodelan. Parameter yang digunakan meliputi nilai α (alpha) sebagai pengatur keseimbangan antara kepadatan internal dan konektivitas eksternal komunitas, serta ambang batas penggabungan komunitas berbasis koefisien Jaccard. Nilai α yang digunakan dalam ilustrasi ini adalah 0,5, 1,0, dan 1,5. Selain itu, ambang batas fitness gain ditetapkan bernilai nol atau tidak negatif sebagai syarat minimal penambahan simpul pada tahap ekspansi. Pada tahap merging, dua komunitas akan digabungkan apabila nilai koefisien Jaccard melebihi 0,333, yang menunjukkan tingkat tumpang tindih yang signifikan.

3.4.1 Seeding Phase

Pada algoritma GLOD, *seeding phase* merupakan tahap awal yang bertujuan menentukan titik awal komunitas secara lokal sebelum proses ekspansi dilakukan. Seeding tidak dilakukan dengan menjadikan setiap simpul sebagai seed utama secara langsung, melainkan melalui pembentukan komunitas awal (*rough communities*) yang merepresentasikan wilayah jaringan dengan kepadatan lokal yang tinggi. Pendekatan ini digunakan untuk memastikan bahwa ekspansi komunitas dimulai dari struktur lokal yang kuat dan tidak bergantung pada simpul dengan keterhubungan yang lemah.

Pada tahap *seeding phase*, algoritma membentuk himpunan NL (*nodes without labels*) yang berisi seluruh simpul dalam jaringan yang belum memiliki label komunitas. Proses seeding dilakukan secara iteratif dengan mengevaluasi setiap simpul dalam himpunan NL sebagai kandidat pusat komunitas awal, bukan sebagai seed final secara langsung, untuk membentuk komunitas awal V_i, yang terdiri atas simpul inti v_i beserta

seluruh tetangganya. Komunitas awal yang potensial umumnya berada pada wilayah graf yang padat dan memiliki keterhubungan internal yang relatif tinggi.

Setiap komunitas awal V_i kemudian diberi skor dengan mempertimbangkan jumlah koneksi simpul pusat, ukuran komunitas, dan banyaknya hubungan internal antar simpul dalam komunitas. Skor ini digunakan sebagai dasar pemeringkatan komunitas awal, di mana komunitas dengan nilai tertinggi dipilih sebagai seed untuk tahap ekspansi selanjutnya.

Berdasarkan informasi pada Tabel 3.5, seluruh simpul dalam jaringan selanjutnya dimasukkan ke dalam himpunan NL (*Nodes without Labels*), yaitu himpunan simpul yang belum memiliki label komunitas. Pada tahap *seeding*, simpul belum diberi label komunitas sehingga seluruh simpul tetap berada dalam himpunan NL. Kondisi awal himpunan NL pada tahap seeding disajikan pada Tabel 3.6.

Tabel 3. 6 Node List Awal (NL)

Fase	**Node List (NL)**	**Status**
Seeding	{A, B, C, D, E, F, G}	Semua simpul belum berlabel

Untuk mendukung pembentukan komunitas awal pada tahap *seeding phase*, terlebih dahulu dilakukan identifikasi derajat dan himpunan tetangga dari setiap simpul berdasarkan struktur jaringan pada Gambar 3.11. Himpunan tetangga ini menjadi dasar dalam mengukur kemiripan struktural antar simpul menggunakan Persamaan (2.2) pada Bab II, yaitu dengan menghitung jumlah elemen pada irisan tetangga bersama (*common neighbors*) dari dua simpul yang saling terhubung. Dengan demikian, setiap pasangan simpul yang memiliki jumlah tetangga bersama lebih besar menunjukkan kedekatan struktural yang lebih tinggi dan berpotensi berada dalam komunitas yang sama. Hasil perhitungan nilai kesamaan antar simpul tersebut disajikan secara rinci pada Tabel 3.7.

Tabel 3. 7 Hasil perhitungan kesamaan simpul (vi)

No	**Simpul (v_i)**	**Derajat**	**Tetangga ($N(v_i)$)**
1	A	2	{B, C}
2	B	2	{A, C}
3	C	3	{A, B, D}
4	D	2	{C, E}
5	E	3	{D, F, G}
6	F	2	{E, G}
7	G	2	{E, F}

Setelah simpul pusat v_i terpilih, dibentuk komunitas awal V_i yang terdiri dari simpul inti dan tetangga-tetangga yang memiliki tingkat kemiripan tinggi. Pembentukan komunitas awal ini dilakukan dengan mempertimbangkan tingkat kemiripan struktur antar simpul menggunakan Common Neighbor Similarity (NC) sebagaimana dirumuskan pada Persamaan (2.2). Nilai NC antara dua simpul dihitung berdasarkan jumlah tetangga bersama (irisan) yang dimiliki oleh kedua simpul tersebut, detail ketetanggaan ditunjukan pada Tabel 3.4 dan Gambar 3.11. Nilai NC yang lebih besar menunjukkan tingkat kemiripan struktural yang lebih tinggi. Pada tahap ini, setiap simpul dalam himpunan NL diperlakukan sebagai kandidat pusat komunitas awal (v_i). Untuk setiap pasangan simpul bertetangga (v_i, v_j), dilakukan perhitungan irisan antara himpunan tetangga $N(v_i)$ dan $N(v_j)$. Apabila nilai NC lebih besar dari nol, maka simpul v_j dianggap memiliki kemiripan struktural yang cukup dan dimasukkan sebagai kandidat anggota komunitas awal V_i. Hasil perhitungan *Common Neighbor Similarity* untuk seluruh pasangan simpul disajikan secara rinci pada Tabel 3.8.

Tabel 3. 8 Perhitungan Common Neighbor Similarity

No	Pusat (v_i)	Pasangan (v_i, v_j)	N(v_i)	N(v_j)	N(v_i) ∩ N(v_j)	NC	Status
1	A	(A, B)	{B, C}	{A, C}	{C}	1	Kandidat
2	A	(A, C)	{B, C}	{A, B, D}	{B}	1	Kandidat
3	B	(B, A)	{A, C}	{B, C}	{C}	1	Kandidat
4	B	(B, C)	{A, C}	{A, B, D}	{A}	1	Kandidat
5	C	(C, A)	{A, B, D}	{B, C}	{B}	1	Kandidat
6	C	(C, B)	{A, B, D}	{A, C}	{A}	1	Kandidat
7	C	(C, D)	{A, B, D}	{C, E}	∅	0	Bukan kandidat
8	D	(D, C)	{C, E}	{A, B, D}	∅	0	Bukan kandidat
9	D	(D, E)	{C, E}	{D, F, G}	∅	0	Bukan kandidat
10	E	(E, F)	{D, F, G}	{E, G}	{G}	1	Kandidat
11	E	(E, G)	{D, F, G}	{E, F}	{F}	1	Kandidat
12	F	(F, E)	{E, G}	{D, F, G}	{G}	1	Kandidat
13	F	(F, G)	{E, G}	{E, F}	{E}	1	Kandidat
14	G	(G, E)	{E, F}	{D, F, G}	{F}	1	Kandidat
15	G	(G, F)	{E, F}	{E, G}	{E}	1	Kandidat

Berdasarkan hasil perhitungan *Common Neighbor Similarity* pada Tabel 3.8, setiap simpul pusat v_i membentuk komunitas awal V_i yang terdiri atas simpul inti dan simpul-simpul tetangga yang memiliki nilai NC lebih besar dari nol. Rincian komunitas awal yang terbentuk pada tahap ini disajikan pada Tabel 3.9.

Tabel 3. 9 Rough Communities

Pusat v_i	Komunitas Awal (V_i)	Jumlah Anggota
A	{A, B, C}	3
B	{A, B, C}	3
C	{A, B, C}	3
D	{D}	1
E	{E, F, G}	3
F	{E, F, G}	3
G	{E, F, G}	3

Sebagai contoh, simpul A memiliki dua tetangga, yaitu B dan C, yang masing-masing memiliki satu tetangga bersama dengan simpul A. Oleh karena itu, komunitas awal yang terbentuk dari pusat A adalah V_a = {A, B, C}. Pola yang sama juga terjadi pada simpul B dan C sehingga ketiga simpul tersebut menghasilkan komunitas awal dengan komposisi yang identik. Sebaliknya, simpul D tidak memiliki tetangga dengan nilai *common neighbor* lebih besar dari nol. Kondisi ini menyebabkan komunitas awal yang terbentuk dari simpul D hanya terdiri atas simpul itu sendiri. Pada bagian graf lainnya, simpul E, F, dan G membentuk komunitas awal yang sama, yaitu {E, F, G}, karena ketiga simpul tersebut saling memiliki tetangga bersama.

Setelah seluruh komunitas awal terbentuk sebagaimana disajikan pada Tabel 3.9, langkah selanjutnya pada seeding phase adalah melakukan pemeringkatan komunitas awal untuk menentukan komunitas mana yang paling layak dijadikan seed pada tahap ekspansi. Penilaian komunitas awal dilakukan dengan menghitung skor untuk setiap komunitas awal V_i. Skor ini dihitung berdasarkan jumlah tiga komponen, yaitu jumlah koneksi simpul pusat (degree(v_i)), jumlah simpul dalam komunitas awal (| V_i |), serta jumlah sisi internal yang menghubungkan simpul-simpul di dalam komunitas tersebut, informasi ini didapatkan dari Tabel 3.4 dan Gambar 3.11. Ketiga komponen ini digunakan untuk merepresentasikan

kepadatan dan kekuatan konektivitas lokal dari komunitas awal. Rincian perhitungan skor untuk setiap komunitas awal disajikan pada Tabel 3.10.

Tabel 3. 10 Ranking Score Rough Communities

v_i	V_i	$\|V_i\|$	Degree (v_i)	Degree (v_i)	Edge internal V_i	Internal Edges	Score
A	{A, B, C}	3	{(A,C), (A,B)}	2	{(A,C), (A,B), (B,C)}	3	8
B	{A, B, C}	3	{(B,A), (B,C)}	2	{(A,C), (A,B), (B,C)}	3	8
C	{A, B, C}	3	{(C,A), (C,B), (C,D)}	3	{(A,C), (A,B), (B,C)}	3	9
D	{D}	1	{(D,C), (D,E)}	2	{ }	0	3
E	{E, F, G}	3	{(E,D), (E,F), (E,G)}	3	{(F,G), (E,F), (E,G)}	3	9
F	{E, F, G}	3	{(F,E), (F,G)}	2	{(F,G), (E,F), (E,G)}	3	8
G	{E, F, G}	3	{(G,E), (G,F)}	2	{(F,G), (E,F), (E,G)}	3	8

Berdasarkan Tabel 3.10, terlihat bahwa komunitas awal dengan pusat simpul C dan E memiliki nilai skor tertinggi, yaitu 9. Nilai ini diperoleh karena kedua komunitas tersebut memiliki jumlah koneksi simpul pusat yang tinggi, jumlah anggota komunitas yang relatif besar, serta jumlah hubungan internal yang maksimal. Sebaliknya, komunitas awal dengan pusat simpul D memiliki skor paling rendah. Hal ini disebabkan oleh tidak adanya hubungan internal di dalam komunitas awal tersebut, sehingga komunitas ini tidak merepresentasikan struktur lokal yang padat.

Setelah seluruh komunitas awal diperingkat berdasarkan nilai skornya, algoritma GLOD memilih komunitas awal dengan skor tertinggi sebagai *seed* untuk tahap ekspansi komunitas. Proses pemilihan ini mengikuti prinsip bahwa komunitas dengan struktur lokal paling kuat akan menghasilkan hasil ekspansi yang lebih stabil dan representatif. Hasil seleksi *seed* berdasarkan pemeringkatan skor komunitas awal disajikan pada Tabel 3.11.

Tabel 3. 11 Hasil Pemilihan Seed pada Seeding Phase

v_i	V_i	$\|V_i\|$	Degree (v_i)	Degree (v_i)	Edge internal V_i	Internal Edges	Score
C	{A, B, C}	3	{(C,A), (C,B), (C,D)}	3	{(A,C), (A,B), (B,C)}	3	9
E	{E, F, G}	3	{(E,D), (E,F), (E,G)}	3	{(F,G), (E,F), (E,G)}	3	9

Meskipun pada Tabel 3.11 diperoleh dua seed awal, yaitu komunitas dengan pusat simpul C dan E, pemilihan tersebut tidak dilakukan secara langsung atau simultan sejak awal proses seeding. Algoritma GLOD terlebih dahulu membentuk dan mengevaluasi seluruh komunitas awal yang berasal dari simpul-simpul dalam himpunan nodes without labels (NL). Komunitas awal dengan pusat simpul C teridentifikasi terlebih dahulu sebagai seed karena memiliki skor maksimum. Namun, algoritma tidak menghapus komunitas awal melainkan memberi label untuk setiap anggota yang ada pada komunitas awal dan kembali mencari dengan mempertimbangkan himpunan nodes without labels (NL). Hasil pemeringkatan selanjutnya menunjukkan bahwa komunitas awal dengan pusat simpul E juga memiliki nilai skor tertinggi, sehingga pada tahap ini diperoleh dua seed akhir yang akan digunakan pada tahap ekspansi komunitas, yaitu:

1) $S_1 = \{A, B, C\}$ dengan pusat simpul C,
2) $S_2 = \{E, F, G\}$ dengan pusat simpul E.

Himpunan *seed* ini selanjutnya digunakan sebagai titik awal pada tahap ekspansi komunitas, di mana setiap *seed* akan diperluas secara lokal untuk membentuk komunitas yang lebih lengkap sesuai dengan fungsi *fitness* yang digunakan oleh algoritma GLOD.

3.4.2 Expansion Phase

Berdasarkan hasil seeding phase yang disajikan pada Tabel 3.11, diperoleh dua komunitas awal dengan skor tertinggi, yaitu komunitas {A, B, C} dengan simpul pusat C dan komunitas {E, F, G} dengan simpul pusat E. Kedua komunitas ini selanjutnya ditetapkan sebagai seed awal pada proses ekspansi dilakukan melalui perhitungan manual dengan menggunakan tiga variasi parameter alpha (α), yaitu α = 0,5, α = 1,0, dan α = 1,5.

Pada tahap *expansion phase*, komunitas awal yang telah terbentuk dari proses seeding digunakan sebagai komunitas sementara. Seluruh simpul yang bertetangga dengan komunitas tersebut, tetapi belum menjadi anggota, diperlakukan sebagai simpul kandidat untuk ditambahkan. Setiap simpul kandidat kemudian dievaluasi secara iteratif dengan mempertimbangkan empat kriteria utama, yaitu nilai *fitness komunitas* (Persamaan (2.3)), *fitness gain simpul* (Persamaan (2.4)), *neighbor similarity* (Persamaan (2.5)), dan *influence function* (Persamaan (2.6)). Ketiga fungsi ini bertujuan untuk menilai apakah penambahan simpul akan meningkatkan kualitas struktur komunitas. Secara umum, simpul hanya akan ditambahkan apabila penambahannya meningkatkan kepadatan internal komunitas dan tidak memperbesar koneksi keluar secara signifikan. Proses ini dilakukan secara iteratif hingga tidak ada lagi simpul yang memenuhi kriteria penambahan.

1. Expansion Phase untuk α = 0,5

1) Ekspansi Seed Pertama {A,B,C}

Komunitas awal pada seed kedua ditetapkan sebagai $C = \{A, B, C\}$ dengan ukuran komunitas $| C | = 3$. Pada tahap awal ini, dilakukan perhitungan jumlah sisi internal dan sisi eksternal komunitas untuk mengevaluasi kualitas struktur komunitas awal, interaksi tiap simpul bisa dilihat pada Tabel 3.4. Hasil perhitungan nilai fitness komunitas awal ditunjukkan pada Tabel 3.12.

Tabel 3. 12 Hasil Perhitungan f(C) Seed {A,B,C}

Sisi Internal	Sisi Eksternal	d_{in}^{C}	d_{out}^{C}	$f(C)$
{(A,B), (A,C), (B,C)}	{(C,D)}	3	1	$f(C) = \frac{3}{(3+1)^{0,5}} = 1.5000$

Berdasarkan Tabel 3.12, komunitas awal {A, B, C} memiliki tiga sisi internal yang menghubungkan seluruh simpul di dalam komunitas, serta satu sisi eksternal yang menghubungkan simpul C dengan simpul D di luar komunitas. Dengan demikian, nilai fitness awal komunitas adalah sebesar 1,5000. Tahap selanjutnya adalah identifikasi *shell*, yaitu himpunan simpul di luar komunitas yang memiliki hubungan langsung dengan simpul-simpul di dalam komunitas. Berdasarkan struktur jaringan (Gambar 3.11), hanya simpul C yang memiliki tetangga di luar komunitas, yaitu simpul D, ditunjukan pada tabel 3.13.

Tabel 3. 13 Identifikasi Shell Seed {A, B, C}

Simpul dalam C	Tetangga di luar C
A	–
B	–
C	D

Berdasarkan Tabel 3.13, diperoleh himpunan $Shell = \{D\}$. Simpul D selanjutnya dievaluasi sebagai kandidat penambahan pada iterasi pertama. Sisi internal maupun eksternal bisa dilihat pada Gambar 3.11.

Tabel 3. 14 Fitness Komunitas Sebelum dan Sesudah Penambahan Simpul D

Komunitas	Sisi Internal	Sisi Eksternal	d_{in}^{C}	d_{out}^{C}	$f(C)$
{A,B,C}	{(A,B), (A,C), (B,C)}	{(C,D)}	3	1	$f(C) = \frac{3}{(3+1)^{0,5}} = 1.5000$
{A,B,C,D}	{(A,B), (A,C), (B,C), (C,D)}	{(D,E)}	4	1	$\frac{4}{(4+1)^{0,5}} = 1.7889$

Berdasarkan Tabel 3.14, penambahan simpul D meningkatkan jumlah sisi internal komunitas dari 3 menjadi 4, sementara jumlah sisi eksternal tetap bernilai 1. Kondisi ini menyebabkan nilai fitness komunitas meningkat dari 1,5000 menjadi 1.7889. Nilai fitness gain simpul D dihitung sebagai selisih antara nilai fitness komunitas setelah dan sebelum penambahan simpul D, ditunjukan pada tabel 3.15.

Tabel 3. 15 Perhitungan Fitness Gain Simpul D

Kandidat	$f(C \cup \{v\})$	$f(C)$	$f(v)$
D	1.5000	1.7889	$1.7889 - 1.5000 = 0.2889$

Karena nilai fitness gain simpul D bernilai positif, maka simpul D layak untuk dievaluasi lebih lanjut menggunakan kriteria neighbor similarity dan influence function. Setelah simpul D dinyatakan layak untuk dievaluasi lebih lanjut karena memiliki nilai fitness gain positif, langkah berikutnya pada tahap ekspansi adalah menghitung nilai *neighbor similarity*, berdasarkan kemiripan struktur ketetanggaannya dengan simpul-simpul di dalam komunitas. Informasi ketetanggaan simpul D ditunjukkan pada Tabel 3.16.

Tabel 3. 16 Informasi Ketetanggaan Simpul D

Keterangan	Himpunan Simpul
Simpul kandidat (v_i)	D
Tetangga langsung $N(D)$	{C, E}
Tetangga orde-2 $N^2(D)$	{A, B, F, G}
Tetangga D di dalam komunitas N_C^D	{C}

Berdasarkan Tabel 3.16, hanya terdapat satu simpul dalam komunitas yang bertetangga langsung dengan simpul D, yaitu simpul C. Oleh karena itu, perhitungan neighbor similarity dilakukan antara simpul kandidat D dan simpul pembanding C. Informasi ketetanggaan simpul C disajikan pada Tabel 3.17.

Tabel 3. 17 Informasi Ketetanggaan Simpul C

Keterangan	Himpunan Simpul
Simpul pembanding (v_j)	C
Tetangga langsung $N(C)$	{A, B, D}
Tetangga orde-2 $N^2(C)$	{E}

Selanjutnya, dilakukan perhitungan irisan ketetanggaan antara simpul D dan simpul C, baik pada ketetanggaan orde-1 maupun orde-2. Hasil perhitungan irisan tersebut ditunjukkan pada Tabel 3.18.

Tabel 3. 18 Hasil Irisan Ketetanggaan Simpul D dan C

Irisan	Hasil
$N(D) \cap N(C)$	$\emptyset$
$N^2(D) \cap N^2(C)$	$\emptyset$

Berdasarkan Tabel 3.18, terlihat bahwa simpul D dan simpul C tidak memiliki tetangga yang sama, baik pada ketetanggaan orde-1 maupun orde-2. Untuk simpul pembanding $v_j = C$, jumlah tetangga langsung simpul C adalah $| N(C) | = 3$, sedangkan

jumlah tetangga orde-2 simpul C adalah $| N^2(C) |= 1$. Nilai Term$_1$ dihitung berdasarkan ketetanggaan orde-1 sebagai berikut:

$$\text{Term}_1 = \frac{| N(D) \cap N(C) | +1}{| N(C) |} = \frac{0+1}{3} = 0{,}3333$$

Selanjutnya, nilai Term$_2$ dihitung berdasarkan ketetanggaan orde-2 sebagai berikut:

$$\text{Term}_2 = \frac{| N^2(D) \cap N^2(C) | +1}{| N^2(C) |} = \frac{0+1}{1} = 1{,}0000$$

Kedua nilai term tersebut kemudian dikombinasikan untuk memperoleh nilai neighbor similarity simpul D terhadap komunitas $C = \{A, B, C\}$ menggunakan Persamaan (2.5), yaitu:

$$\omega(D) = \frac{\text{Term}_1 + 0{,}1 \times \text{Term}_2}{1{,}1} = \frac{0{,}3333 + 0{,}1 \times 1{,}0000}{1{,}1} = 0{,}3939$$

Karena hanya terdapat satu simpul pembanding $v_j \in N_C^D$, maka nilai maksimum neighbor $\omega(D) = 0{,}3939$. Setelah nilai neighbor similarity simpul D diperoleh sebesar $\omega(D) = 0{,}3939$, langkah selanjutnya pada tahap ekspansi adalah menghitung nilai *influence function*. Tujuan dari perhitungan ini adalah mengukur pengaruh struktural seed awal $S = \{A, B, C\}$ terhadap simpul kandidat D, dengan mempertimbangkan keterhubungan langsung antara simpul D dan simpul-simpul dalam seed awal. Informasi keterhubungan simpul D terhadap seed awal disajikan pada Tabel 3.19.

Tabel 3. 19 Perhitungan Influence Function Simpul D

Keterangan	**Nilai**
Seed awal S	{A, B, C}
Tetangga langsung simpul D $N(D)$	{C, E}
Irisan $N(D) \cap S$	{C}
Influence Function $F(D, S)$	1/3 = 0,3333

Setelah seluruh kriteria dievaluasi yaitu fitness gain, neighbor similarity, dan influence function, ringkasannya disajikan pada Tabel 3.20.

Tabel 3. 20 Ringkasan Evaluasi Kriteria Ekspansi Simpul D

Kriteria Evaluasi	**Nilai**	**Keterangan**
Fitness gain $f(D)$	0.2889	Bernilai positif (>0)
Neighbor similarity $\omega(D)$	0,3939	Memenuhi kriteria kemiripan
Influence function $F(D, S)$	0,3333	Terdapat pengaruh dari seed awal

Berdasarkan Tabel 3.20, simpul D memenuhi kriteria ekspansi sehingga ditambahkan ke dalam komunitas, yang semula C={A,B,C} menjadi C={A,B,C,D}. Selanjutnya dilakukan iterasi lanjutan untuk mengevaluasi simpul di luar komunitas yang memiliki keterhubungan langsung. Pada iterasi kedua, simpul D terhubung dengan simpul E sehingga *shell* yang terbentuk adalah {E}. Proses evaluasi dilakukan sama seperti pada iterasi pertama, jika fungsi gain tetap positif, komunitas yang terhubung dengan komunitas yang terbentuk akan diperiksa lebih lanjut. Misalnya, karena E terhubung dengan F dan G, iterasi setelah E akan mengevaluasi F dan G. Hasil evaluasi dari iterasi pertama hingga selesai ditampilkan pada tabel 3.21 berikut:

Tabel 3. 21 Hasil Expansion Phase pada Seed {A, B, C} (α = 0,5)

Iterasi	Kandidat	f(v)	ω(v_i)	F(v,s)	Status	C
1	D	0.2889	0.3939	0.3333	Ditambahkan	A, B, C
2	E	0.1010	0.4773	0.2500	Ditambahkan	A, B, C, D
3	F	0.2315	0.6970	0.2000	Ditambahkan	A, B, C, D, E
4	G	0.7071	1.0909	0.3333	Ditambahkan	A, B, C, D, E, F
5	-	-	-	-	Tidak ada kandidat → STOP	A, B, C, D, E, F, G

Setelah menjalankan lima iterasi, pada iterasi terakhir tidak ada lagi simpul yang tersisa untuk dievaluasi. Semua simpul telah diperiksa, sehingga proses iterasi berhenti. Pada ronde pertama ini, komunitas awal C = {A, B, C} berkembang menjadi C = {A, B, C, D, E, F, G}.

2) Ekspansi Seed Kedua {E,F,G}

Komunitas awal pada seed kedua ditetapkan sebagai $C = \{E, F, G\}$ dengan ukuran komunitas $|\ C\ | = 3$. Pada tahap awal ini, dilakukan perhitungan jumlah sisi internal dan sisi eksternal komunitas untuk mengevaluasi kualitas struktur komunitas awal. Hasil perhitungan nilai fitness komunitas awal ditunjukkan pada Tabel 3.22.

Tabel 3. 22 Hasil Perhitungan f(C)

Sisi Internal	Sisi Eksternal	d_{in}^{C}	d_{out}^{C}	$f(C)$
{(E,F), (E,G), (F,G)}	{(E,D)}	3	1	$f(C) = \frac{3}{(3+1)^{0,5}} = 1.5000$

Berdasarkan Tabel 3.22, komunitas awal $\{E, F, G\}$ memiliki tiga sisi internal yang menghubungkan seluruh simpul di dalam komunitas, serta satu sisi eksternal yang menghubungkan simpul E dengan simpul D di luar komunitas. Dengan demikian, nilai fitness awal komunitas adalah sebesar 1,5000. Tahap selanjutnya adalah identifikasi *shell*, yaitu himpunan simpul di luar komunitas yang memiliki hubungan langsung dengan simpul-simpul di dalam komunitas. Berdasarkan struktur jaringan, hanya simpul E yang memiliki tetangga di luar komunitas, yaitu simpul D.

Tabel 3. 23 Identifikasi Shell Seed {E, F, G}

Simpul dalam C	Tetangga di luar C
E	D
F	–
G	-

Berdasarkan Tabel 3.23, diperoleh himpunan $Shell = \{D\}$. Simpul D selanjutnya dievaluasi sebagai kandidat penambahan pada iterasi pertama dengan menghitung fungsi gain fitness setelah dan sebelum D ditambahkan ke komunitas, bisa dilihat pada tabel 3.24.

Tabel 3. 24 Fitness Komunitas Sebelum dan Sesudah Penambahan Simpul D

Komunitas	Sisi Internal	Sisi Eksternal	d_{in}^{C}	d_{out}^{C}	$f(C)$
{(E,D)}	{(E, F), (E, G), (F, G)}	{(D, E)}	3	1	$f(C) = \frac{3}{(3+1)^{0,5}} = 1.5000$
{D,E,F,G}	{(E,F), (E,G), (F,G), (D,E)}	{(D,C)}	4	1	$\frac{4}{(4+1)^{0,5}} = 1.7889$

Berdasarkan Tabel 3.23, penambahan simpul D meningkatkan jumlah sisi internal komunitas dari 3 menjadi 4, sementara jumlah sisi eksternal tetap bernilai 1. Kondisi ini menyebabkan nilai fitness komunitas meningkat dari 1,5000 menjadi 1.7889. Nilai fitness

gain simpul D dihitung sebagai selisih antara nilai fitness komunitas setelah dan sebelum penambahan simpul D, ditunjukan pada tabel 3.25 sebagai berikut:

Tabel 3. 25 Perhitungan Fitness Gain Simpul D

Kandidat	$f(C \cup \{v\})$	$f(C)$	$f(v)$
D	1.5000	1.7889	1.7889−1.5000=0.2889

Karena nilai fitness gain simpul D bernilai positif, maka simpul D layak untuk dievaluasi lebih lanjut menggunakan kriteria neighbor similarity dan influence function. Setelah simpul D dinyatakan layak untuk dievaluasi lebih lanjut karena memiliki nilai fitness gain positif, langkah berikutnya pada tahap ekspansi adalah menghitung nilai *neighbor similarity*, berdasarkan kemiripan struktur ketetanggaannya dengan simpul-simpul di dalam komunitas. Informasi ketetanggaan simpul D ditunjukkan pada Tabel 3.26

Tabel 3. 26 Informasi Ketetanggaan Simpul D

Keterangan	Himpunan Simpul
Simpul kandidat (v_i)	D
Tetangga langsung $N(D)$	{C, E}
Tetangga orde-2 $N^2(D)$	{A, B, F, G}
Tetangga D di dalam komunitas N_C^D	{E}

Berdasarkan Tabel 3.26, hanya terdapat satu simpul dalam komunitas yang bertetangga langsung dengan simpul D, yaitu simpul E. Oleh karena itu, perhitungan neighbor similarity dilakukan antara simpul kandidat D dan simpul pembanding E. Informasi ketetanggaan simpul E disajikan pada Tabel 3.27.

Tabel 3. 27 Informasi Ketetanggaan Simpul E

Keterangan	Himpunan Simpul
Simpul pembanding (v_j)	E
Tetangga langsung $N(E)$	{D, F, G}
Tetangga orde-2 $N^2(E)$	{C}

Selanjutnya, dilakukan perhitungan irisan ketetanggaan antara simpul D dan simpul E, baik pada ketetanggaan orde-1 maupun orde-2. Hasil perhitungan irisan tersebut ditunjukkan pada Tabel 3.28.

Tabel 3. 28 Hasil Irisan Ketetanggaan Simpul D dan E

Irisan	Hasil
$N(D) \cap N(E)$	∅
$N^2(D) \cap N^2(E)$	∅

Berdasarkan Tabel 3.28, terlihat bahwa simpul D dan simpul E tidak memiliki tetangga yang sama, baik pada ketetanggaan orde-1 maupun orde-2. Untuk simpul pembanding $v_j = E$, jumlah tetangga langsung simpul E adalah $| N(E) | = 3$, sedangkan jumlah tetangga orde-2 simpul E adalah $| N^2(E) | = 1$. Nilai Term$_1$ dihitung berdasarkan ketetanggaan orde-1 sebagai berikut:

$$\text{Term}_1 = \frac{| N(D) \cap N(E) | + 1}{| N(E) |} = \frac{0+1}{3} = 0{,}3333$$

Selanjutnya, nilai Term$_2$ dihitung berdasarkan ketetanggaan orde-2 sebagai berikut:

$$\text{Term}_2 = \frac{| N^2(D) \cap N^2(E) | + 1}{| N^2(E) |} = \frac{0+1}{1} = 1{,}0000$$

Kedua nilai term tersebut kemudian dikombinasikan untuk memperoleh nilai neighbor similarity simpul D terhadap komunitas $C = \{A, B, C\}$ menggunakan Persamaan (2.5), yaitu:

$$\omega(D) = \frac{\text{Term}_1 + 0{,}1 \times \text{Term}_2}{1{,}1} = \frac{0{,}3333 + 0{,}1 \times 1{,}0000}{1{,}1} = 0{,}3939$$

Karena hanya terdapat satu simpul pembanding $v_j \in N_C^D$, maka nilai maksimum neighbor $\omega(D) = 0{,}3939$. Setelah nilai neighbor similarity simpul D diperoleh sebesar $\omega(D) = 0{,}3939$, langkah selanjutnya pada tahap ekspansi adalah menghitung nilai *influence function*. Tujuan dari perhitungan ini adalah mengukur pengaruh struktural seed awal $S = \{A, B, C\}$ terhadap simpul kandidat D, dengan mempertimbangkan keterhubungan langsung antara simpul D dan simpul-simpul dalam seed awal. Informasi keterhubungan simpul D terhadap seed awal disajikan pada Tabel 3.29.

Tabel 3. 29 Perhitungan Influence Function Simpul D

Keterangan	**Nilai**
Seed awal S	{E, F, G}
Tetangga langsung simpul D $N(D)$	{C, E}
Irisan $N(D) \cap S$	{E}
Influence Function $F(D, S)$	1/3 = 0,3333

Setelah seluruh kriteria dievaluasi yaitu fitness gain, neighbor similarity, dan influence function, ringkasannya disajikan pada Tabel 3.30.

Tabel 3. 30 Ringkasan Evaluasi Kriteria Ekspansi Simpul D

Kriteria Evaluasi	**Nilai**	**Keterangan**
Fitness gain $f(D)$	0.2889	Bernilai positif (>0)
Neighbor similarity $\omega(D)$	0,3939	Memenuhi kriteria kemiripan
Influence function $F(D, S)$	0,3333	Terdapat pengaruh dari seed awal

Berdasarkan Tabel 3.30, simpul D memenuhi seluruh kriteria ekspansi, sehingga layak untuk ditambahkan ke dalam komunitas. Dengan penambahan simpul D, komunitas awal $C = \{E, F, G\}$ diperluas menjadi $C = \{D, E, F, G\}$. Setelah simpul D berhasil ditambahkan ke komunitas, proses iterasi berikutnya dilakukan untuk mencari simpul tambahan yang dapat meningkatkan kualitas komunitas. Pada iterasi kedua, simpul-simpul di luar komunitas yang memiliki keterhubungan langsung dengan komunitas, dievaluasi. Berdasarkan struktur jaringan, simpul D memiliki keterhubungan langsung dengan simpul C, sehingga himpunan shell pada iterasi ini adalah Shell={C}. Selanjutnya jika fungsi gain yang dihasilkan tidak negatif, untuk iterasi selanjutnya karena C terhubung dengan A dan B, iterasi setelah C adalah A dan B. Hasil evaluasi dari iterasi pertama hingga selesai ditampilkan pada tabel 2.31 berikut:

Tabel 3. 31 Hasil Iterasi Expansion Phase pada Seed {E, F, G} (α = 0,5)

Iterasi	**Kandidat**	**f(v)**	**ω(v_i)**	**F(v,s)**	**Status**	**C**
1	D	0.2889	0.3939	0.3333	Ditambahkan	E, F, G
2	C	0.1010	0.4773	0.2500	Ditambahkan	D, E, F, G
3	A	0.2315	0.6667	0.2000	Ditambahkan	C, D, E, F, G
4	B	0.7071	1.0909	0.3333	Ditambahkan	A, C, D, E, F, G
5	-	-	-	-	Tidak ada kandidat → STOP	A, B, C, D, E, F, G

Penerapan algoritma GLOD dengan parameter α = 0,5 menunjukkan bahwa komunitas yang terbentuk pada ronde pertama dan kedua bersifat sama. Pada masing-masing ronde, proses ekspansi diawali dari *seed* C = {A, B, C} dan C = {E, F, G}, kemudian berkembang secara bertahap hingga mencakup seluruh simpul jaringan dalam lima iterasi. Proses ekspansi berhenti setelah tidak ditemukan lagi simpul yang memenuhi kriteria penambahan, sehingga komunitas akhir yang dihasilkan adalah C = {A, B, C, D, E, F, G}. Konsistensi hasil ini mengindikasikan bahwa nilai α yang relatif kecil memberikan kelonggaran terhadap keterhubungan eksternal, sehingga mendorong terbentuknya satu komunitas besar dalam jaringan.

2. Expansion Phase untuk α = 1,0

1) Ekspansi Seed Pertama {A,B,C}

Komunitas awal pada seed kedua ditetapkan sebagai $C = \{A, B, C\}$ dengan ukuran komunitas $| C | = 3$. Pada tahap awal ini, dilakukan perhitungan jumlah sisi internal dan sisi eksternal komunitas untuk mengevaluasi kualitas struktur komunitas awal, interaksi tiap simpul bisa dilihat pada Tabel 3.4. Hasil perhitungan nilai fitness komunitas awal ditunjukkan pada Tabel 3.32.

