

IMPLEMENTASI METODE *MARKET BASKET ANALYSIS* PADA DATA TRANSAKSI PENJUALAN MENGGUNAKAN ALGORITMA *CT-PRO*

(Studi Kasus : Toko Bambang, Kebumen)

TUGAS AKHIR

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



Disusun Oleh:

Achmad Davit Adi Supriyatna
123140126

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”
YOGYAKARTA**

2021

HALAMAN PENGESAHAN PEMBIMBING

**IMPLEMENTASI METODE *MARKET BASKET ANALYSIS* PADA
DATA TRANSAKSI PENJUALAN MENGGUNAKAN ALGORITMA
*CT-PRO***

(Studi Kasus : Toko Bambang, Kebumen)

Disusun Oleh :

Achmad Davit Adi Supriatna
123140126

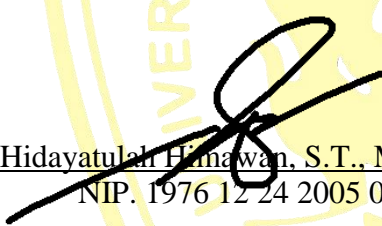
Telah diperiksa dan disetujui oleh pembimbing untuk diseminarkan


Pada tanggal :

Menyetujui,

Pembimbing I

Pembimbing II


Hidayatullah Hamawan, S.T., M.M., M.Eng
NIP. 1976 12 24 2005 011001


Wilis Kaswidjanti, S.Si., M.Kom
NIK. 2 7604 00 0226 1

Mengetahui,
Koordinator Tugas Akhir


Dr. Awang Hendrianto Pratomo, S.T., M.T.
NIP. 1977 07 25 2005 01 1001

ABSTRAK

Toko Bambang merupakan brntuk UMKM setiap harinya harus memenuhi kebutuhan konsumen dan dituntut untuk mengambil keputusan yang tepat dalam menentukan strategi pemasaran dan manajemen usaha yang baik. Dengan memanfaatkan data transaksi penjualan, manajemen dapat mengetahui kebiasaan atau perilaku pelanggan mengenai produk apa saja yang dibeli.

Cara mengetahui produk yang dibeli secara bersamaan dapat dilakukan dengan analisa keranjang belanja (*Market Basket Analisis*), yaitu metode dari data mining untuk menemukan suatu kombinasi kombinasi *item*. Proses pencarian menggunakan algoritma *CT-Pro* dimana algoritma ini lebih unggul dibandingkan algoritma lainnya untuk menghasilkan pola kombinasi *item* dan rules sebagai ilmu pengetahuan dan informasi yang penting.

Hasil penelitian dari data transaksi bulan Januari-Maret 2021 dengan nilai *minimum support* 2-12% dan *confidence* 30-50% didapatkan hasil aturan asosiasi dengan trend pembelian konsumen yang mempunyai keterikatan terbaik ialah produk segitiga terigu → telur 1kg dengan nilai *support* 11,27%, nilai *confidence* 53,01% dan *lift ratio* 1,79%.

Kata kunci: *Market Basket Analysis, Association Rule, Algoritma CT-Pro*

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
ABSTRAK	iii
DAFTAR GAMBAR.....	vi
DAFTAR TABEL	vii
DAFTAR MODUL PROGRAM.....	viii
BAB I.....	1
1.1 Latar Belakang Permasalahan.....	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian dan Perancangan Sistem.....	3
1.7 Sistematika Penulisan	4
BAB II	6
2.1 Pengertian Data Mining	6
2.1.1 <i>Data Mining</i>	6
2.1.2 Pengelompokan <i>Data Mining</i>	6
2.2 Market Basket Analysis	8
2.3 Association Rule	9
2.3.1 Tahapan <i>Association Rule</i>	9
2.4 Algoritma <i>CT-Pro</i>	10
2.4.1 Implementasi Algoritma <i>CT-Pro</i>	10
2.5 Entity Relationship Diagram (ERD).....	11
2.5.1 Simbol <i>ERD</i>	12
2.5.2 Derajat Relasi	12
2.5.3 Pemetaan Kardinalitas Relasi	12
2.5.4 <i>Key</i>	13
2.5.5 Peran (<i>Role</i>).....	13
2.5.6 <i>Weak Entity</i>	14
2.6 Relasi Antar Tabel (<i>RAT</i>).....	14
2.7 Data Flow Diagram (DFD)	15
2.7.1 Kegunaan <i>Data Flow Diagram</i>	15
2.7.2 Simbol-simbol <i>Data Flow Diagram</i>	15
2.8 Flowchart	17

2.9	Studi Pustaka.....	18
BAB III.....		21
3.1	Analisa Sistem	21
3.1.1	Perumusan Masalah.....	21
3.1.2	Pengumpulan Data Awal	21
3.1.3	Analisa Kebutuhan Sistem	21
3.2	Desain Sistem	23
3.3	Arsitektur Sistem	23
3.4	Perancangan Basis Data.....	24
3.4.1	<i>Entity Relation Diagram (ERD)</i>	24
3.4.2	Perancangan Tabel.....	24
3.4.3	Relasi Antar Tabel	26
3.5	Data Flow Diagram (DFD)	26
3.6	Flowchart	28
3.7	Rancangan Interface	33
3.8	Simulasi Hitungan.....	36
3.8.1	Pengumpulan Data.....	36
3.8.2	Pengolahan Data	36
3.8.3	<i>Preprocessing</i>	36
3.8.4	Pemanggilan Data.....	37
3.8.5	Proses Mencari <i>Frequent Item</i>	37
3.8.6	Pembangunan Global <i>CFP-Tree</i>	38
3.8.7	<i>Mining Local Tree</i>	39
3.8.8	<i>Generate Rule</i>	41
BAB IV.....		43
4.1	Implementasi Basis Data	43
4.2	Implementasi Proses	45
4.3	Implementasi Interface	52
4.4	Pengujian Sistem.....	59
4.5	Hasil dan Pembahasan	60
BAB V		62
5.1	Kesimpulan	62
5.2	Saran	62
DAFTAR PUSTAKA.....		63

DAFTAR GAMBAR

Gambar 3. 1 Arsitektur Sistem	23
Gambar 3. 2 Entity Relation Diagram	24
Gambar 3. 3 Relasi Antar Tabel	26
Gambar 3. 4 DFD Level 0	27
Gambar 3. 5 Data Flow Diagram Level 0	28
Gambar 3. 6 Flowchart Algoritma CT-Pro.....	29
Gambar 3. 7 Flowchart Proses Mencari Frequent Item.....	30
Gambar 3. 8 Flowchart Proses Pembentukan Global CFP-Tree	31
Gambar 3. 9 Flowchart Proses Mining.....	32
Gambar 3. 10 Halaman Login	33
Gambar 3. 11 Halaman Utama	33
Gambar 3. 12 Halaman Daftar Admin.....	34
Gambar 3. 13 Halaman Daftar Produk	34
Gambar 3. 14 Halaman Data Transaksi.....	35
Gambar 3. 15 Halaman CT-PRO.....	35
Gambar 3. 16 Global CFP-Tree.....	39
Gambar 3. 17 Local CFP-Tree Node 3.....	40
Gambar 3. 18 Proses Mendapatkan Frequent Itemset	40
Gambar 4. 1 Tabel Products	43
Gambar 4. 2 Tabel Transactions.....	44
Gambar 4. 3 Tabel Transactiondetails	44
Gambar 4. 4 Tabel Users	44
Gambar 4. 5 Tabel Rules	45
Gambar 4. 6 Tabel CFP_Tree.....	45
Gambar 4. 7 Halaman Login	53
Gambar 4. 8 Halaman Dashboard.....	53
Gambar 4. 9 Halaman Admin.....	54
Gambar 4. 10 Halaman Tambah Admin/User	54
Gambar 4. 11 Halaman Edit Admin/User	55
Gambar 4. 12 Halaman Produk	55
Gambar 4. 13 Halaman Tambah Produk	56
Gambar 4. 14 Halaman Edit Produk.....	56
Gambar 4. 15 Halaman Transaksi	57
Gambar 4. 16 Halaman Tambah Transaksi	57
Gambar 4. 17 Lanjutan Halaman Tambah Transaksi	57
Gambar 4. 18 Halaman Edit Transaksi.....	58
Gambar 4. 19 Lanjutan Halaman Edit Transaksi	58
Gambar 4. 20 Halaman Detail Transaksi.....	58
Gambar 4. 21 Halaman CT-Pro.....	59
Gambar 4. 22 Grafik Proses Aturan Asosiasi Bulan Januari-Maret 2021	61

DAFTAR TABEL

Tabel 2. 1 Simbol-simbol DFD	15
Tabel 2. 2 Simbol-simbol DFD (lanjutan).....	16
Tabel 2. 3 Bagan, nama dan fungsi flowchart	17
Tabel 2. 4 Studi Pustaka	18
Tabel 2. 5 Studi Pustaka (lanjutan).....	19
Tabel 2. 6 Studi Pustaka (lanjutan).....	20
Tabel 3. 1 Products	25
Tabel 3. 2 Transactions.....	25
Tabel 3. 3 Transactiondetails	25
Tabel 3. 4 User	25
Tabel 3. 5 Rules	25
Tabel 3. 6 Rules (lanjutan)	26
Tabel 3. 7 CFP_Tree.....	26
Tabel 3. 8 Sampel Data Transaksi	36
Tabel 3. 9 Daftar Item.....	37
Tabel 3. 10 Global Item.....	37
Tabel 3. 11 Koversi Data.....	38
Tabel 3. 12 Koversi Data Secara Ascending	38
Tabel 3. 13 Parent Node 3	39
Tabel 3. 14 Local Item Table	39
Tabel 3. 15 Frequent Itemset	40
Tabel 3. 16 Perhitungan Nilai Support dan Confidence	41
Tabel 3. 17 Aturan Yang Memenuhi Minimum Confidence.....	41
Tabel 3. 18 Nilai Benchmark Confidence Untuk Setiap Aturan Terkuat.....	41
Tabel 3. 19 Nilai Benchmark Confidence Untuk Setiap Aturan Terkuat (lanjutan)	42
Tabel 3. 20 Nilai Lift Ratio Untuk Setiap Aturan Terkuat.....	42
Tabel 4. 1 Hasil Black Box Testing.....	59
Tabel 4. 2 Hasil Black Box Testing.....	60
Tabel 4. 3 Hasil Dari Aturan Asosiasi Bulan Januari-Maret 2021	60

DAFTAR MODUL PROGRAM

Modul Program 4. 1 Source Code Pembentukan tabel Global Item	46
Modul Program 4. 2 Source Code Mencari Frequent Item	46
Modul Program 4. 3 Source Code Mencari Frequent Item (lanjutan).....	47
Modul Program 4. 4 Source Code Pembentukan Global CFP-Tree.....	48
Modul Program 4. 5 Source Code Pembentukan Global CFP-Tree (lanjutan)	49
Modul Program 4. 6 Source Code Pembentukan Global CFP-Tree (lanjutan)	50
Modul Program 4. 7 Source Code Mencari Hasil Asosiasi.....	50
Modul Program 4. 8 Source Code Mencari Hasil Asosiasi (lanjutan)	51
Modul Program 4. 9 Source Code Mencari Hasil Asosiasi (lanjutan)	52

BAB I

PENDAHULUAN

1.1 Latar Belakang Permasalahan

Perkembangan teknologi informai telah memberikan kontribusi pada cepatnya pertumbuhan jumlah data yang dikumpulkan dan disimpan dalam basis data berukuran besar. Dibutuhkan sebuah metode atau teknik yang dapat merubah data tersebut menjadi sebuah informasi berharga atau pengetahuan yang bermanfaat untuk mendukung pengambilan keputusan bisnis (Gunadi 2012).

Banyaknya persaingan dalam dunia usaha/bisnis khususnya dalam bidang perdagangan, menuntut para pengembang untuk menemukan suatu strategi yang dapat meningkatkan penjualan dan pemasaran produk yang dijual salah satunya adalah dengan pemanfaatan data transaksi penjualan. Dengan adanya kegiatan penjualan setiap hari, data semakin lama akan semakin bertambah banyak. Data tersebut tidak hanya berfungsi sebagai arsip bagi perusahaan, data tersebut dapat dimanfaatkan dan diolah menjadi informasi yang berguna untuk peningkatan penjualan dan promosi produk.

Seiring dengan kemajuan ini, maka berkembang pula kemampuan dalam mengumpulkan, menganalisa dan mengolah data pada *database*. Sehingga diperlukan suatu sistem untuk memilah dan memilih data penting dari *database* yang berhubungan dengan persediaan barang dengan pemanfaatan pada data transaksi penjualan.

Toko Bambang, salah satu bentuk UMKM (Usaha Mikro, Kecil dan Menengah) dibidang perdagangan yang bergerak menjual kebutuhan pokok masyarakat. Toko Bambang beralamat di Kebumen, Jawa Tengah, tepatnya di Kecamatan Adimulyo, Jalan Raya Caruban Nomor 1. Aktivitas jual beli pada Toko Bambang menghasilkan data transaksi yang tidak sedikit. Akan sangat disayangkan data tersebut yang seharusnya berguna bagi toko, hanya akan menjadi tumpukan sampah apabila data tersebut tidak dikelola dengan baik.

Salah satu metode yang dapat digunakan adalah analisis keranjang belanja atau *Market Basket Analysis (MBA)*. Analisis ini merupakan salah satu metode dalam penambangan data (*data mining*) yang bertujuan untuk menemukan produk-produk yang sering dibeli bersamaan dari data transaksi (Dewantara, Santosa, and Setyanto 2013). Metode analisa pola perilaku belanja menggunakan bantuan algoritma *CT-Pro*, yaitu algoritma yang merupakan pengembangan dari algoritma *FP-Growth* dengan melakukan

modifikasi pada *tree* yang digunakan (Rohmania Nurlaili Putri, Sihwi Sari Widya n.d.). Algoritma ini menyimpan informasi mengenai *frequent itemset* dalam bentuk struktur *FP-Tree*. Algoritma *CT-Pro* memodifikasi struktur *FP-Tree* ke dalam bentuk *Compressed FP-Tree (CFP-Tree)* dimana informasi dari sebuah *FP-Tree* diringkas dengan struktur yang lebih kecil atau ringan, sehingga algoritma ini paling cepat dibandingkan dengan algoritma *FP-Growth*. Dalam proses pencarian *frequent itemset* algoritma ini tidak melakukan *candidate generation*, sehingga dapat mengurangi *scan database* secara berulang dalam proses *mining*. Keunggulan lain dengan menggunakan algoritma ini adalah dari penggunaan memori yang lebih hemat (Ummah 2015). Dari keunggulan-keunggulan tersebut, maka dari itu digunakan algoritma *CT-Pro* dimana algoritma tersebut lebih efisien dibandingkan algoritma *FP-Growth*.

Hasil yang diharapkan dari penelitian ini yaitu sistem yang dapat memberikan informasi berguna dari pemanfaatan data transaksi penjualan yang dapat dipergunakan Toko Bambang dalam penetapan strategi usaha yang efisien untuk meningkatkan penjualan produknya. Selain itu, bertujuan juga untuk membantu mengantisipasi kekosongan persediaan produk, terutama untuk produk yang sering terjual secara bersamaan dan memiliki keterkaitan.

1.2 Perumusan Masalah

Berdasarkan latar belakang masalah tersebut, maka rumusan masalah yang dapat diambil yaitu bagaimana menganalisa pola pembelian *customer* pada Toko Bambang dengan menggunakan algoritma *ct-pro* untuk mempermudah dalam pembuatan strategi penjualan dan promosi maupun persediaan barang.

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

1. Pengambilan data pada penelitian ini diperoleh dari Toko Bambang, dengan menggunakan data transaksi penjualan.
2. Data yang diolah adalah data transaksi pada periode tertentu saja yaitu data transaksi selama tiga bulan dari tanggal 1 Januari – 31 Maret 2021.
3. Metode yang digunakan *Market Basket Analysis* dengan algoritma *CT-Pro* untuk menemukan pola kombinasi antar *item* yang berkaitan.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah diuraikan di atas, maka tujuan dari penelitian ini adalah menghasilkan aplikasi data *mining* untuk mengetahui kombinasi antar

item dengan frekuensi tertinggi sehingga diketahui pola pembelian *customer*.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini, yaitu:

1. Memudahkan pihak Toko Bambang untuk memperoleh informasi terkait dengan barang-barang yang sering terjual bersamaan, sehingga memudahkan dalam pembuatan strategi penjualan dan promosi.
2. Bisa mengetahui barang apa yang harus disediakan diwaktu yang akan datang, berdasarkan hasil pola hubungan antar *item*.
3. Membantu pihak Toko Bambang dalam hal pengelolaan persediaan barang.

1.6 Metodologi Penelitian dan Perancangan Sistem

Metode penelitian yang diterapkan pada penelitian ini menggunakan metode penelitian deskriptif. Metode deskriptif adalah suatu metode dalam meneliti status sekelompok manusia, suatu objek, suatu kondisi, suatu sistem pemikiran, ataupun suatu kelas peristiwa pada masa sekarang.

1. Metode pengumpulan data

Metode yang digunakan dalam mengumpulkan data dalam penelitian ini adalah sebagai berikut :

- a. Wawancara

Teknik pengumpulan data dengan cara berinteraksi atau berkomunikasi secara langsung kepada responden dengan cara mengajukan pertanyaan yang sesuai dengan topik yang diambil.

- b. Studi Literatur

Teknik pengumpulan data dengan cara mengadakan studi penelaahan terhadap buku-buku, literatur-literatur, catatan-catatan, dan laporan-laporan yang ada hubungannya dengan masalah yang akan dipecahkan.

2. Metode Pengembangan Perangkat Lunak

Metode yang digunakan dalam proses pembangunan perangkat lunak ini adalah dengan model *waterfall*. Model ini melakukan pendekatan pada perkembangan perangkat lunak secara sistematis dan sekuensial.

Tahap-tahap utama dari model ini memetakan kegiatan-kegiatan pembangunan dasar yaitu:

a. *Analisis*

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memesifikasi kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

b. *Design*

Design perangkat lunak adalah proses multi langkah yang fokus pada design pembuatan program perangkat lunak termasuk struktur data, perangkat lunak, representasi antarmuka dan prosedur pengodean.

c. *Code*

Merupakan proses penulisan program yang merealisasikan rancangan aplikasi yang dikembangkan dengan menggunakan bahasa pemrograman.

d. *Test*

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji.

e. *Maintenance*

Biasanya, ini merupakan fase siklus hidup yang paling lama. Sistem di-*install* dan dipakai. Pemeliharaan mencakup koreksi dari berbagai error yang tidak ditemukan pada tahap-tahap terdahulu, perbaikan atas implementasi unit sistem dan pengembangan pelayanan sistem, sementara persyaratan-persyaratan baru ditambahkan.

1.7 Sistematika Penulisan

Sistematika penulisan pada skripsi disusun dalam lima bab yang masing-masing bab akan dijelaskan sebagai berikut:

BAB I. PENDAHULUAN

Bab ini menguraikan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan perancangan sistem serta sistematika penulisan dalam pengimplementasian metode *market basket analysis* pada data transaksi penjualan menggunakan algoritma *ct-pro*.

BAB II. TINJAUAN PUSTAKA

Pada bab ini berisi tentang berbagai uraian teori dari beberapa sumber referensi termasuk dari beberapa referensi penelitian yang sudah dilakukan dan menjadi landasan dalam penelitian ini, teori-teori tersebut meliputi *data mining*, *association rule*, *market basket analysis*, algoritma *ct-pro* dan hal yang berkaitan dalam membangun sistem dengan implementasi *market basket analysis* pada data transaksi penjualan.

BAB III. METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM

Bab ini menggambarkan perancangan sistem, proses analisis data dan arsitektur sistem melalui diagram, serta rancangan antarmuka aplikasi. Penjelasan mengenai hasil dari analisis data dan perancangan sistem akan digunakan untuk memecahkan masalah-masalah yang ada.

BAB IV. HASIL, PENGUJIAN DAN PEMBAHASAN

Bab ini akan menjelaskan tentang perincian dari hasil aplikasi yang telah diteliti yang telah dilakukan analisis dan perancangan diikuti dengan antarmuka, *source code* yang digunakan serta cara kerja dari aplikasi yang akan digunakan.

BAB V. PENUTUP

Bab ini membahas tentang penutup dari penjelasan yang berisi tentang kesimpulan dan saran dari hasil aplikasi.

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Data Mining

Data *mining* adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam basis data. Data *mining* adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai basis data besar (Anggraeni 2014).

2.1.1 Data Mining

Data mining merupakan proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terakit dari berbagai *database* besar atau *data warehouse* (Purba *et al.* 2018). Berdasarkan defenisi-defenisi yang telah disampaikan, hal penting yang terkait dengan data mining adalah:

1. Data mining merupakan suatu proses otomatis terhadap data yang sudah ada.
2. Data yang akan diproses berupa data yang sangat besar.
3. Tujuan data mining adalah mendapatkan hubungan atau pola yang akan mungkin memberikan indikasi yang bermanfaat.

2.1.2 Pengelompokan Data Mining

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu :

1. Deskripsi

Terkadang peneliti dan analis secara sederhana ingin mencoba mencari data untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menentukan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelesan untuk suatu pola atau kecenderungan.

2. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih kearah numerik dari pada kearah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel target sebagai prediksi. Selanjutnya, pada peninjauan

berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi. Sebagai contoh akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan *level* sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya.

3. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasik akan ada dimasa mendatang. Contoh prediksi bisnis dan penelitian adalah:

- a. Prediksi harga beras dalam tiga bulan yang akan datang.
- b. Prediksi persentasi kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikkan.

Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

4. Klasifikasi

Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah. Contoh lain klasifikasi dalam bisnis dan penelitian adalah:

- a. Menentukan apakah suatu transaksi kartu *credit* merupakan transaksi yang curang atau tidak.
- b. Memperkirakan apakah suatu pengajuan hipotek oleh nasabah merupakan suatu *credit* yang baik atau buruk.
- c. Mendiagnosis penyakit seorang pasien untuk mendapatkan termasuk kategori penyakit apa.

5. Pengklusteran (*Clustering*)

Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Kluster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan *record-record* dalam kluster lain. Pengklusteran berbeda dengan klasifikasi yaitu tidak adanya variabel target dalam pengklusteran. Pengklusteran tidak mencoba untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel target. Akan tetapi, algoritma pengklusteran mencoba untuk melakukan pembagian terhadap

keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (*homogeny*), yang mana kemiripan dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal. Contoh pengklusteran dalam bisnis dan penelitian adalah:

- a. Mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari satu suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar.
- b. Untuk tujuan audit akuntansi, yaitu melakukan pemisahan terhadap perilaku financial dalam baik dan mencurigakan.
- c. Melakukan pengklusteran terhadap ekspresi dari gen, untuk mendapatkan kemiripan perilaku dari gen dalam jumlah besar.

6. Asosiasi

Tugas asosiasi dalam data mining adalah menemukan attribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja. Contoh asosiasi dalam bisnis dan penelitian adalah:

- a. Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respon positif terhadap penawaran upgrade layanan yang diberikan.
- b. Menentukan barang dalam supermarket yang dibeli secara bersamaan dan yang tidak pernah dibeli secara bersamaan.

2.2 Market Basket Analysis

Market basket analysis adalah salah satu teknik pemodelan dalam data *mining* berdasarkan teori yang mana jika anda membeli suatu grup *item*, anda akan memiliki kemungkinan membeli *itemset* yang lain (Fatihatul, Setiawan, and Rosadi 2016). *Market Basket* berdasarkan kumpulan *item* yang dibeli oleh konsumen dalam sebuah transaksi. Dalam hal ini, kuantitas dari sebuah *item* yang dibeli konsumen tidak mempengaruhi proses analisis ini. *Market basket analysis* hanya berdasarkan tipe-tipe *item* yang berbeda, tidak peduli seberapa banyak kuantitasnya. Dalam *Market basket analysis* akan dianalisis akumulasi kumpulan transaksi dari sejumlah besar konsumen dalam periode waktu yang telah berlangsung.

Proses ini menganalisis buying habits dari para konsumen dengan menemukan hubungan assosiasi antar *item-item* yang berbeda yang seringkali dibeli oleh konsumen. Hasil dari proses analisis ini nantinya akan sangat berguna bagi perusahaan retail

khususnya seperti toko swalayan dan supermarket untuk mengembangkan strategi pemasaran dan proses pengambilan keputusan dengan melihat *item-item* berbeda yang sering dibeli secara bersamaan oleh konsumen dalam satu waktu.

Beberapa kombinasi *item* yang sering dibeli konsumen memang terkadang sangat mudah untuk ditebak, contohnya seringkali konsumen membeli susu bayi dan popok secara bersamaan. Pola ini sangat biasa terjadi dan mudah ditebak karena susu bayi dan popok memiliki hubungan yang sangat dekat. Namun terkadang pola seperti deterjen dan telur jarang terfikirkan karena deterjen dan telur tidak mempunyai hubungan sama sekali. Melalui *Market basket analysis*, pola – pola yang terkadang tidak terfikirkan ini dapat ditemukan dengan mudah sehingga akan membantu pengambilan keputusan dan proses pengembangan strategi bagi perusahaan retail. Masalah - masalah seperti kehabisan stok akan diminimalisir dengan diketahuinya pola pembelian konsumen melalui *Market basket analysis* sehingga dapat meningkatkan penjualan perusahaan tersebut.

2.3 Association Rule

Aturan asosiasi atau *association rule* merupakan salah satu teknik didalam data mining untuk menentukan hubungan antar *item* dalam suatu dataset (sekumpulan data) yang telah ditentukan (Priyana and Kardianawati 2019). Contoh aturan asosiasi dari analisis pembelian di suatu pasar swalayan adalah dapat diketahuinya berapa besar kemungkinan seorang konsumen membeli roti bersama dengan susu.

Aturan asosiasi adalah bentuk jika “kejadian sebelumnya” kemudian “konsekuensinya”. (*If antecedent, then consequent*), yang diikuti dengan perhitungan aturan *support* dan *confidence*. Suatu pola menggambarkan isi dari *database* yang belum diketahui dan mungkin tidak bisa diungkapkan secara langsung.

Aturan asosiasi ditentukan oleh dua parameter, yaitu:

1. *Support* : Suatu ukuran atau nilai penunjang yang menunjukkan seberapa besar tingkat presentase kombinasi *item* dari keseluruhan transaksi.
2. *Confidence* : Suatu ukuran atau nilai kepastian yang menunjukkan kuatnya hubungan antar *item* dalam aturan asosiatif.

Kedua parameter diatas digunakan untuk menentukan kekuatan suatu pola dan menemukan pola yang memenuhi syarat *minimum* untuk *support* dan syarat *minimum* untuk *confidence*.

2.3.1 Tahapan Association Rule

Metodologi dasar analisis asosiasi (Atmaja, Simaremare, and Rosa 2019) sebagai

berikut :

1. Analisa pola frekuensi tinggi

Tahap ini mencari kombinasi antar *item* yang memenuhi syarat *minimum* dari nilai *support* dalam *database*.

$$Support(A) = \frac{Jumlah\ transaksi\ mengandung\ A}{Total\ transaksi} \times 100\% \quad (2.1)$$

Sedangkan nilai dari *support* dua *item* diperoleh dari persamaan berikut :

$$Support(A, B) = \frac{Jumlah\ transaksi\ mengandung\ A\ dan\ B}{Total\ transaksi} \times 100\% \quad (2.2)$$

2. Pembentukan Aturan Asosiasi

Setelah semua pola frekuensi tertinggi ditemukan, barulah dicari aturan asosiasi yang memenuhi syarat *minimum confidence* dengan menghitung *confidence* aturan asosiatif $A \rightarrow B$. Nilai aturan $A \rightarrow B$ diperoleh persamaan:

$$Confidence(A, B) = P(A|B) = \frac{Jumlah\ transaksi\ mengandung\ A\ dan\ B}{Jumlah\ transaksi\ mengandung\ A} \times 100\% \quad (2.3)$$

3. Lift Ratio

Setelah lolos dari aturan yang memenuhi nilai *minimum confidence*, selanjutnya dihitung nilai *lift ration*nya sekaligus, dengan persamaan :

$$Lift\ Ratio = \frac{Confidence}{Benchmark\ confidence} \times 100\% \quad (2.4)$$

Untuk nilai *benchmark confidencenya* sendiri berasal dari persamaan :

$$Benchmark\ Confidence = \frac{Jumlah\ kemunculan\ consequent}{Total\ transaksi} \times 100\% \quad (2.5)$$

2.4 Algoritma CT-Pro

Algoritma ini merupakan (Rohmania Nurlaili Putri, Sihwi Sari Widya n.d.) pengembangan dari algoritma *FP-Growth* dengan melakukan modifikasi pada *tree* yang digunakan. Algoritma ini menggunakan struktur Compressed *FP-Tree* (*CFP-Tree*) dimana informasi dari sebuah *FP-Tree* diringkas dengan struktur yang lebih kecil atau ringan, sehingga baik pembentukan *tree* maupun *frequent itemset mining* yang dilakukan menjadi lebih cepat.