Tabel 3. 32 Hasil Perhitungan f(C) Seed {A,B,C}

Sisi Internal	Sisi Eksternal	d_{in}^{C}	d_{out}^{C}	$f(C)$
{(A,B), (A,C), (B,C)}	{(C,D)}	3	1	$f(C) = \frac{3}{(3+1)^1} = 0{,}7500$

Berdasarkan Tabel 3.32, komunitas awal {A, B, C} memiliki tiga sisi internal yang menghubungkan seluruh simpul di dalam komunitas, serta satu sisi eksternal yang menghubungkan simpul C dengan simpul D di luar komunitas. Dengan demikian, nilai fitness awal komunitas adalah sebesar 0,7500. Tahap selanjutnya adalah identifikasi *shell*, yaitu himpunan simpul di luar komunitas yang memiliki hubungan langsung dengan simpul-simpul di dalam komunitas. Berdasarkan struktur jaringan, hanya simpul C yang memiliki tetangga di luar komunitas, yaitu simpul D. Interaksi eksternal dari komunitas disajikan pada tabel 3.33.

Tabel 3. 33 Identifikasi Shell Seed {A, B, C}

Simpul dalam C	Tetangga di luar C
A	–
B	–
C	D

Berdasarkan Tabel 3.33, diperoleh himpunan $Shell = \{D\}$. Simpul D selanjutnya dievaluasi sebagai kandidat penambahan pada iterasi pertama.

Tabel 3. 34 Fitness Komunitas Sebelum dan Sesudah Penambahan Simpul D

Komunitas	Sisi Internal	Sisi Eksternal	d_{in}^{C}	d_{out}^{C}	$f(C)$
{A,B,C}	{(A,B), (A,C), (B,C)}	{(C,D)}	3	1	0,7500
{A,B,C,D}	{(A,B), (A,C), (B,C), (C,D)}	{(D,E)}	4	1	$\frac{4}{5} = 0{,}8000$

Evaluasi penambahan simpul D dilakukan dengan membandingkan nilai *fitness* komunitas sebelum dan sesudah penambahan, sebagaimana disajikan pada Tabel 3.34. Hasil perbandingan menunjukkan bahwa jumlah sisi internal meningkat dari 3 menjadi 4, sementara jumlah sisi eksternal tetap bernilai 1. Kondisi ini menyebabkan nilai *fitness*

komunitas meningkat dari 0,7500 menjadi 0,8000. Selisih nilai *fitness* tersebut menghasilkan *fitness gain* sebesar 0,0500, sebagaimana ditunjukkan pada Tabel 3.35.

Tabel 3. 35 Perhitungan Fitness Gain Simpul D

Kandidat	$f(C \cup \{v\})$	$f(C)$	$f(v)$
D	0,8000	0,7500	0,8000 – 0,7500 = 0,0500

Karena nilai fitness gain simpul D bernilai positif, maka simpul D layak untuk dievaluasi lebih lanjut menggunakan kriteria neighbor similarity dan influence function. Setelah simpul D dinyatakan layak untuk dievaluasi lebih lanjut karena memiliki nilai fitness gain positif, langkah berikutnya pada tahap ekspansi adalah menghitung nilai *neighbor similarity*, berdasarkan kemiripan struktur ketetanggaannya dengan simpul-simpul di dalam komunitas. Informasi ketetanggaan simpul D ditunjukkan pada Tabel 3.36.

Tabel 3. 36 Informasi Ketetanggaan Simpul D

Keterangan	**Himpunan Simpul**
Simpul kandidat (v_i)	D
Tetangga langsung $N(D)$	{C, E}
Tetangga orde-2 $N^2(D)$	{A, B, F, G}
Tetangga D di dalam komunitas N_C^D	{C}

Berdasarkan Tabel 3.36, hanya terdapat satu simpul dalam komunitas yang bertetangga langsung dengan simpul D, yaitu simpul C. Oleh karena itu, perhitungan neighbor similarity dilakukan antara simpul kandidat D dan simpul pembanding C. Informasi ketetanggaan simpul C disajikan pada Tabel 3.37.

Tabel 3. 37 Informasi Ketetanggaan Simpul C

Keterangan	**Himpunan Simpul**
Simpul pembanding (v_j)	C
Tetangga langsung $N(C)$	{A, B, D}
Tetangga orde-2 $N^2(C)$	{E}

Selanjutnya, dilakukan perhitungan irisan ketetanggaan antara simpul D dan simpul C pada ketetanggaan orde-1 dan orde-2. Hasil irisan tersebut ditunjukkan pada Tabel 3.38.

Tabel 3. 38 Hasil Irisan Ketetanggaan Simpul D dan C

Irisan	**Hasil**
$N(D) \cap N(C)$	$\emptyset$
$N^2(D) \cap N^2(C)$	$\emptyset$

Berdasarkan Tabel 3.38, terlihat bahwa simpul D dan simpul C tidak memiliki tetangga yang sama, baik pada ketetanggaan orde-1 maupun orde-2. Untuk simpul pembanding $v_j = C$, jumlah tetangga langsung simpul C adalah $| N(C) | = 3$, sedangkan jumlah tetangga orde-2 simpul C adalah $| N^2(C) | = 1$. Nilai Term$_1$ dihitung berdasarkan ketetanggaan orde-1 sebagai berikut:

$$\text{Term}_1 = \frac{| N(D) \cap N(C) | + 1}{| N(C) |} = \frac{0+1}{3} = 0{,}3333$$

Selanjutnya, nilai Term$_2$ dihitung berdasarkan ketetanggaan orde-2 sebagai berikut:

$$\text{Term}_2 = \frac{| N^2(D) \cap N^2(C) | + 1}{| N^2(C) |} = \frac{0+1}{1} = 1{,}0000$$

Kedua nilai term tersebut kemudian dikombinasikan untuk memperoleh nilai neighbor similarity simpul D terhadap komunitas $C = \{A, B, C\}$ menggunakan Persamaan (4), yaitu:

$$\omega(D) = \frac{\text{Term}_1 + 0{,}1 \times \text{Term}_2}{1{,}1} = \frac{0{,}3333 + 0{,}1 \times 1{,}0000}{1{,}1} = 0{,}3939$$

Karena hanya terdapat satu simpul pembanding $v_j \in N_C^D$, maka nilai maksimum neighbor $\omega(D) = 0{,}3939$. Setelah nilai neighbor similarity simpul D diperoleh sebesar $\omega(D) = 0{,}3939$, langkah selanjutnya pada tahap ekspansi adalah menghitung nilai *influence function*. Tujuan dari perhitungan ini adalah mengukur pengaruh struktural seed awal $S = \{A, B, C\}$terhadap simpul kandidat D, dengan mempertimbangkan keterhubungan langsung antara simpul D dan simpul-simpul dalam seed awal. Informasi keterhubungan simpul D terhadap seed awal disajikan pada Tabel 3.39.

Tabel 3. 39 Perhitungan Influence Function Simpul D

Keterangan	**Nilai**
Seed awal S	{A, B, C}
Tetangga langsung simpul D $N(D)$	{C, E}
Irisan $N(D) \cap S$	{C}
Influence Function $F(D, S)$	1/3 = 0,3333

Setelah seluruh kriteria dievaluasi yaitu fitness gain, neighbor similarity, dan influence function, ringkasannya disajikan pada Tabel 3.40.

Tabel 3. 40 Ringkasan Evaluasi Kriteria Ekspansi Simpul D

Kriteria Evaluasi	**Nilai**	**Keterangan**
Fitness gain $f(D)$	0,0500	Bernilai positif (>0)
Neighbor similarity $\omega(D)$	0,3939	Memenuhi kriteria kemiripan
Influence function $F(D, S)$	0,3333	Terdapat pengaruh dari seed awal

Berdasarkan Tabel 3.40, simpul D memenuhi seluruh kriteria ekspansi, sehingga layak untuk ditambahkan ke dalam komunitas. Dengan penambahan simpul D, komunitas awal $C = \{A, B, C\}$ diperluas menjadi $C = \{A, B, C, D\}$. Setelah simpul D ditambahkan ke dalam komunitas, proses iterasi berikutnya dilakukan untuk mencari simpul lain yang dapat meningkatkan kualitas komunitas. Metode yang digunakan sama seperti sebelumnya, setiap simpul kandidat dievaluasi berdasarkan fitness gain, neighbor similarity, dan influence function. Berdasarkan struktur jaringan, simpul D memiliki hubungan langsung dengan simpul E yang berada di luar komunitas. Oleh karena itu, shell pada iterasi kedua Adalah Shell={E}. Simpul E dievaluasi menggunakan ketiga kriteria, dengan hasil perhitungan disajikan pada Tabel 3.41.

Tabel 3. 41 Hasil Perhitungan evaluasi simpul E

Fitness Gain	**Neighbor Similarity**	**Influence Function**
-0,0857	0.4773	0.2500

Nilai fitness gain simpul E negatif $(-0{,}0857)$, sehingga simpul ini tidak memenuhi kriteria penambahan ke komunitas. Sesuai stopping condition, proses ekspansi dihentikan ketika tidak ada simpul kandidat dengan fitness gain positif, mencegah over-expansion.

2) Ekspansi Seed Kedua {E,F,G}

Komunitas awal ditetapkan sebagai $C = \{E, F, G\}$ dengan ukuran komunitas $| C | = 3$. Pada tahap ini, dihitung jumlah sisi internal dan sisi eksternal komunitas untuk mengevaluasi kualitas struktur awal komunitas. Hasil perhitungan f(C) ditunjukan pasa tabel 3.42.

Tabel 3. 42 Hasil perhitungan f(C)

Sisi Internal	**Sisi Eksternal**	d_{in}^{C}	d_{out}^{C}	$f(C)$
{(E, F), (E, G), (F, G)}	{(E, D)}	3	1	$f(C) = \frac{3}{(3+1)^1} = 0{,}7500$

Selanjutnya, dilakukan identifikasi *shell*, yaitu himpunan simpul di luar komunitas yang memiliki hubungan langsung dengan simpul di dalam komunitas. Berdasarkan struktur jaringan, simpul Ememiliki keterhubungan dengan simpul D, sedangkan simpul Fdan Gtidak memiliki tetangga di luar komunitas. Dengan demikian, diperoleh himpunan *shell* sebagai berikut Shell $= \{D\}$. Tahap berikutnya adalah mengevaluasi simpul kandidat Duntuk menentukan apakah simpul tersebut layak ditambahkan ke dalam komunitas C. Hasil perhitungan *fitness* komunitas sebelum dan sesudah penambahan simpul D disajikan pada Tabel 3.43.

Tabel 3. 43 Perhitungan Fitness Setelah Penambahan Simpul D

Komunitas	**Sisi Internal**	**Sisi Eksternal**	$\mathbf{d_{in}^{C}}$	$\mathbf{d_{out}^{C}}$	$\mathbf{f(C)}$
{E,F,G}	{(E,F), (E,G), (F,G)}	{(E,D)}	3	1	$f(C) = \frac{3}{(3+1)^1} = 0{,}7500$
{D,E,F,G}	{(D,E), (E,F), (E,G), (F,G)}	{(D,B)}	4	1	$f(C) = \frac{4}{(4+1)^1} = 0{,}8000$

Berdasarkan hasil perhitungan tersebut, penambahan simpul D meningkatkan jumlah sisi internal menjadi empat, dengan jumlah sisi eksternal tetap satu, sehingga nilai *fitness* komunitas meningkat menjadi 0,8000. Selanjutnya, nilai *fitness gain* simpul D dihitung sebagai selisih antara nilai *fitness* setelah dan sebelum penambahan simpul. Hasil perhitungan *fitness gain* ini ditampilkan pada Tabel 3.44.

Tabel 3. 44 Perhitungan Fitness Gain Simpul D

Kandidat	$\boldsymbol{f(C \cup \{v\})}$	$\boldsymbol{f(C)}$	**Perhitungan** $\boldsymbol{f(v)}$
D	0,8000	0,7500	$0{,}8000 - 0{,}7500 = 0{,}0500$

Setelah simpul D dinyatakan layak dievaluasi karena memiliki nilai *fitness gain* positif, langkah selanjutnya adalah menghitung nilai *neighbor similarity* untuk memastikan kesesuaian simpul tersebut dengan komunitas $C = \{E, F, G\}$. Perhitungan ini dilakukan berdasarkan struktur ketetanggaan simpul kandidat terhadap simpul-simpul di dalam komunitas. Informasi ketetanggaan simpul D disajikan pada Tabel 3.45.

Tabel 3. 45 Informasi Ketetanggaan Simpul D

Keterangan	**Himpunan Simpul**
Simpul kandidat (v_i)	D
Tetangga langsung $N(D)$	{C, E}
Tetangga orde-2 $N^2(D)$	{A, B, F, G}
Tetangga D di dalam komunitas N_C^D	{E}

Karena nilai *fitness gain* simpul D bernilai positif $(f(D) = 0{,}0500 \geq 0)$, maka dilakukan perhitungan *neighbor similarity* antara simpul D dan setiap simpul v_j yang berada

di dalam N_C^D. Pada ontoh ini, hanya terdapat satu simpul, yaitu E. Informasi ketetanggaan simpul E ditunjukkan pada Tabel 3.46.

Tabel 3. 46 Informasi Ketetanggaan Simpul E

Keterangan	Himpunan Simpul
Simpul pembanding (v_j)	E
Tetangga langsung $N(E)$	{D, F, G}
Tetangga orde-2 $N^2(E)$	{C}

Selanjutnya, dihitung irisan ketetanggaan antara simpul D dan E, baik pada tetangga orde-1 maupun orde-2. Hasil perhitungan irisan tersebut disajikan pada Tabel 3.47.

Tabel 3. 47 Hasil Irisan Ketetanggaan Simpul D dan E

Jenis Irisan	Hasil
$N(D) \cap N(E)$	$\emptyset$
$N^2(D) \cap N^2(E)$	$\emptyset$

Berdasarkan hasil irisan ketetanggaan pada Tabel 3.47, terlihat bahwa simpul D dan simpul E tidak memiliki tetangga yang sama, baik pada ketetanggaan orde-1 maupun orde-2. Meskipun demikian, perhitungan *neighbor similarity* tetap memberikan kontribusi nilai melalui mekanisme *smoothing* dengan penambahan konstanta 1 pada pembilang, sebagaimana didefinisikan dalam Persamaan (2.5).

Untuk simpul pembanding $v_j = E$, tetangga langsung simpul E adalah $N(E) = \{D, F, G\}$ dengan jumlah elemen sebanyak 3, sedangkan tetangga orde-2 simpul E adalah $N^2(E) = \{B\}$ dengan jumlah elemen sebanyak 1. Karena tidak terdapat irisan tetangga dengan simpul D, maka jumlah elemen irisan pada ketetanggaan orde-1 dan orde-2 masing-masing bernilai nol. Nilai *term* pertama dihitung berdasarkan rasio jumlah irisan tetangga orde-1 terhadap jumlah tetangga langsung simpul E, dengan penambahan konstanta 1, sehingga diperoleh:

$$\text{Term}_1 = \frac{|\, N(D) \cap N(E) \,| + 1}{|\, N(E) \,|} = \frac{0+1}{3} = 0{,}3333$$

Selanjutnya, nilai *term* kedua dihitung berdasarkan rasio jumlah irisan tetangga orde-2 terhadap jumlah tetangga orde-2 simpul E, juga dengan penambahan konstanta 1, sehingga diperoleh:

$$\text{Term}_2 = \frac{|\, N^2(D) \cap N^2(E) \,| + 1}{|\, N^2(E) \,|} = \frac{0+1}{1} = 1{,}0000$$

Kedua nilai *term* tersebut kemudian dikombinasikan untuk memperoleh nilai *neighbor similarity* simpul D terhadap komunitas C menggunakan Persamaan (2.5), yaitu:

$$\omega(D) = \frac{\text{Term}_1 + 0{,}1 \times \text{Term}_2}{1{,}1} = \frac{0{,}3333 + 0{,}1 \times 1{,}0000}{1{,}1} = 0{,}3939$$

Karena hanya terdapat satu simpul pembanding $v_j \in N_C^D$, maka nilai maksimum *neighbor similarity* simpul D ditetapkan sebesar $\omega(D) = 0{,}3939$. Setelah nilai *neighbor similarity* simpul D diperoleh, langkah selanjutnya pada tahap ekspansi adalah menghitung nilai *influence function*. Pada tahap ini, seed awal yang digunakan adalah komunitas $S = \{E, F, G\}$ dengan ukuran $|\, S \,| = 3$. Informasi keterhubungan simpul kandidat D terhadap seed awal disajikan pada Tabel 3.48.

Tabel 3. 48 Informasi Keterhubungan Simpul D terhadap Seed Awal

Keterangan	Himpunan Simpul
Seed awal S	{E, F, G}
Tetangga langsung simpul D ($N(D)$)	{C, E}
Irisan $N(D) \cap S$	{E}

Berdasarkan Tabel 3.48, terlihat bahwa simpul D hanya memiliki satu tetangga yang berasal dari seed awal, yaitu simpul E. Dengan demikian, nilai *influence function* simpul D terhadap seed awal Sdihitung menggunakan Persamaan (5) sebagai berikut:

$$F(D,S) = \frac{|\, N(D) \cap S \,|}{|\, S \,|} = \frac{1}{3} = 0{,}3333$$

Nilai tersebut menunjukkan bahwa satu dari tiga simpul seed awal memiliki hubungan langsung dengan simpul kandidat D. Meskipun tidak seluruh simpul seed terhubung langsung, nilai *influence function* ini tetap menunjukkan adanya pengaruh struktural seed awal terhadap simpul D.

Tabel 3. 49 Ringkasan Evaluasi Kriteria Ekspansi Simpul D

Kriteria Evaluasi	Nilai	Keterangan
Fitness gain $f(D)$	0,0500	Bernilai positif (> 0)
Neighbor similarity $\omega(D)$	0,3939	Memenuhi kriteria kemiripan
Influence function $F(D,S)$	0,3333	Terdapat pengaruh dari seed awal

Berdasarkan ringkasan evaluasi pada Tabel 3.49, simpul D memenuhi seluruh kriteria pada tahap ekspansi sehingga dinyatakan layak untuk ditambahkan ke dalam komunitas. Dengan penambahan tersebut, komunitas awal C = {E, F, G} berkembang menjadi C = {D, E, F, G}. Setelah simpul D bergabung, dilakukan iterasi kedua untuk mengevaluasi kemungkinan penambahan simpul lain. Berdasarkan struktur jaringan, simpul D memiliki keterhubungan langsung dengan simpul C yang berada di luar komunitas, sehingga himpunan *shell* pada iterasi ini adalah *Shell* = {C}. Hasil evaluasi menunjukkan bahwa nilai *fitness gain* simpul C bernilai negatif, yaitu −0,0857, yang mengindikasikan bahwa penambahan simpul tersebut justru menurunkan kualitas komunitas secara lokal.

Neighbor similarity hanya dilakukan apabila nilai *fitness gain* bernilai non-negatif. Karena simpul C tidak memenuhi kondisi tersebut, maka simpul ini dinyatakan tidak layak untuk diekspansi. Mengacu pada *stopping condition* dalam algoritma *GLOD*, proses ekspansi dihentikan tanpa dilakukan perhitungan *neighbor similarity* dan *influence function* lebih lanjut. Dengan demikian, pada *expansion phase* dengan α = 1,0 diperoleh dua komunitas, yaitu C = {A, B, C, D} dan C = {D, E, F, G}.

3. Expansion Phase untuk α = 1,5

1) Ekspansi Seed Pertama {A,B,C}

Komunitas awal pada seed kedua ditetapkan sebagai $C = \{A, B, C\}$ dengan ukuran komunitas $|\, C \,| = 3$. Pada tahap awal ini, dilakukan perhitungan jumlah sisi internal dan sisi eksternal komunitas untuk mengevaluasi kualitas struktur komunitas awal, interaksi tiap simpul bisa dilihat pada Tabel 3.4. Hasil perhitungan nilai fitness komunitas awal ditunjukkan pada Tabel 3.50.

Tabel 3. 50 Hasil Perhitungan f(C) Seed {A,B,C}

Sisi Internal	Sisi Eksternal	d_{in}^{C}	d_{out}^{C}	$f(C)$
{(A,B), (A,C), (B,C)}	{(C,D)}	3	1	$f(C) = \frac{3}{(3+1)^{1.5}} = 0{,}7500$

Berdasarkan Tabel 3.50, komunitas awal {A, B, C} memiliki tiga sisi internal yang menghubungkan seluruh simpul di dalam komunitas, serta satu sisi eksternal yang menghubungkan simpul C dengan simpul D di luar komunitas. Dengan demikian, nilai fitness awal komunitas adalah sebesar 0,7500. Tahap selanjutnya adalah identifikasi *shell*, yaitu himpunan simpul di luar komunitas yang memiliki hubungan langsung dengan simpul-simpul di dalam komunitas. Berdasarkan struktur jaringan, hanya simpul C yang memiliki tetangga di luar komunitas, yaitu simpul D.

Tabel 3. 51 Identifikasi Shell Seed {A, B, C}

Simpul dalam C	Tetangga di luar C
A	–
B	–
C	D

Berdasarkan Tabel 3.51, diperoleh himpunan $Shell = \{D\}$. Simpul D selanjutnya dievaluasi sebagai kandidat penambahan pada iterasi pertama. Hasil perhitungan setelah penambahan simpul D ditunjukan pada Tabel 3.52.

Tabel 3. 52 Perhitungan Fitness Setelah Penambahan Simpul D

Komunitas	Sisi Internal	Sisi Eksternal	d_{in}^{C}	d_{out}^{C}	$f(C)$
{E,F,G}	{(E,F), (E,G), (F,G)}	{(E,D)}	3	1	$\frac{3}{(3+1)^{1,5}} = 0.3750$
{D,E,F,G}	{(D,E), (E,F), (E,G), (F,G)}	{(D,B)}	4	1	$\frac{4}{(4+1)^{1,5}} = 0.3578$

Berdasarkan hasil perhitungan tersebut, penambahan simpul D meningkatkan jumlah sisi internal menjadi empat, dengan jumlah sisi eksternal tetap satu, sehingga nilai *fitness* komunitas menurun menjadi 0.3578. Selanjutnya, nilai *fitness gain* simpul D dihitung sebagai selisih antara nilai *fitness* setelah dan sebelum penambahan simpul. Hasil perhitungan *fitness gain* ini ditampilkan pada Tabel 3.53.

Tabel 3. 53 Perhitungan Fitness Gain Simpul D

Kandidat	$f(C \cup \{v\})$	$f(C)$	$f(v)$
D	0.3578	0.3750	0.3578 - 0.3750 = -0.0172

Berdasarkan Tabel 3.53, nilai *fitness gain* simpul D bernilai negatif (*f(D) < 0*), sehingga simpul D tidak memenuhi kriteria *fitness* dan tidak dapat ditambahkan ke dalam komunitas. Oleh karena itu, komunitas akhir pada ronde pertama dengan α = 1,5 tetap sama dengan komunitas awal, yaitu C = {A, B, C}.

2) Ekspansi Seed kedua {E,F,G}

Komunitas awal ditetapkan sebagai $C = \{E, F, G\}$ dengan ukuran komunitas $| C | = 3$. Pada tahap ini, dihitung jumlah sisi internal dan sisi eksternal komunitas untuk mengevaluasi kualitas struktur awal komunitas. Hasil perhitungan f(C) ditunjukan pasa tabel 3.54.

Tabel 3. 54 Hasil perhitungan f(C)

Sisi Internal	Sisi Eksternal	d_{in}^{C}	d_{out}^{C}	$f(C)$
{(E, F), (E, G), (F, G)}	{(E, D)}	3	1	$f(C) = \frac{3}{(3+1)^{1,5}} = 0.3750$

Selanjutnya, dilakukan identifikasi *shell*, yaitu himpunan simpul di luar komunitas yang memiliki hubungan langsung dengan simpul di dalam komunitas. Berdasarkan struktur jaringan, simpul Ememiliki keterhubungan dengan simpul D, sedangkan simpul Fdan Gtidak memiliki tetangga di luar komunitas. Dengan demikian, diperoleh himpunan *shell* sebagai berikut Shell $= \{D\}$. Tahap berikutnya adalah mengevaluasi simpul kandidat Duntuk menentukan apakah simpul tersebut layak ditambahkan ke dalam komunitas C. Hasil perhitungan *fitness* komunitas sebelum dan sesudah penambahan simpul D disajikan pada Tabel 3.55.

Tabel 3. 55 Perhitungan Fitness Setelah Penambahan Simpul D

Komunitas	Sisi Internal	Sisi Eksternal	d_{in}^{C}	d_{out}^{C}	$f(C)$
{E,F,G}	{(E,F), (E,G), (F,G)}	{(E,D)}	3	1	$\frac{3}{(3+1)^{1,5}} = 0.3750$
{D,E,F,G}	{(D,E), (E,F), (E,G), (F,G)}	{(D,B)}	4	1	$\frac{4}{(4+1)^{1,5}} = 0.3578$

Berdasarkan hasil perhitungan tersebut, penambahan simpul D meningkatkan jumlah sisi internal menjadi empat, dengan jumlah sisi eksternal tetap satu, sehingga nilai *fitness* komunitas menurun menjadi 0.3578. Selanjutnya, nilai *fitness gain* simpul D dihitung sebagai selisih antara nilai *fitness* setelah dan sebelum penambahan simpul. Hasil perhitungan *fitness gain* ini ditampilkan pada Tabel 3.56.

Tabel 3. 56 Perhitungan Fitness Gain Simpul D

Kandidat	$f(C \cup \{v\})$	$f(C)$	Perhitungan $f(v)$
D	0.3578	0,7500	$0.3578 - 0.3750 = -0.0172$

Berdasarkan Tabel 3.56, nilai *fitness gain* simpul D bernilai negatif (*f(D) < 0*), sehingga simpul tersebut tidak memenuhi kriteria *fitness* dan tidak layak ditambahkan ke dalam komunitas. Dengan demikian, komunitas akhir pada ronde pertama dengan α = 1,5 tetap sama dengan komunitas awal, yaitu C = {E, F, G}. Oleh karena itu, hasil pembentukan komunitas pada α = 1,5 menghasilkan dua komunitas, yakni C = {A, B, C} dan C = {E, F, G}.

3.4.3 Merging Phase

Tahap *merging phase* merupakan tahap akhir dalam algoritma GLOD yang berfungsi untuk menentukan apakah komunitas hasil *expansion phase* perlu digabungkan atau tetap dipertahankan sebagai komunitas yang saling tumpang tindih. Pada tahap ini digunakan Koefisien Jaccard untuk mengukur tingkat kesamaan antara dua komunitas. Koefisien ini menghitung perbandingan antara jumlah simpul yang dimiliki bersama oleh dua komunitas terhadap jumlah total simpul unik dari keduanya. Jika dua komunitas memiliki nilai kesamaan yang tinggi, maka keduanya dianggap merepresentasikan struktur yang hampir sama dan akan digabungkan menjadi satu komunitas yang lebih representatif. Dengan mekanisme ini, algoritma dapat menghindari terbentuknya komunitas-komunitas yang isinya hampir identik sehingga hasil akhir menjadi lebih ringkas dan mudah

diinterpretasikan secara biologis. Semakin besar nilai Jaccard, semakin tinggi tingkat kesamaan struktur kedua komunitas tersebut. Dalam penelitian ini, digunakan ambang batas $J \geq 0{,}3333$ sebagai kriteria penggabungan komunitas. Nilai ambang batas ini didasari oleh tinjauan literatur, yang menyatakan bahwa dua komunitas layak digabungkan apabila tingkat kesamaannya melebihi sepertiga ukuran komunitas terkecil.

Pada tahap *expansion phase* sebelumnya, jumlah komunitas yang dihasilkan bergantung pada nilai parameter alpha yang digunakan. Ketika digunakan nilai $\alpha = 0{,}5$, proses ekspansi menghasilkan satu komunitas. Sementara itu, pada nilai $\alpha = 1{,}0$ dan $\alpha = 1{,}5$, masing-masing dihasilkan dua komunitas yang berbeda. Sebelum menghitung nilai koefisien Jaccard, langkah pertama yang dilakukan adalah mengidentifikasi irisan dan gabungan dari pasangan komunitas hasil *expansion phase*. Identifikasi ini diperlukan karena koefisien Jaccard dihitung berdasarkan perbandingan jumlah anggota pada irisan dan gabungan kedua komunitas. Rincian hasil identifikasi irisan dan gabungan pasangan komunitas ditunjukkan pada Tabel 3.57.

Tabel 3. 57 Informasi Pasangan Komunitas Hasil Expansion Phase

α	**C_1**	**C_2**	**$C_1 \cap C_2$**	**∩**	**$C_1 \cup C_2$**	**∪**
0,5	{A, B, C, D, E, F, G}	{A, B, C, D, E, F, G}	{A, B, C, D, E, F, G}	7	{A, B, C, D, E, F, G}	7
1,0	{A, B, C, D}	{D, E, F, G}	{D}	1	{A, B, C, D, E, F, G}	7
1,5	{A, B, C}	{E, F, G}	∅	0	{A, B, C, E, F, G}	6

Setelah nilai irisan dan gabungan diketahui, selanjutnya dilakukan perhitungan koefisien Jaccard menggunakan Persamaan (2.7). Nilai yang diperoleh kemudian dibandingkan dengan ambang batas $J \geq 0{,}3333$ untuk menentukan apakah pasangan komunitas tersebut layak digabungkan atau tidak. Hasil perhitungan nilai koefisien *Jaccard* dan keputusan penggabungan komunitas ditunjukkan pada Tabel 3.58.

Tabel 3. 58 Hasil Perhitungan Koefisien J dan Keputusan Penggabungan

α	**Perhitungan Jaccard**	**Nilai Jaccard (J)**	**Ambang Batas**	**Keputusan Merge**
0,5	$\frac{7}{7}$	1,0000	0,3333	Digabung
1,0	$\frac{1}{7}$	0,1429	0,3333	Tidak digabung
1,5	$\frac{0}{6}$	0,0000	0,3333	Tidak digabung

Berdasarkan hasil *merging phase*, keputusan akhir komunitas berbeda pada setiap nilai parameter α. Pada α = 0,5, kedua komunitas hasil *expansion phase* memiliki anggota yang identik, yaitu {A, B, C, D, E, F, G}, dengan nilai Jaccard J = 7/7 = 1,0000. Karena nilai tersebut melebihi ambang batas 0,3333, kedua komunitas digabung sehingga hasil akhirnya menjadi satu komunitas besar yang mencakup seluruh simpul jaringan.

Pada α = 1,0, terbentuk dua komunitas, yaitu {A, B, C, D} dan {D, E, F, G}, dengan satu simpul yang sama (D). Nilai Jaccard diperoleh sebesar J = 1/7 = 0,1429, lebih kecil dari 0,3333, sehingga tidak dilakukan penggabungan. Hasil akhirnya tetap dua komunitas yang saling tumpang tindih pada simpul D.

Sementara itu, pada α = 1,5 terbentuk dua komunitas yang sepenuhnya terpisah, yaitu {A, B, C} dan {E, F, G}. Karena tidak ada irisan, nilai Jaccard J = 0/6 = 0,0000 dan berada

di bawah ambang batas, maka kedua komunitas tidak digabung. Dengan demikian, semakin besar nilai α, komunitas yang dihasilkan semakin kecil dan semakin terpisah, sedangkan α yang lebih kecil cenderung menghasilkan komunitas yang lebih besar dan menyatu.

3.5 Evaluation

Selain melakukan evaluasi terhadap hasil deteksi komunitas, pada tahap ini juga dijelaskan secara eksplisit rancangan eksperimen (experimental setup) yang digunakan dalam penelitian. Rancangan eksperimen ini mencakup data yang digunakan, algoritma yang diterapkan, parameter yang diuji, serta skenario pengujian yang dilakukan untuk mengevaluasi performa algoritma GLOD dalam mendeteksi komunitas overlapping.

Dalam penelitian ini, eksperimen dilakukan pada jaringan interaksi protein kanker payudara yang direpresentasikan dalam bentuk graf tidak berarah dan tidak berbobot. Algoritma yang digunakan adalah Local Greedy Extended Dynamic Overlapping Community Detection (GLOD) yang bekerja melalui tiga tahapan utama, yaitu seeding phase, expansion phase, dan merge phase. Parameter utama yang diuji dalam eksperimen ini adalah parameter α (alpha) pada fungsi fitness yang mengontrol proses ekspansi komunitas, serta parameter merging threshold berbasis koefisien Jaccard (J) yang digunakan pada tahap merge phase. Variasi nilai kedua parameter ini bertujuan untuk mengamati pengaruhnya terhadap jumlah komunitas, ukuran komunitas, serta tingkat overlapping antar komunitas yang dihasilkan.

Pendekatan eksperimen yang digunakan adalah kombinasi parameter, di mana setiap nilai parameter α dipasangkan dengan setiap nilai parameter Jaccard threshold. Dengan pendekatan ini, dapat diamati secara sistematis bagaimana interaksi kedua parameter tersebut mempengaruhi hasil deteksi komunitas. Nilai parameter yang digunakan dalam penelitian ini ditentukan berdasarkan rentang yang umum digunakan dalam literatur serta rekomendasi dari penelitian sebelumnya terkait algoritma GLOD. Kombinasi parameter α dan threshold Jaccard yang digunakan dalam setiap skenario eksperimen ditunjukkan pada Tabel 3.59.

Tabel 3. 59 Kombinasi parameter eksperimen

Eksperimen	**α**	**J**	**Eksperimen**	**α**	**J**
Eksperimen 1	0.70	0.20	Eksperimen 9	0.80	0.20
Eksperimen 2	0.70	0.25	Eksperimen 10	0.80	0.25
Eksperimen 3	0.70	0.30	Eksperimen 11	0.80	0.30
Eksperimen 4	0.70	0.33	Eksperimen 12	0.80	0.33
Eksperimen 5	0.75	0.20	Eksperimen 13	0.85	0.20
Eksperimen 6	0.75	0.25	Eksperimen 14	0.85	0.25
Eksperimen 7	0.75	0.30	Eksperimen 15	0.85	0.30
Eksperimen 8	0.75	0.33	Eksperimen 16	0.85	0.33

Proses evaluasi dalam penelitian ini dilakukan berdasarkan rancangan eksperimen (experimental setup) yang telah ditentukan, di mana setiap kombinasi parameter diuji untuk menghasilkan nilai Normalized Node Cut (NNC). Kombinasi parameter yang menghasilkan nilai NNC terbaik selanjutnya dipilih dan digunakan untuk evaluasi lebih lanjut terhadap kualitas komunitas yang dihasilkan oleh algoritma GLOD.

Komunitas yang diperoleh dari setiap skenario percobaan kemudian dianalisis untuk mengetahui kualitas struktur hubungan dalam jaringan serta relevansi biologisnya terhadap

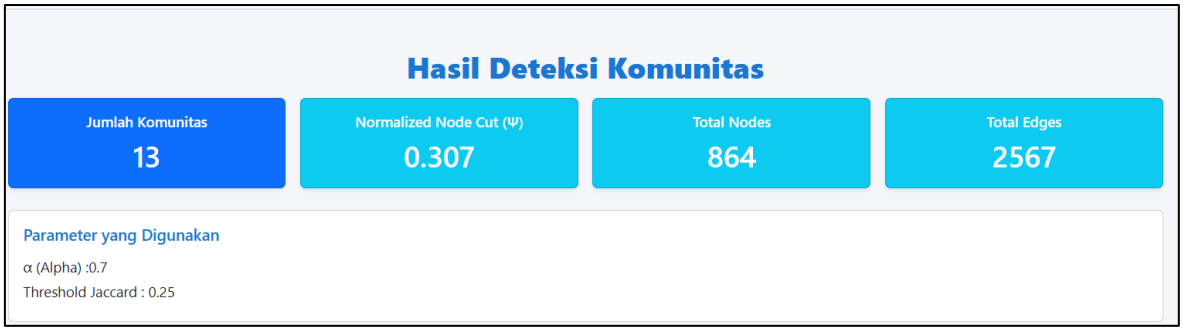
proses kanker payudara. Dengan demikian, tahap evaluasi tidak hanya berfungsi untuk menilai hasil deteksi komunitas, tetapi juga untuk memastikan bahwa analisis lanjutan dilakukan pada konfigurasi parameter yang memberikan kualitas komunitas paling optimal.

3.5.1 Normalized Node Cut (ψ)

Evaluasi struktural pada penelitian ini menggunakan parameter Normalized Node Cut (NNC) untuk menilai kualitas komunitas yang bersifat *overlapping*. Parameter ini dipilih karena mampu mengevaluasi struktur komunitas berdasarkan keberadaan simpul perbatasan (*boundary nodes*), yaitu simpul yang memiliki koneksi baik ke dalam maupun ke luar komunitas. Pada penelitian ini, metrik Normalized Node Cut digunakan untuk mengevaluasi kualitas struktur komunitas yang dihasilkan dari setiap skenario percobaan algoritma GLOD. Nilai metrik ini dihitung untuk setiap komunitas yang terbentuk pada masing-masing eksperimen sehingga memungkinkan perbandingan kualitas komunitas yang dihasilkan dari berbagai kombinasi parameter yang diuji.