2.4.1 Implementasi Algoritma CT-Pro

Berikut ini alur proses yang dilakukan:

1. Preprocessing

Data cleaning: proses menghilangkan noise dan data yang tidak konsisten atau tidak relevan.

- a. Transformasi data: data-data yang telah dibersihkan kemudian diubah menjadi satu format yang sama untuk mempermudah proses *mining* dan analisis pola data.
- b. Transformasi data dilakukan dengan mengubah parameter data-data pribadi menjadi sebuah kode *item*.

2. Menemukan *item-item* yang *frequent*

Membuat *item frequent table* yaitu dengan menghitung kemunculan tiap *item*. Kemudian dilakukan seleksi data dengan menentukan *minimum support* terlebih dahulu. Kemudian data-data yang memenuhi *minimum support* diurutkan berdasarkan jumlah kemunculan masing-masing *item* dengan memberikan *index* berdasarkan *count* dari terbesar ke terkecil, sehingga diperoleh *global item table*.

3. *Mapping* dan Membuat *CFP-Tree*

Setelah ditemukan *item-item* yang *frequent* kemudian dilakukan pembangunan *CFP-Tree*. Untuk membangun *CFP-Tree* sebelumnya, data hasil transformasi *dimapping* berdasarkan *index* pada *global item table*. Selanjutnya adalah pembangunan *CFP-Tree*. Langkah yang pertama yaitu pembacaan transaksi *ID* yang pertama akan membuat sebuah simpul sehingga terbentuk lintasan transaksi. *Support count* dari setiap simpul bernilai satu. Lalu pembacaan transaksi kedua, jika terjadi prefik transaksi sama dengan transaksi pertama, maka lintasan transaksi kedua dapat ditimpakan pada transaksi yang sama sambil menambah *support count* dari transaksi yang sama tersebut dan selanjutnya membuat lintasan baru sesuai dengan transaksi kedua, dan seterusnya untuk transaksi berikutnya. Proses ini dilanjutkan sampai *CFP-Tree* berhasil dibangun berdasarkan tabel data transaksi yang diberikan.

4. Tahap *Mining*

Algoritma *CT-Pro* bekerja dengan *bottom-up mining* sehingga *global item table* diurutkan mulai dari *item* berfrekuensi terkecil hingga terbesar. Masing-masing *global item table* dilakukan pencarian *node* yang berkaitan dengan *item* tersebut maka diperoleh *local item table*. Selanjutnya dari *local item table* kemudian dibangun *local CFP-Tree* berdasarkan jumlah *minimum support* yang telah ditentukan. Dari *local CFP-Tree*, kemudian dilakukan *mining* dengan rumus penghitungan *confidence*.

2.5 Entity Relationship Diagram (ERD)

Entity Relationship Diagram adalah suatu model penyajian data dengan menggunakan *Entity* dan *Relationship*. *ERD* menggambarkan model konseptual untuk

menggambarkan struktur logis dari basisdata berbasis grafis (Satrio Agung W, Ari Kusyanti 2011). Tujuan dari penyajian ini adalah agar *database* dapat dipahami dan dirancang dengan mudah.

2.5.1 Simbol ERD

ERD memiliki beberapa simbol, berikut adalah macam-macam simbolnya :

1. *Entity*

Entity adalah objek yang dapat dibedakan dalam dunia nyata. Sedangkan *entity set* adalah kumpulan dari entitty yang sejenis. *Entity set* dapat berupa objek secara fisik (Rumah, Kendaraan, Peralatan) atau objek secara konsep (Pekerjaan, Perusahaan). *Entity* disimbolkan dengan persegi panjang.

2. *Relationship*

Relationship adalah hubgan yang terjadi antara satu atau lebih *entity*. Sedangkan *Relationship set* adalah kumpulan *relationship* yang sejenis. *Relationship* disimbolkan dengan jajar genjang.

3. *Attribute*

Attribute adalah karakteristik dari tiap *entity* atau *relationship* yang menyediakan penjelasan detail mengenai *entity* atau *relationship* tersebut. Nilai dari *attribute* adalah data aktual atau informasi yang disimpan pada suatu attribut di dalam *entity* atau *relationship*, dimana tiap *attribute* memiliki domain (*value set*) tersendiri. Domain (*value set*) adalah batas-batas nilai yang diperbolehkan bagi suatu *attribute*. *Attribute* disimbolkan dengan elips.

2.5.2 Derajat Relasi

Derajat Relasi menunjukkan banyaknya himpunan entitas yang saling berelasi. Jenis derajat himpunan relasi adalah:

1. *Unary degree* (derajat satu), melibatkan sebuah entitas yang berelasi dengan dirinya sendiri.
2. *Binary degree*(derajat dua), himpunan relasi melibatkan dua himpunan entitas. Secara umum himpunan relasi dalam sistem basis data adalah *binary*.
3. *Ternary degree* (derajat tiga), himpunan relasi memungkinkan untuk melibatkan lebih dari dua himpunan entitas.

2.5.3 Pemetaan Kardinalitas Relasi

Pemetaan kardinalitas relasi menggambarkan banyaknya jumlah maksimum entitas dapat berelasi dengan entitas pada himpunan entitas yang lain. Untuk Himpunan relasi

biner pemetaan kardinalitasnya dapat merupakan salah satu dari tipe2 berikut :

1. Satu ke satu (*one to one*). Artinya, sebuah *entity* hanya dapat berelasi dengan satu buah *object* di *entity* yang lain. Kardinalitas ini disimbolkan dengan 1-1.
2. Satu ke banyak (*one to many*), disimbolkan dengan 1-m. Artinya, sebuah *entity* dapat berelasi dengan banyak *object* di *entity* yang lain. Kardinalitas ini disimbolkan dengan 1-m.
3. Banyak ke satu (*many to one*), disimbolkan dengan m-1. Ini adalah kebalikan dari *one to many*, maksudnya banyak *entity* akan berelasi dengan satu objek yang sama pada *entity* yang lain. Kardinalitas ini disimbolkan dengan 1-m.
4. Banyak ke banyak (*many to many*), disimbolkan dengan m-n. Artinya, akan ada banyak *entity* yang akan berelasi dengan banyak *object* di *entity* yang lain. Kardinalitas ini disimbolkan dengan m-n.

2.5.4 Key

Penggunaan *key* merupakan cara untuk membedakan suatu entitas didalam himpunan entitas dengan entitas lain. Secara konsep, masing-masing entitas (nilainya) berbeda, perbedaannya terlihat pada isi dari masing-masing atributnya. Oleh karena itu, dibutuhkan suatu atribut yang memiliki nilai yang menjadi pembeda dengan entitas lain. Key adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua row dalam relasi secara unik.

Ada 3 macam *key* yang dapat diterapkan pada suatu relasi :

1. *Super key*, merupakan satu atau lebih atribut (kumpulan atribut) yang dapat membedakan setiap baris data dalam sebuah relasi secara unik.
2. *Candidate key*, merupakan kumpulan atribut minimal yang dapat membedakan setiap baris data dalam sebuah relasi secara unik.
3. *Primary key*, merupakan salah satu dari *candidate key* yang terpilih. Pemilihan *primary key* dari sejumlah *candidate key* umumnya didasari oleh :
 - a. Key tersebut lebih sering (lebih natural) untuk dijadikan sebagai acuan.
 - b. Key tersebut lebih ringkas.
 - c. Jaminan keunikan *key* tersebut lebih baik

2.5.5 Peran (Role)

Relasi himpunan entitas tidak harus dalam bentuk yang berbeda, contoh label manager dan worker disebut *roles* (peran), yang menspesifikasi bagaimana entitas employe berinteraksi melalui relasi *works-for*. Peran dalam *ERD* diindikasikan dengan memberikan

label (nama) pada garis yang menghubungkan relasi dengan entitas. Label peran bersifat optional dan digunakan untuk mengklarifikasi semantik suatu relasi.

Dalam menggambarkan kardinalitas pada Diagram *ER*, digunakan :

- a. Garis panah (\rightarrow) yang menunjukkan satu.
- b. Garis biasa (---) yang menunjukkan banyak, antara relasi dengan entitas.

2.5.6 *Weak Entity*

Weak entity adalah suatu *entity* dimana keberadaan dari *entity* tersebut tergantung dari keberadaan *entity* lain. *Entity* yang merupakan induknya disebut *identifying owner* dan *relationship*-nya disebut *identifying relationship*. *Weak entity* selalu mempunyai total participation constraint dengan *identifying owner*.

2.6 Relasi Antar Tabel (*RAT*)

Relasi adalah hubungan antara tabel yang mempresentasikan hubungan antar objek di dunia nyata. Relasi merupakan hubungan yang terjadi pada suatu tabel dengan lainnya yang mempresentasikan hubungan antar objek di dunia nyata dan berfungsi untuk mengatur operasi suatu *database*. Hubungan yang dapat dibentuk dapat mencakup 3 macam hubungan, yaitu :

1. *One-To-One* (1-1)

Mempunyai pengertian setiap baris data pada tabel pertama dihubungkan hanya ke satu baris data pada tabel ke dua. Contohnya, relasi antara tabel mahasiswa dan tabel orang tua. Satu baris mahasiswa hanya berhubungan dengan satu baris orang tua begitu juga sebaliknya.

2. *One-To-Many* (1-N)

Mempunyai pengertian setiap baris data dari tabel pertama dapat dihubungkan ke satu baris atau lebih data pada tabel ke dua. Contohnya, relasi perwalian antara tabel dosen dan tabel mahasiswa. Satu baris dosen atau satu dosen bisa berhubungan dengan satu baris atau lebih mahasiswa.

3. *Many-To-Many* (N-M)

Mempunyai pengertian satu baris atau lebih data pada tabel pertama bisa dihubungkan ke satu atau lebih baris data pada tabel ke dua. Artinya ada banyak baris di tabel satu dan tabel dua yang saling berhubungan satu sama lain. Contohnya, relasi antar tabel mahasiswa dan tabel mata kuliah. Satu baris mahasiswa bisa berhubungan dengan banyak baris mata kuliah begitu juga sebaliknya.

2.7 Data Flow Diagram (DFD)

Data flow diagram (*DFD*) atau diagram alir data adalah alat yang meng-gambarkan aliran data melalui sistem dan atau pengolahan yang dilakukan oleh sistem tersebut (Pendekatan, Sistem, and Dynamics 2014). *DFD* juga dapat dikatakan sebagai penggambaran grafis atas sumber dan tujuan data, yang dapat memperlihatkan data berasal dari mana dan menuju ke mana. Dalam arti lain data kedalam sistem atau keluar dari sistem. *DFD* dapat melihatkan proses/kejadian (*event*) yang dilakukan oleh suatu sistem dari organisasi terhadap data yang masuk ke dalam sistem ataupun terhadap data yang keluar dari sistem, dan pada akhirnya dapat dilihat data tersebut disimpan.

Data Flow Diagram dimulai dari diagram konteks yaitu diagram yang berisi gambaran umum dari sistem. Jadi pada diagram konteks yang perlu kita tahu adalah sistem yang kita buat menghasilkan informasi apa untuk pihak luar dan sistem tersebut memerlukan data apa dari pihak luar. Atau dengan kata lain siapa yang memberikan data dan kepada siapa saja informasi perlu diberikan. Selanjutnya setelah konteks dibuat maka perlu buat diagram yang lebih detail yaitu diagram nol dan diagram detail.


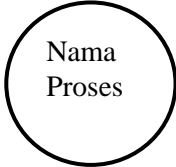
2.7.1 Kegunaan Data Flow Diagram

DFD digunakan untuk mendokumentasikan sistem yang ada serta digunakan merencanakan rancangan dan desain sistem yang baru.

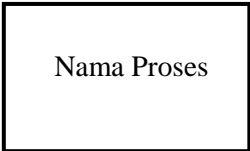
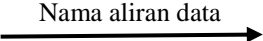
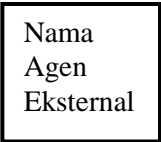

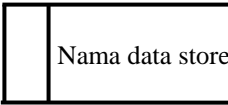
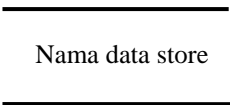
2.7.2 Simbol-simbol Data Flow Diagram

Dalam penggambaran diagram alir data dapat digunakan empat simbol yang lazim dipakai. Simbol yang akan dipakai adalah proses, aliran data, agen eksternal, dan data store atau penyimpanan. Keempat simbol tersebut dipadukan guna melihat bagaimana suatu sistem tersebut berjalan untuk mengolah data dan menghasilkan informasi. Untuk lebih jelasnya kita dapat lihat dalam tabel dibawah ini.