Normalized Node Cut mengukur keseimbangan antara koneksi internal dan koneksi eksternal pada suatu komunitas. Nilai tersebut merepresentasikan kepadatan koneksi di dalam komunitas dibandingkan dengan keterhubungannya dengan bagian jaringan di luar komunitas. Nilai yang lebih kecil menunjukkan bahwa komunitas memiliki koneksi internal yang lebih kuat dibandingkan koneksi eksternal, sehingga struktur komunitas dianggap lebih baik dan lebih terpisah dari jaringan di sekitarnya. Oleh karena itu, nilai NNC digunakan sebagai indikator utama dalam menentukan kualitas komunitas yang dihasilkan pada setiap skenario percobaan serta untuk menentukan kombinasi parameter yang menghasilkan hasil deteksi komunitas yang paling optimal.

Rentang nilai Normalized Node Cut berada pada interval 0 hingga 1. Nilai yang semakin mendekati 0 menunjukkan bahwa komunitas memiliki struktur internal yang lebih kohesif dan pemisahan yang lebih jelas dari jaringan luar, sedangkan nilai yang mendekati 1 menunjukkan komunitas kurang terpisah secara struktural. Hasil evaluasi Normalized Node Cut secara keseluruhan ditampilkan pada Gambar 3.12. Gambar tersebut menunjukkan nilai NNC agregat yang dihasilkan dari setiap skenario eksperimen, sehingga dapat digunakan untuk membandingkan kualitas struktur komunitas secara umum pada berbagai kombinasi parameter yang diuji.



Gambar 3. 12 Normalized node cut

Sementara itu, nilai Normalized Node Cut pada masing-masing komunitas ditunjukkan pada Gambar 3.13. Gambar ini memberikan gambaran lebih rinci mengenai kualitas setiap komunitas

yang terbentuk, sehingga dapat diketahui komunitas mana yang memiliki struktur paling kohesif serta batas yang paling jelas dalam jaringan.

Komunitas ID	Jumlah Anggota	Jumlah Overlap	Anggota Protein	Protein Overlap	Normalized Node Cut (Ψ)
Komunitas 1	142	126	▶ Lihat 142 anggota	▶ Lihat 126 overlap	0.3057
Komunitas 2	90	86	▶ Lihat 90 anggota	▶ Lihat 86 overlap	0.214
Komunitas 3	200	144	▶ Lihat 200 anggota	▶ Lihat 144 overlap	0.1798
	74	70	▶ Lihat 74 anggota	▶ Lihat 70 overlap	0.3788
Komunitas 5	91	89	▶ Lihat 91 anggota	▶ Lihat 89 overlap	0.3421
	105	95	▶ Lihat 105 anggota	▶ Lihat 95 overlap	0.2782

Gambar 3. 13 *Normalized node cut perkomunitas*

Ilustrasi berikut digunakan untuk menjelaskan proses perhitungan nilai ψ pada komunitas yang dihasilkan. Perhitungan ini melibatkan jumlah seluruh interaksi, interaksi di dalam komunitas, serta interaksi yang menghubungkan komunitas dengan simpul di luar komunitas. Informasi interaksi antar protein yang digunakan dalam perhitungan ini dapat dilihat pada Tabel 3.4.

1. α = 0,5

Pada α = 0,5, seluruh simpul tergabung menjadi satu komunitas besar yaitu $C = \{A, B, C, D, E, F, G\}$. Berdasarkan Tabel 3.4, interaksi dalam komunitas ini meliputi seluruh relasi yang ada pada jaringan, yaitu A-B, A-C, B-C, C-D, D-E, E-F, E-G, dan F-G. Karena seluruh simpul jaringan masuk ke dalam satu komunitas, maka tidak terdapat simpul di luar komunitas. Dengan demikian seluruh interaksi bersifat internal, tidak ada interaksi keluar komunitas, untuk setiap simpul berlaku $k_i^{out} = 0$, Akibatnya, nilai Normalized Node Cut = 0. Nilai ini menunjukkan bahwa komunitas tidak memiliki batas eksternal karena mencakup seluruh jaringan.

2. α = 1,0

Pada nilai parameter α = 1,0 terbentuk dua komunitas yang saling tumpang tindih (overlap), yaitu $C_1 = \{A, B, C, D\}$, $C_2 = \{D, E, F, G\}$. Simpul D menjadi simpul overlapping karena termasuk ke dalam kedua komunitas tersebut. Struktur interaksi antar protein yang digunakan dalam perhitungan ini mengacu pada Tabel 3.4.

1) Evaluasi Komunitas $C_1 = \{A, B, C, D\}$

Berdasarkan Tabel 3.4, interaksi internal komunitas ini adalah A-B, A-C, B-C, dan C-D. Interaksi eksternal komunitas adalah D-E.Dengan demikian, derajat masing-masing simpul adalah:

1. A: $k^{in} = 2, k^{out} = 0, k = 2$
2. B: $k^{in} = 2, k^{out} = 0, k = 2$
3. C: $k^{in} = 3, k^{out} = 0, k = 3$
4. D: $k^{in} = 1, k^{out} = 1, k = 2$

Total derajat internal komunitas:

$$k_{in}(C_1) = 2 + 2 + 3 + 1 = 8$$

Kontribusi masing-masing simpul terhadap nilai $\boldsymbol{\psi}$:

Untuk A, B, dan C, karena tidak memiliki koneksi eksternal, maka:

$$\frac{k_i^{in} k_i^{out}}{k_i} = 0$$

Untuk D:

$$\frac{1 \times 1}{2} = 0{,}5$$

Sehingga jumlah kontribusi seluruh simpul:

$$\sum = 0 + 0 + 0 + 0{,}5$$

Nilai Normalized Node Cut untuk komunitas C_1:

$$\Psi(C_1) = \frac{1}{8} \times 0{,}5$$
$$\Psi(C_1) = 0{,}0625$$

2) Evaluasi Komunitas $C_2 = \{D, E, F, G\}$

Berdasarkan Tabel 3.4, interaksi internal komunitas ini adalah D-E, E-F, E-G, dan F-G. Interaksi eksternal komunitas adalah C-D. Derajat masing-masing simpul:

1. D: $k^{in} = 1$, $k^{out} = 1$, $k = 2$
2. E: $k^{in} = 3$, $k^{out} = 0$, $k = 3$
3. F: $k^{in} = 2$, $k^{out} = 0$, $k = 2$
4. G: $k^{in} = 2$, $k^{out} = 0$, $k = 2$

Total derajat internal:

$$k_{in}(C_2) = 1 + 3 + 2 + 2 = 8$$

Kontribusi simpul:

Untuk E, F, dan G, karena tidak memiliki koneksi eksternal, maka:

$$\frac{k_i^{in} k_i^{out}}{k_i} = 0$$

Untuk D:

$$\frac{1 \times 1}{2} = 0{,}5$$

Sehingga jumlah kontribusi seluruh simpul:

$$\sum = 0 + 0 + 0 + 0{,}5$$

Nilai Normalized Node Cut:

$$\Psi(C_2) = \frac{1}{8} \times 0{,}5$$
$$\Psi(C_2) = 0{,}0625$$

Berdasarkan hasil perhitungan diperoleh nilai Normalized Node Cut pada komunitas pertama sebesar $\Psi(C_1) = 0{,}0625$ dan pada komunitas kedua sebesar $\Psi(C_2) = 0{,}0625$. Dengan demikian, nilai NNC keseluruhan jaringan dihitung sebagai rata-rata dari kedua komunitas, yaitu

$$\Psi_{total} = \frac{\Psi(C_1) + \Psi(C_2)}{2} = \frac{0{,}0625 + 0{,}0625}{2} = 0{,}0625$$

Maka, hasil perhitungan NNC dari dua komunitas tersebut adalah 0,0625.

3. α = 1,5

Pada parameter α = 1,5, hasil deteksi komunitas menghasilkan dua komunitas yang terpisah, yaitu C_1 = {A, B, C} dan C_2 = {E, F, G}, tanpa adanya simpul overlapping. Berdasarkan perhitungan Normalized Node Cut diperoleh nilai $\Psi(C_1) = 0$ dan $\Psi(C_2) = 0$. Dengan demikian, nilai NNC keseluruhan jaringan dihitung sebagai rata-rata dari kedua komunitas, yaitu

$$\Psi_{total} = \frac{\Psi(C_1) + \Psi(C_2)}{2} = \frac{0 + 0}{2} = 0$$

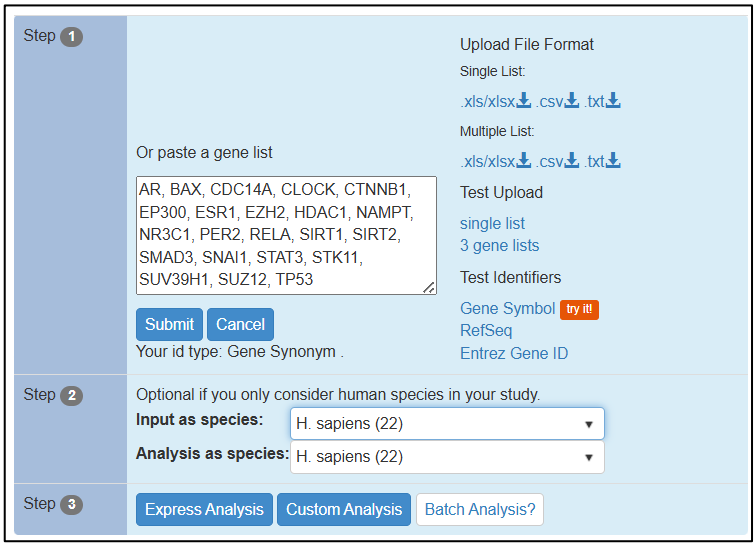
Maka, hasil perhitungan NNC pada α = 1,5 adalah 0. Nilai ini menunjukkan bahwa tidak terdapat simpul batas yang memiliki koneksi eksternal antar komunitas.

3.5.2 Enrichment Analysis

Setelah kualitas struktur komunitas dievaluasi menggunakan Normalized Node Cut, tahap selanjutnya adalah melakukan evaluasi biologis terhadap komunitas protein yang dihasilkan. Evaluasi ini bertujuan untuk mengetahui apakah komunitas protein yang terbentuk memiliki keterkaitan dengan fungsi biologis atau jalur molekuler yang relevan dengan kanker payudara.

Evaluasi fungsional dilakukan menggunakan pendekatan Enrichment Analysis dengan memanfaatkan platform Metascape. Pada tahap ini, daftar protein dari komunitas yang dihasilkan pada eksperimen dengan kualitas terbaik dimasukkan ke dalam sistem Metascape untuk dianalisis lebih lanjut. Analisis ini digunakan untuk mengidentifikasi proses biologis (*biological process*), fungsi molekuler (*molecular function*), serta jalur pensinyalan (*pathway*) yang secara signifikan diperkaya pada komunitas tersebut.

Melalui pendekatan ini, penelitian dapat memvalidasi bahwa komunitas yang dihasilkan oleh algoritma GLOD tidak hanya memiliki struktur jaringan yang baik secara topologis, tetapi juga memiliki relevansi biologis terhadap mekanisme kanker payudara. Proses analisis diawali dengan memasukkan daftar protein dari setiap komunitas dan memilih spesies *Homo sapiens*, seperti ditunjukkan pada Gambar 3.14.



Gambar 3. 14 Input metascape

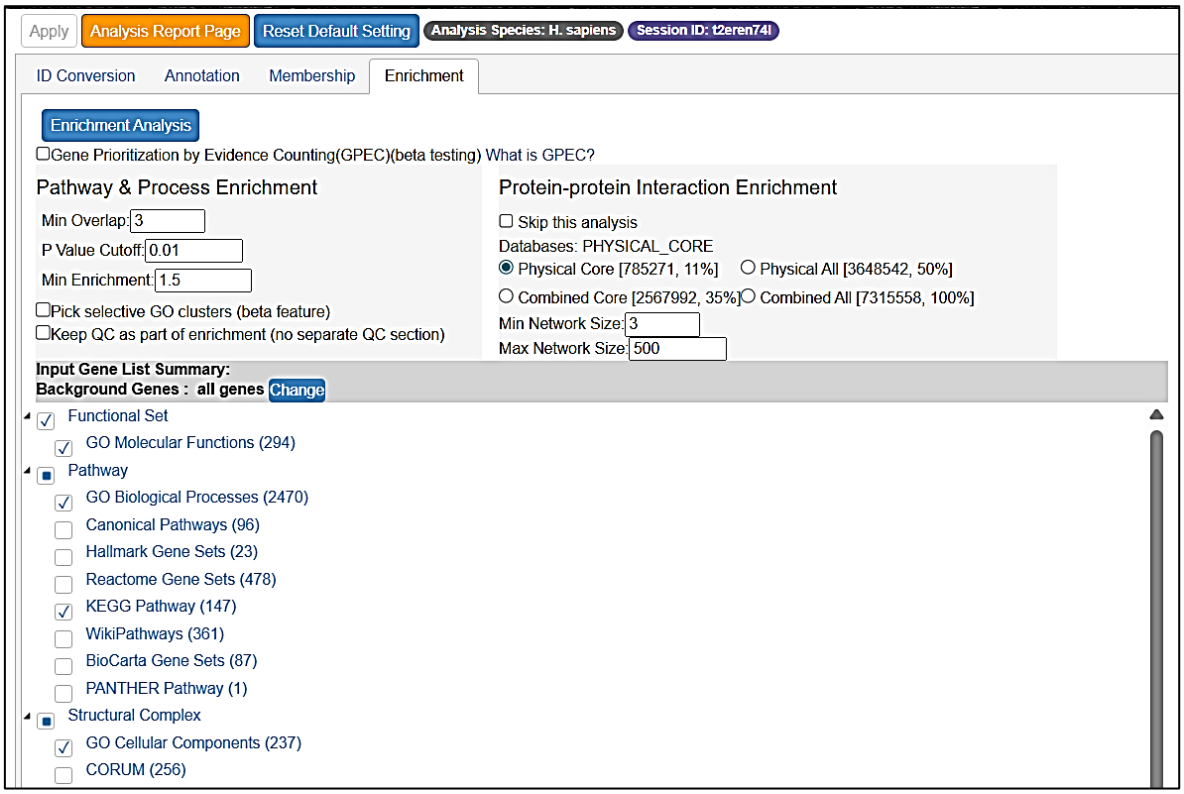
Selanjutnya klik *custom analysis* dan atur parameter analisis sebagai berikut, untuk parameter *pathway & process enrichment*:

1. Min Overlap = 3
2. P Value Cutoff = 0.01
3. Min Enrichment = 1.5

Sedangkan untuk parameter Protein-protein Interaction Enrichment atur sebagai berikut:

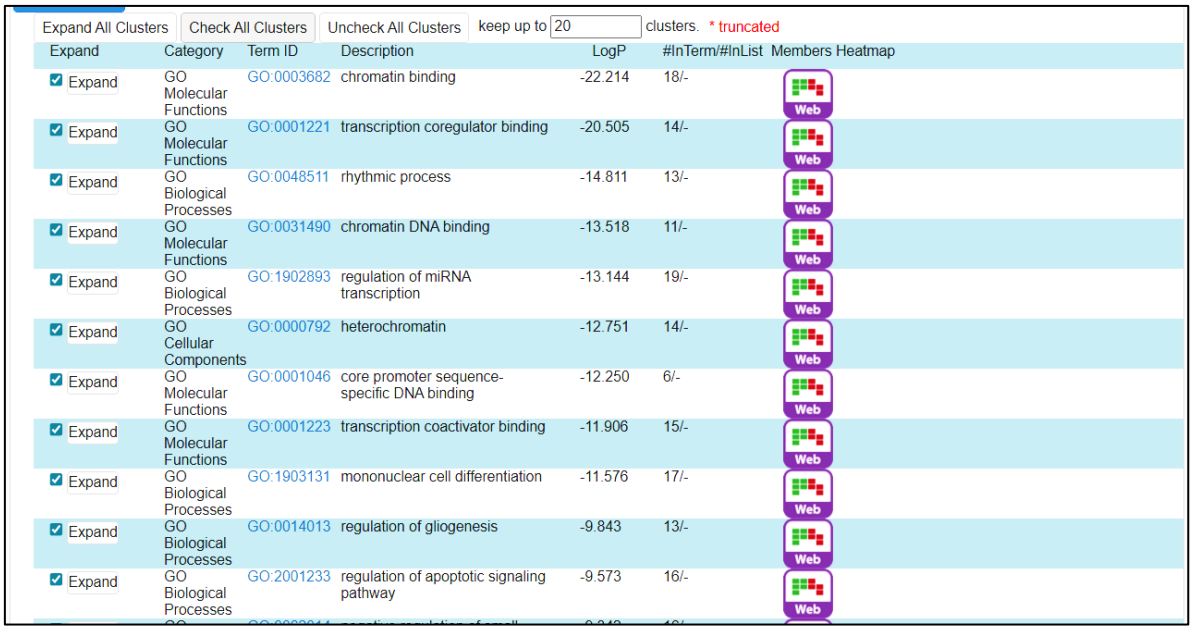
1. Databases: Physical Core [785271, 11%]
2. Min Network Size = 3
3. Max Network Size = 500

Setelah semua parameter diatur, pilih 4 jenis analisis yaitu GO Molecular Function, GO Biological Processes, KEGG Pathway, dan GO Cellular Components seperti yang ditunjukkan oleh Gambar 3.15.



Gambar 3. 15 Parameter enrichment

Validitas biologis komunitas ditentukan berdasarkan nilai signifikansi statistik (*p-value* atau *LogP*), di mana nilai yang lebih kecil menunjukkan tingkat signifikansi yang lebih tinggi. Hasil analisis menunjukkan bahwa protein dalam komunitas yang terbentuk memiliki keterlibatan yang signifikan dalam mekanisme biologis yang berkaitan dengan kanker payudara, sebagaimana ditampilkan pada Gambar 3.16.



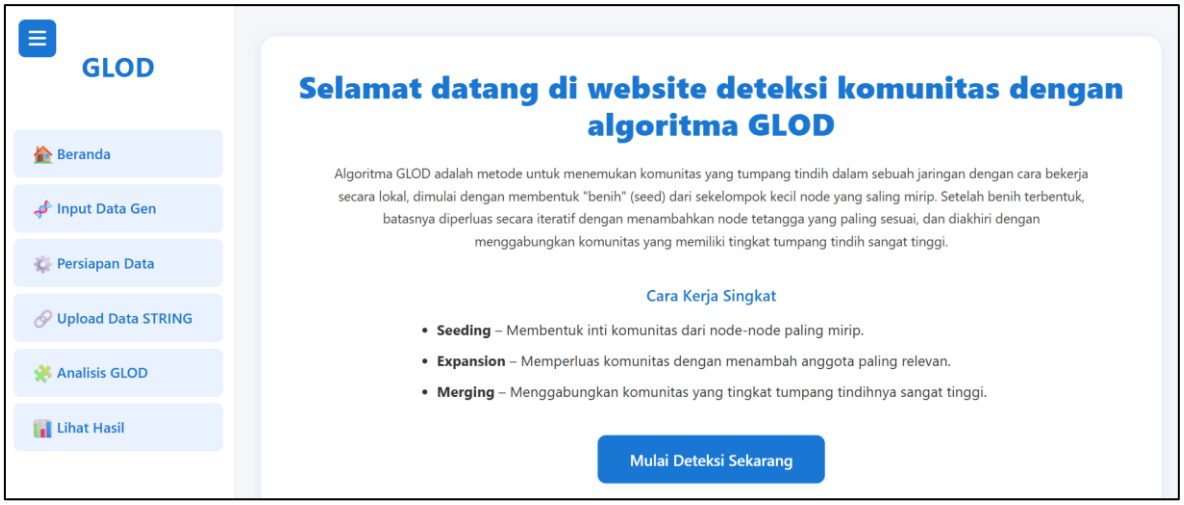
Gambar 3. 16 Hasil *enrichment*

3.6 Deployment

Proses deployment ini menggunakan framework Django untuk mengembangkan backend dan menggunakan bootstrap untuk antarmukanya. Pada penelitian ini deployment dibagi menjadi dua tahap yaitu pembuatan user interface dan integrasi sistem.

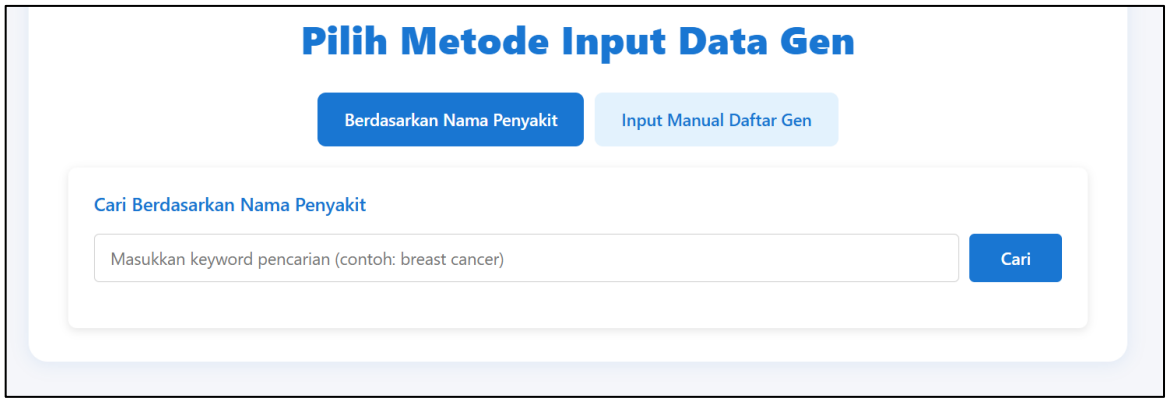
3.6.1 Pembuatan *User Interface*

Proses ini dimulai dengan membuat beberapa halaman menu aplikasi, yang pertama adalah beranda yang berisi informasi singkat tentang cara kerja program ini dan sidebar yang berisi menu aplikasi seperti yang ditunjukkan oleh Gambar 3.17.



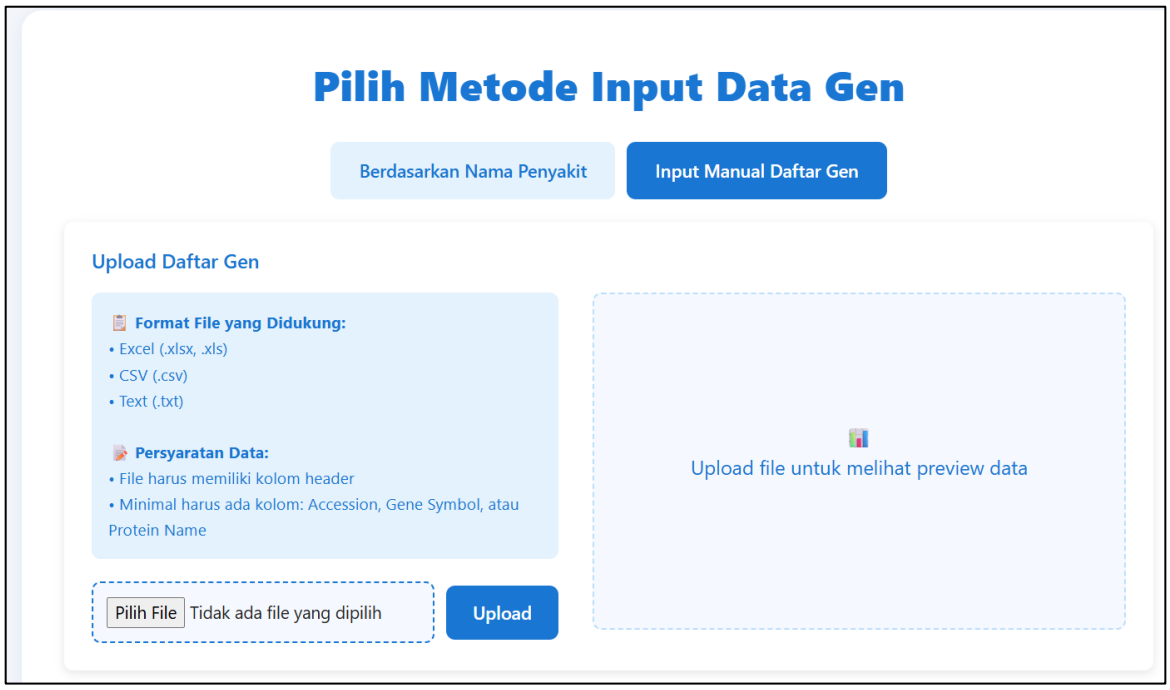
Gambar 3. 17 Halaman beranda

Selanjutnya, sistem menyediakan halaman input data gen yang memungkinkan pengguna untuk memasukkan data yang akan dianalisis. Pengguna dapat mengambil data secara langsung dari database UniProt melalui fitur pencarian yang ditunjukkan pada Gambar 3.18. Halaman ini memfasilitasi pencarian dan pemilihan data protein berdasarkan kata kunci tertentu.



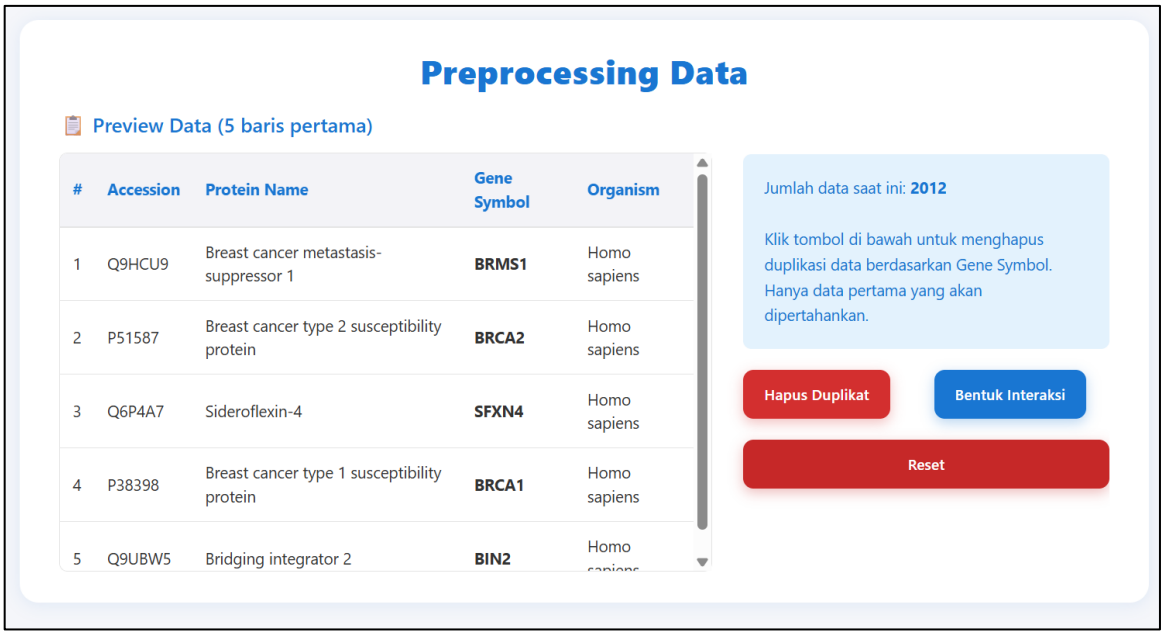
Gambar 3. 18 Halaman search Uniprot

Selain itu, pengguna juga dapat mengunggah data gen secara manual melalui halaman input yang ditunjukkan pada Gambar 3.19. Halaman ini mendukung berbagai format file, seperti xlsx, xls, csv, dan txt, sehingga memberikan fleksibilitas dalam proses input data.



Gambar 3. 19 Halaman input data gen

Selanjut, halaman persiapan data, halaman ini berfungsi untuk mengolah data dari halaman input data agar siap digunakan untuk proses selanjutnya. Pada halaman ini data akan diseleksi, dan dibersihkan dari duplikasi (Gambar 3.20).

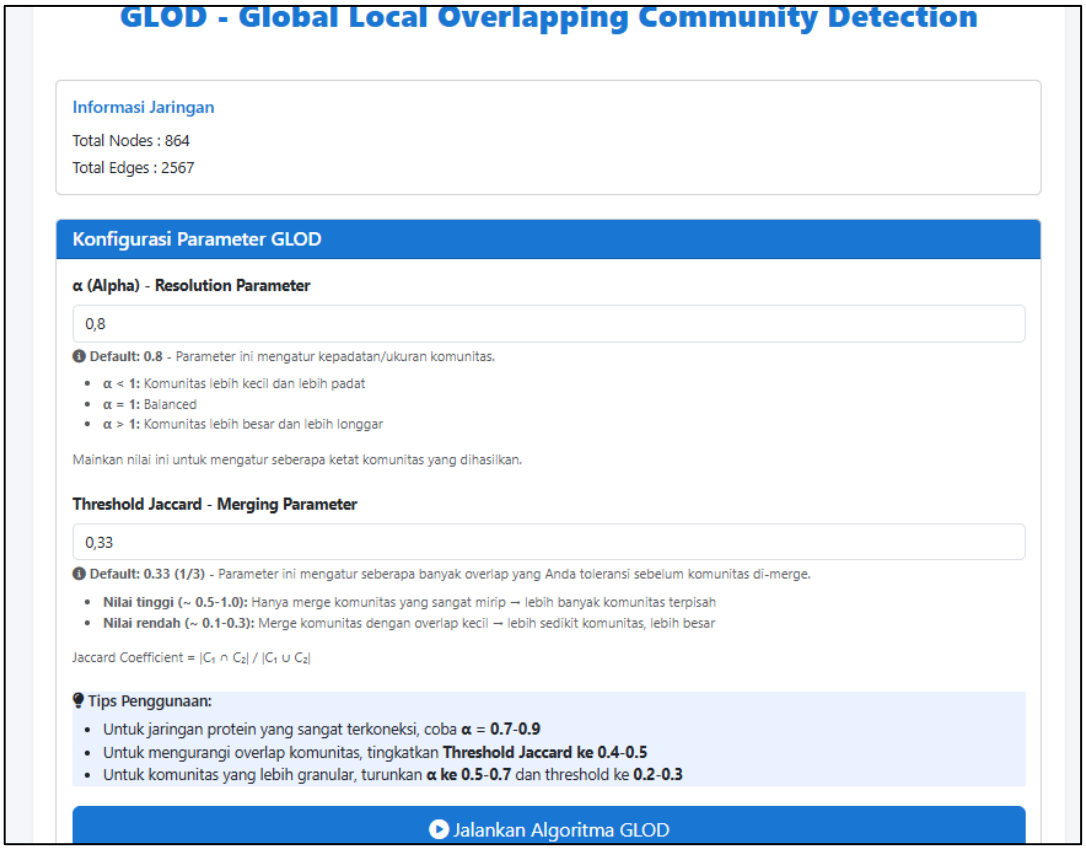


Gambar 3. 20 Halaman *preprocessing data*

Untuk membangun jaringan menggunakan STRING dibuatlah halaman Pembangunan jaringan protein (Gambar 3.21) yang terdapat inputan untuk parameter *confidence score* dan organisme sebagai parameter dalam membangun jaringan.

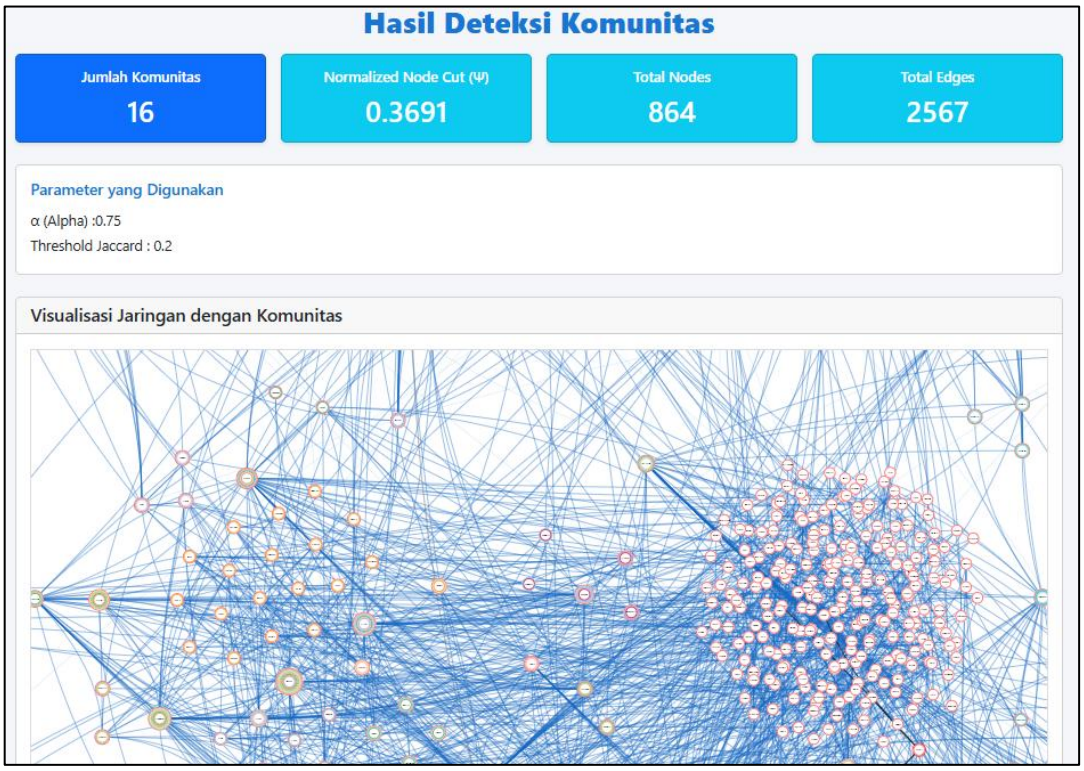
Gambar 3. 21 Halaman pembangunan jaringan

Setelah jaringan berhasil dibangun, data selanjutnya diproses menggunakan algoritma GLOD melalui halaman analisis GLOD yang ditunjukkan pada Gambar 3.22. Halaman ini menyediakan input parameter yang digunakan dalam proses deteksi komunitas, yaitu parameter α (alpha) dan threshold Jaccard (J). Kedua parameter ini berperan dalam mengatur proses ekspansi dan penggabungan komunitas pada algoritma GLOD.



Gambar 3. 22 Halaman analisis GLOD

Setelah proses analisis selesai dilakukan, hasil deteksi komunitas akan ditampilkan beserta informasi terkait pada halaman hasil, seperti yang ditunjukkan pada Gambar 3.23.



Gambar 3. 23 Halaman lihat hasil

3.6.2 Integrasi Sistem

Pada sistem ini, antarmuka pengguna dan logika algoritma deteksi komunitas dikembangkan secara terpisah menggunakan kerangka kerja *Django*. Pemisahan ini diterapkan untuk mendukung pengembangan sistem yang bersifat modular sehingga lebih mudah dikelola dan dikembangkan. Antarmuka berfungsi menerima masukan dari pengguna, seperti unggahan berkas daftar gen atau pengambilan data dari basis data *UniProt*, serta parameter algoritma berupa *Alpha* dan *Threshold Jaccard*, kemudian menampilkan hasil deteksi komunitas. Sementara itu, proses deteksi dijalankan oleh modul algoritma GLOD pada sisi *backend*. Antarmuka mengirimkan data masukan ke modul algoritma, memprosesnya, dan menampilkan hasil visualisasi serta metrik evaluasi kepada pengguna. Dengan arsitektur ini, perubahan pada logika algoritma dapat dilakukan tanpa memengaruhi tampilan antarmuka, dan sebaliknya.

Untuk memastikan bahwa sistem yang dikembangkan dapat berjalan sesuai dengan kebutuhan fungsional, dilakukan proses pengujian terhadap seluruh alur sistem, mulai dari tahap input data hingga penyajian hasil deteksi komunitas. Pengujian ini mencakup validasi proses unggah dan pengolahan data gen, konstruksi jaringan protein menggunakan STRING, eksekusi algoritma GLOD berdasarkan parameter yang diberikan, serta visualisasi hasil komunitas. Pengujian dilakukan secara menyeluruh terhadap setiap komponen sistem untuk memastikan bahwa setiap tahapan proses berjalan tanpa kesalahan dan menghasilkan keluaran yang sesuai dengan yang diharapkan. Hasil dari pengujian tersebut tidak disajikan dalam bentuk tabel pada bab ini, melainkan ditunjukkan secara langsung melalui hasil implementasi sistem pada Bab IV, khususnya pada Subbab 4.2, yang memperlihatkan keluaran dari setiap tahapan proses secara runtut dan terintegrasi.