Tabel 2. 1 Simbol-simbol *DFD*

Nama Simbol	Bentuk Simbol	Bentuk Menurut	Keterangan
Proses		Gane & Sarson	Kerja yang dilakukan pada atau sebagai respon terhadap aliran data masuk atau kondisi. Proses dilakukan oleh orang atau mesin komputer. Contoh ; Pengiriman, Penagihan Kas, Penjualan, membuat order, proses pesanan anggota
		DeMarco/Yourdon	







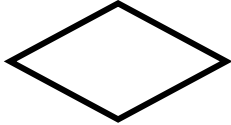
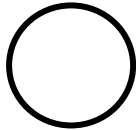
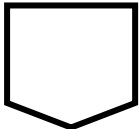
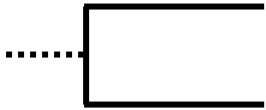
Tabel 2. 2 Simbol-simbol *DFD* (lanjutan)

Nama Simbol	Bentuk Simbol	Bentuk Menurut	Keterangan
Proses		SSADM/IDEFO	Kerja yang dilakukan pada atau sebagai respon terhadap aliran data masuk atau kondisi. Proses dilakukan oleh orang atau mesin komputer. Contoh ; Pengiriman, Penagihan Kas, Penjualan, membuat order, proses pesanan anggota
Aliran Data			Menunjukkan input data ke proses atau <i>output</i> data.
Agen Eksternal		Gane & Sarson	Orang, unit organisasi, sistem atau organisasi luar yang berinteraksi dengan sistem. Agen eksternal ini memberikan input dan menerima <i>Output</i> dari sistem Contoh : Yayasan, Mahasiswa, Dosen, Pelanggan.
		DeMarco/Yourdon	
<i>Data Store</i>		Gane & Sarson	Penyimpanan data yang ditunjuk untuk penggunaan selanjutnya. Data store atau simpanan data dapat berupa suatu file atau suatu sistem <i>database</i> dari suatu komputer, suatu arsip/dokumen, buku.
		DeMarco/Yourdon	

2.8 Flowchart

Flowchart adalah serangkaian bagan-bagan yang menggambarkan alir program. *Flowchart* atau diagram alir memiliki bagan-bagan yang melambangkan fungsi tertentu (Astuti 2015). Bagan, nama dan fungsinya seperti yang disajikan pada tabel berikut :

Tabel 2. 3 Bagan, nama dan fungsi *flowchart*

Bagan	Nama	Fungsi
	Terminator	Awal atau akhir program
	<i>Flow</i>	Arah aliran program
	<i>Preparation</i>	Inisialisasi/pemberian nilai awal
	<i>Proses</i>	Proses/pengolahan data
	<i>Input/output data</i>	<i>Input/output data</i>
	<i>Sub program</i>	<i>Sub program</i>
	<i>Decision</i>	Seleksi Atau Kondisi On
	<i>On connector page</i>	Penghubung bagian-bagian <i>flowchart</i> pada halaman yang sama off
	<i>Off connector page</i>	Penghubung bagian-bagian <i>flowchart</i> pada halaman yang berbeda
	<i>Comment</i>	Tempat Komentar Tentang Suatu Proses

2.9 Studi Pustaka

Judul Implementasi Metode *Market Basket Analysis* Pada Data Transaksi Penjualan Menggunakan Algoritma *CT-Pro* ini belum ada yang mengangkat di UPN “Veteran” Yogyakarta. Adapun penelitian lainnya dari Universitas lain yaitu Universitas Trunojoyo Madura oleh Khoirul Ummah (Ummah 2015) dengan judul “Sistem Rekomendasi Pembelajaran Pada E-Learning Menggunakan Algoritma Ct-Pro”, namun dalam penelitian ini untuk mencari informasi tentang perkembangan siswa dengan melihat jawaban-jawaban siswa yang salah dalam suatu ujian. Selain penelitian tersebut adapun penelitian lain dari Universitas lain yaitu Universitas Sebelas Maret oleh Rohmania Putri Nurlaili, Sari Widya Sihwi, Meiyanto Eko Sulisty (Rohmania Nurlaili Putri, Sihwi Sari Widya n.d.) dengan judul “Implementasi Algoritma Ct-Pro Untuk Menemukan Pola Pada Data Siswa Sma (Studi Kasus: Madrasah Aliyah Negeri (Man) Karanganon Klaten)”, yang membedakan penelitian ini adalah penelitian ini mencari keterkaitan antara data pribadi dengan nilai UN SMA dengan studi kasus di MAN Karangnom Klaten. Adapun penelitian sebelumnya ada beberapa variasi dan keunggulan masing-masing, berikut penelitian sebelumnya diantaranya :

Tabel 2. 4 Studi Pustaka

No.	Karya Ilmiah	Hasil	Teknologi
1	Application Of Ct-Pro Algorithm For Crime Analysis (Atmaja <i>et al.</i> 2019)	Algoritma CT-PRO dapat diterapkan untuk menganalisis tindak kejahatan yang terjadi di Polresta Yogyakarta dengan menemukan aturan asosiasi antar <i>item</i> . Dengan menggunakan data tindak kejahatan di Polresta Yogyakarta pada tahun 2016- 2018, batas tertinggi <i>minimum support</i> dan confidence yang dapat menghasilkan aturan asosiasi adalah 8,59% dan 70% dengan hasil satu aturan yaitu: “Jika tindak kejahatan yang dilakukan adalah CURAT maka tindak kejahatan terjadi di MUKIM.” Aturan tersebut memiliki tingkat kepercayaan sebesar 70.5%, <i>support count</i> sebesar 275 dan memiliki <i>lift ratio</i> senilai 1.66 yang berarti aturan termasuk kategori kuat. Aturan asosiasi yang dihasilkan sudah sesuai dengan kondisi di lapangan.	<ul style="list-style-type: none"> - PHP - MySQL
2	Implementasi Algoritma <i>Apriori</i> Untuk Mencari Relasi Pada Transaksi Pembelian Alat-Alat Kesehatan (Studi Kasus: Rs. Estomihi) (Purba <i>et al.</i> 2018)	Dalam mencari transaksi pembelian alat kesehatan dapat menghasilkan suatu data yang benar dengan mengetahui hasil dari data yang nantinya dapat dijadikan laporan rumah sakit. Dapat menyelesaikan hasil dari data transaksi pembelian alat kesehatan dengan perhitungan yang telah dibuat, sehingga dapat hasil dari data perhitungan yang benar.melakukan pengujian data dengan tangra dengan hasil yang dapat diketahui dengan <i>support</i> dan confidence.	

Tabel 2. 5 Studi Pustaka (lanjutan)

No.	Karya Ilmiah	Hasil	Teknologi
3	Perbandingan Algoritma <i>Apriori</i> dan Algoritma <i>FP-Growth</i> untuk Perekomendasi Pada Transaksi Peminjaman Buku di Perpustakaan Universitas Dian Nuswantoro (Anggraeni 2014)	Hasil dari sistem tersebut yaitu sebuah informasi rekomendasi buku dengan membandingkan kinerja algoritma <i>apriori</i> dan <i>fp-growth</i> .	
4	Perancangan Aplikasi <i>Data Mining</i> Dengan Algoritma <i>Apriori</i> Untuk Frekuensi Analisis Keranjang Belanja Pada Data Transaksi Penjualan (Dewantara <i>et al.</i> 2013)	Hasil penelitian ini adalah sebuah prototipe aplikasi MBA. Pengujian prototipe dilakukan dengan batasan <i>minimum</i> transaksi (<i>support</i>) sebesar 7 transaksi dan <i>minimum confidence</i> sebesar 5%. Dengan batasan tersebut, aplikasi MBA membentuk 11 aturan asosiasi. Salah satu aturan asosiasi yang terbentuk adalah jika membeli gula pasir lokal 1kg, indofood bmb.racik sayur sop 20gr 9117, maka membeli indofood bmb.racik sy.asem 20gr rsah.463 dengan nilai <i>support</i> = 0,52% dan nilai <i>confidence</i> = 90,91% yang merupakan aturan dengan nilai <i>confidence</i> tertinggi. Proses selanjutnya adalah mengkategorikan <i>item</i> yang digunakan sebagai acuan perbaikan layout, sehingga mendapatkan rekomendasi perbaikan layout yang menyatakan bahwa gula didekatkan dengan telur, bumbu masak jadi, dan minyak goreng; minyak goreng didekatkan dengan bumbu masak jadi; telur didekatkan dengan beras dan mie instant serta minuman didekatkan dengan roti. Dengan demikian, penataan barang dagangan bisa disesuaikan dengan aturan asosiasi agar sesuai dengan pola konsumsi konsumen.	
5	Asosiasi <i>Data Mining</i> Menggunakan Algoritma <i>FP-Growth</i> Untuk <i>Market Basket Analysis</i> (Fatihatul <i>et al.</i> 2016)	Penelitian ini menghasilkan association rules yang ditemukan dengan bantuan software WEKA 3.6.4. Penelitian ini menggunakan data sekunder berupa data transaksi penjualan sebuah supermarket dalam suatu periode waktu. Hasil dari penelitian ini berupa aturan (rules) yang merupakan kumpulan frequent <i>itemset</i> dengan nilai <i>confidence</i> yang tinggi. Hasil implementasi telah dilakukan dengan menghasilkan nilai strong <i>confidence</i> paling tinggi 92 %.	- WEKA 3.6.4
6	Penerapan Metode <i>Data Mining Market Basket Analysis</i> Terhadap Data Penjualan Produk Buku Dengan Menggunakan Algoritma <i>Apriori</i> Dan <i>Frequent Pattern Growth (FP-Growth)</i> : Studi Kasus Percetakan PT. Gramedia. (Gunadi 2012)	Dalam penelitian ini algoritma <i>Apriori</i> dan frequent pattern growth (FP-growth) digunakan untuk membantu menemukan sejumlah aturan asosiasi dari basis data transaksi penjualan produk buku di Percetakan PT. Gramedia, sehingga untuk selanjutnya dapat digunakan sebagai pertimbangan dalam membuat strategi pemasaran dan penjualan yang efektif.	

Tabel 2. 6 Studi Pustaka (lanjutan)

No.	Karya Ilmiah	Hasil	Teknologi
7	<i>Data Mining</i> Asosiasi Untuk Menentukan Cross-Selling Produk Menggunakan Algoritma <i>Frequent Patern-Growth</i> Pada Koperasi Karyawan PT . Phapros Semarang Sistem informasi yang terkomputerisasi organisasi dalam mengumpulkan berbagai data dalam suatu basis data (Priyana and Kardianawati 2019)	Menghasilkan informasi tentang paket pembelian produk dengan menangkap fenomena yang terjadi dalam transaksi penjualan.	- PHP

BAB III

METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM

3.1 Analisa Sistem

Pada tahap analisa sistem, didefinisikan seluruh bagian perangkat lunak, mengidentifikasi secara kebutuhan, dan garis besar aplikasi data mining yang akan dibuat. Dalam tahap ini dibagi menjadi tiga yaitu perumusan masalah yang timbul, pengumpulan data, dan juga analisis kebutuhan aplikasi pada penelitian ini.

3.1.1 Perumusan Masalah

Masalah yang timbul yaitu sering terjadi kehabisan *stock* barang atau bahkan barang yang tidak kunjung habis. Teknik data mining dapat membantu permasalahan *stock* barang dengan melihat pola penjualan barang pada toko bambang. Pada penelitian ini akan dibahas bagaimana memberikan solusi untuk meningkatkan kualitas *stock* barang ataupun promosi barang pada toko bambang.

3.1.2 Pengumpulan Data Awal

Pada penelitian ini, menggunakan studi pustaka. Studi pustaka dilakukan dalam penelitian untuk menguatkan literatur dan landasan teori penelitian. Metode ini dilakukan dengan cara mempelajari sumber-sumber tertulis, seperti buku-buku, jurnal-jurnal, skripsi, dan bahan-bahan tertulis lainnya yang berhubungan dengan pengetahuan teoritis yang terkait dengan data mining, dan algoritma *ct-pro*. Studi pustaka telah dijelaskan pada BAB II.

3.1.3 Analisa Kebutuhan Sistem

Penelitian data mining yang berjudul Implementasi *Metode Market Basket Analysis* Pada Data Transaksi Penjualan Menggunakan *Algoritma CT-Pro* ini merupakan suatu analisis yang berfungsi untuk membantu memprediksikan penjualan yang berlangsung pada toko bambang serta membantu dalam hal pengambilan keputusan dalam penjualan. Informasi ataupun data diperoleh dari data transaksi penjualan yang terjadi di buku bon. Prediksi dilakukan dengan cara mengolah data transaksi yang sudah ada yang nantinya diproses dengan metode *market basket analysis* algoritma *ct-pro*. Dalam hal ini variabel yang dianalisis adalah pola pembelian barang yang keluar bersamaan atau dibeli secara bersamaan.

Mengetahui pola pembelian barang yang dibeli bersamaan oleh konsumen terhadap variabel jenis barang merupakan informasi yang sangat penting. Dengan diketahuinya

pola-pola antar produk, maka dapat diperkirakan apakah nantinya produk A akan dibeli bersamaan dengan produk B atau tidak, dan begitu juga sebaliknya. Oleh karena itu, toko bambang memerlukan analisis yang dapat membantu pengambilan keputusan dalam penjualan.

Hasil dari analisis ini berupa pola penjualan antara dua buah barang atau lebih yang diperoleh dari data-data transaksi yang diolah menggunakan data *mining* dengan menganalisa keterkaitan konsumen terhadap jenis-jenis barang yang terdapat di toko bambang. Hasil diperoleh setelah memasukan data-data yang diperlukan seperti data transaksi yang telah diinputkan oleh *user*. Penelitian data *mining* ini digunakan oleh satu *user*.

Terdapat analisis kebutuhan fungsional dan non-fungsional pada analisis kebutuhan sistem terhadap penelitian ini. Kebutuhan fungsional berisi proses-proses apa saja yang nantinya dilakukan oleh sistem. Kebutuhan non-fungsional menitikberatkan pada properti perilaku yang dimiliki oleh sistem. Analisis kebutuhan sistem dilakukan dengan melakukan observasi. Analisis kebutuhan sistem merupakan langkah awal untuk melakukan investigasi terhadap pengembangan aplikasi, dengan melakukan pengidentifikasian masalah, serta rencana konsep gambaran secara umum dari aplikasi, dan analisis kebutuhan sistem dalam membangun aplikasi ini menggunakan *software engineering* dengan metode *waterfall*. Analisis sistem ini bertujuan untuk membuat sistem yang lebih efektif dan efisien yang terdiri atas kebutuhan fungsional dan non-fungsional.

a. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan gambaran dari informasi yang dapat terjadi pada aplikasi dan apa saja yang dilakukan oleh aplikasi yang bersangkutan serta berisi tentang informasi apa saja yang harus ada yang dihasilkan oleh aplikasi. Perancangan yang dilakukan yaitu perancangan program aplikasi diperuntukkan kepada *user*. Perancangan ini merupakan suatu perancangan baru yang telah didesain sebaik mungkin sehingga diharapkan mampu memberikan suatu solusi baru yang efektif sesuai dengan kebutuhan.

b. Analisis Kebutuhan Non-Fungsional

Adapun analisis kebutuhan non-fungsional dilakukan untuk mengetahui spesifikasi kebutuhan pada aplikasi. Spesifikasi kebutuhan mencakup analisis kebutuhan perangkat keras, kebutuhan perangkat lunak, dan analisis pengguna. Secara umum akan dijabarkan sebagai berikut :

1) Analisis kebutuhan perangkat keras

Perangkat keras yang digunakan untuk membangun aplikasi ini adalah perangkat komputer dengan spesifikasi sebagai berikut :

Processor: AMD A4-5000 APU 4CPUs (1.5GHz)

Memory : 2 GB DDR3

Hardisk : 500 GB

Display : 14 inch, max-resolution 1366 x 768

Video Type : AMD Radeon HD 8330

2) Analisis kebutuhan perangkat lunak

Analisis yang telah dilakukan dalam kebutuhan perangkat lunak untuk pengembangan dan menjalankan aplikasi yang disarankan agar dapat berjalan dengan baik adalah sebagai berikut :

a) Sublime Text 3

b) XAMPP

c) Browser

3) Analisis pengguna

Analisis kebutuhan pada pengguna dalam aplikasi ini adalah :

a) *User* dapat menginput/mengimport data yang akan diproses

b) *User* dapat menentukan nilai *support* dan *confidence*

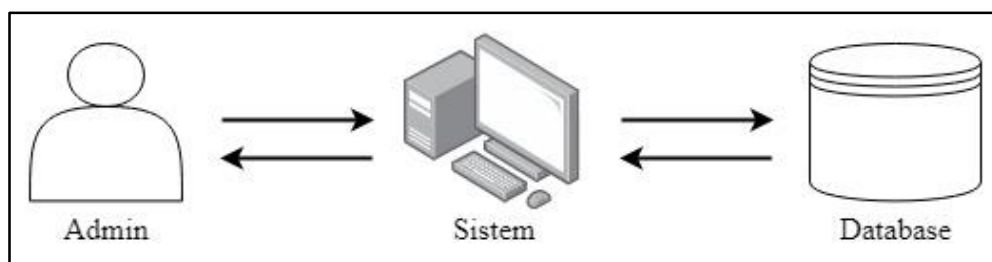
c) *User* dapat menghapus atau membatalkan proses

d) *User* dapat mengetahui hasil yang sudah diproses

3.2 Desain Sistem

Pada desain sistem berisi tentang desain aplikasi data mining secara umum yang dibagi menjadi empat yaitu arsitektur sistem, perancangan basis data, perancangan proses yang berupa UML, serta rancangan interface aplikasi.

3.3 Arsitektur Sistem



Gambar 3. 1 Arsitektur Sistem

Arsitektur pada sistem ini menjelaskan bahwa data diperoleh dari data transaksi

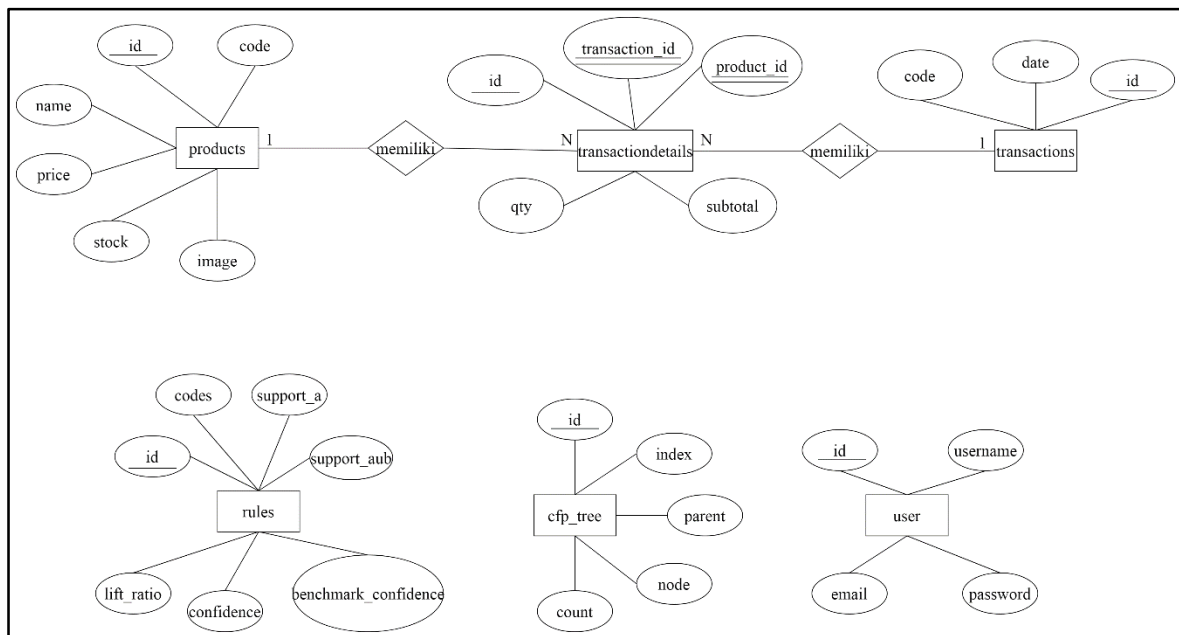
pada toko bambang yang digunakan sebagai studi kasus penelitian, data yang diperoleh dimasukkan kedalam aplikasi. Admin aplikasi akan memasukkan data beserta atribut aplikasi yang digunakan untuk menyeleksi data. Data akan diproses dan selanjutnya hasil dapat ditampilkan pada halaman yang sudah diatur.

3.4 Perancangan Basis Data

Perancangan basis data adalah proses untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem. Adapun perancangannya sebagai berikut :

3.4.1 Entity Relation Diagram (ERD)

Untuk *Entity Relation Diagram (ERD)* ditunjukkan pada Gambar 3.2. Pada Gambar 3.2 terdapat 5 entitas, yaitu *rules*, *cfp_tree*, *user*, *products*, *transactions* dan *transactiondetails*. Entitas *products* memiliki relasi dengan *transactiondetails*, dengan rasio kardinalitas *one to many*, dimana satu *products* memiliki banyak *transactiondetails*. Sedangkan entitas *transactions* pun sama, memiliki relasi dengan *transactiondetails* dengan rasio kardinalitas *one to many*, dimana satu *transactions* bias memiliki banyak *transactiondetails*.



Gambar 3. 2 Entity Relation Diagram

3.4.2 Perancangan Tabel

a. Tabel Products

Tabel *products* berfungsi sebagai media penyimpanan data barang atau data yang diinputkan oleh pengguna. Berikut struktur tabel *products* :

Tabel 3. 1 Products

Nama Field	Tipe Data	Constraint	Keterangan
id	Bigint(20)	Primary Key	Id Produk
code	Varchar(255)	-	Kode Produk
name	Varchar(255)	-	Nama Proudruk
price	Int(11)	-	Harga Produk
stock	Int(10)	0	Stok Produk
image	Varchar(255)	-	Gambar Produk

b. Tabel Transactions

Tabel transactions berfungsi sebagai media penyimpanan data transaksi yang diinputkan oleh pengguna. Berikut struktur tabel transactions :

Tabel 3. 2 Transactions

Nama Field	Tipe Data	Constraint	Keterangan
id	Bigint(20)	Primary Key	Id Transaksi
code	Varchar(255)	-	Kode Transaksi
date	Date	-	Tanggal Transaksi

c. Tabel Transactiondetails

Tabel transactiondetails berfungsi sebagai media penyimpanan detail transaksi. Berikut struktur tabel transactiondetails :

Tabel 3. 3 Transactiondetails

Nama Field	Tipe Data	Constraint	Keterangan
id	Bigint(20)	Primary key	Id detail transaksi
transaction_id	Bigint(20)	Foreign key	Id transaksi
product_id	Bigint(20)	Foreign key	Id produk
qty	Int(11)	1	Jumlah produk yang dibeli
Subtotal	Int(11)	-	Total harga transaksi

d. Tabel Users

Tabel users berfungsi sebagai media penyimpanan data pengguna. Berikut struktur table users :

Tabel 3. 4 User

Nama Field	Tipe Data	Constraint	Keterangan
id	Bigint(20)	Primary key	Id pengguna
username	Varchar(225)	Unique	Username pengguna
email	Varchar(225)	Unique	Email pengguna
password	Varchar(225)	-	Password akun pengguna

e. Tabel Rules

Tabel rules berfungsi sebagai media penyimpanan data hasil asosiasi dengan aturan yang telah ditentukan untuk proses data mining. Berikut struktur tabel rules :

Tabel 3. 5 Rules

Nama Field	Tipe Data	Constraint	Keterangan
id	Bigint(20)	Primary Key	Id rule
codes	Varchar(255)	-	Kode barang

Tabel 3. 6 Rules (lanjutan)

Nama Field	Tipe Data	Constraint	Keterangan
<i>support a</i>	Double	-	Nilai <i>support a</i>
<i>support aub</i>	Double	-	Nilai <i>support aub</i>
<i>benchmark_confidence</i>	Double	-	Nilai <i>benchmark confidence</i>
<i>confidence</i>	Double	-	Nilai <i>confidence</i>
<i>lift_ratio</i>	Double	-	Nilai <i>lift ratio</i>

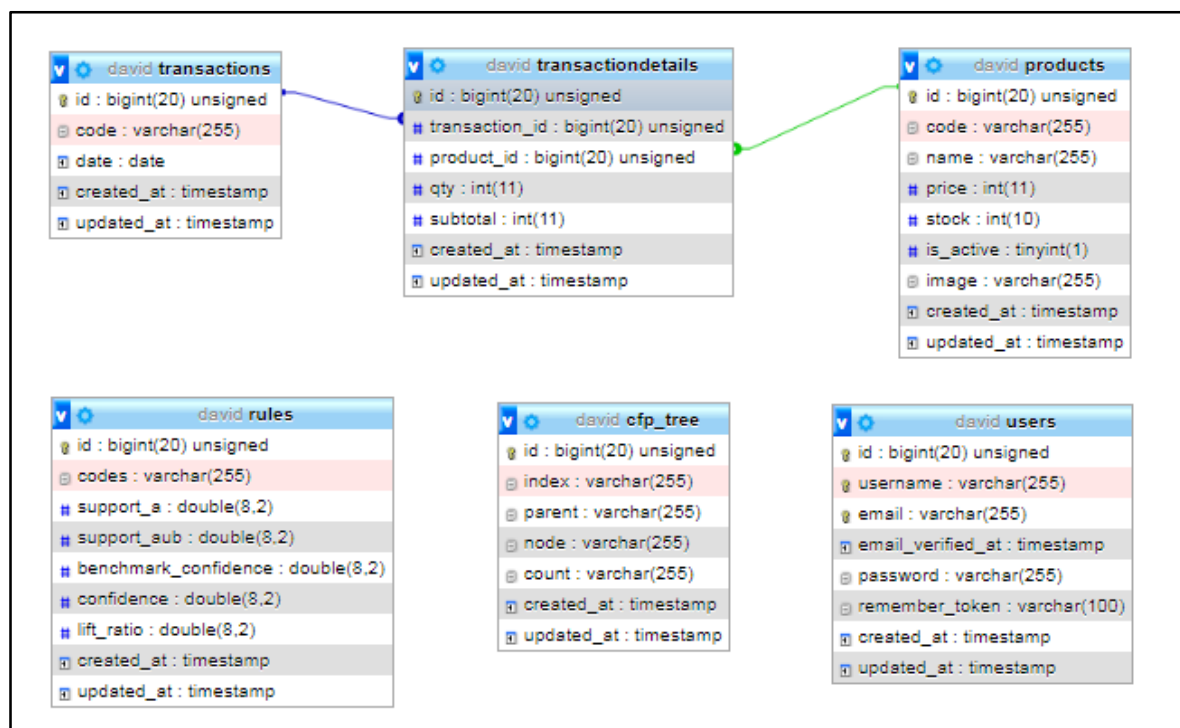
f. Tabel CFP_Tree

Tabel *cfp_tree* berfungsi sebagai media penyimpanan data *global tree*. Berikut struktur tabel *cfp_tree* :

Tabel 3. 7 CFP_Tree

Nama Field	Tipe Data	Constraint	Keterangan
id	Bigint(20)	Primary Key	Id <i>global tree</i>
index	Varchar(255)	-	Indeks tree
parent	Varchar(255)	-	Parent tree
node	Varchar(255)	-	Node tree
count	Varchar(255)	-	Jumlah tree