BAB IV

HASIL PENGUJIAN DAN PEMBAHASAN

4.1 Implementasi

Bagian ini menjelaskan proses implementasi sistem deteksi komunitas overlap pada jaringan interaksi protein kanker payudara menggunakan algoritma Local Community Detection berbasis GLOD. Implementasi mencakup Data Preparation, penerapan algoritma GLOD untuk memperoleh komunitas protein dan Pengembangan sistem. Pengembangan sistem dilakukan berbasis web menggunakan framework Django dengan bahasa pemrograman Python.

4.1.1 Deskripsi Alat dan Lingkungan Eksekusi

Pengembangan sistem memanfaatkan beberapa library Python yang disesuaikan dengan tahapan proses penelitian, mulai dari pengumpulan data, preprocessing, pembangunan jaringan interaksi protein, hingga deteksi komunitas menggunakan algoritma GLOD. Daftar library yang digunakan pada setiap proses ditunjukkan pada Tabel 4.1.

Tabel 4. 1 Daftar library yang digunakan

Proses	Modul / Library
Pengumpulan Data	requests, pandas, urllib.parse, urllib.request, urllib.error, ssl, re, time, logging
Data Preprocessing	pandas, logging, Django (messages dan session), requests, json, typing, datetime
Deteksi Komunitas	networkx, math, typing, json
Evaluasi	networkx, math, typing
Ekspor Hasil	csv, openpyxl, datetime, Django (HttpResponse)

4.1.2 Tahapan Implementasi Sistem

Tahapan implementasi sistem mengikuti alur metodologi pada Bab III, yang terdiri dari pengambilan data protein, preprocessing, pembangunan jaringan PPI, dan deteksi komunitas menggunakan algoritma GLOD.

4. Pengumpulan Data

Tahap pengumpulan data bertujuan untuk memperoleh daftar protein yang berkaitan dengan kanker payudara. Data diambil dari basis data UniProt menggunakan REST API. Implementasi pengambilan data dilakukan pada berkas views_uniprot.py. Kode Program 1 menunjukkan proses permintaan data ke API UniProt dengan menentukan parameter pencarian berupa kata kunci penyakit, format keluaran JSON, jumlah data yang diambil, serta field informasi protein yang dibutuhkan, seperti accession, nama protein, simbol gen, organisme, dan status reviewed.

Kode Program 1: API Request UniProt

```
base_url = "https://rest.uniprot.org/uniprotkb/search"
params = {
    "query": query,
    "format": "json",
    "size": str(size),
    # Explicitly request all fields we need
    "fields":
"accession,id,protein_name,gene_names,gene_primary,organism_name,reviewed",
}
resp = requests.get(base_url, params=params, headers=headers, timeout=60)

next_link = _get_next_link(resp.headers)
return resp, next_link
```

5. Data Preparation

Data Preparation bertujuan untuk menyiapkan data protein agar siap dianalisis lebih lanjut. Proses ini meliputi pengambilan simbol gen, penghapusan data duplikat, pemetaan gen ke ID STRING, pengambilan interaksi protein, serta pemilihan *giant component*. Tahap ini penting karena kualitas jaringan sangat bergantung pada kebersihan dan keterhubungan data. Kode Program 2 digunakan untuk mengekstraksi simbol gen (*gene symbol*) dari data protein hasil UniProt, kemudian menyimpannya ke dalam *session* Django. Kode ini mengambil kolom *gene_symbol* dari setiap entri protein, menghapus nilai kosong, serta menyaring simbol gen agar menjadi unik. Hasilnya berupa daftar gen tanpa pengulangan yang disimpan dalam *session* sebagai input tahap berikutnya. Dengan cara ini, sistem memastikan bahwa hanya gen valid yang digunakan dalam pembangunan jaringan protein.

Kode Program 2: Ambil kolom *gene symbol*

```
request.session['preprocessing_data'] = valid_data
gene_symbols = [row.get('gene_symbol') for row in valid_data if
row.get('gene_symbol')]
request.session['preprocessing_genes'] =
list(sorted(set(gene_symbols)))
request.session.modified = True
```

Kode Program 3 berfungsi untuk menghilangkan data protein yang memiliki simbol gen sama (duplikat). Proses ini dilakukan menggunakan fungsi drop_duplicates dari *pandas*. Melalui kode ini, sistem hanya mempertahankan satu entri untuk setiap simbol gen. Jumlah data yang terhapus dicatat, kemudian daftar gen unik diperbarui kembali ke dalam *session*. Penghapusan duplikat bertujuan mencegah satu protein dihitung lebih dari sekali, yang dapat memengaruhi struktur jaringan dan hasil deteksi komunitas.

Kode Program 3: Penghapusan Duplikat

```
if 'gene_symbol' in df.columns:
    df_unique = df.drop_duplicates(subset=['gene_symbol'], keep='first')
    removed_count = original_count - len(df_unique)
    unique_data = df_unique.to_dict('records')

    request.session['preprocessing_data'] = unique_data
    request.session['preprocessing_duplicates_removed'] = True

    gene_symbols = [row.get('gene_symbol') for row in unique_data if
row.get('gene_symbol')]
    request.session['preprocessing_genes'] =
list(sorted(set(gene_symbols)))

    if 'network_genes' in request.session:
        del request.session['network_genes']

    request.session.modified = True
    print(f"[DEBUG] remove_duplicates: Updated preprocessing_genes with
{len(request.session['preprocessing_genes'])} unique genes")
    messages.success(request, f'Berhasil menghapus {removed_count} data
duplikat. Data sekarang: {len(unique_data)} entries.')
else:
    messages.error(request, 'Kolom gene_symbol tidak ditemukan pada
data. Tidak dapat menghapus duplikat.')
```

Kode Program 4 digunakan untuk mengubah simbol gen menjadi *STRING ID*, yaitu identitas protein yang dikenali oleh STRING Database. Pada tahap ini, daftar gen dikirim ke API STRING menggunakan metode POST. Sistem kemudian membaca respons dalam bentuk teks tabular dan mengambil ID STRING untuk setiap gen. Pemetaan ini diperlukan karena STRING tidak menggunakan simbol gen secara langsung, melainkan ID internal untuk merepresentasikan protein.

Kode Program 4: Pemetaan Gene Symbol ke STRING ID

```
def _get_string_ids(gene_names: List[str], species: str = "9606",
chunk_size: int = 500, limit: int = 5) -> Dict[str, str]:
    all_string_ids: Dict[str, str] = {}
    request_url = f"{STRING_API_URL}/tsv/get_string_ids"
    params = {
        "identifiers": "\r".join(gene_names),
        "species": species,
        "limit": limit,
        "echo_query": 1
    }
    try:
        response = requests.post(request_url, data=params, timeout=60)
        if response.status_code == 200 and response.text.strip():
            lines = response.text.strip().splitlines()
            for line in lines[1:]:
                if not line.strip():
                    continue
                parts = line.split('\t')
                if len(parts) >= 3:
                    query_name = parts[0].strip()
                    string_id = parts[2].strip()
                    if query_name and string_id and query_name not in
all_string_ids:
                        all_string_ids[query_name] = string_id
    except Exception as e:
        print(f"Terjadi kesalahan saat mendapatkan ID STRING: {e}")
    return all_string_ids
```

Kode Program 5 berfungsi mengambil pasangan interaksi antar protein berdasarkan STRING ID yang telah diperoleh sebelumnya. Kode ini meminta data jaringan dari STRING Database dan mengekstraksi pasangan protein beserta nilai skor kepercayaannya. Skor tersebut menunjukkan tingkat keyakinan hubungan antar protein. Interaksi yang duplikat kemudian disatukan sehingga setiap pasangan protein hanya muncul satu kali dengan skor tertinggi. Tahap ini menghasilkan daftar hubungan protein yang akan digunakan untuk membentuk graf jaringan

Kode Program 5: Pengambilan Interaksi Protein

```
def _get_protein_interactions(string_ids: List[str], species: str =
"9606", required_score: int = 400,
                              chunk_size: int = 500, network_type: str =
"full") -> List[Dict]:
    all_interactions: List[Dict] = []
    request_url = f"{STRING_API_URL}/tsv/network"
    params = {
        "identifiers": "\r".join(string_ids),
        "species": species,
```

Kode Program 5: Pengambilan Interaksi Protein (Lanjutan)

```
        "required_score": required_score,
        "network_type": network_type
    }
    try:
        response = requests.post(request_url, data=params, timeout=120)
        if response.status_code == 200 and response.text.strip():
            lines = response.text.strip().splitlines()
            for line in lines[1:]:
                if not line.strip():
                    continue
                parts = line.split('\t')
                if len(parts) >= 3:
                    p1 = parts[0].strip()
                    p2 = parts[1].strip()
                    score = _safe_float(parts[5])
                    all_interactions.append({'protein1':            p1,
'protein2': p2, 'score': score})
    except Exception as e:
        print(f"Terjadi kesalahan saat mendapatkan interaksi: {e}")
    consolidated = _consolidate_interactions(all_interactions)
    return consolidated
```

Kode Program 6 digunakan untuk memilih *giant component*, yaitu komponen jaringan terbesar yang saling terhubung. Pada tahap ini, sistem terlebih dahulu membangun daftar ketetanggaan antar simpul berdasarkan hubungan edge yang tersedia, kemudian melakukan penelusuran jaringan menggunakan pendekatan *Breadth First Search* (BFS) untuk mengelompokkan jaringan menjadi beberapa komponen terpisah. Dari seluruh komponen yang terbentuk, sistem memilih komponen dengan jumlah simpul terbesar sebagai *giant component*, sementara simpul yang berada di luar komponen utama tersebut akan dihapus dari jaringan. Proses ini bertujuan untuk memastikan bahwa analisis komunitas hanya dilakukan pada jaringan yang benar-benar saling terhubung, sehingga protein yang terisolasi tidak memengaruhi hasil deteksi komunitas. Dengan memilih *giant component*, jaringan yang dianalisis menjadi lebih representatif terhadap sistem biologis yang sebenarnya, karena hanya protein yang memiliki interaksi yang dipertahankan dalam proses selanjutnya.

Kode Program 6: *Giant Component*

```
function removeIsolatedNodes() {
    if (!network || !nodes || !edges) {alert('Network belum tersedia');
        return;    }
    const allNodes = nodes.get();
    const allEdges = edges.get();
    const adjacencyList = {};
    allNodes.forEach(node => {
        adjacencyList[node.id] = [];    });
    allEdges.forEach(edge => {
        if (adjacencyList[edge.from]) {
            adjacencyList[edge.from].push(edge.to);        }
        if (adjacencyList[edge.to]) {
            adjacencyList[edge.to].push(edge.from); }    });
    const visited = new Set();
    const components = [];
```

Kode Program 6: *Giant Component* (Lanjutan)

```
    function bfs(startNode) {
        const queue = [startNode];
        const component = new Set();
        visited.add(startNode);
        component.add(startNode);
        while (queue.length > 0) {
            const node = queue.shift();
            const neighbors = adjacencyList[node] || [];
            for (const neighbor of neighbors) {
                if (!visited.has(neighbor)) {
                    visited.add(neighbor);
                    component.add(neighbor);
                    queue.push(neighbor); }}}
        return component;    }
    for (const node of allNodes) {
        if (!visited.has(node.id)) {
            const component = bfs(node.id);
            components.push(component); }}
    let giantComponent = new Set();    let maxSize = 0;
    components.forEach(component => {
        if (component.size > maxSize) {
            maxSize = component.size;
            giantComponent = component; }    });
    const       nodesToRemove       =       allNodes.filter(node       =>
!giantComponent.has(node.id));
    if (nodesToRemove.length === 0) {
        alert('Semua node sudah terhubung dalam satu komponen utama.');
        return;    }
    const  confirmMessage  =  `Ditemukan  ${components.length}  komponen
terpisah.\n` +
                          `Komponen           terbesar           memiliki
${giantComponent.size} node.\n` +
                          `Akan  menghapus  ${nodesToRemove.length}  node
yang tidak terhubung ke komponen utama. Lanjutkan?`;
    if (!confirm(confirmMessage)) {return; }
    const nodeIdsToRemove = nodesToRemove.map(node => node.id);
    nodes.remove(nodeIdsToRemove);
    const edgesToRemove = allEdges.filter(edge =>
        nodeIdsToRemove.includes(edge.from)                              ||
nodeIdsToRemove.includes(edge.to)    );
    edges.remove(edgesToRemove.map(edge => edge.id));
    const currentNodesCount = nodes.get().length;
    const currentEdgesCount = edges.get().length;
    const  nodesCountElement  =  document.getElementById('current-nodes-
count');
    const  edgesCountElement  =  document.getElementById('current-edges-
count');
    if      (nodesCountElement)      nodesCountElement.textContent      =
currentNodesCount;
    if      (edgesCountElement)      edgesCountElement.textContent      =
currentEdgesCount;
    alert(`Berhasil  menghapus  ${nodesToRemove.length}  node  yang  tidak
terhubung ke komponen utama.\n` +
          `Tersisa  ${currentNodesCount}  node  dan  ${currentEdgesCount}
edge.`);}
```

6. Pembentukan kelompok protein menggunakan algoritma GLOD

Tahap pembentukan kelompok protein dilakukan menggunakan algoritma GLOD yang terdiri dari tiga fase utama, yaitu *seeding phase*, *expansion phase*, dan *merging phase*. Ketiga fase ini bertujuan untuk membentuk komunitas protein yang bersifat *overlapping*, di mana satu protein dapat menjadi anggota lebih dari satu komunitas.

Kode Program 7 menjelaskan proses pembentukan *rough seed*, yaitu komunitas awal yang digunakan sebagai titik awal deteksi komunitas. Pada tahap ini, seluruh simpul dalam jaringan dimasukkan ke dalam himpunan kandidat (*NL*). Dari simpul NL tersebut, sistem membentuk *rough seed* dengan menambahkan tetangga yang memiliki tingkat keterhubungan tinggi, kemudian menghitung skor seed berdasarkan jumlah simpul, derajat, dan edge internal. Proses ini dilakukan secara iteratif hingga diperoleh sejumlah kandidat seed terbaik. Seeding phase bertujuan untuk menghasilkan komunitas awal yang secara topologi sudah cukup kuat sebelum diperluas lebih lanjut pada tahap berikutnya

Kode Program 7: *Seeding Phase*

```
NL = set(self.graph.nodes())
all_nodes = set(self.graph.nodes())
candidate_seeds = []

iteration = 0
max_iterations = 1000

while NL and iteration < max_iterations:
    iteration += 1

    best_center = min(
        NL,
        key=lambda node: (-self.graph.degree(node), node)
    )

    print(f"\nIteration {iteration}: Processing center node
{best_center} (degree: {self.graph.degree(best_center)}, NL size:
{len(NL)})")

    rough_seed = self.create_rough_seed(best_center)
    score = self.calculate_seed_score(rough_seed)

    print(f"  Rough seed created: size {len(rough_seed)}, score
{score:.2f}")

    candidate_seeds.append((rough_seed, score, best_center))
    NL.discard(best_center)

    if len(candidate_seeds) >= 100:
        print(f"  Reached 100 candidate seeds, processing them now...")
        break

candidate_seeds.sort(key=lambda x: x[1], reverse=True)
print(f"\nSeeding phase complete: {len(candidate_seeds)} candidate seeds
found")
```

Kode Program 8 menjelaskan tahap perluasan komunitas (*expansion phase*). Pada fase ini, setiap seed diperluas dengan menambahkan protein di sekitar komunitas (*shell*

nodes) secara bertahap. Untuk setiap kandidat protein, sistem menghitung tiga nilai utama, yaitu *fitness gain*, *omega* (tingkat kemiripan tetangga), dan *influence* (pengaruh simpul terhadap komunitas). Protein akan ditambahkan ke dalam komunitas apabila memberikan peningkatan kualitas komunitas berdasarkan salah satu dari ketiga kriteria tersebut. Proses ekspansi dihentikan ketika tidak ada lagi protein yang memenuhi ambang batas peningkatan fitness atau ketika ukuran komunitas mencapai batas maksimum. Dengan mekanisme ini, komunitas berkembang secara adaptif berdasarkan struktur jaringan, bukan berdasarkan ukuran tetap.

Kode Program 8: *Expansion Phase*

```
community = seed.copy()
improved = True
initial_fitness = self.fitness_function(community)
iterations = 0
min_fitness_gain_threshold = 0.0001
max_community_size_ratio = 0.5
max_community_size   =   max(3,   int(self.graph.number_of_nodes()   *
max_community_size_ratio))

print(f"    Expanding  seed  (size:  {len(seed)},  initial  fitness:
{initial_fitness:.4f})")
print(f"  Max community size allowed: {max_community_size} nodes")

while improved:
    improved = False
    current_fitness = self.fitness_function(community)

    shell_nodes = set()
    for node in community:
        for neighbor in self.graph.neighbors(node):
            if neighbor not in community:
                shell_nodes.add(neighbor)

    if not shell_nodes:
        print(f"    Stopping: No shell nodes found")
        break

    if len(community) >= max_community_size:
        print(f"    Stopping: Community size ({len(community)}) reached
max limit ({max_community_size})")
        break

    candidate_scores = {}
    for candidate in sorted(shell_nodes):
        test_community = community.copy()
        test_community.add(candidate)

        fitness_gain    =    self.fitness_function(test_community)    -
current_fitness
        omega_val = self.omega(candidate, community)
        influence = self.influence_function(candidate, community)

        candidate_scores[candidate] = {
            'fitness': fitness_gain,
            'omega': omega_val,
            'influence': influence
```

Kode Program 8: *Expansion Phase* (Lanjutan)

```
        }

    best_by_fitness = max(candidate_scores.items(), key=lambda x:
x[1]['fitness'])
    best_by_omega = max(candidate_scores.items(), key=lambda x:
x[1]['omega'])
    best_by_influence = max(candidate_scores.items(), key=lambda x:
x[1]['influence'])

    argmax_nodes_set = {best_by_fitness[0], best_by_omega[0],
best_by_influence[0]}
    argmax_nodes = sorted(argmax_nodes_set)

    best_candidate = None
    best_score = float('-inf')
    best_criterion = None
    best_fitness_gain = 0.0

    for candidate in argmax_nodes:
        scores = candidate_scores[candidate]
        max_score_for_candidate = max(scores['fitness'],
scores['omega'], scores['influence'])

        if max_score_for_candidate > best_score or (
            max_score_for_candidate == best_score and
            (best_candidate is None or candidate < best_candidate)
        ):
            best_score = max_score_for_candidate
            best_candidate = candidate
            best_fitness_gain = scores['fitness']
            if scores['fitness'] == max_score_for_candidate:
                best_criterion = 'fitness'
            elif scores['omega'] == max_score_for_candidate:
                best_criterion = 'omega'
            else:
                best_criterion = 'influence'

    add_node = False
    reason = ""

    if best_candidate is not None:
        if best_fitness_gain >= min_fitness_gain_threshold:
            add_node = True
            reason = f"by {best_criterion} (fitness_gain:
{best_fitness_gain:.6f})"
        elif best_fitness_gain < 0 and iterations < 10:
            scores = candidate_scores[best_candidate]
            if scores['omega'] > 0.8:
                add_node = True
                reason = f"by omega={scores['omega']:.4f} (exceptional
high similarity despite negative fitness)"
            elif scores['influence'] > 0.8:
                add_node = True
                reason = f"by influence={scores['influence']:.4f}
(exceptional high influence despite negative fitness)"
```

Kode Program 8: *Expansion Phase* (Lanjutan)

```
        if      not      add_node      and      best_fitness_gain      <
min_fitness_gain_threshold:
            print(f"   Stopping expansion: No candidate meets criteria")
            print(f"      Best: {best_candidate} by {best_criterion}")
            print(f"             fitness_gain:  {best_fitness_gain:.6f}
(threshold: {min_fitness_gain_threshold})")
            print(f"                                                    omega:
{candidate_scores[best_candidate]['omega']:.4f}")
            print(f"                                               influence:
{candidate_scores[best_candidate]['influence']:.4f}")
            break

    if add_node and best_candidate is not None:
        community.add(best_candidate)
        improved = True
        iterations += 1
        print(f"   Added node {best_candidate} {reason} (community size:
{len(community)})")
    else:
        break

final_fitness = self.fitness_function(community)
print(f"     Expansion complete: {len(seed)} → {len(community)} nodes,
fitness:  {initial_fitness:.4f}  →  {final_fitness:.4f}  ({iterations}
iterations)")
return community
```

Kode Program 9 menjelaskan tahap penggabungan komunitas (*merging phase*). Pada fase ini, sistem mengevaluasi tingkat kemiripan antar komunitas menggunakan koefisien Jaccard yang telah dimodifikasi. Apabila dua komunitas memiliki tingkat irisan anggota yang tinggi dan melebihi nilai ambang batas, maka kedua komunitas tersebut digabung menjadi satu. Proses penggabungan dilakukan secara berulang hingga tidak ditemukan lagi pasangan komunitas yang memenuhi kriteria penggabungan. Tahap ini bertujuan untuk mengurangi redundansi komunitas dan memastikan bahwa komunitas akhir benar-benar merepresentasikan kelompok protein yang saling berhubungan erat.

Kode Program 9: *Merging Phase*

```
print(f"\nStarting  merge  phase  with  improved  Jaccard  coefficient
(threshold: {self.jaccard_threshold:.4f})")
print(f"Communities before merging: {len(self.communities)}")

merge_count = 0
merged = True

while merged:
    merged = False
    new_communities = []
    communities_to_skip = set()

    for i in range(len(self.communities)):
```

Kode Program 9: *Merging Phase*

```
        if i in communities_to_skip:
            continue
        current_comm = self.communities[i]
        communities_to_merge = [i]

        for j in range(i + 1, len(self.communities)):
            if j in communities_to_skip:
                continue

            other_comm = self.communities[j]
            improved_jaccard = self.improved_jaccard_coefficient(i, j)

            if improved_jaccard >= self.jaccard_threshold:
                print(
                    f"  Merging C{i} (size={len(current_comm)}) and C{j}
(size={len(other_comm)}) "
                    f"(Improved   Jaccard:   {improved_jaccard:.4f}   >=
{self.jaccard_threshold:.4f})"
                )
                communities_to_merge.append(j)
                communities_to_skip.add(j)
                merged = True
                merge_count += 1

        if len(communities_to_merge) > 1:
            merged_community = current_comm.copy()
            for idx in communities_to_merge[1:]:
                merged_community                                        =
merged_community.union(self.communities[idx])
            new_communities.append(merged_community)
        else:
            new_communities.append(current_comm)

        communities_to_skip.add(i)

    self.communities = new_communities

print(f"Merging complete: {merge_count} merges performed")
print(f"Communities after merging: {len(self.communities)}\n")
```

7. Pengembangan Sistem

Pengembangan antarmuka pengguna bertujuan untuk memudahkan pengguna dalam menjalankan seluruh tahapan analisis, mulai dari input data hingga melihat hasil komunitas protein.

A. Implementasi User Interface

Kode Program 10 menjelaskan struktur navigasi utama sistem yang ditampilkan dalam bentuk sidebar. Menu yang tersedia meliputi halaman beranda, input data gen, persiapan data, pembangunan jaringan STRING, serta analisis GLOD. Struktur navigasi ini dirancang agar pengguna dapat berpindah antar tahapan secara berurutan sesuai alur penelitian, sehingga proses analisis menjadi lebih terarah dan mudah diikuti.

Kode Program 10: Struktur Navigasi Utama

```
<nav class="sidebar" id="sidebar">
  <div class="logo">GLOD</div>
  <div class="menu">
    <a href="/" class="nav-link" data-menu="beranda">
      <span class="icon">🏠</span>Beranda    </a>
    <a href="/uniprot/input/" class="nav-link" data-menu="input-data">
      <span class="icon">💉</span>Input Data Gen    </a>
    <a       href="/preprocessing/"       class="nav-link"       data-
menu="preprocessing">
      <span class="icon">⚙️</span>Persiapan Data    </a>
    <a href="/string/" class="nav-link" data-menu="string">
      <span class="icon">🔗</span>STRING    </a>
    <a href="/glod/process/" class="nav-link" data-menu="glod">
      <span class="icon">🧩</span>Analisis GLOD
    </a>  </div> </nav>
```

Kode Program 11 menjelaskan form input parameter algoritma GLOD, yaitu nilai *alpha* dan *jaccard threshold*. Melalui form ini, pengguna dapat menentukan sensitivitas fungsi fitness serta ambang batas penggabungan komunitas. Setelah parameter diisikan, pengguna dapat menjalankan algoritma dengan menekan tombol "Jalankan Algoritma GLOD". Form ini menjadi penghubung antara antarmuka pengguna dan proses komputasi di sisi backend, sehingga pengguna tanpa latar belakang pemrograman tetap dapat mengatur parameter analisis secara langsung melalui halaman web.

Kode Program 11: Form *GLOD* dan pembuatan tabel

```
<form method="post" action="{% url 'glod_result' %}">
  {% csrf_token %}
  <input
    type="number"
    class="form-control"
    id="alpha"
    name="alpha"
    value="{{ default_alpha }}"
    step="0.01"
    min="0.1"
    max="2.0"
    required
  />

  <input
    type="number"
    class="form-control"
    id="jaccard_threshold"
    name="jaccard_threshold"
    value="{{ default_threshold }}"
    step="0.01"
    min="0.0"
    max="1.0"
    required
  />

  <button type="submit" class="btn btn-primary btn-lg">
    <i class="fas fa-play-circle"></i> Jalankan Algoritma GLOD
  </button>
</form>
```

B. Integrasi Sistem

Kode Program 12 menjelaskan proses penampilan hasil komunitas protein dalam bentuk tabel pada halaman hasil. Setiap baris tabel merepresentasikan satu komunitas, yang memuat informasi berupa ID komunitas, jumlah anggota, jumlah protein overlap, daftar anggota protein, protein yang tumpang tindih antar komunitas, serta nilai *Normalized Node Cut (Ψ)*. Selain itu, disediakan tombol aksi untuk menampilkan visualisasi komunitas tertentu. Penyajian dalam bentuk tabel ini bertujuan agar hasil analisis mudah dibaca dan dibandingkan antar komunitas.

Kode Program 12: Backend untuk Menampilkan Hasil Komunitas

```
<table class="table table-striped table-hover">
  <thead class="table-dark">
    <tr>
      <th>Komunitas ID</th>
      <th>Jumlah Anggota</th>
      <th>Jumlah Overlap</th>
      <th>Anggota Protein</th>
      <th>Protein Overlap</th>
      <th>Normalized Node Cut (Ψ)</th>
      <th>Aksi</th>
    </tr></thead>
  <tbody>
    {% for community in communities %}
    <tr id="community-row-{{ community.id }}">
      <td>Komunitas {{ community.id }}</td>
      <td>{{ community.size }}</td>
      <td>{{ community.overlap_count }}</td>
      <td>
        {% for member in community.members %}
        {{ member }}
        {% endfor %}
      </td>
      <td>
        {% for member in community.overlap_members %}
        {{ member }}
        {% endfor %}
      </td>
      <td>{{ community.psi }}</td>
      <td>
        <button          class="btn          btn-sm          btn-primary"
onclick="toggleCommunityVisualization({{ community.id }})">
          Tampilkan Komunitas
        </button>
      </td>
    </tr>
    {% endfor %}
  </tbody>
</table>
```

Kode Program 13 menjelaskan integrasi antara antarmuka pengguna dan algoritma GLOD di sisi backend. Pada tahap ini, sistem membaca nilai *alpha* dan *jaccard threshold* dari form, kemudian membangun graf jaringan menggunakan NetworkX berdasarkan data node dan edge yang tersimpan dalam session. Selanjutnya, algoritma GLOD dijalankan untuk menghasilkan komunitas protein.

Kode Program 13: Frontend untuk Menampilkan Hasil Komunitas

```
alpha = float(request.POST.get('alpha', 0.8))
jaccard_threshold = float(request.POST.get('jaccard_threshold', 0.33))

network_data = request.session.get('glod_network_data')
if not network_data:
    return render(request, 'glod_app/result.html', {'error': 'Data
jaringan tidak ditemukan. Silakan ulangi dari awal.'})

G = nx.Graph()
for node in network_data['nodes']:
    G.add_node(node['id'])
for edge in network_data['edges']:
    G.add_edge(edge['source'], edge['target'])

glod = GLODAlgorithm(G, alpha=alpha, jaccard_threshold=jaccard_threshold)
communities, shen_mod, lazar_mod, nicosia_mod = glod.run(seed_value=42)

psi_scores = [glod.calculate_psi_normalized_node_cut(c) for c in
communities]
avg_psi = sum(psi_scores) / len(psi_scores) if psi_scores else 0.0

community_results = []
node_community_count = {}
for community in communities:
    for node in community:
        node_community_count[node] = node_community_count.get(node, 0) +
1
overlapping_nodes = set(node for node, count in
node_community_count.items() if count > 1)

for idx, community in enumerate(communities, 1):
    psi_value = glod.calculate_psi_normalized_node_cut(community)
    overlap_in_community = community & overlapping_nodes
    community_results.append({
        'id': idx,
        'size': len(community),
        'members': sorted(list(community)),
        'overlap_count': len(overlap_in_community),
        'overlap_members': sorted(list(overlap_in_community)),
        'psi': round(psi_value, 4)
    })

context = {
    'num_communities': len(communities),
    'shen_modularity': round(shen_mod, 4),
    'lazar_modularity': round(lazar_mod, 4),
    'nicosia_modularity': round(nicosia_mod, 4),
    'avg_psi_node_cut': round(avg_psi, 4),
    'alpha': alpha,
    'jaccard_threshold': jaccard_threshold,
    'communities': community_results,
    'communities_json': json.dumps(community_results),
}
return render(request, 'glod_app/result.html', context)
```

4.2 Hasil

Pada bagian ini disajikan hasil dari seluruh tahapan penelitian yang telah dilakukan berdasarkan kerangka kerja yang dijelaskan pada bab sebelumnya. Hasil penelitian dipaparkan secara bertahap mulai dari *business understanding*, *data understanding*, *data preparation*, proses pemodelan menggunakan algoritma GLOD, hingga tahap evaluasi dan *deployment*. Setiap tahapan menampilkan hasil yang diperoleh dari proses pengolahan data jaringan interaksi protein kanker payudara, mulai dari proses pengumpulan dan pemrosesan data, pembentukan jaringan interaksi protein, hingga deteksi komunitas protein yang saling tumpang tindih. Penyajian hasil pada bagian ini bertujuan untuk menunjukkan bagaimana metode yang digunakan dalam penelitian mampu menghasilkan struktur komunitas protein yang kemudian dianalisis lebih lanjut baik secara struktural maupun biologis.

Hasil yang disajikan pada Subbab 4.2 merupakan implementasi langsung dari seluruh tahapan sistem yang telah dirancang pada Bab III, termasuk proses pengujian yang dilakukan terhadap setiap komponen sistem. Setiap tahap, mulai dari data understanding, preprocessing, konstruksi jaringan, hingga pemodelan dan evaluasi komunitas, menghasilkan keluaran yang sesuai dengan yang diharapkan. Hal ini menunjukkan bahwa sistem yang dikembangkan telah berhasil berjalan secara terintegrasi dan mampu mengimplementasikan algoritma GLOD untuk mendeteksi komunitas overlap pada jaringan interaksi protein kanker payudara.

4.2.1. Business Understanding

Pada tahap awal penelitian ini, dilakukan analisis mendalam mengenai permasalahan biologis dan teknis yang melatarbelakangi perlunya deteksi komunitas pada jaringan interaksi protein kanker payudara. Hasil dari tahap *Business Understanding* ini merumuskan bahwa:

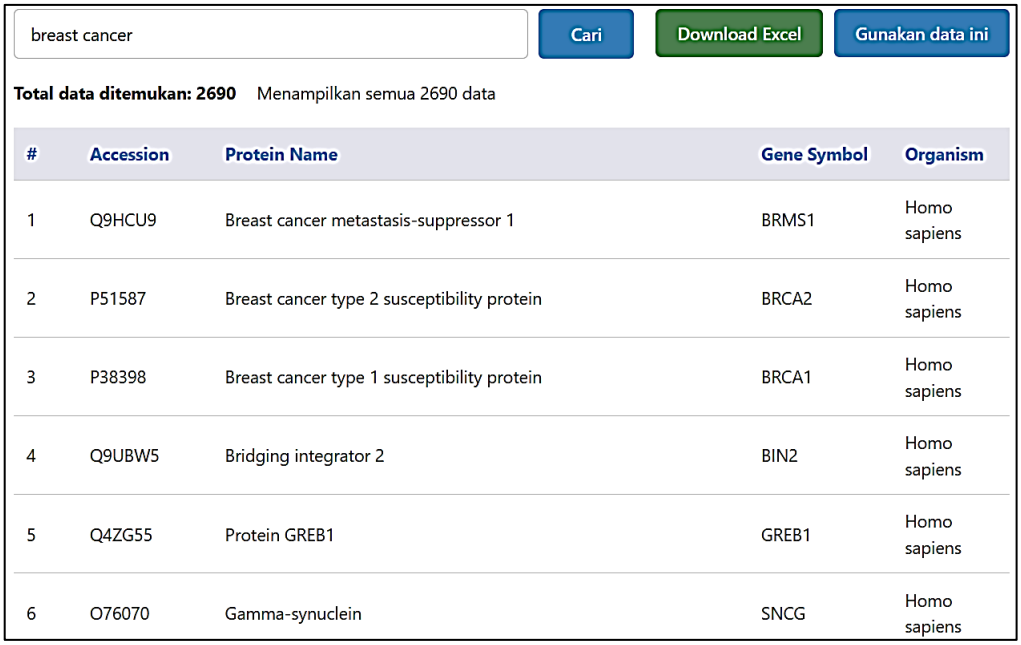
1. Metode deteksi komunitas lokal yang ada saat ini memiliki kecenderungan menghasilkan komunitas yang tidak stabil dan redundan.
2. Algoritma berbasis ekspansi benih sering kali sensitif terhadap pemilihan simpul awal, yang menyebabkan hasil deteksi menjadi bias dan sulit diinterpretasikan secara biologis.
3. Tumpang tindih yang berlebihan (*excessive overlapping*) sering kali mengaburkan batas antar fungsi biologis yang sebenarnya.

Oleh karena itu, tujuan utama dari penelitian ini ditetapkan untuk menerapkan algoritma Local Greedy Extended Dynamic Overlapping Community Detection (GLOD). Pemilihan algoritma ini didasarkan pada kemampuannya untuk menangani masalah ketidakstabilan melalui mekanisme seeding yang kuat dan mengurangi redundansi melalui tahap merging. Dengan indikator keberhasilan penelitian yang mencakup validasi struktural dan relevansi biologis.

4.2.2. Data Understanding

Pada tahap Data Understanding, dilakukan pengumpulan data yang menjadi objek penelitian. Data diperoleh melalui API dari basis data UniProt. Pengambilan data dilakukan dengan menggunakan kata kunci pencarian breast cancer. Hasil dari proses akuisisi data ini berhasil mengumpulkan total 2.690 data protein yang berasosiasi dengan kanker payudara. Setiap entri data yang diperoleh memiliki struktur atribut yang *mencakup Entry ID*

(Accession), Protein Name, Gene Symbol dan Organism seperti yang ditunjukkan oleh Gambar 4.1. Dari atribut-atribut tersebut, Gene Symbol diidentifikasi sebagai atribut kunci yang akan digunakan sebagai identitas utama dalam pembentukan jaringan interaksi protein.