3.4.3 Relasi Antar Tabel



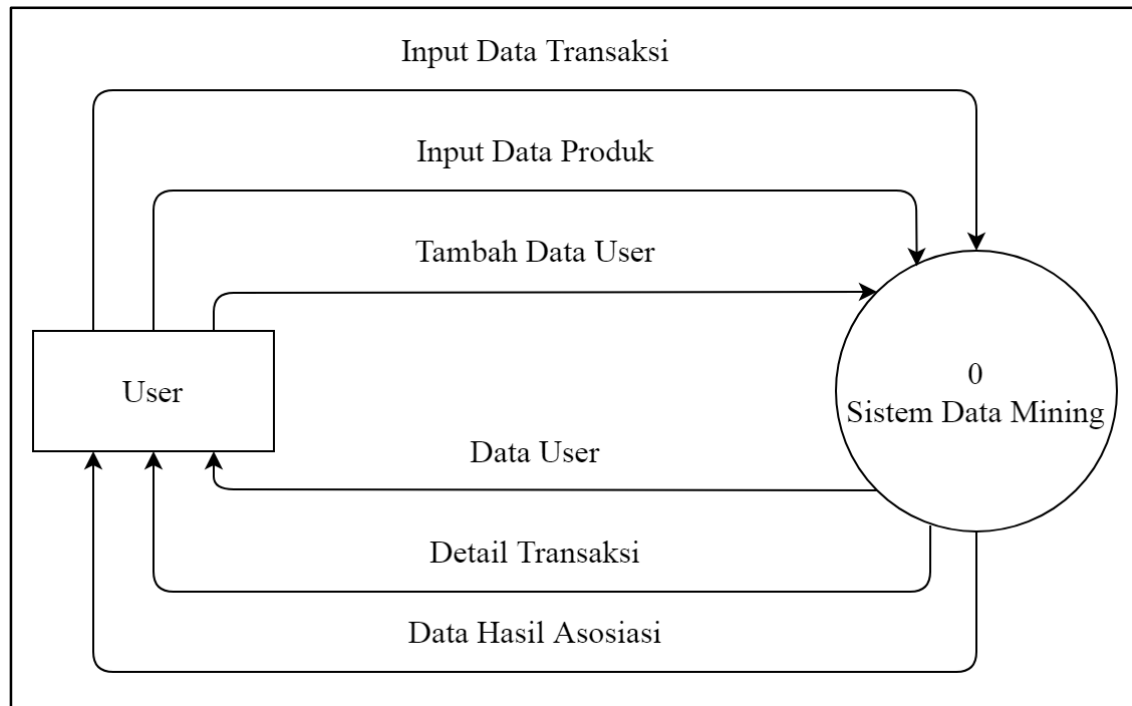
Gambar 3. 3 Relasi Antar Tabel

3.5 Data Flow Diagram (DFD)

Pada rancangan alur data ada 2 perancangan, yaitu *DFD level 0* dan *DFD level 1*. Berikut adalah perancangannya :

a. *DFD level 0*

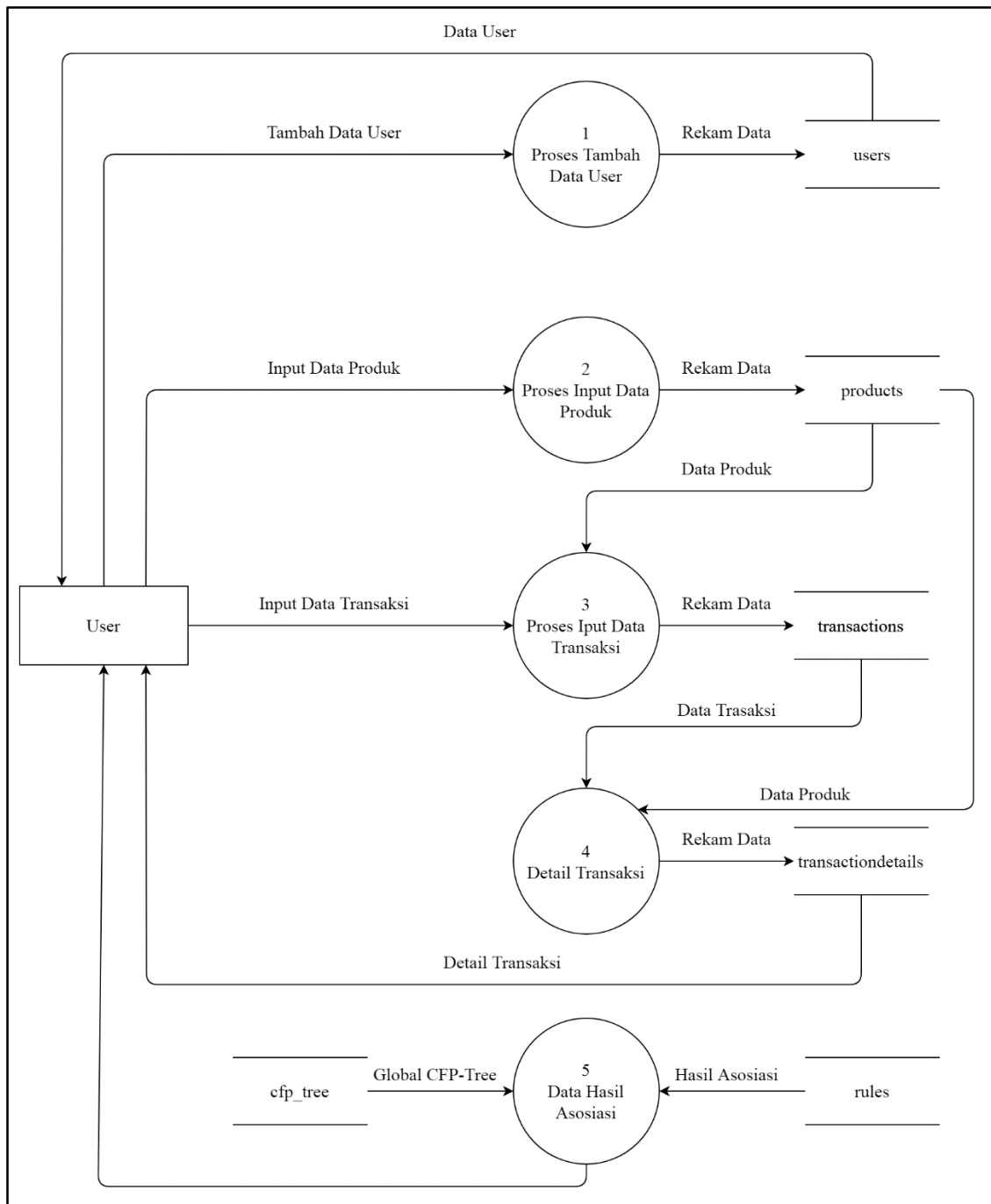
Pada *dfd level 0* terdapat satu buah entitas yaitu *user*. Pada *dfd level 0* terdapat alur mengirimkan data berupa data input transaksi, data input produk dan data tambah *user* serta memiliki alur masuk berupa data *user*, detail transaksi dan data hasil asosiasi. Untuk *dvd level 0* dapat dilihat secara lebih jelas pada Gambar 3.4.



Gambar 3. 4 DFD Level 0

b. *DFD level 1*

Pada *DFD level 1* terdapat 5 buah proses yaitu proses tambah data *user*, proses input data produk, proses input data transaksi, proses menampilkan detail transaksi dan proses menampilkan hasil asosiasi. Pada *dfd level 1*, data *user* disimpan di tabel *users*, data produk disimpan di tabel *products*, data transaksi disimpan di tabel *transactions*. Untuk memudahkan pengimplementasian, data transaksi dan data produk disimpan tabel pivot *transactiondetails*. Sehingga *user* bisa melihat detail transaksi. Untuk proses menampilkan hasil asosiasi, didapat dari tabel *rules*.

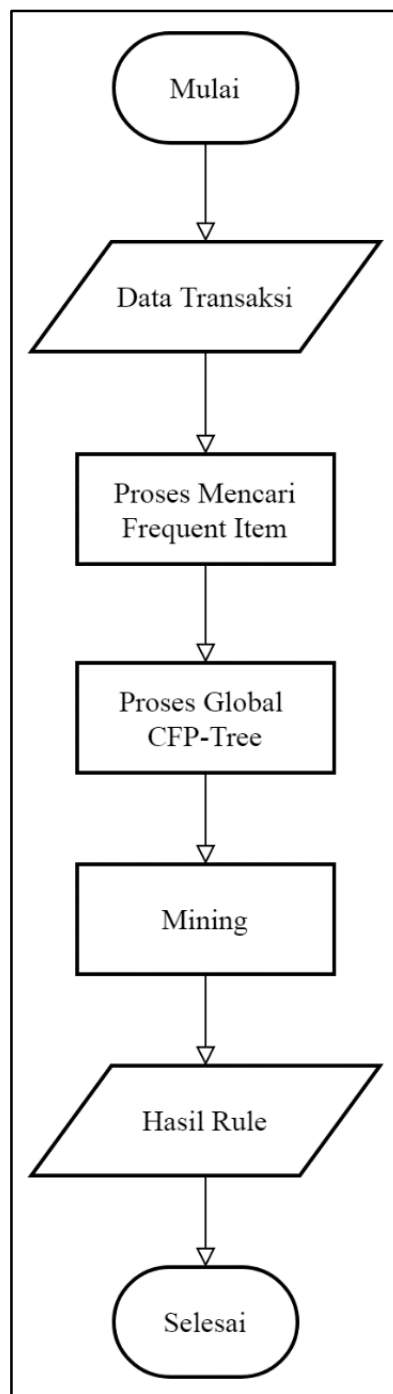


Gambar 3.5 Data Flow Diagram Level 0

3.6 Flowchart

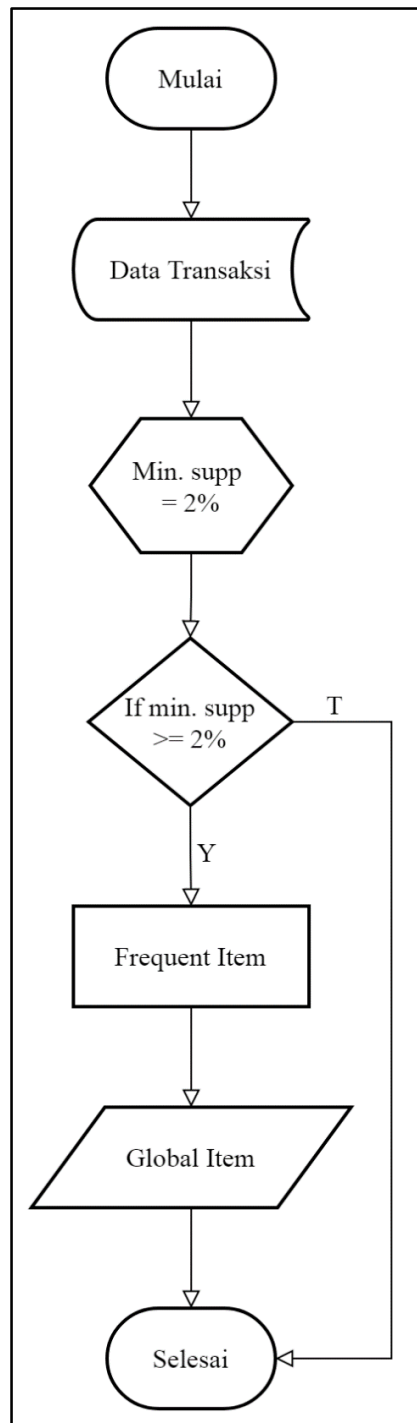
Perancangan proses logika berupa *flowchart* yang menggambarkan bagaimana langkah-langkah proses data mining dengan algoritma *ct-pro*. Pada perancangan *flowchart* akan dipecah berdasarkan langkah-langkah proses algoritma *ct-pro*. Ada empat perancangan *flowchart* pada laporan ini. Yang pertama, *flowchart* keseluruhan algoritma *ct-pro*, kedua *flowchart* mencari frequent item, yang kedua *flowchart* proses *global cfp-tree* dan yang terakhir adalah *flowchart* proses mining. Pada *flowchart* pertama menggambarkan proses

secara singkatnya. Untuk *flowchart* kedua sampai keempat untuk penjelasan setiap langkah secara detailnya. Berikut adalah penjelasannya :



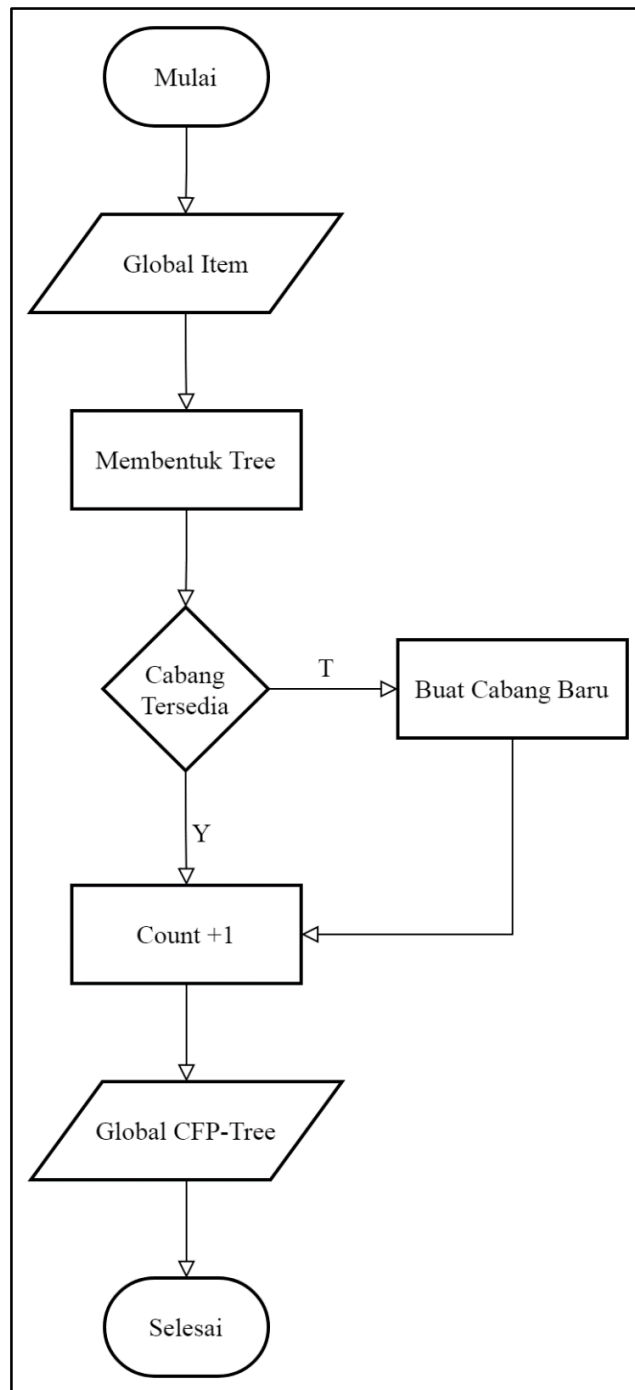
Gambar 3. 6 Flowchart Algoritma CT-Pro

Gambar 3.6 menunjukkan *flowchart* mengenai proses kerja algoritma *ct-pro* dimana proses ini dimulai dengan proses *input* data transaksi. Kemudian data tersebut dicari frequent *item*nya. Setelah itu dilakukan proses *global cfp-tree*. Setelah *global cfp-tree* dibentuk, dilanjutkan langkah mining, pada proses ini menghasilkan aturan asosiasi.



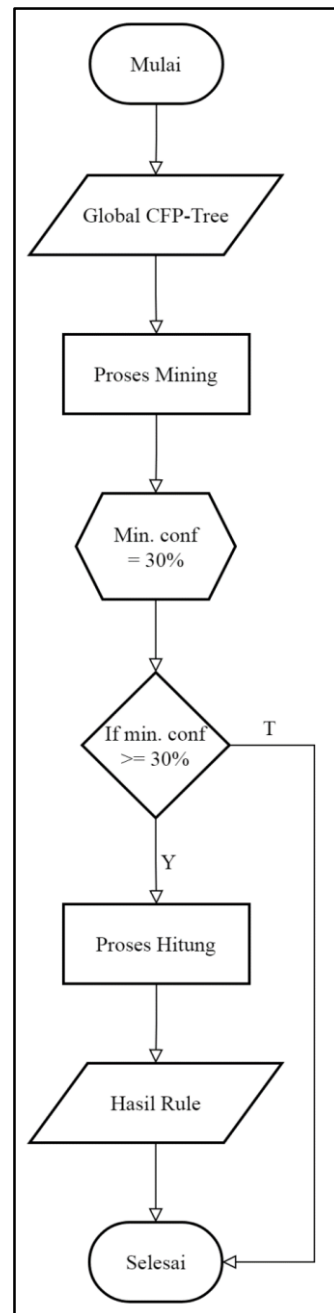
Gambar 3. 7 Flowchart Proses Mencari *Frequent Item*

Pada *flowchart* Gambar 3.7 proses diawali dengan mengambil data dari *database*. Kemudian setiap data dihitung nilai *support*-nya. Di sini nilai awal *minimum support*-nya adalah 2%. Disini data diseleksi berdasarkan nilai *minimum support*, sehingga menghasilkan *frequent item*. Dari *frequent item* yang telah terbentuk, dihitung frekuensi kemunculan setiap *item* sehingga menghasilkan *global item*.



Gambar 3. 8 Flowchart Proses Pembentukan Global CFP-Tree

Pada *flowchart* Gambar 3.8 data pada *global item* diproses untuk dibentuk tree. Apabila pada tree sudah memiliki cabang, maka count ditambah satu, apabila tidak tersedia cabang, maka dibuat cabang baru dan count ditambah satu. Dari proses tree akan menghasilkan *global cfp-tree*.



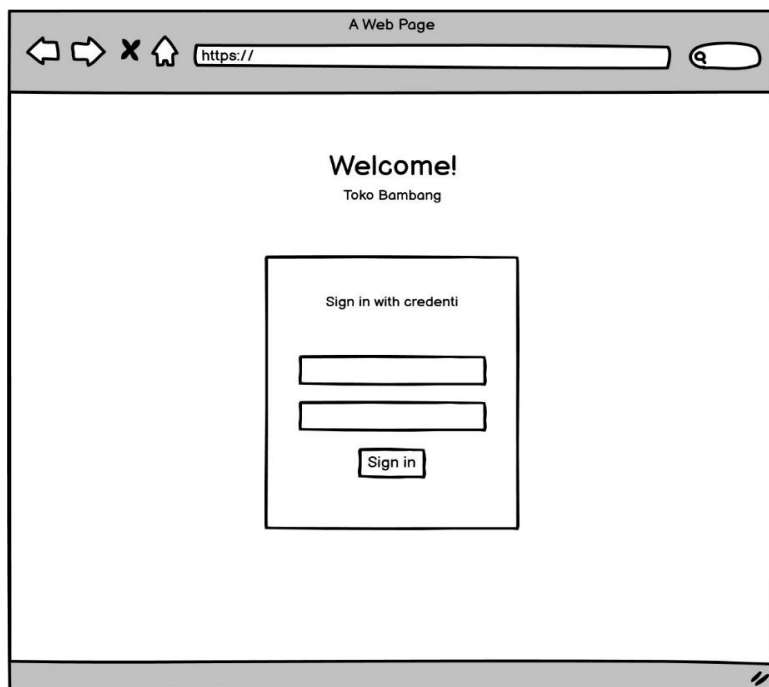
Gambar 3. 9 Flowchart Proses Mining

Flowchart pada Gambar 3.9 dilakukan proses mining pada *global cfp-tree*. Dimana pada proses mining algoritma *ct-pro* bekerja dengan melakukan bottom-up mining. Sehingga tabel *global item* diurutkan mulai dari frekuensi terkecil hingga ke terbesar. Kemudian dilakukan pencarian node yang berkaitan dengan *item* pada *global cfp-tree*. Semua node yang ditemukan untuk setiap *item* inilah yang disebut dengan local frequent *item* dan digunakan untuk membuat tabel local *item*. Kemudian dibuat local *cfp-tree* berdasarkan tabel local *item* yang terbentuk. Dari local *cfp-tree* dibentuk frequent pattern. Kemudian dari frequent pattern dihitung nilai confidencenya, apabila memenuhi nilai *minimum confidence*, maka akan ditampilkan sebagai hasil rule.

3.7 Rancangan Interface

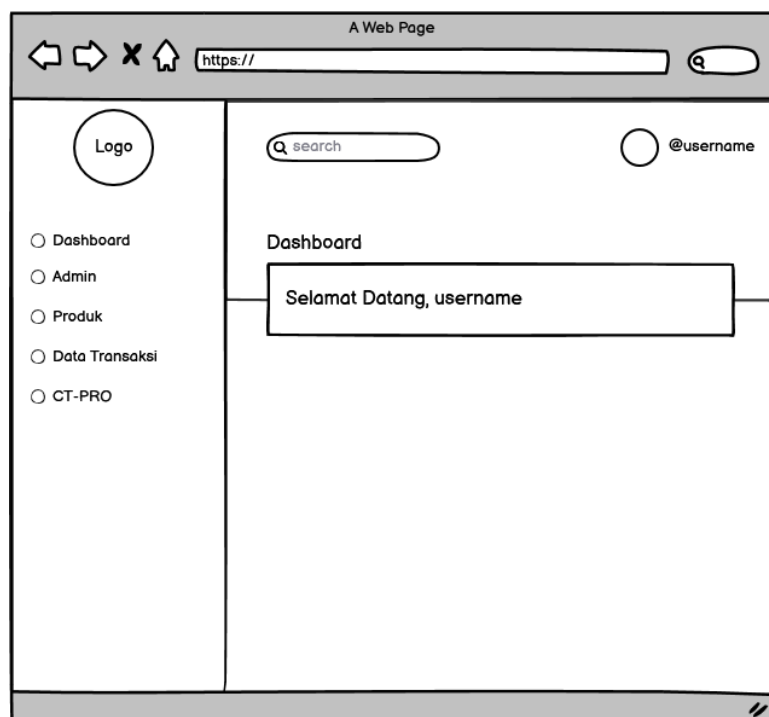
Pada implementasi perancangan antar muka atau *interface* akan ditampilkan halaman-halaman yang akan ada dalam aplikasi data *mining*. Perancangan antar muka dapat digambarkan sebagai berikut :

1. Rancangan Halaman *Login*



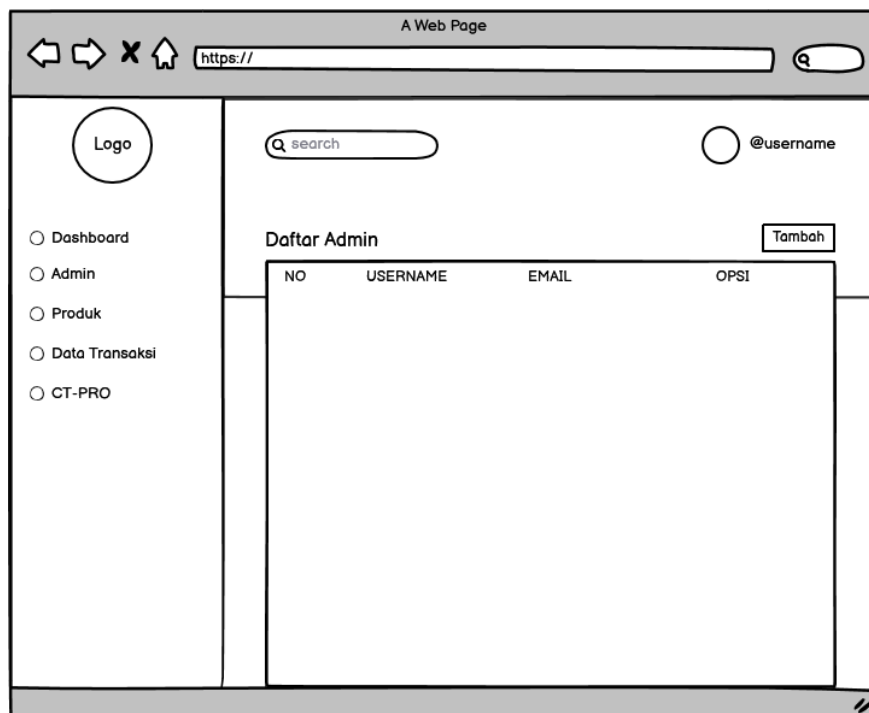
Gambar 3. 10 Halaman Login

2. Rancangan Halaman Utama



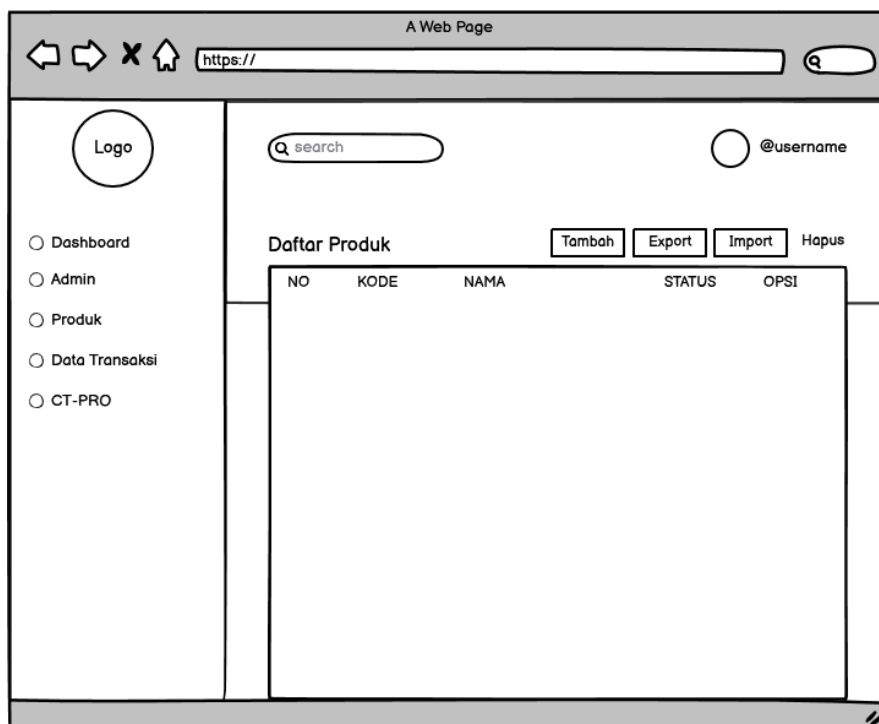
Gambar 3. 11 Halaman Utama

3. Rancangan Halaman Daftar Admin



Gambar 3. 12 Halaman Daftar Admin

4. Rancangan Halaman Daftar Produk



Gambar 3. 13 Halaman Daftar Produk

5. Rancangan Halaman Data Transaksi

The wireframe shows a web page titled "A Web Page" with a browser address bar containing "https://". On the left is a sidebar with a "Logo" and a menu with radio buttons for "Dashboard", "Admin", "Produk", "Data Transaksi", and "CT-PRO". The main content area has a search bar labeled "search" and a user profile icon labeled "@username". Below the search bar is a section titled "Data Transaksi" with buttons for "Tambah", "Export", "Import", and "Hapus". There are date pickers for "Dari Tanggal" and "Sampai Tanggal", and a "Filter" button. Below these is a table with the following headers: "NO", "NO TRANSAKSI", "DATE", "HARGA", and "STATUS".

Gambar 3. 14 Halaman Data Transaksi

6. Rancangan Halaman *CT-PRO*

The wireframe shows a web page titled "A Web Page" with a browser address bar containing "https://". On the left is a sidebar with a "Logo" and a menu with radio buttons for "Dashboard", "Admin", "Produk", "Data Transaksi", and "CT-PRO". The main content area has a search bar labeled "search" and a user profile icon labeled "@username". Below the search bar is a section titled "CT-PRO" with input fields for "Min Support (%)" and "Min Confidence (%)", and a button labeled "Cari Aturan Asosiasi". Below this is a section titled "Global Item" with a table with columns: "No.", "Kode Produk", and "Frekuensi". Below that is a section titled "Aturan Asosiasi" with a table with columns: "No", "Kode Produk", "Support AUB", "Support A", "Confidence", and "Lift Ratio". At the bottom is a section titled "Kesimpulan" with a table with columns: "No", "Aturan Asosiasi", "Confidence", "Lift Ratio", and "Saran".

Gambar 3. 15 Halaman CT-PRO

3.8 Simulasi Hitungan

Pada simulasi hitungan ini merupakan contoh simulasi perhitungan dari proses yang dilakukan pada sistem. Simulasi hitungan ini berfungsi untuk mempermudah gambaran penyelesaian masalah sekaligus menjelaskan bagaimana alur kerja dari proses asosiasi yang sudah dijelaskan pada perancangan *flowchart*.

3.8.1 Pengumpulan Data

Sumber data utama dalam penelitian ini berasal dari data transaksi toko ritel yang dijadikan studi kasus. Untuk simulasi hitungan ini data transaksi yang digunakan hanya beberapa sampel yang bertujuan untuk membuat penjelasan yang lebih mendalam. Data tersebut nantinya akan diolah sesuai aturan yang sudah dibuat untuk mendapatkan pengetahuan berupa pola asosiasi pembelian *item* oleh konsumen yang biasa digunakan sebagai penghitungan operasional inventori.

3.8.2 Pengolahan Data

Pada toko ritel yang dijadikan studi kasus terdapat beraneka ragam produk yang disediakan untuk bisa dibeli oleh konsumen. Dalam penelitian ini produk yang diambil berjumlah lebih dari 400 produk, tetapi untuk simulasi hitung manual produk yang dipakai 14 untuk membuat penjelasan yang lebih mendalam. Produk dan data transaksi yang akan diambil untuk sampel hitungan antara lain:

Tabel 3. 8 Sampel Data Transaksi

No	Data Transaksi
1	Pop Mie, Surya 12, Kapal Api
2	Roti, Aqua Botol, GG Filter
3	Tolak Angin, Fresh Care, GG Filter
4	Teh Gelas, Surya 12, Bodrex
5	Pop Mie, Surya 12, Kapal Api
6	Pop Mie, Korek Api, Kapal Api
7	Roti, Lanting, GG Filter
8	Roti, Korek Api, GG Filter
9	Teh Gelas, Surya 12, Battery
10	Pop Mie, Permen Davos, Kapal Api

3.8.3 Preprocessing

Data transaksi penjualan memiliki atribut yang banyak, tetapi tidak semua atribut pada data transaksi dibutuhkan dalam proses mining, oleh karena itu dilakukan proses *cleaning*. Tujuan dilakukannya proses ini adalah untuk memilih atribut yang menjadi fokus penelitian dan menghilangkan atribut yang tidak dibutuhkan. Dari beberapa atribut yang terdapat pada data transaksi yang digunakan dalam penelitian hanya tanggal transaksi,

kode transaksi, nama/id produk, quantiti dan subtotal harga. Jika sudah dipilih kemudian data transaksi akan diproses yang sudah ditentukan pada sistem.