#	Accession	Protein Name	Gene Symbol	Organism
1	Q9HCU9	Breast cancer metastasis-suppressor 1	BRMS1	Homo sapiens
2	P51587	Breast cancer type 2 susceptibility protein	BRCA2	Homo sapiens
3	P38398	Breast cancer type 1 susceptibility protein	BRCA1	Homo sapiens
4	Q9UBW5	Bridging integrator 2	BIN2	Homo sapiens
5	Q4ZG55	Protein GREB1	GREB1	Homo sapiens
6	O76070	Gamma-synuclein	SNCG	Homo sapiens

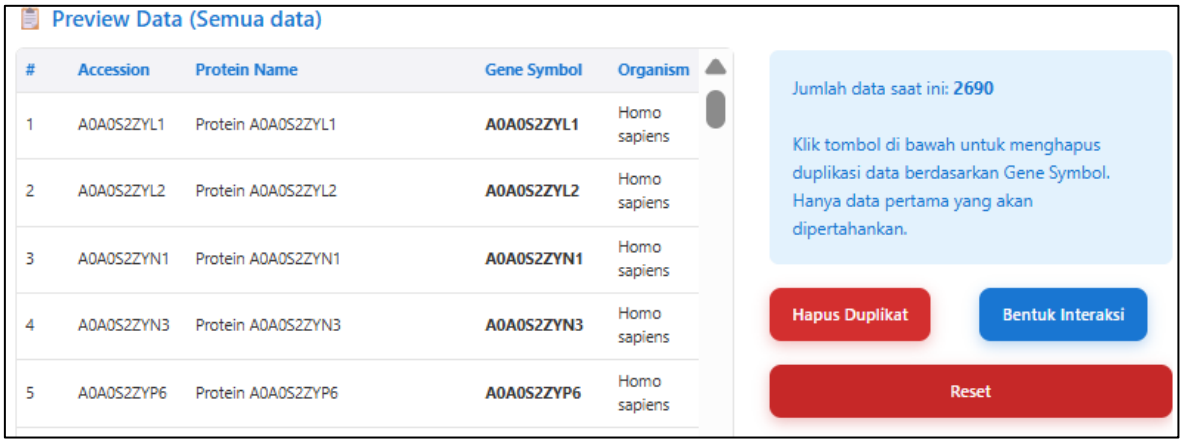
Gambar 4. 1 Daftar protein

4.2.3. Data Preparation

Pada tahap ini dilakukan tahap persiapan data agar siap digunakan untuk proses selanjutnya. Tahapan ini terdiri dari:

1. Data Selection

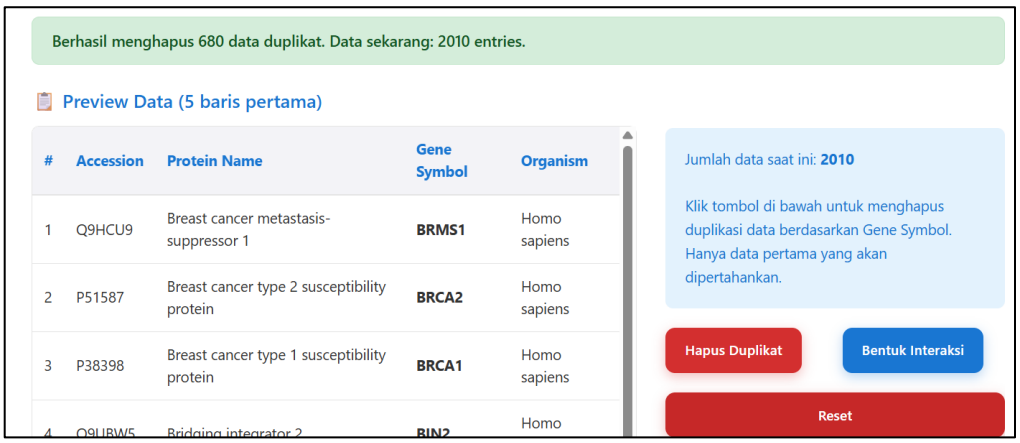
Langkah pertama adalah menyeleksi atribut yang relevan dari 2.690 data mentah yang diperoleh dari UniProt. Aatribut Gene Symbol dipilih sebagai representasi unik setiap protein. Atribut ini diekstraksi dari keseluruhan dataset karena Gene Symbol merupakan identitas standar yang digunakan dalam basis data interaksi protein seperti STRING DB, sehingga memudahkan proses pemetaan pada tahap selanjutnya. Berikut adalah hasil dari tahapan data selection ditunjukkan oleh Lampiran A.1 sampai A.7 dan Gambar 4.2.



Gambar 4. 2 Hasil data selection

2. Data Deduplication

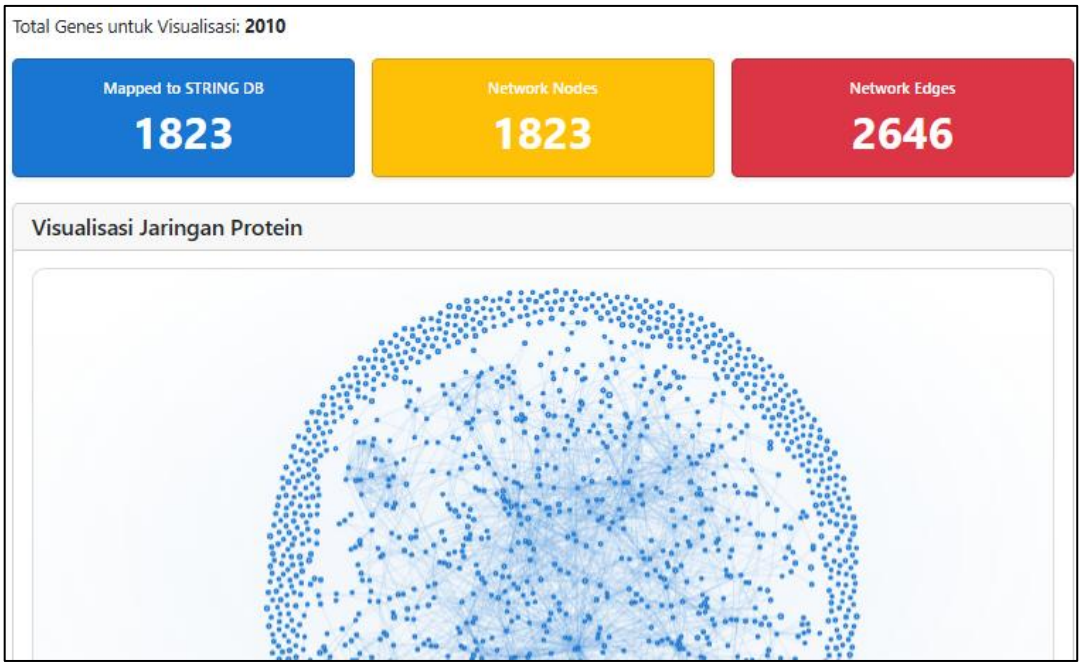
Setelah atribut Gene Symbol dipilih, dilakukan proses pembersihan data untuk menghilangkan data duplikat. Pada dataset awal, sering kali ditemukan duplikasi protein yang dapat menyebabkan bias dalam analisis struktur jaringan. Proses deduplikasi dilakukan dengan menghapus entri yang memiliki Gene Symbol yang sama, sehingga memastikan setiap simpul dalam jaringan merepresentasikan satu entitas biologis yang unik. Hasil dari proses ini mereduksi jumlah data dari 2.690 data mentah menjadi 2.010 data protein unik (Gambar 4.3). Data lengkap hasil proses ini bisa dilihat pada Lampiran B.1 sampai B.4. Data protein unik inilah yang kemudian digunakan sebagai masukan untuk membangun jaringan interaksi.



Gambar 4. 3 Hasil tahap *preprocessing data*

3. Network Construction

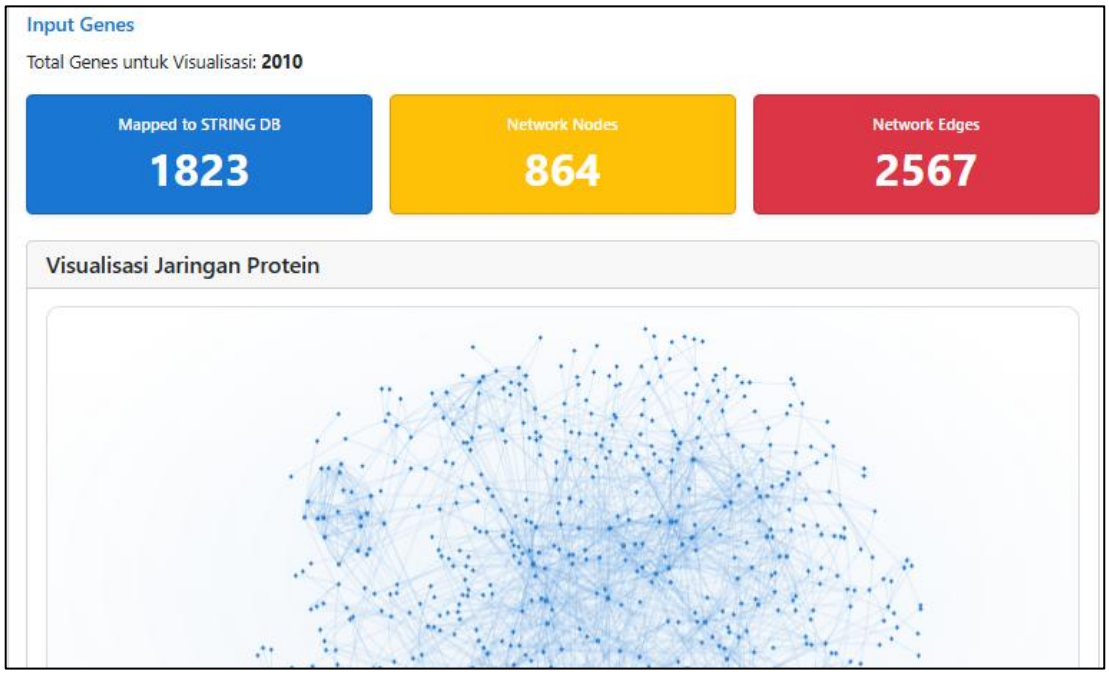
Konstruksi jaringan interaksi protein dilakukan dengan memetakan 2.010 data protein unik ke dalam basis data STRING DB melalui API. Untuk menjamin relevansi biologis yang tinggi dan meminimalkan interaksi positif palsu (*false positives*), parameter *confidence score* ditetapkan pada tingkat tertinggi (*highest confidence*), yaitu 0,900, dengan organisme referensi *Homo sapiens*. Berdasarkan parameter tersebut, STRING DB mengidentifikasi interaksi fisik dan fungsional antar protein. Dari 2.010 gen yang dipetakan, tidak semuanya memiliki interaksi yang memenuhi ambang batas skor kepercayaan 0,900. Hasil konstruksi jaringan awal menghasilkan graf yang terdiri dari 1.823 simpul dan 2.646 sisi. Visualisasi jaringan pada tahap ini (Gambar 4.4) menunjukkan adanya struktur yang masih menyebar, di mana terdapat banyak protein yang terisolasi atau hanya membentuk kelompok-kelompok kecil yang terpisah dari jaringan utama.



Gambar 4. 4 *Hasil network Construction*

4. Pemilihan Giant Component

Jaringan biologis alami sering kali memiliki komponen-komponen kecil yang terpisah (terisolasi) yang tidak memberikan informasi signifikan mengenai struktur topologi global jaringan. Oleh karena itu, dilakukan penyaringan dengan mengambil Giant Component, yaitu sub-graf terhubung terbesar dalam jaringan. Proses ekstraksi ini menghilangkan simpul-simpul yang terisolasi dan kelompok kecil yang tidak terhubung ke komponen utama. Hasil akhir dari pemilihan Giant Component menghasilkan jaringan final yang lebih padat dan terhubung sepenuhnya, yang terdiri dari 864 simpul dan 2.567 sisi (Gambar 4.5).



Gambar 4. 5 Hasil pemilihan *Giant component*

4.2.4. Modelling Algoritma GLOD

Berdasarkan rancangan eksperimen (*experimental setup*) yang telah dijelaskan pada Bab III, pemilihan parameter terbaik dilakukan melalui serangkaian percobaan menggunakan berbagai kombinasi nilai α (*alpha*) dan *merging threshold* berbasis koefisien Jaccard (J). Proses evaluasi untuk menentukan parameter terbaik tersebut dipaparkan secara rinci pada Subbab 4.2.5. Berdasarkan hasil evaluasi tersebut, diperoleh konfigurasi parameter terbaik yaitu α = 0,75 dan Jaccard threshold = 0,20, dengan nilai Normalized Node Cut (NNC) sebesar 0,3691. Konfigurasi parameter ini selanjutnya digunakan pada tahap pemodelan untuk menghasilkan struktur komunitas pada jaringan interaksi protein kanker payudara. Dengan menggunakan parameter terbaik tersebut, algoritma Local Greedy Extended Dynamic Overlapping Community Detection (GLOD) diterapkan pada jaringan hasil *giant component* yang terdiri dari 864 simpul (*nodes*) dan 2.567 sisi (*edges*). Hasil penerapan algoritma menunjukkan bahwa jaringan berhasil dibagi ke dalam 16 komunitas protein yang saling tumpang tindih (*overlapping communities*). Parameter yang digunakan pada proses deteksi komunitas adalah sebagai berikut:

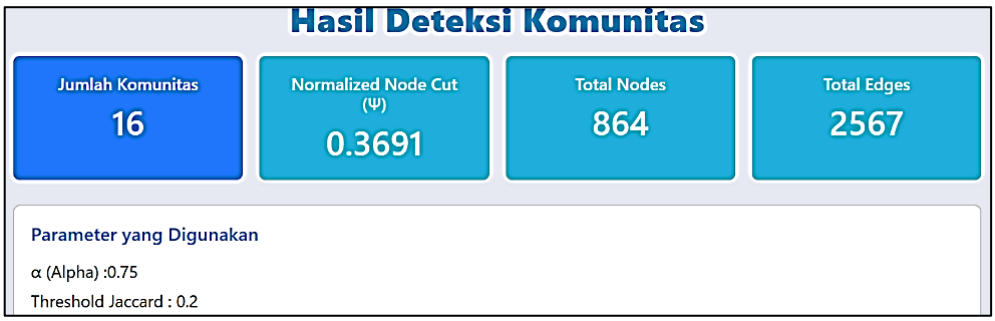
1. Konfigurasi Parameter

   Berdasarkan hasil pengujian pada antarmuka sistem, deteksi komunitas yang optimal diperoleh dengan menggunakan konfigurasi parameter sebagai berikut:

1) Resolution Parameter (alpha): Ditetapkan sebesar 0,75. Nilai ini dipilih untuk menyeimbangkan kepadatan internal komunitas dengan konektivitas eksternalnya, sehingga komunitas yang terbentuk tidak terlalu terpecah (granular) maupun terlalu besar.
2) Merging Threshold (Jaccard): Ditetapkan sebesar 0,2. Nilai ambang batas ini digunakan pada fase merging untuk menggabungkan komunitas-komunitas yang memiliki tingkat tumpang tindih anggota di atas 20% guna mengurangi redundansi hasil.

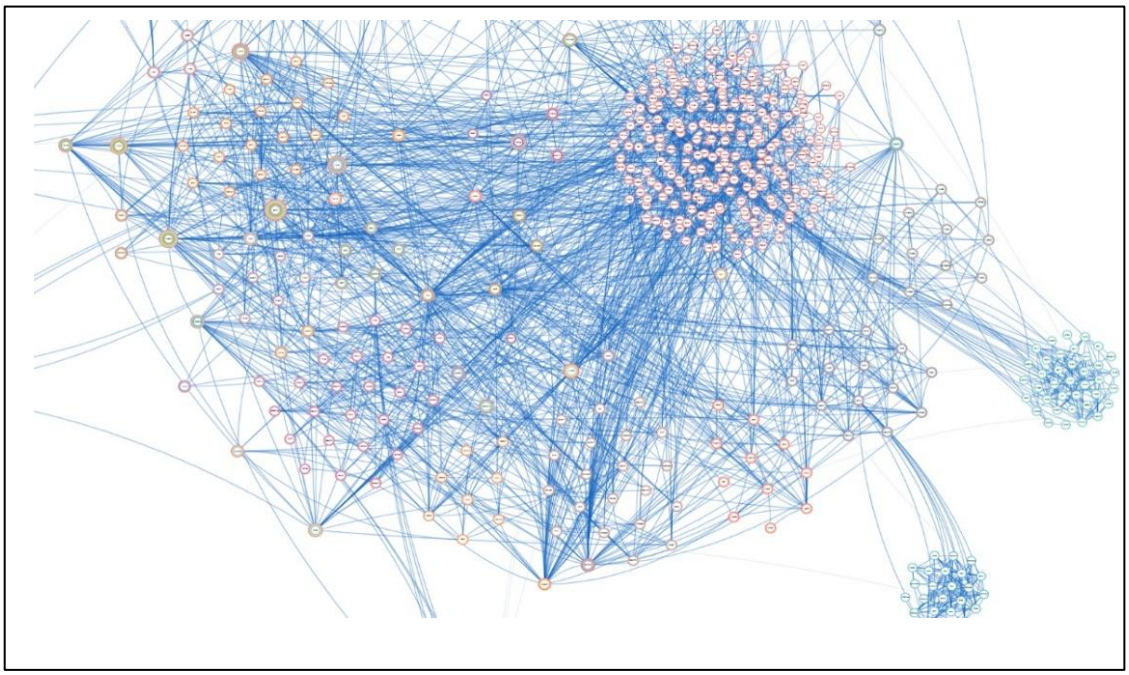
2. Hasil Deteksi

Penerapan algoritma GLOD dengan konfigurasi parameter terbaik berhasil mengidentifikasi struktur topologi jaringan yang terbagi ke dalam 16 komunitas yang saling tumpang tindih (*overlapping communities*). Kualitas struktur komunitas dievaluasi menggunakan metrik Normalized Node Cut (NNC) dan menghasilkan nilai sebesar 0,3691.Hasil deteksi komunitas secara keseluruhan ditunjukkan pada Gambar 4.6. Gambar tersebut memperlihatkan pembagian jaringan ke dalam komunitas-komunitas yang terbentuk berdasarkan hasil penerapan algoritma GLOD.



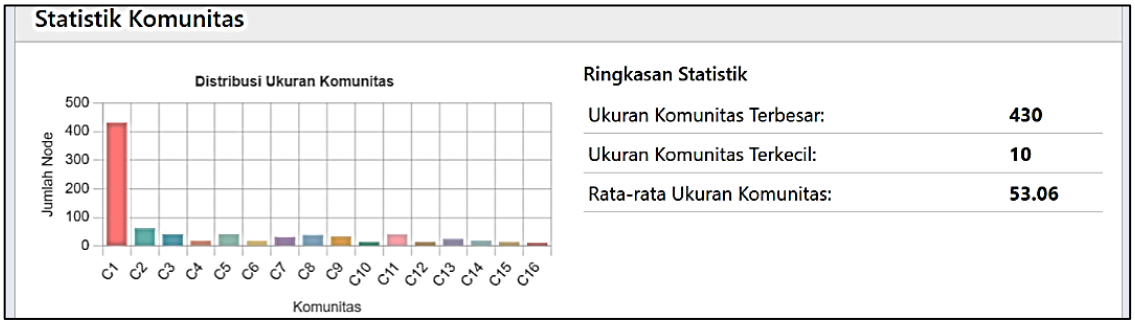
Gambar 4. 6 Hasil deteksi komunitas

Visualisasi jaringan hasil deteksi komunitas ditampilkan pada Gambar 4.7. Pada gambar ini terlihat struktur jaringan protein beserta keterhubungan antar simpul yang membentuk komunitas-komunitas yang berbeda. Warna yang berbeda menunjukkan keanggotaan komunitas yang berbeda, sehingga memudahkan dalam mengidentifikasi pola pengelompokan dalam jaringan.



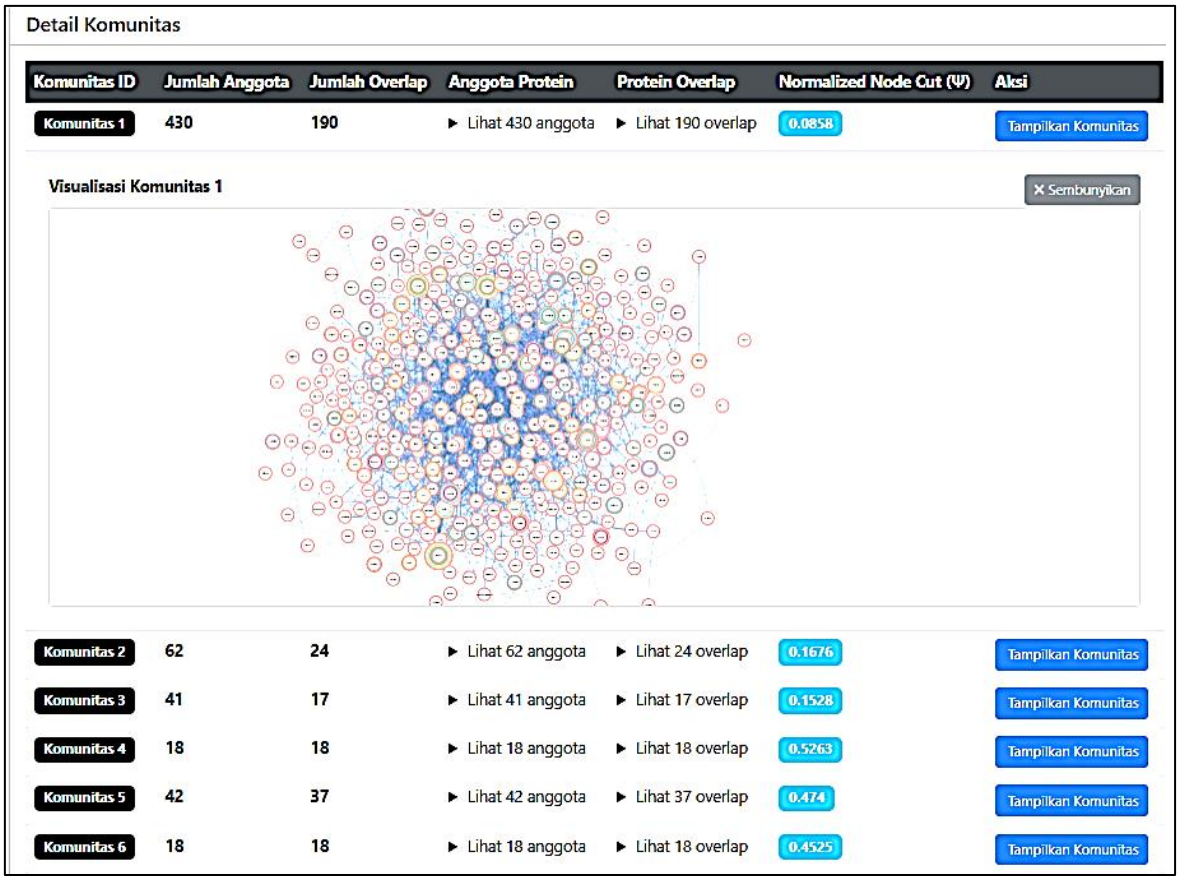
Gambar 4. 7 Visualisasi jaringan

Statistik komunitas yang dihasilkan ditunjukkan pada Gambar 4.8. Gambar ini menyajikan distribusi ukuran komunitas, yang menggambarkan variasi jumlah anggota pada setiap komunitas. Informasi ini penting untuk memahami karakteristik komunitas yang terbentuk, apakah cenderung kecil, besar, atau bervariasi.



Gambar 4. 8 Statistik komunitas

Selanjutnya, detail anggota pada setiap komunitas ditampilkan pada Gambar 4.9. Gambar ini memberikan informasi mengenai komposisi anggota pada masing-masing komunitas, termasuk jumlah protein dan keterlibatan simpul dalam lebih dari satu komunitas (*overlapping nodes*).



Gambar 4. 9 Detail komunitas

Berdasarkan hasil visualisasi tersebut, dapat dilihat bahwa jaringan protein berhasil dibagi menjadi beberapa komunitas dengan ukuran dan tingkat tumpang tindih yang berbeda-beda. Hasil ini menunjukkan bahwa algoritma GLOD mampu mendeteksi komunitas overlapping pada jaringan interaksi protein kanker payudara, sehingga menjawab rumusan masalah pertama dalam penelitian ini.

Selanjutnya, untuk mendukung proses evaluasi kualitas komunitas, dilakukan identifikasi anggota protein pada setiap komunitas berdasarkan keluaran program algoritma GLOD dengan menggunakan konfigurasi parameter terbaik. Hasil identifikasi ini mencakup informasi jumlah komunitas, jumlah protein pada setiap komunitas, serta anggota protein yang tergabung di dalamnya. Karena jumlah data yang cukup besar, detail anggota komunitas tidak disajikan pada bagian utama, melainkan disertakan pada lampiran. Daftar lengkap anggota komunitas dengan nilai Normalized Node Cut (NNC) terbaik dapat dilihat pada Lampiran C, yang terdiri dari Lampiran C.1 hingga Lampiran C.3.

4.2.5. Evaluation

Tahap evaluasi dilakukan untuk menilai kualitas komunitas yang dihasilkan sekaligus menentukan konfigurasi parameter terbaik berdasarkan *experimental setup* yang telah dirancang pada Bab III. Eksperimen dilakukan dengan menguji berbagai kombinasi parameter α (*alpha*) dan *merging threshold* berbasis koefisien Jaccard (J). Pada setiap kombinasi parameter, algoritma GLOD dijalankan dan menghasilkan struktur komunitas yang kemudian dievaluasi menggunakan metrik Normalized Node Cut (NNC). Nilai NNC

yang lebih kecil menunjukkan kualitas komunitas yang lebih baik. Hasil dari seluruh percobaan berdasarkan kombinasi parameter dirangkum pada Tabel 4.2.

Tabel 4. 2 Hasil kombinasi parameter experimental setup

Eksperimen	α	J	NNC (Ψ)
Eksperimen 1	0.70	0.20	0.3807
Eksperimen 2	0.70	0.25	0.4071
Eksperimen 3	0.70	0.30	0.3949
Eksperimen 4	0.70	0.33	0.3933
Eksperimen 5	0.75	0.20	0.3691
Eksperimen 6	0.75	0.25	0.3934
Eksperimen 7	0.75	0.30	0.3863
Eksperimen 8	0.75	0.33	0.3957
Eksperimen 9	0.80	0.20	0.3693
Eksperimen 10	0.80	0.25	0.3902
Eksperimen 11	0.80	0.30	0.3948
Eksperimen 12	0.80	0.33	0.4017
Eksperimen 13	0.85	0.20	0.3695
Eksperimen 14	0.85	0.25	0.4168
Eksperimen 15	0.85	0.30	0.3953
Eksperimen 16	0.85	0.33	0.4017

Tabel tersebut menunjukkan bahwa setiap kombinasi parameter menghasilkan nilai NNC yang berbeda, yang mengindikasikan bahwa pemilihan parameter berpengaruh terhadap kualitas struktur komunitas yang dihasilkan. Berdasarkan hasil tersebut, konfigurasi parameter terbaik ditentukan dengan memilih nilai NNC terkecil. Dari seluruh kombinasi yang diuji, diperoleh bahwa nilai NNC minimum sebesar 0,3691 dicapai pada kombinasi parameter α = 0,75 dan Jaccard threshold = 0,20. Nilai NNC yang lebih kecil menunjukkan bahwa komunitas memiliki kepadatan koneksi internal yang lebih tinggi dibandingkan koneksi eksternal, sehingga struktur komunitas menjadi lebih kohesif dan terdefinisi dengan baik. Oleh karena itu, kombinasi parameter tersebut dipilih sebagai konfigurasi optimal dalam penelitian ini. Konfigurasi parameter terbaik ini selanjutnya digunakan pada tahap pemodelan (Subbab 4.2.4) untuk menghasilkan struktur komunitas akhir, serta digunakan pada evaluasi lanjutan yang meliputi evaluasi struktural pada tingkat komunitas dan validasi biologis melalui *enrichment analysis*.

Tahap evaluasi ini secara langsung menjawab rumusan masalah kedua, yaitu bagaimana mengevaluasi kualitas struktur komunitas overlap yang dihasilkan oleh algoritma GLOD pada jaringan PPI kanker payudara, yang dilakukan melalui dua pendekatan, yaitu evaluasi struktural menggunakan metrik Normalized Node Cut serta evaluasi biologis menggunakan enrichment analysis.

1. Evaluasi Struktural (Normalized Node Cut)

Evaluasi struktural dilakukan untuk menilai kualitas komunitas yang dihasilkan berdasarkan pola keterhubungan antar protein dalam jaringan. Metrik yang digunakan adalah Normalized Node Cut (NNC), yang mengukur keseimbangan antara koneksi internal dalam komunitas dan koneksi eksternal ke komunitas lain. Nilai NNC yang lebih kecil menunjukkan bahwa komunitas memiliki keterhubungan internal yang lebih kuat dibandingkan koneksi eksternal, sehingga struktur komunitas lebih jelas dan terdefinisi dengan baik. Berdasarkan hasil experimental setup yang dipaparkan pada Subbab 4.2.5, diperoleh bahwa konfigurasi parameter terbaik menghasilkan 16 komunitas dengan nilai NNC keseluruhan sebesar 0.3691. Nilai ini menunjukkan bahwa komunitas yang terbentuk memiliki kualitas struktur yang cukup baik, di mana interaksi antar protein di dalam komunitas lebih dominan dibandingkan dengan interaksi ke luar komunitas. Untuk analisis

yang lebih rinci, dilakukan perhitungan nilai NNC pada setiap komunitas sebagaimana ditunjukkan pada Tabel 4.3.

Tabel 4. 3 Normalized node cut tiap komunitas

Komunitas Ke-	Total Protein	Jumlah Protein Overlap	Normalized Node Cut
1	430	190	0.0858
2	62	24	0.1676
3	41	17	0.1528
4	18	18	0.5263
5	42	37	0.474
6	18	18	0.4525
7	31	31	0.2326
8	38	35	0.5194
9	33	31	0.5116
10	14	14	0.6503
11	41	18	0.3625
12	14	13	0.5782
13	24	13	0.3073
14	19	12	0.2527
15	14	4	0.1143
16	10	9	0.5181

Dari hasil pada Tabel 4.3 menunjukkan bahwa nilai NNC pada masing-masing komunitas bervariasi, yaitu berkisar antara 0.0858 hingga 0.6503. Komunitas dengan nilai NNC rendah, seperti komunitas 1, 2, 3, dan 15, menunjukkan struktur yang lebih padat dan terhubung kuat karena didominasi oleh interaksi internal. Sebaliknya, komunitas dengan nilai NNC yang lebih tinggi menunjukkan bahwa masih terdapat keterhubungan yang cukup besar dengan komunitas lain. Variasi nilai ini menunjukkan bahwa struktur komunitas dalam jaringan biologis bersifat kompleks dan tidak seragam, di mana beberapa komunitas memiliki batas yang sangat jelas, sementara komunitas lain cenderung lebih terbuka dan saling terhubung. Secara keseluruhan, hasil evaluasi struktural ini menunjukkan bahwa algoritma GLOD mampu menghasilkan komunitas overlap dengan kualitas struktur yang baik pada jaringan interaksi protein kanker payudara, sehingga berhasil menjawab rumusan masalah kedua dari sisi evaluasi struktural.

2. Enrichment Analysis

Setelah evaluasi struktural dilakukan, tahap selanjutnya adalah evaluasi biologis menggunakan *enrichment analysis*. Tahap ini bertujuan untuk mengetahui apakah komunitas protein yang dihasilkan tidak hanya memiliki kualitas struktur jaringan yang baik, tetapi juga memiliki keterkaitan fungsi biologis yang relevan dengan mekanisme kanker payudara. Tahap ini merupakan implementasi langsung dari metode yang telah dirancang pada Bab III, khususnya pada Subbab 3.5.2. Pada bab tersebut dijelaskan bahwa analisis dilakukan menggunakan platform Metascape dengan memasukkan daftar protein dari setiap komunitas, memilih spesies *Homo sapiens*, serta mengatur parameter analisis seperti *min overlap*, *p-value cutoff*, dan *min enrichment*. Selain itu, digunakan empat kategori utama dalam analisis, yaitu *Gene Ontology Biological Process (GO-BP)*, *Gene Ontology Molecular Function (GO-MF)*, *Gene Ontology Cellular Component (GO-CC)*, serta *KEGG Pathway*.

Pada tahap ini, seluruh prosedur tersebut diterapkan pada hasil komunitas terbaik yang diperoleh dari experimental setup, yaitu pada kombinasi parameter $\alpha = 0.75$ dan *Jaccard threshold* = 0.20, yang menghasilkan 16 komunitas protein. Dengan demikian, tahap ini merupakan kelanjutan langsung dari metode yang telah dirancang sebelumnya, sehingga menunjukkan keterkaitan yang jelas antara perancangan metode pada Bab III dan hasil yang diperoleh pada Bab IV. Hasil *enrichment analysis* untuk setiap komunitas kemudian dirangkum pada Tabel 4.4 dan Tabel 4.5, yang menyajikan hasil paling signifikan dari masing-masing kategori anotasi biologis.

Tabel 4. 4 Hasil enrichment analysis paling signifikan

Komunitas	GO Biological Process	GO Cellular Component	GO Molecular Function	KEGG Pathway
1	Regulasi siklus sel mitotik (19,39%, −59,48)	–	Aktivitas protein kinase (17,06%, −46,50)	Jalur dalam kanker (21,50%, −71,63)
2	Translasi sitoplasmik (25,81%, −23,59)	Pre-ribosom subunit kecil (4,84%, −5,53)	Pengikatan protein bergantung poliubiquitin (8,06%, −6,93)	Nekroptosis (11,29%, −7,26)
3	Perakitan kompleks NADH dehidrogenase (17,50%, −13,05)	Kompleks ATPase pengangkut proton (10,00%, −6,20)	–	Fosforilasi oksidatif (50,00%, −36,33)
4	Regulasi adhesi antar sel (58,82%, −13,35)	Kompleks regulator transkripsi (47,06%, −9,65)	Pengikatan reseptor sitokin (47,06%, −11,97)	Jalur pensinyalan JAK-STAT (82,35%, −28,99)
5	Respon sel terhadap stimulus hormon (47,62%, −23,17)	Rak membran plasma (19,05%, −11,38)	Aktivitas protein kinase serin/treonin (30,95%, −13,70)	Jalur dalam kanker (54,76%, −29,03)
6	Regulasi positif transisi epitel-mesenkimal (50,00%, −19,98)	–	Pengikatan beta-katenin (33,33%, −10,49)	Kanker kolorektal (66,67%, −26,57)
7	Proses siklus sel mitotik (66,67%, −28,23)	Midbody (26,67%, −11,00)	Aktivitas histone kinase (13,33%, −7,87)	Siklus sel (46,67%, −24,07)
8	Regulasi sitoskeleton aktin (47,37%, −27,89)	Korteks sel (36,84%, −17,82)	–	Jalur dalam kanker (52,63%, −24,83)
9	Regulasi autofagi (39,39%, −16,23)	Kompleks fosfatidilinositol 3-kinase (15,15%, −10,89)	Aktivitas protein kinase serin (39,39%, −16,41)	Autofagi – hewan (45,45%, −25,09)
10	Regulasi proliferasi sel epitel (42,86%, −7,97)	Rak membran (57,14%, −12,72)	–	Proteoglikan dalam kanker (64,29%, −16,33)
11	Regulasi negatif adhesi sel (29,27%, −13,84)	Matriks ekstraseluler (41,46%, −19,25)	Pengikatan kinase (31,71%, −10,79)	Proteoglikan dalam kanker (41,46%, −26,08)
12	Organisasi sambungan sel (57,14%, −10,38)	Membran plasma basolateral (57,14%, −12,95)	–	Kanker endometrium (50,00%, −15,60)

Tabel 4. 5 Hasil enrichment analysis untuk seluruh komunitas (Lanjutan)

Komunitas	GO Biological Process	GO Cellular Component	GO Molecular Function	KEGG Pathway
13	Organisasi sambungan sel (29,17%, −6,66)	Sitoskeleton aktin (41,67%, −11,63)	Pengikatan kalmodulin (20,83%, −6,28)	Regulasi sitoskeleton aktin (37,50%, −13,03)
14	Remodeling kromatin (89,47%, −25,46)	–	Penyusun struktural kromatin (57,89%, −22,76)	Karsinogenesis virus (21,05%, −5,13)
15	Perkembangan embrio kordata (21,43%, −2,46)	Kompleks ekspor transkripsi (57,14%, −23,12)	Aktivitas isomerase konformasi asam nukleat (28,57%, −6,15)	–
9	Regulasi autofagi (39,39%, −16,23)	Kompleks fosfatidilinositol 3-kinase (15,15%, −10,89)	Aktivitas protein kinase serin (39,39%, −16,41)	Autofagi – hewan (45,45%, −25,09)
10	Regulasi proliferasi sel epitel (42,86%, −7,97)	Rak membran (57,14%, −12,72)	–	Proteoglikan dalam kanker (64,29%, −16,33)
11	Regulasi negatif adhesi sel (29,27%, −13,84)	Matriks ekstraseluler (41,46%, −19,25)	Pengikatan kinase (31,71%, −10,79)	Proteoglikan dalam kanker (41,46%, −26,08)
12	Organisasi sambungan sel (57,14%, −10,38)	Membran plasma basolateral (57,14%, −12,95)	–	Kanker endometrium (50,00%, −15,60)
13	Organisasi sambungan sel (29,17%, −6,66)	Sitoskeleton aktin (41,67%, −11,63)	Pengikatan kalmodulin (20,83%, −6,28)	Regulasi sitoskeleton aktin (37,50%, −13,03)
14	Remodeling kromatin (89,47%, −25,46)	–	Penyusun struktural kromatin (57,89%, −22,76)	Karsinogenesis virus (21,05%, −5,13)
15	Perkembangan embrio kordata (21,43%, −2,46)	Kompleks ekspor transkripsi (57,14%, −23,12)	Aktivitas isomerase konformasi asam nukleat (28,57%, −6,15)	–
16	Regulasi positif kaskade MAPK (40,00%, −4,91)	Adhesi fokal (70,00%, −11,08)	–	Infeksi sitomegalovirus manusia (70,00%, −12,84)

Berdasarkan Tabel 4.4 dan Tabel 4.5, terlihat bahwa setiap komunitas memiliki karakteristik fungsi biologis yang berbeda. Sebagai contoh, Komunitas 1 menunjukkan keterkaitan dengan proses regulasi siklus sel mitotik, yang merupakan mekanisme penting dalam pembelahan sel dan sering mengalami gangguan pada sel kanker. Komunitas 5 berkaitan dengan respon terhadap stimulus hormon, yang relevan dengan kanker payudara yang dipengaruhi oleh faktor hormonal. Selain itu, pada kategori KEGG Pathway terlihat bahwa beberapa komunitas terlibat dalam jalur yang berkaitan langsung dengan kanker, seperti *pathways in cancer*, *cell cycle*, dan *JAK-STAT signaling pathway*. Hal ini menunjukkan bahwa komunitas yang dihasilkan oleh algoritma GLOD tidak bersifat acak, melainkan memiliki keterkaitan dengan mekanisme biologis tertentu.

Nilai signifikansi ditunjukkan dalam bentuk Log10(P), di mana nilai yang semakin kecil (lebih negatif) menunjukkan hasil yang semakin signifikan secara statistik. Selain itu,

persentase (%) menunjukkan proporsi protein dalam komunitas yang terlibat dalam fungsi biologis tersebut. Hasil ini juga menunjukkan adanya protein yang muncul pada lebih dari satu komunitas, yang menandakan adanya *overlapping community*. Hal ini sesuai dengan karakteristik jaringan biologis, di mana satu protein dapat berperan dalam lebih dari satu proses biologis. Secara keseluruhan, hasil enrichment analysis ini menunjukkan bahwa komunitas yang dihasilkan oleh algoritma GLOD tidak hanya memiliki kualitas struktur yang baik (berdasarkan evaluasi NNC), tetapi juga memiliki relevansi biologis yang jelas terhadap mekanisme kanker payudara. Dengan demikian, hasil ini menunjukkan bahwa metode yang digunakan berhasil menjawab rumusan masalah penelitian, yaitu tidak hanya mendeteksi komunitas overlap, tetapi juga memvalidasi kualitas dan makna biologis dari komunitas tersebut.

Hasil penelitian ini menunjukkan bahwa algoritma GLOD mampu mendeteksi komunitas overlap pada jaringan interaksi protein kanker payudara serta menghasilkan struktur komunitas yang memiliki kualitas baik secara topologis dan relevan secara biologis. Dengan demikian, seluruh rumusan masalah dalam penelitian ini telah berhasil dijawab melalui pendekatan experimental setup dan evaluasi yang dilakukan.

4.2.6. Deployment

Tahap deployment pada penelitian ini merepresentasikan implementasi akhir dari sistem yang telah dirancang pada Bab III, di mana seluruh komponen sistem diintegrasikan dan dijalankan secara menyeluruh. Proses ini mencakup alur lengkap mulai dari input data gen, preprocessing, konstruksi jaringan interaksi protein, deteksi komunitas menggunakan algoritma GLOD, hingga penyajian hasil dalam bentuk visualisasi dan metrik evaluasi.

Berdasarkan hasil yang telah dipaparkan pada Subbab 4.2, setiap tahapan dalam sistem menunjukkan keluaran yang sesuai dengan yang diharapkan. Proses input data, pembentukan jaringan, pemilihan giant component, serta deteksi komunitas overlap dapat berjalan tanpa kendala, dan menghasilkan struktur komunitas yang dapat dianalisis lebih lanjut. Selain itu, sistem juga mampu menampilkan hasil deteksi komunitas dalam bentuk visualisasi jaringan, statistik komunitas, serta nilai Normalized Node Cut sebagai indikator kualitas struktur komunitas. Hal ini menunjukkan bahwa integrasi antara antarmuka pengguna dan modul algoritma pada backend telah berjalan dengan baik. Dengan demikian, dapat disimpulkan bahwa sistem yang dikembangkan mampu menjalankan seluruh tahapan proses secara konsisten dan menghasilkan keluaran yang valid. Implementasi algoritma GLOD dalam sistem ini dinyatakan berhasil, baik dari sisi fungsional sistem maupun dari sisi hasil analisis komunitas yang dihasilkan.

4.3 Pembahasan

Penelitian ini bertujuan untuk menjawab dua rumusan masalah utama, yaitu mendeteksi komunitas overlap pada jaringan interaksi protein kanker payudara menggunakan algoritma GLOD, dan mengevaluasi kualitas komunitas yang dihasilkan secara struktural dan biologis. Untuk mencapai tujuan tersebut, seluruh tahapan penelitian dilakukan secara runtut mulai dari pengumpulan data, persiapan data, konstruksi jaringan, pemodelan, hingga evaluasi, sebagaimana disajikan pada Subbab 4.2.

Pada tahap awal, proses pengumpulan data dilakukan melalui API UniProt menggunakan kata kunci *breast cancer*, yang menghasilkan 2.690 entri protein. Hasil ini menunjukkan bahwa data yang diperoleh masih bersifat mentah dan mengandung redundansi. Hal ini dibuktikan pada tahap data selection dan deduplikasi, di mana hanya atribut *Gene Symbol* yang dipilih sebagai identitas utama, dan jumlah data berkurang menjadi 2.010 protein unik. Bukti proses ini ditunjukkan pada Lampiran A serta hasil preprocessing pada Subbab 4.2.3. Reduksi sebanyak 680 entri mengindikasikan adanya duplikasi yang signifikan pada data awal, sehingga proses pembersihan menjadi krusial untuk memastikan validitas representasi jaringan.

Selanjutnya, data yang telah bersih dipetakan ke dalam STRING DB menggunakan parameter confidence score sebesar 0,900 untuk menjamin kualitas interaksi biologis yang tinggi. Hasil konstruksi jaringan menghasilkan graf dengan 1.823 simpul dan 2.646 sisi, yang menunjukkan bahwa tidak semua protein memiliki interaksi dengan tingkat kepercayaan tinggi. Visualisasi pada Gambar 4.4 memperlihatkan adanya banyak simpul terisolasi, sehingga dilakukan ekstraksi *giant component* untuk memperoleh struktur jaringan yang lebih representatif. Hasilnya, jaringan akhir terdiri dari 864 simpul dan 2.567 sisi, sebagaimana ditunjukkan pada Gambar 4.5 . Hal ini menunjukkan bahwa interaksi protein cenderung terkonsentrasi pada komponen utama jaringan.

Pada tahap pemodelan, algoritma GLOD diterapkan menggunakan konfigurasi parameter terbaik yang diperoleh dari hasil experimental setup pada Tabel 4.2, yaitu $\alpha = 0{,}75$ dan Jaccard threshold = 0,20 dengan nilai NNC sebesar 0,3691. Nilai ini merupakan yang paling kecil dibandingkan kombinasi parameter lainnya, sehingga dipilih sebagai konfigurasi optimal. Penerapan algoritma pada jaringan menghasilkan 16 komunitas yang saling tumpang tindih. Bukti keberhasilan deteksi overlap ditunjukkan Lampiran C, yang terdiri dari Lampiran C.1 hingga Lampiran C.3, di mana terdapat protein yang muncul pada lebih dari satu komunitas. Hal ini secara langsung menjawab rumusan masalah pertama, yaitu kemampuan algoritma GLOD dalam mendeteksi komunitas overlapping pada jaringan PPI kanker payudara.

Dari sisi struktural, kualitas komunitas dianalisis menggunakan metrik Normalized Node Cut (NNC) yang disajikan pada Tabel 4.3. Nilai NNC yang dihasilkan bervariasi antara 0,0858 hingga 0,6503, dengan rata-rata sebesar 0,3691. Komunitas dengan nilai NNC rendah menunjukkan keterhubungan internal yang lebih kuat dibandingkan eksternal, sehingga memiliki batas yang jelas. Sebaliknya, nilai NNC yang tinggi menunjukkan adanya konektivitas lintas komunitas yang lebih besar. Variasi ini mengindikasikan bahwa struktur jaringan tidak homogen, melainkan terdiri dari komunitas dengan tingkat kohesivitas yang berbeda, yang merupakan karakteristik umum dari jaringan biologis kompleks.

Dari sisi biologis, hasil enrichment analysis yang disajikan pada Tabel 4.4 dan Tabel 4.5 menunjukkan bahwa komunitas yang terbentuk memiliki keterkaitan dengan fungsi biologis yang relevan dengan kanker payudara. Sebagai contoh, Komunitas 1 memiliki keterkaitan dengan regulasi siklus sel mitotik dan jalur kanker dengan tingkat signifikansi yang tinggi. Selain itu, ditemukan juga proses seperti transisi epitel-mesenkimal, regulasi adhesi sel, dan aktivitas protein kinase pada komunitas lainnya. Hasil ini menunjukkan

bahwa komunitas yang dihasilkan tidak bersifat acak, melainkan merepresentasikan modul biologis yang memiliki fungsi spesifik.

Keberadaan protein overlap antar komunitas menunjukkan bahwa satu protein dapat terlibat dalam lebih dari satu jalur biologis. Hal ini mencerminkan kompleksitas sistem kanker, di mana protein dapat berperan sebagai penghubung antar proses molekuler yang berbeda. Dengan demikian, hasil ini tidak hanya menunjukkan keberhasilan algoritma secara komputasional, tetapi juga relevansi biologis dari komunitas yang dihasilkan.

Secara keseluruhan, hasil experimental setup (Tabel 4.2), evaluasi struktural (Tabel 4.3), serta evaluasi biologis (Tabel 4.4 dan Tabel 4.5) menunjukkan bahwa algoritma GLOD mampu mendeteksi komunitas overlap dengan kualitas struktur yang baik dan makna biologis yang jelas. Selain itu, seluruh tahapan proses yang disajikan pada Subbab 4.2 juga menunjukkan bahwa sistem yang dikembangkan mampu berjalan secara terintegrasi dan menghasilkan keluaran yang sesuai dengan tujuan penelitian. Dengan demikian, penelitian ini berhasil menjawab kedua rumusan masalah yang telah ditetapkan.

BAB V
PENUTUP

5.1. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa:

1. Algoritma GLOD mampu mendeteksi komunitas overlap melalui mekanisme seeding, expansion, dan merging berbasis kesamaan anggota (Jaccard). Proses ini memungkinkan satu protein masuk ke lebih dari satu komunitas, sehingga struktur overlap dapat terbentuk secara alami, yang dapat dilihat pada hasil deteksi komunitas di Lampiran C (C.1–C.3) yang menunjukkan adanya protein yang muncul pada lebih dari satu komunitas.
2. Pemilihan parameter berpengaruh terhadap kualitas komunitas yang dihasilkan, dan kombinasi optimal mampu menghasilkan struktur komunitas yang lebih baik. Parameter α dan threshold Jaccard mengontrol proses ekspansi dan penggabungan komunitas, yang terbukti dari hasil experimental setup pada Tabel 4.2, di mana kombinasi $\alpha = 0{,}75$ dan $J = 0{,}20$ menghasilkan nilai NNC terbaik.
3. Komunitas yang dihasilkan memiliki kualitas struktur yang cukup baik secara topologis. Hal ini ditunjukkan oleh nilai Normalized Node Cut yang menunjukkan bahwa koneksi internal komunitas lebih dominan dibandingkan koneksi eksternal, sebagaimana terlihat pada distribusi nilai NNC pada Tabel 4.3 dengan rata-rata sebesar 0,3691.
4. Komunitas yang terbentuk memiliki makna biologis yang relevan dengan kanker payudara. Komunitas merepresentasikan modul biologis seperti siklus sel, jalur kanker, dan proses metastasis, serta menunjukkan adanya protein yang berperan dalam lebih dari satu proses biologis, yang dapat dilihat pada hasil enrichment analysis pada Tabel 4.4 dan Tabel 4.5.

5.2. Saran

Beberapa saran yang dapat dipertimbangkan untuk pengembangan selanjutnya adalah sebagai berikut:

1. Pengembangan metode dapat dilakukan dengan membandingkan performa GLOD terhadap algoritma deteksi komunitas overlap lainnya, sehingga diperoleh gambaran komparatif mengenai keunggulan dan keterbatasan pendekatan yang digunakan.
2. Analisis jaringan dapat dikembangkan dengan menggunakan variasi *confidence score* pada STRING DB (misalnya medium atau high confidence), sehingga dapat diamati perbedaan struktur jaringan, jumlah simpul dan sisi, serta dampaknya terhadap hasil deteksi komunitas.
3. Meskipun hasil enrichment analysis menunjukkan keterkaitan fungsi biologis yang signifikan pada setiap komunitas, temuan ini masih bersifat komputasional (*in silico*). Oleh karena itu, diperlukan penelitian lanjutan secara eksperimental, baik melalui uji *in vitro* maupun *in vivo*, untuk memvalidasi peran biologis protein-protein dalam komunitas yang teridentifikasi, khususnya yang berkaitan dengan jalur kanker dan regulasi siklus sel pada kanker payudara.

DAFTAR PUSTAKA

Baltsou, G., Christopoulos, K. and Tsichlas, K. (2022) 'Local Community Detection: A Survey', *IEEE Access*, 10, pp. 110701–110726. Available at: https://doi.org/10.1109/ACCESS.2022.3213980.

Betz, M. *et al.* (2025) 'Translational Oncology Decoding mutational signatures in breast cancer : Insights from a multi-cohort study', 53(November 2024). Available at: https://doi.org/10.1016/j.tranon.2025.102315.

Castellanos-Girouard, X., Serohijos, A.W. and Michnick, S.W. (2024) 'Protein-protein interaction is a major source of epistasis in genetic interaction networks'. Available at: https://doi.org/10.1101/2024.11.11.623081.

Cheng, F. *et al.* (2021) 'A Local-Neighborhood Information Based Overlapping Community Detection Algorithm for Large-Scale Complex Networks', *IEEE/ACM Transactions on Networking*, 29(2), pp. 543–556. Available at: https://doi.org/10.1109/TNET.2020.3038756.

Dilmaghani, S. *et al.* (2022) 'From communities to protein complexes: A local community detection algorithm on PPI networks', *PLoS ONE*, 17(1 January). Available at: https://doi.org/10.1371/journal.pone.0260484.

Ferlay, J. *et al.* (2021) 'Cancer statistics for the year 2020: An overview', *International Journal of Cancer*, 149(4), pp. 778–789. Available at: https://doi.org/10.1002/ijc.33588.

Guo, K. *et al.* (2022) 'Local community detection algorithm based on local modularity density', *Applied Intelligence*, 52(2), pp. 1238–1253. Available at: https://doi.org/10.1007/s10489-020-02052-0.

Havemann, F. *et al.* (2012) 'Evaluating Overlapping Communities with the Conductance of their Boundary Nodes', (Fortunato 2010). Available at: http://arxiv.org/abs/1206.3992.

Ma, Y., Liu, Y. and Tao, J. (2020) 'Detecting the overlapping community by using extract method', in *Journal of Physics: Conference Series*. IOP Publishing Ltd. Available at: https://doi.org/10.1088/1742-6596/1673/1/012020.

Ni, L. *et al.* (2019) 'Local overlapping community detection', *ACM Transactions on Knowledge Discovery from Data*, 14(1). Available at: https://doi.org/10.1145/3361739.

Platos, J., Id, P.P. and Dra, P. (2024) 'Overlapping community detection in weighted networks via hierarchical clustering', pp. 1–22. Available at: https://doi.org/10.1371/journal.pone.0312596.

Prayoga, A. (2025) 'DETEKSI KOMUNITAS JARINGAN INTERAKSI PROTEIN PENYAKIT KANKER PARU-PARU MENGGUNAKAN ALGORITMA ANT COLONY OPTIMIZATION PENYAKIT KANKER PARU-PARU MENGGUNAKAN ALGORITMA'.

Song, Y. *et al.* (2023) 'GLOD: The Local Greedy Expansion Method for Overlapping Community Detection in Dynamic Provenance Networks', *Mathematics*, 11(15). Available at: https://doi.org/10.3390/math11153284.

Sri Suharini, Y. *et al.* (2023) *Pendekatan Teori Graf untuk Analisis Jaringan*

Interaksi Protein-Protein (Graph Theory Approach to Network Analysis of Protein-Protein Interactions).

Sung, H. *et al.* (2021) 'Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries', *CA: A Cancer Journal for Clinicians*, 71(3), pp. 209–249. Available at: https://doi.org/10.3322/caac.21660.

Szklarczyk, D. *et al.* (2023) 'The STRING database in 2023: protein-protein association networks and functional enrichment analyses for any sequenced genome of interest', *Nucleic Acids Research*, 51(1 D), pp. D638–D646. Available at: https://doi.org/10.1093/nar/gkac1000.

Vieira, V. da F., Xavier, C.R. and Evsukoff, A.G. (2020) 'A comparative study of overlapping community detection methods from the perspective of the structural properties', *Applied Network Science*, 5(1). Available at: https://doi.org/10.1007/s41109-020-00289-9.

Wang, Y. *et al.* (2021) 'Overlapping Structures Detection in Protein-Protein Interaction Networks Using Community Detection Algorithm Based on Neighbor Clustering Coefficient', *Frontiers in Genetics*, 12. Available at: https://doi.org/10.3389/fgene.2021.689515.

Zhang, X. and Liu, Q. (2025a) 'A graph neural network approach for hierarchical mapping of breast cancer protein communities', *BMC Bioinformatics*, 26(1). Available at: https://doi.org/10.1186/s12859-024-06015-x.

Zhang, X. and Liu, Q. (2025b) 'A graph neural network approach for hierarchical mapping of breast cancer protein communities', *BMC Bioinformatics*, 26(1). Available at: https://doi.org/10.1186/s12859-024-06015-x.

Zhao, Y. *et al.* (2023) 'Overlapping Community Detection Algorithm Based on High-Quality Subgraph Extension in Local Core Regions of Network', *Wireless Communications and Mobile Computing*, 2023. Available at: https://doi.org/10.1155/2023/4988601.

Zhou, Y. *et al.* (2019) 'Metascape provides a biologist-oriented resource for the analysis of systems-level datasets', *Nature Communications*, 10(1). Available at: https://doi.org/10.1038/s41467-019-09234-6.

LAMPIRAN

Lampiran A Data Selection

Lampiran A.1 Hasil Data Selection

Dataset	Nama Kolom	Isi Kolom
Uniprot - Breast Cancer	Gene Symbol	BRMS1, BRCA2, BRCA1, BIN2, GREB1, SNCG, CALML4, BRMS1L, BCAS1, MUC1, SGO1, DLEC1, GREB1L, CEP85L, TGFBR2, BLTP2, TFF1, SYTL2, BCAR1, BCAR3, RHOBTB2, SFXN4, KRAS, PTEN, EBAG9, AKT1, SEPTIN1, MAGED2, SMG8, CDH1, CASP8, AKIP1, TOP1, RBBP8, FAM83D, EI24, ERGIC3, MUCL1, ANKRD17, STRADA, MYLK, FBLN1, CDKN2C, MTA1, CYP1B1, KIF15, UQCC2, ZNHIT6, SLX4, RB1CC1, CUEDC2, LHCGR, FAM3C, ZNF703, SH3BGRL, AXL, CDKN1B, ITIH4, PALB2, RNF146, AGR3, PSMD6, NRG1, TRERF1, ABCG2, LETMD1, CTAG1A, PRPF31, MFGE8, TOMM6, CENPW, PRSS50, TACC1, BCAS3, PTK6, NCOA6, PIK3CA, ATAD2, CHEK2, LDOC1, SMIM22, LMO4, PPHLN1, BRDT, BCAS4, CCAR2, LZTS1, MAGEA8, PASD1, LRATD2, MLH3, LRRC26, EPCAM, AR, ADAM29, STARD8, MAGEA11, SVEP1, MLH1, NDFIP1, PLEKHA8, CEMIP, MSH6, PMS2, ARID4B, MAGEC2, MAGEA1, MSH2, MAGEA12, MAGEA4, STK11, FLT1, AFAP1, TNS4, BRINP1, EMSY, MAGEA3, OLA1, MAD1L1, TP53, MSMB, TP63, TPTE, MAGEA6, RPL13, HIC1, FLT4, DCC, KDM5B, MRE11, MAGEA2, VTCN1, PPP1R14C, ERBB4, PRICKLE4, DPH1, CASC3, PHF11, CTCF, MANF, BCAS2, KCNK9, PBOV1, MELK, CABLES1, BRAF, BRAP, ZFHX3, CTNNB1, EHBP1, AKAP13, HEATR6, MAPK12, VWA2, CTU1, FGFR4, SLC67A1, ANLN, UNC5A, NCOA3, UNC5B, KLK8, MAP3K9, UHRF1, DPP4, EIF3D, UNC5C, VWA5A, STARD13, RLIM, TP53BP2, SEPTIN9, USP32, NEDD9, ANKRD30A, BLID, BPHL, ERBB2, CHMP2A, EP300, PTK2, UBE3A, ASAP3, LEF1, RBM5, SYNE1, SMAD4, EGFR, AIM2, ING1, SMAD3, HMGB1, NCOA1, KDR, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, TMEM33, PLK1, KLK10, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, STARD10, PPM1D, VOPP1, SUSD2, SRARP, CLCA2, ENOX1, ZNF668, ABRAXAS1, MAGEA10, ELF3, PBRM1, TPM3, NLRP8, SH2D3A, XRCC3, POTEF, Q5YLB2, XRRA1, BOD1L1, B3GNT8, ANKRD30B, TPM2, ACRBP, POU5F1B, TP53I3, RAD51D, BAG1, FAM83A, ST18, BRIP1, CXCL17, IRX5, HOXA3, SNAI2, ZNF202, RAD51C, TPM1, AKAP3, PYHIN1, DGAT2L6, MYH1, CDK1, ARMC12, PKDREJ, TOP2B, UBE2V1, TRPM8, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, GYG2, GLS2, TRIM24, BRCA1, AVPR2, SLC9A2, NUPR1, ADAMTSL3, SEMA6B, SLC39A10, FN1, PMPCA, ICE2, ARHGAP35, AURKA, ATM, MYH9, FAIM2, ZNF432, COL19A1, ZNF318, SEMA5B, BIRC5, RAB25, MDM4, ETV5, EPG5, PHB2, WDCP, SLC2A5, RPAP1, GKN1, RABL3, PEPD, LGALS1, TUBG1, RIC3, WWP1, PDIA3, TNN, CTIF, CCNG1, ARAP3, UTP20, MAP3K6, KRT7, RFX2, XRCC2, MAP1B, LMAN1, SIGMAR1, KIT, SCARB2, PPP1R8, FGFR2, TRA, XAGE1A, TRB, SIX4, PLAUR, KRT16, CHRNA7, SGK3, TXN, BEX2, HNRNPK, HDAC1, CTTN, RHOA, SCGB2A1, RAPH1, APC2, YBX1, SLC3A2, KPNA2, POLQ, LCK, RASGRF2, DNASE1L3, RAP1GAP, DERPC, PRMT2, PRRC2A, CRIPTO3, ENTREP3, ROBO1, NENF, BPIFA4P, MCF2L2, SUSD3, CUX1, ECT2, DEPP1, NBN, RERG, RRP9, GPER1, MTMR8, CIC, CNGA2, NET1, LIMD2, PHKB, SECTM1, TMEM161A, CHD5, POSTN, ENOX2, MACROD1, LACTB, RASAL2, LYPD3, MTUS1, RAD51, HOXB13, KCNH4, RNF115, PTPN14, XIRP1, ZFP64, BCL2L14, SP110, RAD54L, FASTKD3, TMEM39A, CST4, LGALS2, SLC25A51,

Lampiran A.2 Hasil Data Selection (Lanjutan)

Dataset	Nama Kolom	Isi Kolom
Uniprot - Breast Cancer	Gene Symbol	NIP7, TSKS, ARHGAP11A, PRLR, GPNMB, TMEM170B, DHRS2, BMP10, RAB6C, SRA1, SETD4, MIEN1, SUDS3, MINAR1, ENPEP, CSMD3, FRK, LIMD1, NAA25, YY2, ARFRP1, HOXA4, RNF149, NOA1, EDRF1, TENM1, GNB1L, NPAP1, GOLIM4, KLK15, FAM217B, MADD, NEK3, CDKN3, TSTD1, XDH, ENAH, GNAS, SRCIN1, NECTIN4, ZNF804A, MMP11, ADAM19, ACTL9, THOC5, FEM1C, SIK3, APC, PEG10, MYOD1, SMAGP, HDLBP, TRARG1, ADGRF4, RHNO1, BCL9L, KRT73, PPM1F, GPC6, FOLR2, TXNDC15, RCE1, SPAG6, CCNQ, COLGALT2, PRR5, AKAP6, AFF1, CLASRP, CDH20, MAGEE1, CCAR1, DSE, ELAPOR1, CYP4Z1, TECTA, SHANK1, HSPA14, DNAJC24, CDH10, DCLK3, GOLGB1, DHX32, JADE3, NPBWR1, SHROOM2, N4BP2, SORCS1, CUTC, UTS2R, PACC1, TSC22D4, OBSCN, CREB3L4, SYT9, SLC5A8, MIA, SLC22A9, TTLL3, MTHFD1L, ADNP, CDCP1, ANAPC5, ITIH5, CUBN, HK3, BATF2, ADAMTS15, CLSTN2, PIK3R5, GABRP, FABP4, RASEF, IRF8, SMTN, SBNO1, TREML1, MAP7D2, RACK1, ZDHHC4, ZNF281, THBS3, MFAP5, SLC39A12, TCF7L1, EFS, OVGP1, CPNE3, ZBTB8B, NDUFA3, INSL3, GPC2, HDC, CBFA2T3, MROH7, DAZAP1, ST6GALNAC1, SDHD, PLXDC1, SLC39A6, RYBP, AGAP2, FERD3L, PRUNE2, WNT7B, CASP10, GRIN2D, PRDM5, VSTM4, EREG, DNAH9, ZFYVE26, ALS2CL, NLRP14, ADHFE1, CNTN6, GRIK3, SKIC2, RNF11, EPHA3, WFDC2, HEPACAM2, ZNF646, MMP10, CXCR5, MPLKIP, ADAM12, APOC4, DDX10, INA, FOXP4, ZNF276, IL1RAPL2, SERPINE2, SAFB2, WNT8B, PARP8, MMP17, ZSCAN16, CFAP77, OLAH, IRX4, MAK, PREX2, CLIC3, TESK2, XBP1, CAPN6, TBX22, ELOA2, SIPA1L1, TLL1, RANBP1, DOK1, RGCC, MAP3K19, TPD52, SRGAP3, NDUFA2, ARHGAP29, DNAJC21, SPOCD1, NME8, TRMT10A, PPM1E, LRRC7, POP1, SYT3, UQCC1, TRAPPC8, IFNA2, PANX2, STRBP, KCNJ15, PRRT1, DDX18, NLE1, DENR, PCDH8, PFKFB4, FRMPD1, CNNM4, KBTBD8, TRAPPC12, ZNF541, CSNK1A1L, PAQR4, CCN4, TMED1, SH3PXD2A, EFHD1, ARFGEF3, CDKN1C, DSCAML1, CCDC170, ECM1, H2AC21, ANKK1, KLK14, SASH1, SRMS, LAMP3, WNT16, ZNF652, GAB1, KDF1, BARD1, FKTN, PHB1, TFAP2D, ABCB5, MTDH, OTOF, CD300E, MIA2, NF1, SLC6A3, THOC1, SLC66A2, TBRG1, PRDM14, TTLL10, CCNB3, CD248, SALL3, AMER2, CELA1, FAM110B, HMMR, PTHLH, CPO, PODXL, AMTN, PDILT, WWOX, SLC1A7, WWC1, ADARB2, LIN28B, TPM4, PPP1R3A, ARHGEF4, FOXA1, HID1, SCARF2, ZNF436, MED14, GALNT5, TRMT2A, JPT2, TACC2, BGN, ITGA9, ISG20, RPRD1A, BOC, ZNF644, OSTC, CFL2, CD2, DDX59, CYB5R4, CLPTM1L, OSBPL11, NRK, PRAF2, PER1, ARID1A, CORO2B, LONRF2, AMDHD2, HTR3C, NPAS2, RNF182, HOXC9, RHOB, ACAP1, EVC2, NCAPG, COL11A1, TIMELESS, NUP214, TMPRSS6, BPIFB2, RAC2, FOLH1, AMPD1, SREBF2, PLPPR2, CLIP1, MLLT11, ATP2A3, TCHP, ADAMTS18, ZNF217, RASSF1, MAP2, ZNF350, PHF7, AMPH, NDUFAF4, PLD2, WBP4, APCS, NDUFA8, ABCB8, KCNIP3, PEX5L, GMCL2, ASTN2, EOMES, NQO2, EIF5, NID2, TRIM29, LRRFIP1, NLRP9, P3H2, EIF1AD, ADAMTS19, CAVIN3, PADI3, FARP1, PES1, SSNA1, PRG2, EFNA1, PIGS, TREM1, ZNF480, ZNF540, GUCY2F, VPS72, ACO2, KDM3A, FAM210A, PAK6, SAFB, GFRAL, DAW1, COG3, CDC27, GSE1, PDZRN4, CAMK1G, MOK, AGR2, SPAG17, PDCD4, SEC23B, EPHB6, TMEM140, KCNB2, SNRK, ZNF232, DLG5, PELP1, GGA1, KLK5, SMG1, DCHS1, RNPEPL1, FBXW7, GLI1, ZNF155, STK32C, SUCO, ACTN4, ECD, CORO1B, NPR1, NCEH1, PTPRU, NODAL, HPS3, CDK15, CD109, NDST3, PBX4, ODAM, NDUFA1, RUSF1, SULF1, NCR1, LZTFL1, HIPK1, KALRN, LMTK3, TTC3, PNLIPRP1, KTN1, B3GALNT2, ABCC12, KLK7, EYA4.

Lampiran A.3 Hasil Data Selection (Lanjutan)

Dataset	Nama Kolom	Isi Kolom
Uniprot - Breast Cancer	Gene Symbol	IFNB1, LRBA, PLOD1, SAMD9L, NSMCE2, FCRL5, SULF2, PRICKLE3, KRT76, TMTC4, ADAT3, GABRA6, DUSP21, NCDN, RPS6KC1, TSPAN4, LYPLA2, GEN1, ZMIZ1, JAKMIP2, MYLK3, HTR5A, ARHGEF10L, KCNA10, GUCY1A2, REST, MYO1B, DENND4A, EXOC4, CHL1, EPHA7, KL, MMP2, EPHA10, ATXN3L, CHORDC1, PSMB5, EHMT1, LRRC4, TMEM97, TFEC, SLC44A4, FGF8, PKNOX1, ATR, RABL6, RELT, TRIM37, ADGRL2, PTPRD, CDCA7, SEMA3A, SMOX, NELL1, ZNF572, RIF1, NHS, KCNT2, ATF7IP, TFAP2C, GIMAP1, NPBWR2, PCDH11X, KLK6, TRAF3IP3, SPEN, SLC17A6, PTPRN2, PSMG1, HOOK3, NLRC5, UNC45B, CFHR5, ANAPC4, LDB1, DSTN, PDGFD, ADCY8, ACACA, TIGAR, MIIP, USP17L2, KDM8, CIB1, OLFM2, EML2, PCSK2, CRIPTO, ATP8B1, SLC30A2, CDO1, BAIAP3, LIFR, UNC13B, DGKG, STX12, SH2D3C, NT5C1A, DDX47, IGFBP3, LGR6, CCDC62, KPNA5, PRDX5, HELQ, ARFGAP3, RPS9, KCNJ1, RUFY1, CPSF3, MMRN2, C1QB, ACY1, KLHL22, SMC6, ATP6V0B, TNS1, ZDHHC7, GRM6, PLS3, RFC4, ROR1, SURF1, BCORL1, TOR1AIP1, ARFGEF2, USP54, NAMPT, ETAA1, EVL, CHFR, CHST8, FLNB, DEFB106A, TEKT4, CPE, NRBP1, RPS6KA4, MAP3K12, EPB41L3, ALX4, ENTREP1, NID1, GSDME, ITGB3BP, ANK2, KLF5, TUBB2A, FOXC1, SIK1, ZNF438, DMKN, KIF16B, CASD1, PRKCE, BEND7, CLSPN, FZD7, RARG, GPC1, RUNX1T1, RALGDS, CNTN4, ARHGEF5, DMD, POU2F1, PRP4K, SLC6A5, BCAR1, GABRA4, DBN1, EIF3M, OXSM, ICAM5, G3BP2, PHIP, LIPE, HIVEP3, RNF126, TSPEAR, ABCC11, KCTD1, JAKMIP1, CTSH, PZP, SDF4, COL7A1, PDS5A, BMP6, WRN, UBE2T, CSNK1A1, SCAF1, GRK6, SMAD2, SCNN1B, KIF14, TCP1, FNDC3B, CAVIN1, SUV39H2, DACT1, PLEKHM1, CATSPERT, ACKR3, KAT6B, PIK3R4, TRPS1, FRMPD2, PRODH, NFYC, HSD17B8, RASSF2, CNKSR2, APOL1, GTF2A1, MTMR3, TESK1, PIBF1, LIG1, FPR1, RPRD1B, PLXNB1, MICALL1, SMARCAL1, PRUNE1, PEG3, DDR1, GPATCH2, KCNQ5, AREG, ERCC6, SFRP1, TLR9, TNFRSF9, RCN1, HDAC4, KCNJ3, ZFP91, PTK2B, SAMD9, ZMYM4, PCM1, IFI16, SETBP1, LMO7, PPIL2, NUAK2, CAPN10, AKAP8, AP2A1, SLC7A7, HDAC5, CDK10, MSI2, ALDH1L1, TNKS2, RHD, UHRF2, SIGLEC7, EDNRA, NUP133, CDS1, MAGOHB, CCDC93, PUS10, G6PC1, SLC8A3, EIF4A2, NRCAM, PLCB1, HOOK1, FSCB, NOP9, STK36, WARS1, PDLIM7, IQUB, DEFA4, KMT2C, PRKD1, FGF14, CYREN, KRTAP10-8, TUBB6, PALLD, AKAP9, CTSV, HRH1, C8orf17, PPP1R13B, ZZZ3, SRSF6, DAB2, SLC33A1, ADORA3, ATAD1, ST8SIA4, CNOT11, MAST2, CST6, FOXRED1, RADIL, CD24, KSR1, CNTN1, VRK2, THOC6, H1-5, WNT4, BRF1, DNPH1, BACE2, LAS1L, CENPH, MYOT, GHRHR, TRIB3, COL4A6, GATA3, THBS2, HEPH, TNK2, CSF1, ZMYND11, KMT5B, ST8SIA1, UQCRC2, KRT20, NLGN4X, CSE1L, GSDMB, BMP1, DGKE, CBFB, SORBS1, ABCB10, TRPC4, AQP8, CTSD, LAMA2, POLR2F, STAT4, TWIST1, HAT1, PSMA1, FCN2, TBX1, PAN2, CFP, HADHB, DTX3L, MICAL1, EGFL6, OSGIN1, CLDN7, DPAGT1, ESR1, IFI6, RAB38, MTA3, SLC39A9, ARGLU1, ADORA1, TTC4, SPTBN2, MIER1, SLCO1B3, DNAH8, AKAP12, MAMDC4, ACCS, NISCH, KLF17, EXOC3L1, RUNDC3B, TNFRSF10A, FCN1, EPHA8, PXDN, ZMYND8, CD93, EPHA1, LAMC1, CDC42BPB, HIPK2, PANK4, PIGO, ME1, ST3GAL1, AP1M1, PLD6, CDC14A, WNK1, IL24, PKN1, DPEP1, THOC3, FAP, H2AC12, INPP5K, GGA3, PLK4, S100A7, S100A8, MAOA, VPS13B, KDM6A, SNRPB2, HR, CINP, TRPM7, RAB11FIP3, CHRND, STX5, LOXL2, DNAH3, ETV4, ADRA1A, LGR4, APPL1, AMFR, WNT7A, CACNB1, TNNI3K, BCL11B, PGAP3, H2AC25, TCF7L2, ARHGEF1, PRL, SDCBP2, PER2, ELOC, MKRN3, WNT3A, BAX, INTS13, IRS4, TPD52L1, FANCG.

Lampiran A.4 Hasil Data Selection (Lanjutan)

Dataset	Nama Kolom	Isi Kolom
Uniprot - Breast Cancer	Gene Symbol	CTNND2, WBP2, RARB, H2AC1, RPL6, VPS18, GRM1, CLSTN3, STAM, PLA2G3, DPYSL2, OSBP, SNAI1, MYB, RBP3, AGTPBP1, KLK4, LAMB3, MFN1, ATP6V1E1, ETFDH, PRKAA2, RAB5C, ANGPTL4, TPX2, S100A2, HOXA1, HSD17B7, CLOCK, ZBTB7A, CDK6, CYP1A1, CSF1R, ADAR, FBXL2, ACSL5, KCNA5, SLAMF1, FCRL3, DDO, MACF1, LATS1, ASL, ABL2, OCA2, CDH13, DCAF7, ZEB2, CCN5, UBR4, NIPBL, TFAP2A, BAP1, TBXAS1, SPTAN1, HDAC7, DNALI1, SLC12A5, OMA1, JAK3, CAMSAP2, GFI1B, PKHD1, JAK1, DEFB1, BRCC3, ALK, SLC17A5, LZTS2, METTL5, GABPB1, WAC, SCN3B, S100A11, FOXR2, NEK2, NSD3, FADS2, H2AC4, CITED4, GNA12, TRIP4, CAD, PDCD6, S100A9, ACADM, GRIK2, TIAM1, SLC35A2, ARFGEF1, SRSF1, RAD50, ZNHIT1, B4GALNT2, NOS2, IFI27, SACS, WNT10B, H2AC14, VPS13A, TBX2, MAD2L1, ERVK-6, POU4F2, ST7, H2AC20, EPHB4, GLI3, GRID2, RECQL, STRAP, STAT5A, SCP2, CEBPA, TAX1BP1, ARID1B, PTPRS, TCOF1, PRF1, CACNB2, EPSTI1, MICB, DVL3, CCNDBP1, POLH, NF2, TRIM25, RPS6KA3, EIF2AK2, PRPS1, FAAH, SUZ12, CD46, CLU, RET, IDH1, ACIN1, LIG3, GNA13, H2BC14, DHX40, H2AC7, SORL1, ERVK-18, SF3A1, H2AC6, H2AC18, UBE2D1, SLC29A1, LTF, RHEB, DFFA, BCCIP, NOS3, RPL19, NFATC1, EED, PHGDH, H2AC11, SIVA1, ASAP1, NR1D1, PDGFRB, LEP, MPO, BCL11A, BCL10, PTPRC, ATP11C, CRX, MMP14, KNG1, CDKL5, ESR2, PIN1, RPS6KA5, RBM10, UIMC1, THOC7, SCNN1G, GRB7, ACTR3, SNX6, KCND3, RBP1, FUS, NOTCH2, KRT17, WDR26, PLK3, SPTLC1, NEDD4, EXTL3, LMNB2, MCM3AP, NONO, PIP, IRAK1, IL6ST, BANP, CLCN1, DRC4, ATF2, SYNE2, ABCA3, MCOLN1, CAV1, XRCC1, DLG3, AKT2, H1-4, SERPINA5, MST1R, SLCO1A2, GPSM2, MKI67, GLI2, PNPLA2, PXN, BAK1, SMURF2, JPT1, EPN1, ARID4A, PDGFB, PFN1, AURKB, TFG, VWF, TICAM1, PSMA2, DDX17, TSG101, HDAC6, PGAM1, MFF, UGT1A9, HSD11B1, NUP98, ERCC3, DYSF, ANK1, TEK, RNF31, NR3C2, PRSS1, TCF7, RASSF5, ERCC4, MMP13, FZD4, ACSL4, SOS1, LRP1, STAT3, TTN, TRAF4, MDK, PSMC5, RRP1B, DHX15, KRT18, ABCB1, PQBP1, JAK2, PGR, PRKCA, CSNK1D, KRT19, LIMK1, DDRGK1, STK4, ADGRG6, ATG5, FXR1, PDPK1, ZFP36, HEXIM1, GSK3B, CA2, PYCARD, VAT1, KEAP1, QKI, DDIT3, PRKACA, SPOP, CDH2, SEM1, CYP19A1, YEATS4, FOXK2, FABP3, PDCD10, ALYREF, LASP1, VMP1, UXT, STARD3, PSME3, TCF3, ABCA1, PLA2G4A, GALNS, DCAF1, IKBKB, EIF2AK3, PRKAA1, CLDN1, CRYAA, CAV2, PTCH1, SLC11A2, RORA, CBX1, TIMP3, F5, SNRNP200, DDX5, KLK3, EZR, MAP2K1, PARP1, SIRT1, STAT5B, EZH2, PROS1, ARHGEF38, GSDMD, P2RX7, RPS6KB1, PRKCZ, ABCB6, ODC1, LPIN1, SPAST, MAP2K4, SLC19A1, UBA5, SLCO2B1, LYN, LGALS3, LRP2, FBXO31, PUS1, APOB, HSPD1, MAGEA5P, TPT1, RTN4, GSN, THOC2, MAGEA9, EDA, SOD2, CUL4A, TRIP10, CDK2, RANBP9, GTF2B, SIRT2, MCRS1, SCRIB, SPP1, VIM, HLCS, PAK2, UFSP2, DDX21, RPS27A, H4C1, TGM2, ITCH, STAT1, IMMT, EIF4G1, FGFR1, AKR1C3, TRIM28, NME1, TJP1, UFM1, YAP1, FAM72A, TBK1, SUV39H1, CCR2, UFL1, BLM, RPSA, FASN, HSPA9, MICA, COL3A1, CASP7, DDX3X, NR3C1, RAB7A, ACTB, RELA, PTN, DDX39B, GRB2, HSPA5, NLRP1, CD44, BTK, KRT9, CYB5R3, DAG1, PPA1, PCDH20, STPG1, EIF3E, TPI1, CHD4, RTP1, TCEAL7, SCN9A, BECN1, CUL3, INSR, PML, CSAG2, HSPB1, RAC1, KHDRBS1, CXCR4, SQSTM1, BAGE, ABCA4, PNLIPRP3, ZNF442, PRRG1, PURG, TRIML1, COL1A1, TMEM219, MYLK4, CYB5D2, ZNHIT2, ANKEF1, FAM83F, PCDHB15, ZNF22, ANKRD29, KIF6, SCGB3A2, PGCKA1, MSMP, ERVK-7, ERVK-24, GJD4, RASL10B, ABHD12B, ZCCHC14, TAS2R13. RGL1, MOSPD1, ZNF25, CSRNP3, SERPINI2, DSEL, MSI1, LONRF3,

Lampiran A.5 Hasil Data Selection (Lanjutan)

Dataset	Nama Kolom	Isi Kolom
Uniprot - Breast Cancer	Gene Symbol	LMF2, ZNF277, SLC26A10P, DSCR8, VRTN, KLK13, ZNF546, TBC1D9B, INHBE, PRIMA1, SNX8, OTOGL, IGSF21, DMRTA1, PPIAL4A, SPNS3, ZSCAN5A, CATSPERE, MANEA, P2RY14, ERI2, FBXO30, OR1E2, UGT3A2, FSTL5, BRCA2, MLF2, ERVK-21, ZNF624, HAPLN1, ERVK-9, ZNF471, ZNF43, RELCH, ERVK-8, LRRC47, BCL2L12, HERVK_113, GPR87, BRCA1, KLLN, ERVK-19, DIP2C, MT1HL1, MYEOV, SCGB2A2, SCGB1D2, SEPTIN14P20, DEPDC1B, CYP4Z2P, ZNF569, ZNF385D, GPR180, JAKMIP3, KIAA1671, ERVK-5, CDV3, B2RBL9, AHSA2P, PGBD3, PDZD4, WFDC1, EFCAB13, C1orf87, SLC67A2, ZNF75A, CST9, DOP1A, SPATA21, MARVELD1, HEATR9, FBXO8, WDR88, OR12D3, GPR45, OR1N1, ZBTB3, TCERG1L, FMR1NB, ITPRID1, SPACA7, OTOP2, ZNF532, ZNF560, CFAP337, ISLR, TOMM40L, C8orf34, MS4A5, TRERF1, TMEM164, CCDC9B, DMXL1, TTC36, MAOA, SRRM3, cytb, ABCB1, TMEM125, APRG1, PXDC1, C4orf50, FAM131A, KRTAP21-1, WDR53, KRTAP20-1, Q14267, DKFZp727E011, COI, COX1, BRCA1, COX1, BRCA1, ND5, PGR, PGR, PGR, BRCA1, SH3TC1, BRCA2, BRCA1, BRCA1, BRCA2, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA2, BRCA1, BRCA2, ABCG2, GDAP1L1, BRCA1, ERICH1, GAS8-AS1, BRCA1, BRCA1, BRCA2, BRCA1, BRCA1, BCTP1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA1, BRCA2, BRCA2, BRCA1, BRCA1, BRCA2, BRCA1, BRCA2, BRCA1, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA1, BRCA2, BRCA2, BRCA2, BRCA2, BRCA1, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA1, BRCA1, BRCA2, BRCA1, BRCA1, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA1, BRCA2, BRCA1, BRCA2, BRCA2, BRCA2, BRCA2, BRCA1, BRCA2, BRCA2, BRCA1, BRCA1, BRCA1, BRCA1, BRCA2, BRCA1, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA1, BRCA1, BRCA1, BRCA2, BRCA2, BRCA1, BRCA2, BRCA1, BRCA2, BRCA1, BRCA1, BRCA2, BRCA1, BRCA1, BRCA2, BRCA2, BRCA1, BRCA1, BRCA2, BRCA1, BRCA2, BRCA2, BRCA1, BRCA1, BRCA1, BRCA2, BRCA2, BRCA1, BRCA1, BRCA2, BRCA2, BRCA1, BRCA2, BRCA1, BRCA2, BRCA1, BRCA2, BRCA1, BRCA1, BRCA1, BRCA2, BRCA2, BRCA2, BRCA1, BRCA1, BRCA2, BRCA2, BRCA1, BRCA2, BRCA2, BRCA1, BRCA2, BRCA1, BRCA1, BRCA2, BRCA1, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA1, Q9H287, BRCA1, BRCA1, BRCA2, Q9H281, BRCA2, Q9H273, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, Q13559, BRCA2, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA2, BRCA1, Q9H279, Q9H282, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA2, BRCA2, Q96CD4, HMGB1, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA1, BRCA2, BRCA1, BRCA2, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA1, BRCA2, BRCA2, BRCA2, BRCA2, BRCA1, BRCA1, BRCA2, BRCA1, BRCA2, BRCA1, BRCA1, BRCA2, BRCA2, BRCA1, hCG_1993240, B4DWK2, PRKCA, COX1, POU5F1P4, BCAR1, BRCA1, B4DLQ5, B3KWS6, TRGC1, B3KNL6, B3KP06, B3KWE2, FAK, BRCA2, BRCA2, B3KU43, BRMS1, B7ZA85, ND5, CYTB, COX2, ND5, ATP6, COX2, ND5, COX3. POU5F1P4, ND5, ND2, POU5F1P4, POU5F1P4, POU5F1P4, CYTB, ND2, ATP6, ND2, PR, ND6, BCAR4, BRCA1, BRCA1, ND1, CYTB,

Lampiran A.6 Hasil Data Selection (Lanjutan)

Dataset	Nama Kolom	Isi Kolom
Uniprot - Breast Cancer	Gene Symbol	CYTB, POU5F1P4, POU5F1P4, NADH4, COX3, COX3, ND5, ATP6, ND2, ND2, ND5, ND5, POU5F1P4, POU5F1P4, POU5F1P4, POU5F1P4, POU5F1P4, POU5F1P4, POU5F1P4, CYTB, Q14268, ND2, ND2, CYTB, ND5, CYTB, CYTB, ND5, ND1, ND5, Q8IWP6, COX1, cytb, ND2, ND5, ND3, ND4L, BRCA1, PBR, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, MELKV4, TMEM16A, TP53, CYTB, ESR1, COX3, ND5, KCNJ3, KCNJ3, ND5, CYTB, RQCD1, SGA56M, ATP6, ND2, B2ZAH2, B2ZAH3, F2YQ21, B2ZAH4, TP53, ND1, NRG4, ND4L, CYTB, C5orf34, ESRP2, ESRP2, CD47, C12orf76, SMG8, J3QRE1, ND4, CYTB, ND4, CYTB, ND5, ND2, COX2, ND5, ND5, ND2, ND4, ND5, COX3, YB035, YK038, COX1, ND1, ND2, ATP6, CYTB, Q9NUA2, FHIT, CYB, NEDD9, RARB, COX2, COI, COI, COI, COI, COI, COI, COI, COI, COI, COI, COI, ATP6, Pact, LAGE-2, ND6, ND5, NDI, NDI, COX3, ND5, ND1, ND6, TSGA10, CYP2D6, ND4, COX3, ND2, ND1, ND2, ND5, ATP6, ATP6, ND6, B7ZM70, LIBC, B3KXB6, NADH5, ND4, TNN, ND4, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, TP53, COX3, TP53, cytb, COX1, cytb, TPM3, ATF, TP53, ATP6, ND6, NDI, ATP8, POU5F1P3, Q6RJW6, POU5F1P3, CYTB, CYTB, BRCA2, CYTB, CYTB, CYTB, CYTB, COX1, C6YB46, COX1, BRCA2, BRCA2, ND5, ND5, p53, CYTB, CYTB, BRCA2, ND5, CYTB, BRCA1, BRCA2, CYTB, BRCA2, BRCA1, BRCA1, BRCA1, BRCA1, CYTB, Q7Z2X2, CYTB, Q6RJW2, DSC3, BRCA2, BRCA2, BRCA1, CMTM1, Q6PKT8, AGER, SAT1, Q6PKT5, COII, BRCA2, BRCA2, DKFZp727A051, O14738, Q6RJW4, Q7KYZ0, BRCA2, ND6, ND4, COX1, ND1, COX2, ND5, BRCA1, BRCA2, BRCA2, BRCA1, BRCA2, BRCA2, DKFZp564A063, BRCA2, BRCA2, BRCA2, BRCA2, B2ZAH0, BRCA2, BRCA2, PIK3CA, PRLR, BRMS1, ATP6, BRMS1, ATP6, ERBB2, BRCA1, BRCA1, BRCA1, HER2, env, env, Q8NC57, env, ATP6, A0A3Q8VQF5, A9QVW2, A0A384MTS3, Q16095, Q6PKT6, Q8WYZ6, Q86XS2, PRTN3, ERBB2, Q8TD76, SCN5A, ESR1, BRCA1, A0A2S0RQF7, A0A2S0RQT0, A0A2S0RQA8, A0A2S0RQI2, BRCA1, BRCA1, A0A2S0RQH0, A0A2S0RQI1, A0A2S0RQK6, A0A2S0RQF6, A0A2S0RPZ9, A0A2S0RQ09, A0A2S0RQH1, A0A2S0RQN4, Q9NP17, SCN5A, SCN5A, A0A0S2ZYV0, Q7M4M7, A0A2S0R9G4, A0A2S0RBI6, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, BRCA2, A0A2S0R9C8, A0A2S0R9E6, A0A2S0R9H0, A0A2S0RAN4, A0A2S0RBI9, BRCA2, BRCA2, BRCA2, BRCA2, HMGB1, A0A2S0R766, A0A2S0R789, A0A2S0R9E5, A0A2S0R9F0, A0A2S0R9F4, A0A2S0R9G3, A0A2S0RA90, A0A2S0RBL0, BRCA2, A0A2S0R781, A0A2S0R999, A0A2S0R9D5, A0A2S0R9D6, A0A2S0R9F2, A0A2S0R9F7, A0A2S0R9F8, A0A2S0RA61, A0A2S0RA74, A0A2S0RA82, A0A2S0RAA0, BRCA2, A0A2S0R991, A0A2S0R9A8, A0A2S0R9E7, A0A2S0RA94, A0A2S0RAP4, A0A2S0RAQ9, SUDS3, SUDS3, AKT1, AKT1, CYP2D6, B2ZAH1, env, F2YBS5, F2YBS4, F2YBS6, env, ATP6, BRMS1L, ATP6, PALB2, ND6, PALB2, CYTB, ABCB1, SH2D3C, BRMS1, BCAR1, BRCA1, BRCA1, BRCA1, p53, BRCA1, BRCA1, BRCA1, ND5, BCAR1, ERBB2, BCAR1, BCAR1, TP53, BCAR1, BCAR1, BCAR1, BCAR1, ATP8, SHINC3, RHOA, THBS1, PLXNB1, CTTN, RHOD, RAP1GAP, JAG2, LGALS1, CASK, SH2D3C, SMG8, ND1, B3KUE2, NADH6, EAF2, B4DG32, SH2D3A, A8K2M8, COX3, A0A0S2ZYP9, BRCA2, BRCA1, A0A0S2ZYL1, A0A0S2ZYN3, A0A0S2ZYP6, BRCA1, A0A0S2ZYL2, BRCA2, A0A0S2ZYN1, A0A0S2ZZE6, D2D4A4, ABCB1, Q9BZG5, Q9BZG7, Q9BZG6. CCDC74AV2, CCDC74AV3, Q6T424, PTK2, ND1, A8K7M6, A8K9K3, RPS16, ND5, LPHH1, LPHH1, LPHH1, LPHH1, LPHH1, ND2, CDK4, ATP6, NADH5, ND6, ND1, ND4, COX2, ND4L, ATP6, MAP3K11,

Lampiran A.7 Hasil Data Selection (Lanjutan)

Dataset	Nama Kolom	Isi Kolom
Uniprot - Breast Cancer	Gene Symbol	BCAS3, ATP8, CYTB, E7CCH4, COX2, ND2, ND2, ATP6, Q14801, COX2, ND2, ATP6, ND4, ATP6, COX3, ATP8, ND5, SCN8A, SCN5A, ATP6, ATP8, TGFB1, COX3, ND3, Q16464, RPL27, Q76N35, LMO7, PLU-1, p53, keratin 19, TP53, TP53, TP53, TP53, TP53, TP53, SCN8A, TP53.

Lampiran B Data Deduplication

Lampiran B.1 Hasil Data Deduplication

Dataset	Nama Kolom	Isi Kolom
Uniprot – Breast Cancer	Gene Symbol	BRMS1, BRCA2, BRCA1, BIN2, GREB1, SNCG, CALML4, BRMS1L, BCAS1, MUC1, SGO1, DLEC1, GREB1L, CEP85L, TGFBR2, BLTP2, TFF1, SYTL2, BCAR1, BCAR3, RHOBTB2, SFXN4, KRAS, PTEN, EBAG9, AKT1, SEPTIN1, MAGED2, SMG8, CDH1, CASP8, AKIP1, TOP1, RBBP8, FAM83D, EI24, ERGIC3, MUCL1, ANKRD17, STRADA, MYLK, FBLN1, CDKN2C, MTA1, CYP1B1, KIF15, UQCC2, ZNHIT6, SLX4, RB1CC1, CUEDC2, LHCGR, FAM3C, ZNF703, SH3BGRL, AXL, CDKN1B, ITIH4, PALB2, RNF146, AGR3, PSMD6, NRG1, TRERF1, ABCG2, LETMD1, CTAG1A, PRPF31, MFGE8, TOMM6, CENPW, PRSS50, TACC1, BCAS3, PTK6, NCOA6, PIK3CA, ATAD2, CHEK2, LDOC1, SMIM22, LMO4, PPHLN1, BRDT, BCAS4, CCAR2, LZTS1, MAGEA8, PASD1, LRATD2, MLH3, LRRC26, EPCAM, AR, ADAM29, STARD8, MAGEA11, SVEP1, MLH1, NDFIP1, PLEKHA8, CEMIP, MSH6, PMS2, ARID4B, MAGEC2, MAGEA1, MSH2, MAGEA12, MAGEA4, STK11, FLT1, AFAP1, TNS4, BRINP1, EMSY, MAGEA3, OLA1, MAD1L1, TP53, MSMB, TP63, TPTE, MAGEA6, RPL13, HIC1, FLT4, DCC, KDM5B, MRE11, MAGEA2, VTCN1, PPP1R14C, ERBB4, PRICKLE4, DPH1, CASC3, PHF11, CTCF, MANF, BCAS2, KCNK9, PBOV1, MELK, CABLES1, BRAF, BRAP, ZFHX3, CTNNB1, EHBP1, AKAP13, HEATR6, MAPK12, VWA2, CTU1, FGFR4, SLC67A1, ANLN, UNC5A, NCOA3, UNC5B, KLK8, MAP3K9, UHRF1, DPP4, EIF3D, UNC5C, VWA5A, STARD13, RLIM, TP53BP2, SEPTIN9, USP32, NEDD9, ANKRD30A, BLID, BPHL, ERBB2, CHMP2A, EP300, PTK2, UBE3A, ASAP3, LEF1, RBM5, SYNE1, SMAD4, EGFR, AIM2, ING1, SMAD3, HMGB1, NCOA1, KDR, TMEM33, PLK1, KLK10, STARD10, PPM1D, VOPP1, SUSD2, SRARP, CLCA2, ENOX1, ZNF668, ABRAXAS1, MAGEA10, ELF3, PBRM1, TPM3, NLRP8, SH2D3A, XRCC3, POTEF, Q5YLB2, XRRA1, BOD1L1, B3GNT8, ANKRD30B, TPM2, ACRBP, POU5F1B, TP53I3, RAD51D, BAG1, FAM83A, ST18, BRIP1, CXCL17, IRX5, HOXA3, SNAI2, ZNF202, RAD51C, TPM1, AKAP3, PYHIN1, DGAT2L6, MYH1, CDK1, ARMC12, PKDREJ, TOP2B, UBE2V1, TRPM8, GYG2, GLS2, TRIM24, AVPR2, SLC9A2, NUPR1, ADAMTSL3, SEMA6B, SLC39A10, FN1, PMPCA, ICE2, ARHGAP35, AURKA, ATM, MYH9, FAIM2, ZNF432, COL19A1, ZNF318, SEMA5B, BIRC5, RAB25, MDM4, ETV5, EPG5, PHB2, WDCP, SLC2A5, RPAP1, GKN1, RABL3, PEPD, LGALS1, TUBG1, RIC3, WWP1, PDIA3, TNN, CTIF, CCNG1, ARAP3, UTP20, MAP3K6, KRT7, RFX2, XRCC2, MAP1B, LMAN1, SIGMAR1, KIT, SCARB2, PPP1R8, FGFR2, TRA, XAGE1A, TRB, SIX4, PLAUR, KRT16, CHRNA7, SGK3, TXN, BEX2, HNRNPK, HDAC1, CTTN, RHOA, SCGB2A1, RAPH1, APC2, YBX1, SLC3A2, KPNA2, POLQ, LCK, RASGRF2, DNASE1L3, RAP1GAP, DERPC, PRMT2, PRRC2A, CRIPTO3, ENTREP3, ROBO1, NENF, BPIFA4P, MCF2L2, SUSD3, CUX1, ECT2, DEPP1, NBN, RERG, RRP9, GPER1, MTMR8, CIC, CNGA2, NET1, LIMD2, PHKB, SECTM1, TMEM161A, CHD5, POSTN, ENOX2, MACROD1, LACTB, RASAL2. LYPD3, MTUS1, RAD51, HOXB13, KCNH4, RNF115, PTPN14, XIRP1, ZFP64, BCL2L14, SP110, RAD54L, FASTKD3, TMEM39A, CST4, LGALS2, SLC25A51, NIP7, TSKS, ARHGAP11A, PRLR,

Lampiran B.2 Hasil Data Deduplication (Lanjutan)

Dataset	Nama Kolom	Isi Kolom
Uniprot – Breast Cancer	Gene Symbol	MIEN1, SUDS3, MINAR1, ENPEP, CSMD3, FRK, LIMD1, NAA25, YY2, ARFRP1, HOXA4, RNF149, NOA1, EDRF1, TENM1, GNB1L, NPAP1, GOLIM4, KLK15, FAM217B, MADD, NEK3, CDKN3, TSTD1, XDH, ENAH, GNAS, SRCIN1, NECTIN4, ZNF804A, MMP11, ADAM19, ACTL9, THOC5, FEM1C, SIK3, APC, PEG10, MYOD1, SMAGP, HDLBP, TRARG1, ADGRF4, RHNO1, BCL9L, KRT73, PPM1F, GPC6, FOLR2, TXNDC15, RCE1, SPAG6, CCNQ, COLGALT2, PRR5, AKAP6, AFF1, CLASRP, CDH20, MAGEE1, CCAR1, DSE, ELAPOR1, CYP4Z1, TECTA, SHANK1, HSPA14, DNAJC24, CDH10, DCLK3, GOLGB1, DHX32, JADE3, NPBWR1, SHROOM2, N4BP2, SORCS1, CUTC, UTS2R, PACC1, TSC22D4, OBSCN, CREB3L4, SYT9, SLC5A8, MIA, SLC22A9, TTLL3, MTHFD1L, ADNP, CDCP1, ANAPC5, ITIH5, CUBN, HK3, BATF2, ADAMTS15, CLSTN2, PIK3R5, GABRP, FABP4, RASEF, IRF8, SMTN, SBNO1, TREML1, MAP7D2, RACK1, ZDHHC4, ZNF281, THBS3, MFAP5, SLC39A12, TCF7L1, EFS, OVGP1, CPNE3, ZBTB8B, NDUFA3, INSL3, GPC2, HDC, CBFA2T3, MROH7, DAZAP1, ST6GALNAC1, SDHD, PLXDC1, SLC39A6, RYBP, AGAP2, FERD3L, PRUNE2, WNT7B, CASP10, GRIN2D, PRDM5, VSTM4, EREG, DNAH9, ZFYVE26, ALS2CL, NLRP14, ADHFE1, CNTN6, GRIK3, SKIC2, RNF11, EPHA3, WFDC2, HEPACAM2, ZNF646, MMP10, CXCR5, MPLKIP, ADAM12, APOC4, DDX10, INA, FOXP4, ZNF276, IL1RAPL2, SERPINE2, SAFB2, WNT8B, PARP8, MMP17, ZSCAN16, CFAP77, OLAH, IRX4, MAK, PREX2, CLIC3, TESK2, XBP1, CAPN6, TBX22, ELOA2, SIPA1L1, TLL1, RANBP1, DOK1, RGCC, MAP3K19, TPD52, SRGAP3, NDUFA2, ARHGAP29, DNAJC21, SPOCD1, NME8, TRMT10A, PPM1E, LRRC7, POP1, SYT3, UQCC1, TRAPPC8, IFNA2, PANX2, STRBP, KCNJ15, PRRT1, DDX18, NLE1, DENR, PCDH8, PFKFB4, FRMPD1, CNNM4, KBTBD8, TRAPPC12, ZNF541, CSNK1A1L, PAQR4, CCN4, TMED1, SH3PXD2A, EFHD1, ARFGEF3, CDKN1C, DSCAML1, CCDC170, ECM1, H2AC21, ANKK1, KLK14, SASH1, SRMS, LAMP3, WNT16, ZNF652, GAB1, KDF1, BARD1, FKTN, PHB1, TFAP2D, ABCB5, MTDH, OTOF, CD300E, MIA2, NF1, SLC6A3, THOC1, SLC66A2, TBRG1, PRDM14, TTLL10, CCNB3, CD248, SALL3, AMER2, CELA1, FAM110B, HMMR, PTHLH, CPO, PODXL, AMTN, PDILT, WWOX, SLC1A7, WWC1, ADARB2, LIN28B, TPM4, PPP1R3A, ARHGEF4, FOXA1, HID1, SCARF2, ZNF436, MED14, GALNT5, TRMT2A, JPT2, TACC2, BGN, ITGA9, ISG20, RPRD1A, BOC, ZNF644, OSTC, CFL2, CD2, DDX59, CYB5R4, CLPTM1L, OSBPL11, NRK, PRAF2, PER1, ARID1A, CORO2B, LONRF2, AMDHD2, HTR3C, NPAS2, RNF182, HOXC9, RHOB, ACAP1, EVC2, NCAPG, COL11A1, TIMELESS, NUP214, TMPRSS6, BPIFB2, RAC2, FOLH1, AMPD1, SREBF2, PLPPR2, CLIP1, MLLT11, ATP2A3, TCHP, ADAMTS18. ZNF217, RASSF1, MAP2, ZNF350, PHF7, AMPH, NDUFAF4, PLD2, WBP4, APCS, NDUFA8, ABCB8, KCNIP3, PEX5L, GMCL2, ASTN2, EOMES, NQO2, EIF5, NID2, TRIM29, LRRFIP1, NLRP9, P3H2, EIF1AD, ADAMTS19, CAVIN3, PADI3, FARP1, PES1, SSNA1, PRG2, EFNA1, PIGS, TREM1, ZNF480, ZNF540, GUCY2F, VPS72, ACO2, KDM3A, FAM210A, PAK6, SAFB, GFRAL, DAW1, COG3, CDC27, GSE1, PDZRN4, CAMK1G, MOK,

Lampiran B.3 Hasil Data Deduplication (Lanjutan)

Dataset	Nama Kolom	Isi Kolom
Uniprot – Breast Cancer	Gene Symbol	AGR2, SPAG17, PDCD4, SEC23B, EPHB6, TMEM140, KCNB2, SNRK, ZNF232, DLG5, PELP1, GGA1, KLK5, SMG1, DCHS1, RNPEPL1, FBXW7, GLI1, ZNF155, STK32C, SUCO, ACTN4, ECD, CORO1B, NPR1, NCEH1, PTPRU, NODAL, HPS3, CDK15, CD109, NDST3, PBX4, APPL1, AMFR, WNT7A, CACNB1, TNNI3K, BCL11B, PGAP3, H2AC25, TCF7L2, ARHGEF1, PRL, SDCBP2, PER2, ELOC, MKRN3, WNT3A, BAX, INTS13, IRS4, TPD52L1, FANCG, CTNND2, WBP2, RARB, H2AC1, RPL6, VPS18, GRM1, CLSTN3, STAM, PLA2G3, DPYSL2, OSBP, SNAI1, MYB, RBP3, AGTPBP1, KLK4, LAMB3, MFN1, ATP6V1E1, ETFDH, PRKAA2, RAB5C, ANGPTL4, TPX2, S100A2, HOXA1, HSD17B7, CLOCK, ZBTB7A, CDK6, CYP1A1, CSF1R, ADAR, FBXL2, ACSL5, KCNA5, SLAMF1, FCRL3, DDO, MACF1, LATS1, ASL, ABL2, OCA2, CDH13, DCAF7, ZEB2, CCN5, UBR4, NIPBL, TFAP2A, BAP1, TBXAS1, SPTAN1, HDAC7, DNALI1, SLC12A5, OMA1, JAK3, CAMSAP2, GFI1B, PKHD1, JAK1, DEFB1, BRCC3, ALK, SLC17A5, LZTS2, METTL5, GABPB1, WAC, SCN3B, S100A11, FOXR2, NEK2, NSD3, FADS2, H2AC4, CITED4, GNA12,TRIP4, CAD, PDCD6, S100A9, ACADM, GRIK2, TIAM1, SLC35A2, ARFGEF1, SRSF1, RAD50, ZNHIT1, B4GALNT2, NOS2, IFI27, SACS, WNT10B, H2AC14, VPS13A, TBX2, MAD2L1, ERVK-6, POU4F2, ST7, H2AC20, EPHB4, GLI3, GRID2, RECQL, STRAP, STAT5A, SCP2, CEBPA, TAX1BP1, ARID1B, PTPRS, TCOF1, PRF1, CACNB2, EPSTI1, MICB, DVL3, CCNDBP1, POLH, NF2, TRIM25, RPS6KA3, EIF2AK2, PRPS1, FAAH, SUZ12, CD46, CLU, RET, IDH1, ACIN1, LIG3, GNA13, H2BC14, DHX40, H2AC7, SORL1, ERVK-18, SF3A1, H2AC6, H2AC18, UBE2D1, SLC29A1, LTF, RHEB, DFFA, BCCIP, NOS3, RPL19, NFATC1, EED, PHGDH, H2AC11, SIVA1, ASAP1, NR1D1, PDGFRB, LEP, MPO, BCL11A, BCL10, PTPRC, ATP11C, CRX, MMP14, KNG1, CDKL5, ESR2, PIN1, RPS6KA5, RBM10, UIMC1, THOC7, SCNN1G, GRB7, ACTR3, SNX6, KCND3, RBP1, FUS, NOTCH2, KRT17, WDR26, PLK3, SPTLC1, NEDD4, EXTL3, LMNB2, MCM3AP, NONO, PIP, IRAK1, IL6ST, BANP, CLCN1, DRC4, ATF2, SYNE2, ABCA3, MCOLN1, CAV1, XRCC1, DLG3, AKT2, H1-4, SERPINA5, MST1R, SLCO1A2, GPSM2, MKI67, GLI2, PNPLA2, PXN, BAK1, SMURF2, JPT1, EPN1, ARID4A, PDGFB, PFN1, AURKB, TFG, VWF, TICAM1, PSMA2, DDX17, TSG101, HDAC6, PGAM1, MFF, UGT1A9, HSD11B1, NUP98, ERCC3, DYSF, ANK1, TEK, RNF31, NR3C2, PRSS1, TCF7, RASSF5, ERCC4, MMP13, FZD4, ACSL4, SOS1, LRP1, STAT3, TTN, TRAF4, MDK, PSMC5, RRP1B, DHX15, KRT18, ABCB1, PQBP1, JAK2, PGR, PRKCA, CSNK1D, KRT19, LIMK1, DDRGK1, STK4, ADGRG6, ATG5, FXR1, PDPK1, ZFP36, HEXIM1, GSK3B, CA2, PYCARD, VAT1, KEAP1, QKI, DDIT3, PRKACA, SPOP, CDH2, SEM1, CYP19A1, YEATS4, FOXK2, FABP3, PDCD10, ALYREF, LASP1, VMP1, UXT, STARD3, PSME3, TCF3, ABCA1, PLA2G4A, GALNS, DCAF1, IKBKB, EIF2AK3, PRKAA1, CLDN1, CRYAA, CAV2, PTCH1, SLC11A2, RORA, CBX1, TIMP3, F5, SNRNP200, DDX5, KLK3, EZR, MAP2K1, PARP1, SIRT1, STAT5B, EZH2, PROS1, ARHGEF38, GSDMD, P2RX7, RPS6KB1, PRKCZ, ABCB6, ODC1, LPIN1, SPAST, MAP2K4, SLC19A1, UBA5, SLCO2B1, LYN, LGALS3, LRP2, FBXO31, PUS1, APOB, HSPD1, MAGEA5P, TPT1, RTN4, GSN, THOC2, MAGEA9, EDA, SOD2. CUL4A, TRIP10, CDK2, RANBP9, GTF2B, SIRT2, MCRS1, SCRIB, SPP1, VIM, HLCS, PAK2, UFSP2, DDX21, RPS27A, H4C1, TGM2, ITCH, STAT1, IMMT, EIF4G1, FGFR1, AKR1C3, TRIM28, NME1, TJP1, UFM1, YAP1, FAM72A, TBK1, SUV39H1, CCR2, UFL1, BLM, RPSA, FASN, HSPA9, MICA, COL3A1, CASP7, DDX3X, NR3C1, RAB7A, ACTB, RELA, PTN, DDX39B, GRB2, HSPA5, NLRP1, CD44, BTK, KRT9, CYB5R3, DAG1, PPA1, PCDH20,

Lampiran B.4 Hasil Data Deduplication (Lanjutan)

Dataset	Nama Kolom	Isi Kolom
Uniprot – Breast Cancer	Gene Symbol	STPG1, EIF3E, TPI1, CHD4, RTP1, TCEAL7, SCN9A, BECN1, CUL3, INSR, PML, CSAG2, HSPB1, RAC1, KHDRBS1, CXCR4, SQSTM1, BAGE, ABCA4, PNLIPRP3, ZNF442, PRRG1, PURG, TRIML1, COL1A1, TMEM219, MYLK4, CYB5D2, ZNHIT2, ANKEF1, FAM83F. PCDHB15, ZNF22, ANKRD29, KIF6, SCGB3A2, PGCKA1, MSMP, ERVK-7, ERVK-24, GJD4, RASL10B, ABHD12B, ZCCHC14, TAS2R13, RGL1, MOSPD1, ZNF25, CSRNP3, SERPINI2, DSEL, MSI1, LONRF3, LMF2, ZNF277, SLC26A10P, DSCR8, VRTN, KLK13, ZNF546, TBC1D9B, INHBE, PRIMA1, SNX8, OTOGL, IGSF21, DMRTA1, PPIAL4A, SPNS3, ZSCAN5A, CATSPERE, MANEA, P2RY14, ERI2, FBXO30, OR1E2, UGT3A2, FSTL5, MLF2, ERVK-21, ZNF624, HAPLN1, ERVK-9, ZNF471, ZNF43, RELCH, ERVK-8, LRRC47, BCL2L12, HERVK_113, GPR87, KLLN, ERVK-19, DIP2C, MT1HL1, MYEOV, SCGB2A2, SCGB1D2, SEPTIN14P20, DEPDC1B, CYP4Z2P, ZNF569, ZNF385D, GPR180, JAKMIP3, KIAA1671, ERVK-5, CDV3, B2RBL9, AHSA2P, PGBD3, PDZD4, WFDC1, EFCAB13, C1orf87, SLC67A2, ZNF75A, CST9, DOP1A, SPATA21, MARVELD1, HEATR9, FBXO8, WDR88, OR12D3, GPR45, OR1N1, ZBTB3, TCERG1L, FMR1NB, ITPRID1, SPACA7, OTOP2, ZNF532, ZNF560, CFAP337, ISLR, TOMM40L, C8orf34, MS4A5, TMEM164, CCDC9B, DMXL1, TTC36, SRRM3, cytb, TMEM125, APRG1, PXDC1, C4orf50, FAM131A, KRTAP21-1, WDR53, KRTAP20-1, Q14267, DKFZp727E011, COI, COX1, ND5, SH3TC1, GDAP1L1, ERICH1, GAS8-AS1, BCTP1, Q9H287, Q9H281, Q9H273, Q13559, Q9H279, Q9H282, Q96CD4, hCG_1993240, B4DWK2, POU5F1P4, B4DLQ5, B3KWS6, TRGC1, B3KNL6, B3KP06, B3KWE2, FAK, B3KU43, B7ZA85, COX2, ATP6, COX3, ND2, PR, ND6, BCAR4, ND1, NADH4, Q14268, Q8IWP6, ND3, ND4L, PBR, MELKV4, TMEM16A, RQCD1, SGA56M, B2ZAH2, B2ZAH3, F2YQ21, B2ZAH4, NRG4, C5orf34, ESRP2, CD47, C12orf76, J3QRE1, ND4, YB035, YK038, Q9NUA2, FHIT, CYB, Pact, LAGE-2, NDI, TSGA10, CYP2D6, B7ZM70, LIBC, B3KXB6, NADH5, ATF, ATP8, POU5F1P3, Q6RJW6, C6YB46, p53, Q7Z2X2, Q6RJW2, DSC3, CMTM1, Q6PKT8, AGER, SAT1, Q6PKT5, COII, DKFZp727A051, O14738, Q6RJW4, Q7KYZ0, DKFZp564A063, B2ZAH0, HER2, env, Q8NC57, A0A3Q8VQF5, A9QVW2, A0A384MTS3, Q16095, Q6PKT6, Q8WYZ6, Q86XS2, PRTN3, Q8TD76, SCN5A, A0A2S0RQF7, A0A2S0RQT0, A0A2S0RQA8, A0A2S0RQI2, A0A2S0RQH0, A0A2S0RQI1, A0A2S0RQK6, A0A2S0RQF6, A0A2S0RPZ9, A0A2S0RQ09, A0A2S0RQH1, A0A2S0RQN4, Q9NP17, A0A0S2ZYV0, Q7M4M7, A0A2S0R9G4, A0A2S0RBI6, A0A2S0R9C8, A0A2S0R9E6, A0A2S0R9H0, A0A2S0RAN4, A0A2S0RBI9, A0A2S0R766, A0A2S0R789, A0A2S0R9E5, A0A2S0R9F0, A0A2S0R9F4, A0A2S0R9G3, A0A2S0RA90, A0A2S0RBL0, A0A2S0R781, A0A2S0R999, A0A2S0R9D5, A0A2S0R9D6, A0A2S0R9F2, A0A2S0R9F7, A0A2S0R9F8, A0A2S0RA61, A0A2S0RA74, A0A2S0RA82, A0A2S0RAA0, A0A2S0R991, A0A2S0R9A8, A0A2S0R9E7, A0A2S0RA94, A0A2S0RAP4, A0A2S0RAQ9, B2ZAH1, F2YBS5, F2YBS4, F2YBS6, SHINC3, THBS1, RHOD, JAG2, CASK, B3KUE2, NADH6, EAF2, B4DG32, A8K2M8, A0A0S2ZYP9, A0A0S2ZYL1, A0A0S2ZYN3, A0A0S2ZYP6, A0A0S2ZYL2, A0A0S2ZYN1, A0A0S2ZZE6, D2D4A4, Q9BZG5, Q9BZG7, Q9BZG6, CCDC74AV2, CCDC74AV3, Q6T424, A8K7M6, A8K9K3, RPS16, LPHH1, CDK4, MAP3K11, E7CCH4, Q14801, SCN8A, TGFB1, Q16464, RPL27, Q76N35, PLU-1, keratin 19

Lampiran C Hasil anggota komunitas dengan NNC terbaik

Lampiran C.1 Anggota tiap komunitas

Komunitas Ke-	Total Protein	Jumlah Protein Overlap	Anggota Komunitas
1	430	190	ABRAXAS1, ACSL4, ACSL5, ACTB, ADNP, AGAP2, AKAP12, AKR1C3, AKT1, AKT2, ALK, ANAPC4, ANAPC5, ANLN, AP2A1, APC, APC2, APOL1, APPL1, AR, AREG, ARFGEF3, ARHGEF1, ARHGEF4, ARID1A, ARID1B, ARID4A, ARID4B, ASAP1, ASL, ATF2, ATG5, ATM, ATR, AURKA, AURKB, AXL, BAK1, BANP, BAP1, BARD1, BAX, BCAR1, BCL11A, BCL11B, BCL9L, BECN1, BIRC5, BLM, BOC, BRAF, BRAP, BRCA2, BRCC3, BRIP1, BRMS1L, BTK, CALML4, CAMK1G, CASP10, CASP8, CAV1, CAV2, CAVIN1, CAVIN3, CBFB, CCAR2, CCNB3, CD24, CD44, CDC14A, CDC27, CDH1, CDH13, CDH2, CDK1, CDK2, CDK4, CDK6, CDKN1B, CDKN1C, CDKN2C, CDKN3, CEBPA, CENPH, CENPW, CHD4, CHD5, CHEK2, CLDN1, CLDN7, CLOCK, CLU, CNGA2, CNKSR2, CREB3L4, CSF1, CSNK1A1, CSNK1A1L, CSNK1D, CTCF, CTNNB1, CTNND2, CTTN, CUBN, CUL3, CUL4A, CYP19A1, CYP1A1, CYP1B1, CYP2D6, DCAF1, DCAF7, DCC, DDX17, DDX5, DKFZp564A063, DLG3, DLG5, DVL3, ECT2, EED, EGFR, EHMT1, EIF2AK2, EOMES, EP300, EPCAM, EPN1, ERBB4, ERCC3, ERCC4, EREG, ESR1, ESR2, ETV4, EZH2, EZR, FABP4, FAM83D, FANCG, FASN, FBXW7, FGF8, FGFR1, FGFR2, FGFR4, FLNB, FLT1, FLT4, FN1, FOXA1, FOXK2, FXR1, G3BP2, G6PC1, GAB1, GATA3, GEN1, GLI1, GLI2, GLI3, GPER1, GPNMB, GRB2, GRB7, GSK3B, GTF2A1, GTF2B, H2AC11, HDAC7, HEATR6, HER2, HEXIM1, HIPK2, HK3, HMGB1, HMMR, HNRNPK, HSD11B1, HSD17B7, HSD17B8, HSPB1, IFI16, IFNA2, IFNB1, IGFBP3, IKBKB, IL6ST, ING1, INSR, IRS4, ITCH, ITGB3BP, JAK1, JAK2, JAK3, KBTBD8, KCNQ5, KDM6A, KDR, KEAP1, KIF14, KIF15, KIT, KL, KLF5, H2AC20, H2AC4, H2AC6, H4C1, HDAC1, HDAC6, KLHL22, KMT2C, KRAS, KSR1, LCK, LDB1, LEF1, LEP, LGALS3, LIFR, LIG1, LIG3, LIPE, LOXL2,LPHH1, LRP1, LRP2, LTF, LYN, MAD1L1, MAD2L1, MAGEC2, MAOA, MAP1B, MAP2K1, MAP2K4, MAPK12, MCM3AP, MDM4, MED14, MELK, MKI67, MLH1, MLH3, MPO, MRE11, MSH2, MSH6, MTA1, MTA3, MUC1, MYOD1, NAMPT, NBN, NCAPG, NCOA1, NCOA3, NCOA6, NEDD4, NEK2, NF1, NFATC1, NLRC5, NOS2, NOS3, NPAS2, NR3C1, NR3C2, NRG1, NRG4, NT5C1A, NUP133, NUP214, NUP98, ODC1, OLAH, OXSM, PALB2, PARP1, PBRM1, PDGFB, PDGFD, PDGFRB, PDPK1, PER2, PGR, PHB1, PHB2, PHKB, PIK3CA, PIK3R4, PIK3R5, PIN1, PLK1, PLK4, PML, PMS2, POLR2F, POU2F1, PPM1D, PRKAA1, PRKAA2, PRKACA, PRKCA, PRKCE, PRKCZ, PRKD1, PRL, PRLR, PRTN3, PSMA1, PSMA2, PSMB5, PSMC5, PSMD6, PSME3, PTCH1, PTEN, PTK2, PTK2B, PTK6, PTN, RASSF1, RASSF5, RB1CC1, RBBP8, RELA, RFC4,

Lampiran C.2 Anggota tiap komunitas (Lanjutan)

Komunitas Ke-	Total Protein	Jumlah Protein Overlap	Anggota Komunitas
1	430	190	RHEB, RHOA, RHOBTB2, RORA, RPS27A, RPS6KA4, RPS6KA5, RPS6KB1, RUNX1T1, SAT1, SCRIB, SEM1, SGK3, SGO1, SIRT1, SIRT2, SLC35A2, SLX4, SMAD2, SMAD3, SMAD4, SMURF2, SNAI1, SNAI2, SORBS1, SOS1, SPOP, SQSTM1, STAM, STAT1, STAT3, STAT4, STAT5A, STAT5B, STK11, STK36, STRADA, SUDS3, SUV39H1, SUZ12, TBK1, TCF3, TCF7, TCF7L1, TCF7L2, TCHP, TENM1, TGFB1, TGFBR2, TIAM1, TJP1, TMEM219, TNFRSF10A, TOP1, TOP2B, TP53, TP53BP2, TPX2, TRA, TRIB3, TRIM28, TUBG1, TWIST1, UBE2D1, UBE2T, UBE3A, UGT1A9, UGT3A2, UHRF1, UIMC1, UNC13B, UNC5A, UNC5B, UNC5C, VMP1, WNT3A, WRN, WWP1, XRCC1, XRCC2, XRCC3, YAP1, YBX1, ZEB2, ZNF350, ZNF436, ZNF569
2	62	24	ABRAXAS1, BARD1, CDK2, CDKN1B, CHMP2A, CSNK1A1, DDX10, DDX18, DDX21, DDX47, DENR, EIF3D, EIF3E, EIF3M, EIF4A2, EIF4G1, EIF5, H2AC11, H2AC18, H2AC4, H2AC6, HDAC6, MDM4, MFF, MICALL1, NEDD4, NIP7, NLE1, NOP9, PDCD4, PDCD6, PES1, POP1, PSMA1, PSMA2, PSMB5, PSMC5, PSMD6, RACK1, RLIM, RNF31, RPL13, RPL19, RPL27, RPL6, RPS16, RPS27A, RPS9, RPSA, RRP1B, RRP9, SMURF2, SQSTM1, STAM, TP53, TPT1, TRIM25, TSG101, UBE2D1, UBE2V1, UIMC1, UTP20
3	41	17	ATM, ATP6, ATP6V0B, ATP6V1E1, ATP8, BARD1, BLM, BRCA2, BRIP1, CHEK2, COX1, COX2, CYTB, FOXRED1, MRE11, NBN, ND1, ND2, ND3, ND4, ND4L, ND5, ND6, NDUFA1, NDUFA2, NDUFA3, NDUFA8, NDUFAF4, PALB2, PMPCA, PMS2, Q5YLB2, RAD51, RAD51C, RAD51D, RBBP8, SDHD, SEM1, UQCRC2, WBP2, XRCC3
4	18	18	AKT1, EGFR, EP300, ESR1, HDAC1, IFNA2, IFNB1, IL6ST, JAK1, JAK2, JAK3, PML, Q5YLB2, STAT1, STAT3, STAT4, STAT5A, STAT5B
5	42	37	ADCY8, AKAP12, AKAP13, AKT1, AR, BRAF, CALML4, CAV1, CAV2, CSNK1A1L, CSNK1D, CTNNB1, EP300, ESR1, ESR2, GLI1, GLI2, GLI3, GSK3B, KRAS, LIPE, MAP2K1, MAPK12, MYLK, MYLK3, NCOA1, NCOA3, NFATC1, NOS2, NOS3, NR3C1, PGR, PRKAA1, PRKAA2, PRKACA, PTK2, RASGRF2, RELA, RHOA, RPS6KB1, SCN3B, SCN5A
6	18	18	AKT1, CTNNB1, EP300, HDAC1, KRAS, LEF1, PTEN, SMAD2, SMAD3, SMAD4, SMURF2, SNAI1, TCF7, TCF7L1, TCF7L2, TGFB1, TGFBR2, YAP1
7	31	31	ANAPC4, ANAPC5, AURKA, AURKB, BIRC5, CCNB3, CDC27, CDK1, CDK2, CDKN1B, CDKN1C, CDKN3, ECT2, HMMR, KIF14, KIF15, MAD1L1, MAD2L1, MCM3AP, MELK, MKI67, NCAPG, NEK2, NUP98, PLK1, PLK4, Q5YLB2, SGO1, TP53, TPX2, TRA

Lampiran C.3 Anggota tiap komunitas (Lanjutan)

Komunitas Ke-	Total Protein	Jumlah Protein Overlap	Anggota Komunitas
8	38	35	ACTB, AKAP13, AKT1, ANLN, ARHGEF1, ASAP1, BCAR1, CDKN1B, CFL2, CTNNB1, CTTN, ECT2, EGFR, ESR1, FN1, GNA12, GNA13, GRB2, KRAS, LIMK1, LYN, PFN1, PIK3CA, PLD2, PRKACA, PRKCZ, PTEN, PTK2, PTK2B, PTK6, PXN, RAC2, RALGDS, RB1CC1, RHOA, SMURF2, TGFBR2, TIAM1
9	33	31	AKT1, AKT2, ATG5, ATM, ATR, BECN1, BTK, CASP8, CTNNB1, CUL3, EIF2AK2, GSK3B, HDAC6, HMGB1, IKBKB, IRAK1, IRS4, KEAP1, MAP1B, MUC1, PDPK1, PIK3CA, PIK3R4, PIK3R5, PRKCZ, PTEN, RB1CC1, RELA, RPS27A, SQSTM1, TAX1BP1, TBK1, TP53
10	14	14	AKT1, CALML4, CAV1, CAV2, CAVIN1, CTNNB1, EGFR, ESR1, ESR2, KDR, NOS3, PIK3CA, PRKACA, PTEN
11	41	18	ANK1, ANK2, BGN, BMP1, CD2, CD44, CDH1, COL11A1, COL1A1, COL3A1, CXCR4, EGFR, EPCAM, ESR1, EZR, FN1, HER2, IGFBP3, ITGA9, LAMA2, LAMC1, LCK, LGALS3, MMP14, MMP2, NF2, NID1, NID2, OBSCN, POSTN, PTK2, PTPRC, PXN, SPP1, STAT3, TGFB1, THBS1, TIAM1, TIMP3, TNN, TTN
12	14	13	AKT1, APC, CD44, CDH1, CDH2, CLDN7, CTNNB1, EGFR, EPCAM, HER2, PTEN, S100A8, SNAI1, TJP1
13	24	13	ACTB, ACTN4, AKT1, ARID1A, ARID1B, CFL2, DSTN, ENAH, ESR1, ESRP2, EVL, EZR, HER2, LMO7, MYH1, MYH9, PBRM1, PFN1, PRKCA, RHOA, TJP1, VPS72, YEATS4, ZNHIT1
14	19	12	BARD1, CHD4, EP300, EZH2, H1-4, H2AC11, H2AC12, H2AC14, H2AC18, H2AC20, H2AC21,
14	19	12	H2AC4, H2AC6, H2AC7, H2BC14, H4C1, NSD3, RPS27A, SUZ12
15	14	4	ALYREF, DDX39B, DDX5, GLI2, MAGOHB, SRSF1, THOC1, THOC2, THOC3, THOC5, THOC6, THOC7, TOP1, TOP2B
16	10	9	AKAP13, ARHGEF1, CD44, CXCR4, DPP4, GNA12, GNA13, PTK2B, PTPRC, RHOA

Lampiran D Hasil Enrichment

Lampiran D.1 Hasil Enrichment Komunitas 1

No	GO Biological Process
1	Regulasi siklus sel mitosis (19.39%, -59.48)
2	Jalur pensinyalan protein reseptor terkait enzim (18.46%, -49.90)
3	Perkembangan kelenjar (14.49%, -42.98)
4	Remodeling kromatin (17.52%, -41.75)
5	Regulasi proses modifikasi protein (16.59%, -40.56)
6	Aktivasi sel (17.29%, -38.79)
7	Regulasi negatif proliferasi populasi sel (17.29%, -38.14)
8	Respons seluler terhadap stimulus hormon (14.95%, -37.67)
9	Organisasi kromosom (13.55%, -37.04)
10	Regulasi proliferasi sel epitel (12.15%, -34.58)
11	Regulasi proses metabolisme DNA (13.08%, -32.48)
12	Regulasi kaskade MAPK (14.95%, -32.21)
13	Respons terhadap radiasi (12.38%, -32.06)
14	Proliferasi populasi sel (13.32%, -30.94)
	GO Cellular Component
1	-
No	**GO Molecular Function**
1	Aktivitas protein kinase (17.06%, -46.50)
2	Pengikatan kromatin (16.82%, -42.82)
3	Pengikatan kinase (17.76%, -40.58)
No	**KEGG Pathway**
1	Jalur kanker (21.50%, -71.63)
2	Jalur pensinyalan PI3K-Akt (13.08%, -40.30)
3	Hepatitis B (9.11%, -35.74)

Lampiran D.2 Hasil Enrichment Komunitas 2

No	GO Biological Process
1	Translasi sitoplasmik (25.81%, -23.59)
2	Biogenesis kompleks ribonukleoprotein (33.87%, -21.33)
3	Proses katabolisme protein yang bergantung pada modifikasi (27.42%, -14.32)
4	Inisiasi translasi sitoplasmik (11.29%, -11.89)
5	Proses virus (14.52%, -8.43)
6	Regulasi transisi fase siklus sel mitosis (16.13%, -8.29)
7	Ubiquitinasi protein (17.74%, -7.50)
8	Regulasi positif biogenesis komponen seluler (16.13%, -7.02)
9	Regulasi positif proses katabolisme protein (9.68%, -5.30)
10	Regulasi negatif aktivitas katalitik (8.06%, -4.82)
11	Biogenesis subunit besar ribosom (6.45%, -4.65)
12	Regulasi proses modifikasi protein (12.90%, -4.25)
13	Perakitan organel tanpa membran (9.68%, -3.83)
14	Proliferasi populasi sel (13.32%, -30.94)
No	**GO Cellular Component**
1	Granula ribonukleoprotein (14.52%, -4.84)
2	Preribosom, prekursor subunit kecil (4.84%, -5.53)
No	**GO Molecular Function**
1	Pengikatan protein bergantung pada modifikasi poliubiquitin (8.06%, -6.93)
2	Aktivitas RNA helicase (8.06%, -6.38)
3	Pengikatan cadherin (9.68%, -4.20)
	KEGG Pathway
1	Necroptosis (11.29%, -7.26)
2	Shigellosis (9.68%, -4.87)

Lampiran D.3 Hasil Enrichment Komunitas 3

No	GO Biological Process
1	Perakitan kompleks NADH dehidrogenase (17.50%, -13.05)
2	Pemrosesan patahan ganda DNA (15.00%, -12.71)
3	Respons seluler terhadap radiasi pengion (15.00%, -9.24)
4	Respons terhadap stres oksidatif (15.00%, -4.85)
	GO Cellular Component
1	Kompleks ATPase dua sektor pengangkut proton (10.00%, -6.20)
	GO Molecular Function
1	Aktivitas protein kinase (17.06%, -46.50)
	KEGG Pathway
1	Fosforilasi oksidatif (50.00%, -36.33)
2	Rekombinasi homolog (35.00%, -30.88)
3	Penyakit hati berlemak non-alkoholik (22.50%, -12.28)
4	Jalur anemia Fanconi (17.50%, -12.10)

Lampiran D.4 Hasil Enrichment Komunitas 4

No	GO Biological Process
1	Regulasi adhesi sel-ke-sel (58.82%, -13.35)
2	Regulasi positif adhesi sel-ke-sel (41.18%, -9.43)
3	Regulasi komitmen nasib sel (23.53%, -7.85)
4	Regulasi jalur pensinyalan apoptosis (35.29%, -7.23)
5	Morfogenesis kelenjar (23.53%, -6.50)
6	Modifikasi peptidil-asam amino (29.41%, -6.38)
7	Diferensiasi sel epitel (29.41%, -4.69)
8	Gangguan proses simbion yang dimediasi inang (17.65%, -4.50)
	GO Cellular Component
1	Kompleks regulator transkripsi (47.06%, -9.65)
	GO Molecular Function
1	Pengikatan reseptor sitokin (47.06%, -11.97)
2	Pengikatan ligase protein mirip ubiquitin (23.53%, -4.61)
	KEGG Pathway
1	Jalur pensinyalan JAK-STAT (82.35%, -28.99)
2	Jalur kanker (94.12%, -26.93)
3	Campak (52.94%, -16.87)
4	Diferensiasi sel Th17 (47.06%, -15.28)
5	Ekspresi PD-L1 dan jalur checkpoint PD-1 pada kanker (35.29%, -11.15)
6	Jalur pensinyalan PI3K-Akt (41.18%, -9.23)
7	Penyakit radang usus (17.65%, -5.18)
8	miRNA pada kanker (23.53%, -4.58)

Lampiran D.5 Hasil Enrichment Komunitas 5

No	GO Biological Process
1	Respons seluler terhadap stimulus hormon (47.62%, -23.17)
2	Perkembangan kelenjar (35.71%, -17.01)
3	Perkembangan jantung (33.33%, -13.60)
4	Regulasi positif adhesi sel (28.57%, -11.60)
5	Proses ritmik (23.81%, -11.55)
	GO Cellular Component
1	Raft membran plasma (19.05%, -11.38)

Lampiran D.6 Hasil Enrichment Komunitas 5 (Lanjutan)

	GO Molecular Function
1	Aktivitas protein kinase serin/treonin (30.95%, -13.70)
	KEGG Pathway
1	Jalur kanker (54.76%, -29.03)
2	Jalur pensinyalan Apelin (33.33%, -22.30)
3	Infeksi sitomegalovirus manusia (35.71%, -21.15)
4	Jalur pensinyalan estrogen (28.57%, -18.27)
5	Stres geser fluida dan aterosklerosis (28.57%, -18.15)
6	Jalur pensinyalan oksitosin (28.57%, -17.68)
7	Jalur pensinyalan cAMP (30.95%, -17.47)
8	Adhesi fokal (26.19%, -14.47)
9	Kanker prostat (21.43%, -13.64)
10	Jalur pensinyalan prolaktin (19.05%, -13.17)
11	Melanogenesis (19.05%, -11.90)
12	Jalur pensinyalan Hedgehog (16.67%, -11.89)

Lampiran D.7 Hasil Enrichment Komunitas 6

No	**GO Biological Process**
1	Regulasi positif transisi epitel ke mesenkimal (50.00%, -19.98)
2	Perkembangan jantung (61.11%, -14.48)
3	Regulasi positif lokalisasi protein ke nukleus (33.33%, -10.86)
4	Regulasi proliferasi sel otot polos (33.33%, -9.42)
5	Regulasi aktivasi sel T (38.89%, -8.65)
6	Regulasi perkembangan sistem saraf (38.89%, -8.40)
7	Regulasi perkembangan jaringan otot lurik (22.22%, -8.10)
8	Gangguan transkripsi virus yang dimediasi inang (16.67%, -6.19)
9	Regulasi pertumbuhan (33.33%, -5.97)
10	Spesifikasi sumbu embrio (16.67%, -5.67)
11	Regulasi proses metabolisme glukosa (16.67%, -4.51)
	GO Cellular Component
1	-
	GO Molecular Function
1	Pengikatan beta-katenin (33.33%, -10.49)
2	Aktivitas represor transkripsi pengikat DNA (27.78%, -6.09)
	KEGG Pathway
1	Kanker kolorektal (66.67%, -26.57)
2	Adherens junction (61.11%, -23.40)
3	Jalur pensinyalan TGF-beta (44.44%, -15.00)
4	Infeksi virus leukemia sel T manusia tipe 1 (50.00%, -14.58)
5	Jalur pensinyalan yang mengatur pluripotensi sel punca (38.89%, -11.83)

Lampiran D.8 Hasil Enrichment Komunitas 7

No	**GO Biological Process**
1	Proses siklus sel mitosis (66.67%, -28.23)
2	Pembelahan sel (60.00%, -23.99)
3	Transisi fase siklus sel (40.00%, -19.80)
4	Regulasi positif pemisahan kromosom (16.67%, -10.02)
5	Siklus sel meiosis (23.33%, -8.05)
6	Regulasi proses berbasis mikrotubulus (20.00%, -6.53)

Lampiran D.9 Hasil Enrichment Komunitas 7 (Lanjutan)

7	Regulasi aktivitas transferase (20.00%, -6.34)
8	Regulasi keluar dari mitosis (10.00%, -6.15)
9	Kondensasi kromosom (10.00%, -5.01)
10	Regulasi negatif proliferasi populasi sel (16.67%, -3.01)
11	Proliferasi populasi sel (13.33%, -2.66)
	GO Cellular Component
1	Midbody (26.67%, -11.00)
2	Keranjang nuklir pori nukleus (13.33%, -9.25)
3	Kromosom nuklir terkondensasi (10.00%, -4.08)
	GO Molecular Function
1	Aktivitas histone kinase (13.33%, -7.87)
2	Pengikatan mikrotubulus (13.33%, -3.75)
	KEGG Pathway
1	Siklus sel (46.67%, -24.07)
2	Pematangan oosit yang dimediasi progesteron (33.33%, -17.08)

Lampiran D.10 Hasil Enrichment Komunitas 8

No	**GO Biological Process**
1	Regulasi positif lokomosi (36.84%, -13.68)
2	Regulasi organisasi proyeksi sel (36.84%, -13.46)
3	Morfogenesis tabung (36.84%, -13.18)
4	Regulasi ukuran komponen seluler (28.95%, -12.33)
5	Organisasi sitoskeleton aktin (31.58%, -11.62)
6	Morfogenesis epitel (28.95%, -10.91)
	GO Cellular Component
1	Korteks sel (36.84%, -17.82)
2	Sitoskeleton aktin (26.32%, -9.33)
	GO Molecular Function
1	Aktivitas protein kinase (28.95%, -10.08)
2	Pengikatan protein fosfatase (18.42%, -9.15)
	KEGG Pathway
1	Regulasi sitoskeleton aktin (47.37%, -27.89)
2	Jalur pada kanker (52.63%, -24.83)
3	Proteoglikan pada kanker (42.11%, -24.70)
4	Kanker kolorektal (26.32%, -17.00)
5	Jalur pensinyalan sfingolipid (26.32%, -15.36)
6	Aktivasi trombosit (23.68%, -13.38)
7	Fagositosis yang dimediasi reseptor Fc gamma (21.05%, -12.35)
8	Infeksi human immunodeficiency virus 1 (23.68%, -11.29)
9	Gaya geser fluida dan aterosklerosis (21.05%, -11.07)
10	Adherens junction (18.42%, -10.62)

Lampiran D.11 Hasil Enrichment Komunitas 9

No	**GO Biological Process**
1	Regulasi autofagi (39.39%, -16.23)
2	Regulasi respons imun bawaan (42.42%, -16.15)
3	Respons seluler terhadap stres kimia (30.30%, -12.38)
4	Regulasi negatif organisasi komponen seluler (39.39%, -12.11)
5	Kaskade pensinyalan intraseluler (39.39%, -12.05)

Lampiran D.12 Hasil Enrichment Komunitas 9 (Lanjutan)

No	GO Biological Process
6	Regulasi proses modifikasi protein (36.36%, -11.64)
7	Regulasi positif fungsi molekuler (30.30%, -10.29)
8	Fosforilasi peptidil-serin (15.15%, -9.20)
9	Regulasi makroautofagi (21.21%, -8.75)
10	Peksophagi (12.12%, -8.67)
11	Regulasi positif adhesi sel (27.27%, -8.64)
	GO Cellular Component
1	Kompleks fosfatidilinositol 3-kinase (15.15%, -10.89)
	GO Molecular Function
1	Aktivitas protein serin kinase (39.39%, -16.41)
2	Ikatan ligase protein ubiquitin (27.27%, -10.66)
3	Ikatan kinase (33.33%, -9.49)
	KEGG Pathway
1	Autofagi – hewan (45.45%, -25.09)
2	Infeksi herpesvirus terkait sarkoma Kaposi (42.42%, -21.97)
3	Shigellosis (42.42%, -20.39)
4	Penuaan seluler (27.27%, -13.12)
5	Aktivasi trombosit (18.18%, -8.33)

Lampiran D.13 Hasil Enrichment Komunitas 10

No	GO Biological Process
1	Regulasi proliferasi sel epitel (42.86%, -7.97)
2	Regulasi perakitan sambungan sel (35.71%, -6.88)
3	Respons terhadap estradiol (28.57%, -6.79)
4	Proses metabolisme oksida nitrat (21.43%, -6.64)
5	Respons terhadap tegangan geser fluida (21.43%, -5.53)
6	Jalur pensinyalan reseptor insulin (21.43%, -5.33)
7	Homeostasis seluler (35.71%, -4.91)
8	Proses ritmik (21.43%, -3.64)
	GO Cellular Component
1	Rakit membran (57.14%, -12.72)
	GO Molecular Function
1	-
	KEGG Pathway
1	Proteoglikan pada kanker (64.29%, -16.33)
2	Jalur pensinyalan estrogen (57.14%, -15.32)
3	Tegangan geser fluida dan aterosklerosis (57.14%, -15.24)
4	Adhesi fokal (57.14%, -13.98)
5	Jalur pensinyalan hormon tiroid (35.71%, -8.72)
6	Invasi bakteri pada sel epitel (28.57%, -7.40)
7	Jalur pensinyalan kalsium (35.71%, -7.12)

Lampiran D.14 Hasil Enrichment Komunitas 11

No	GO Biological Process
1	Regulasi negatif adhesi sel (29.27%, -13.84)
2	Adhesi sel–substrat (24.39%, -13.05)
3	Regulasi positif migrasi sel (31.71%, -12.17)
4	Regulasi pertumbuhan (26.83%, -9.37)

Lampiran D.15 Hasil Enrichment Komunitas 11 (Lanjutan)

No	GO Biological Process
5	Perkembangan endoderm (14.63%, -8.93)
6	Perkembangan sistem rangka (24.39%, -8.79)
7	Osifikasi (19.51%, -8.53)
8	Regulasi negatif migrasi sel (19.51%, -7.34)
9	Perkembangan jantung (21.95%, -7.17)
	GO Cellular Component
1	Matriks ekstraseluler (41.46%, -19.25)
2	Tepi depan sel (21.95%, -8.23)
3	Pita M (9.76%, -7.17)
	GO Molecular Function
1	Pengikatan kinase (31.71%, -10.79)
2	Pengikatan matriks ekstraseluler (14.63%, -9.65)
3	Pengikatan ATPase (14.63%, -8.67)
	KEGG Pathway
1	Proteoglikan pada kanker (41.46%, -26.08)
2	Pensinyalan integrin (34.15%, -21.87)
3	Jalur pada kanker (29.27%, -11.40)
4	Amebiasis (17.07%, -10.05)
5	MicroRNA pada kanker (19.51%, -7.99)

Lampiran D.16 Hasil Enrichment Komunitas 12

No	GO Biological Process
1	Organisasi perlekatan antar sel (57.14%, -10.38)
2	Regulasi adhesi antar sel (50.00%, -8.82)
3	Regulasi negatif adhesi sel (35.71%, -6.55)
4	Perkembangan jantung (35.71%, -5.35)
5	Perkembangan mesenkim (28.57%, -5.35)
	GO Cellular Component
1	Membran plasma basolateral (57.14%, -12.95)
2	Sambungan antar sel (57.14%, -10.61)
3	Bagian apikal sel (50.00%, -8.87)
4	Membran plasma apikolateral (21.43%, -6.92)
	GO Molecular Function
1	-
	KEGG Pathway
1	Kanker endometrium (50.00%, -15.60)
2	Sambungan adherens (42.86%, -11.67)
3	Virion – virus hepatitis (21.43%, -5.87)
4	Pembentukan amplop terkorneifikasi (21.43%, -3.90)

Lampiran D.17 Hasil Enrichment Komunitas 13

No	GO Biological Process
1	Organisasi sambungan sel (29.17%, -6.66)
2	Morfogenesis tabung (29.17%, -6.07)
3	Morfogenesis sel (29.17%, -6.00)
4	Organisasi struktur aktomiosin (16.67%, -5.63)
5	Disassembly komponen seluler (16.67%, -4.02)
	GO Cellular Component
1	Sitokeleton aktin (41.67%, -11.63)
2	Kompleks histon asetiltransferase NuA4 (12.50%, -5.67)
	GO Molecular Function
1	Pengikatan kalmodulin (20.83%, -6.28)

Lampiran D.18 Hasil Enrichment Komunitas 13 (Lanjutan)

	KEGG Pathway
1	Regulasi sitoskeleton aktin (37.50%, -13.03)
2	Pensinyalan IgSF CAM (37.50%, -12.00)
3	Remodeling kromatin yang bergantung pada ATP (29.17%, -11.45)
4	Tight junction (29.17%, -10.30)
5	Proteoglikan pada kanker (29.17%, -9.74)
6	Jalur pensinyalan Rap1 (29.17%, -9.63)
7	Panduan akson (16.67%, -4.89)

Lampiran D.19 Hasil Enrichment Komunitas 14

No	**GO Biological Process**
1	Remodeling kromatin (89.47%, -25.46)
2	Lokalisasi protein ke kromatin (15.79%, -5.39)
	GO Cellular Component
1	Tidak terdapat istilah GO Cellular Component yang signifikan
	GO Molecular Function
1	Komponen struktural kromatin (57.89%, -22.76)
2	Pengikatan DNA kromatin (26.32%, -7.61)
	KEGG Pathway
1	Karsinogenesis virus (21.05%, -5.13)

Lampiran D.20 Hasil Enrichment Komunitas 15

No	**GO Biological Process**
1	Perkembangan embrio kordata (21.43%, -2.46)
	GO Cellular Component
1	Kompleks ekspor transkripsi (57.14%, -23.12)
2	Kompleks spliceosom (35.71%, -7.43)
	GO Molecular Function
1	Aktivitas isomerase konformasi asam nukleat (28.57%, -6.15)
	KEGG Pathway
1	Karsinogenesis virus (21.05%, -5.13)

Lampiran D.21 Hasil Enrichment Komunitas 16

No	**GO Biological Process**
1	Regulasi positif kaskade MAPK (40.00%, -4.91)
	GO Cellular Component
1	Adhesi fokal (70.00%, -11.08)
2	Tepi depan sel (50.00%, -6.83)
3	Membran plasma apikal (40.00%, -5.15)
	GO Molecular Function
1	Aktivitas isomerase konformasi asam nukleat (28.57%, -6.15)
	KEGG Pathway
1	Infeksi sitomegalovirus manusia (70.00%, -12.84)
2	Sintesis, sekresi, dan aksi hormon paratiroid (50.00%, -9.74)
3	Infeksi *Yersinia* (30.00%, -4.96)