3.8.4 Pemanggilan Data

Proses ini merupakan proses yang dilakukan untuk pemanggilan data dari beberapa tabel basis data *MySQL* yang sebelumnya sudah dimasukkan oleh admin dengan ketentuan memasukkan produk *item* dan jumlah masing-masing produknya, sedangkan untuk tanggal transaksi jika tidak pilih maka akan otomatis terpilih pada tanggal saat pemasukan data dilakukan dan untuk id transaksi secara otomatis.

3.8.5 Proses Mencari *Frequent Item*

Pada langkah ini dilakukan proses pencarian *item* yang kemunculannya tidak kurang dari *threshold* yang ditentukan yaitu *min support count* 2 (20%). Mula-mula semua *item* dari Tabel 3.8 diambil dengan menghilangkan *item* ganda. Selanjutnya dihitung jumlah kemunculan masing-masing *item* pada Tabel 3.8. Daftar dari *item-item* tersebut ditunjukkan pada Tabel 3.9.

Tabel 3. 9 Daftar Item

No	Item	Count
1	Pop Mie	4
2	Surya 12	4
3	Kapal Api	4
4	Roti	3
5	Aqua Botol	1
6	GG Filter	4
7	Tolak Angin	1
8	Fresh Care	1
9	Teh Gelas	2
10	Bodrex	1
11	Korek Api	2
12	Lanting	1
13	Battery	1
14	Permen Davos	1

Selanjutnya dilakukan penghapusan terhadap *item-item* pada Tabel 3.9 yang tidak memenuhi *threshold*. *Item-item* tersebut selanjutnya diurutkan berdasarkan *count* terbanyak dan diberi id *global* seperti yang terlihat pada Tabel 3.10.

Tabel 3. 10 Global Item

ID Global	Item	Count
1	Pop Mie	4
2	Surya 12	4
3	Kapal Api	4
4	GG Filter	4
5	Roti	3
6	Teh Gelas	2
7	Korek Api	2

Setelah diperoleh tabel *global item*, selanjutnya adalah melakukan konversi data pada tabel 3.8 sesuai dengan id *global* pada tabel 3.10 menjadi tabel 3.11.

Tabel 3. 11 Koversi Data

No	Item
1	1,2,3
2	5,4
3	4
4	6,2
5	1,2,3
6	1,7,3
7	5,4
8	5,7,4
9	6,2
10	1,3

Hasil konversi data pada tabel 3.11 diurutkan secara *ascending* sesuai dengan urutan id *global* yang ditunjukkan pada tabel 3.12.

Tabel 3. 12 Koversi Data Secara Ascending

No	Item
1	1,2,3
2	4,5
3	4
4	2,6
5	1,2,3
6	1,3,7
7	4,5
8	4,5,7
9	2,6
10	1,3

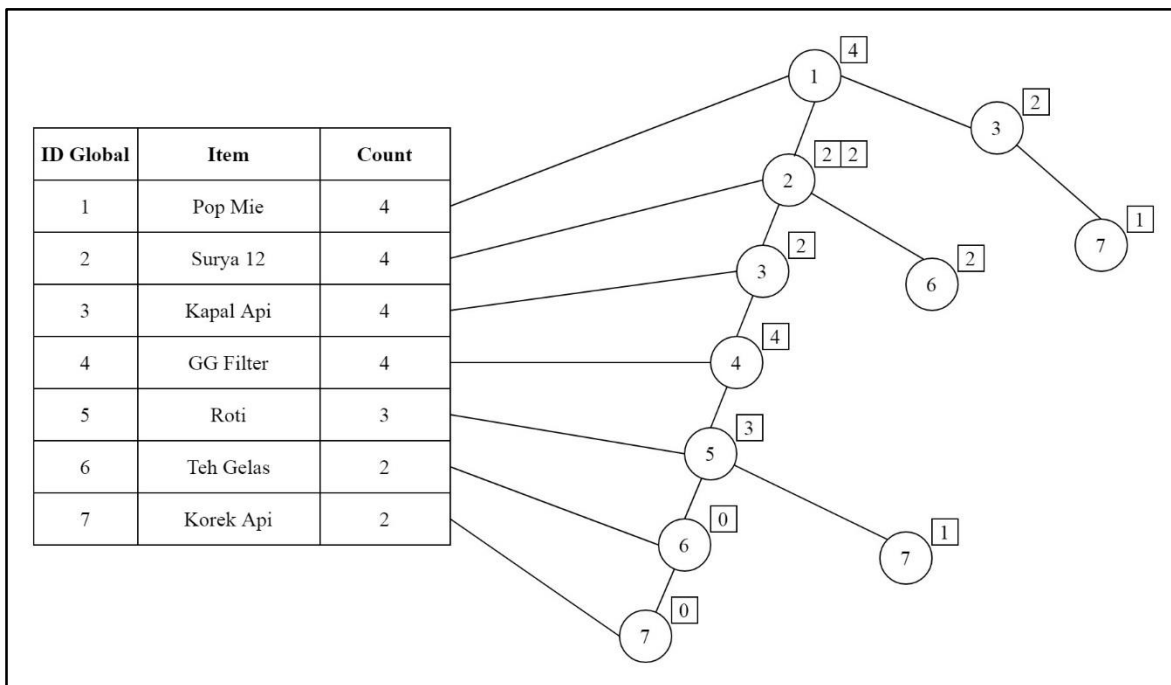
3.8.6 Pembangunan Global CFP-Tree

Proses pembangunan *global CFP-Tree* membutuhkan tabel *global item* dan tabel konversi *itemset*. Adapun algoritma pembentukan *global CFP-Tree* adalah sebagai berikut:

- Inisialisasi *tree*, buat *node* baru untuk setiap *item* yang ada pada tabel *global item* (Tabel 3.11) dan diatur sebagai anak kiri dari setiap *node* yang terbentuk.
- Mengakses setiap *item* yang ada pada *itemset* (Tabel 3.13).
- Jika *item* yang ada pada *itemset* adalah *currentNode* maka *count* yang ada pada *currentNode* ditambah satu.
- Jika *item* tersebut tidak sama dengan *currentNode* maka lakukan penelusuran untuk anak dari *currentNode*.
- Jika *item* tersebut ada pada anak dari *currentNode* maka *count* yang ada pada anak *currentNode* ditambah satu. Jika tidak buat cabang baru untuk *item* baru dan *count* pada cabang baru tersebut ditambah satu.

- f. Ulangi langkah b sampai e untuk setiap *itemset* hingga seluruh *itemset* sudah diproses.

Hasil dari proses pembangunan *global CFP-Tree* dapat dilihat pada Gambar 3.16:



Gambar 3. 16 Global CFP-Tree

3.8.7 Mining Local Tree

Konsep dari *algoritma CT-PRO* adalah *bottom-up mining* sehingga tabel *global item* diurutkan mulai dari *item* yang memiliki frekuensi terkecil hingga terbesar. Proses mining dilakukan satu persatu dimulai dari *global item* terkecil untuk menjadi *local tree*. Pada pembuatan *local item table*, dilakukan seleksi berdasarkan nilai *threshold* yang telah ditentukan.

Sebagai contoh akan dibangun *local CFP-Tree node 3* dengan cara *bottom-up*. Berdasarkan *global CFP-Tree* terdapat dua *item* yang menjadi *parent* dari *node 3* yaitu *item 1* dan *2*. Kedua *item* tersebut dihitung jumlah kemunculannya yang mengandung *node 3* seperti yang diperlihatkan pada tabel 3.13 dan tabel 3.14.

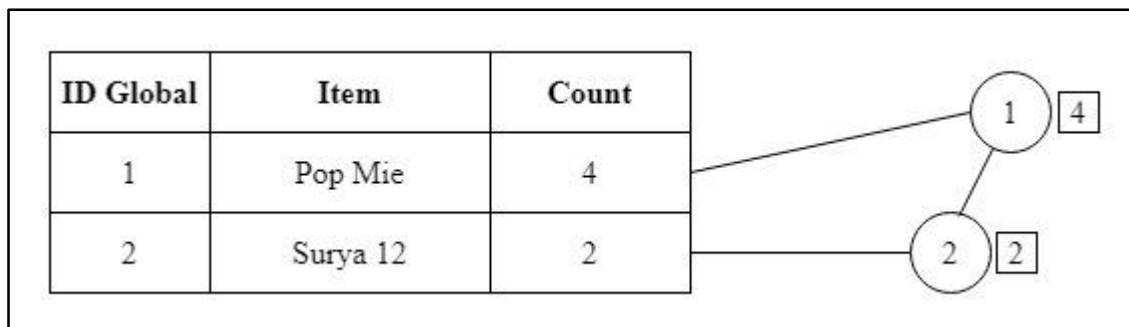
Tabel 3. 13 Parent Node 3

ID Global	Count
1	4
2	2

Tabel 3. 14 Local Item Table

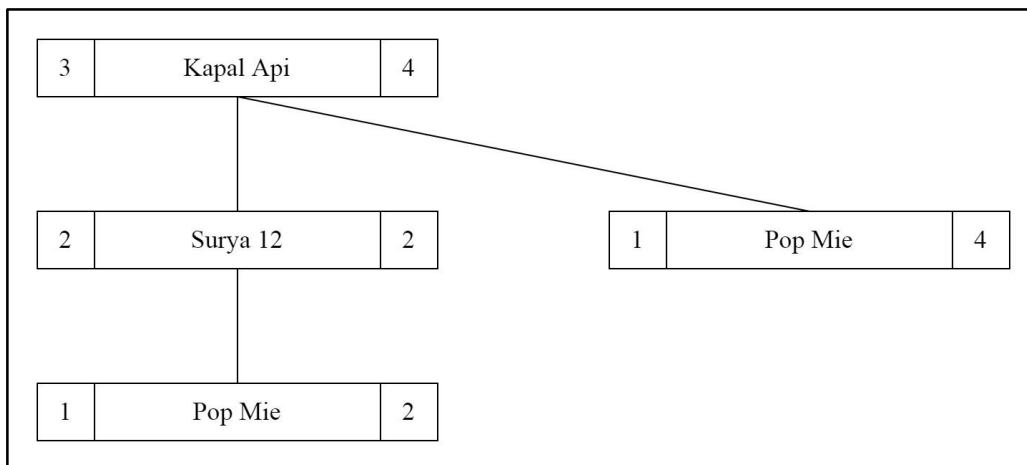
ID local	Item	Count
1	Pop Mie	4
2	Surya 12	2

Seluruh *item* yang terdapat pada tabel 3.13 diseleksi berdasarkan *threshold* yaitu 2 (20%). Lalu *item* yang telah diseleksi akan digunakan untuk membuat *local item table* pada tabel 3.14. Langkah selanjutnya adalah membangun *local CFP-Tree* dari *local item table* yang telah terbentuk. Gambar 3.17 menunjukkan jumlah *item* yang mengandung *item* 1 (Pop Mie) adalah 4. Sedangkan jumlah *item* yang mengandung *itemset* 1 dan 2 (Pop Mie dan Surya 12) adalah 2.



Gambar 3. 17 Local CFP-Tree Node 3

Langkah selanjutnya adalah membuat *frequent itemset* berdasarkan *node* 3 untuk dijadikan *root* yang diikuti dengan *node-node* lain dari *local CFP-Tree* sebagai *subtree*. Sebagai contoh, *node* 3 memiliki *subtree* *node* 2 dan *node* 1 sesuai dengan *node* yang ada pada *local CFP-Tree*. Proses membangun *frequent itemset* untuk *node* 3 ditunjukkan pada gambar 3.18.



Gambar 3. 18 Proses Mendapatkan *Frequent Itemset*

Selanjutnya dilakukan transformasi ke dalam bentuk tabel berdasarkan gambar 3.18 menjadi *frequent itemset* yang ditunjukkan pada tabel 3.15.

Tabel 3. 15 *Frequent Itemset*

No	Itemset	Count
1	Kapal Api, Surya 12, Pop Mie	2
2	Kapal Api, Surya 12	2
3	Kapal Api, Pop Mie	4

Langkah selanjutnya adalah membentuk aturan yang akan diperoleh dari proses

subset frequent item. Dari contoh di atas diperoleh *subset* setiap *item* yaitu {Kapal Api, Surya 12}, {Kapal Api, Pop Mie}, {Surya 12, Pop Mie}, {Kapal Api}, {Surya 12} dan {Pop Mie}.

3.8.8 Generate Rule

Setelah *frequent itemset* untuk *node* 3 ditemukan, langkah selanjutnya adalah membentuk aturan yang diperoleh dari proses *subset frequent itemset* berdasarkan tabel 3.15. Dari hasil *subset* tersebut dilakukan perhitungan nilai *support*, *confidence* dan *lift ratio* seperti yang diperlihatkan pada tabel 3.16.

Tabel 3. 16 Perhitungan Nilai Support dan Confidence

No	A	B	Support Count (A)	Support Count (AUB)	Confidence
1	Kapal Api, Surya 12	Pop Mie	2	2	$2/2=1$
2	Kapal Api, Pop Mie	Surya 12	4	2	$2/4=0,5$
3	Surya 12, Pop Mie	Kapal Api	2	2	$2/2=1$
4	Kapal Api	Surya 12, Pop Mie	4	2	$2/4=0,5$
5	Kapal Api	Surya 12	4	2	$2/4=0,5$
6	Kapal Api	Pop Mie	4	4	$4/4=1$
7	Surya 12	Kapal Api, Pop Mie	4	2	$2/4=0,5$
8	Surya 12	Kapal Api	4	2	$2/4=0,5$
9	Surya 12	Pop Mie	4	2	$2/4=0,5$
10	Pop Mie	Kapal Api, Surya 12	4	2	$2/4=0,5$
11	Pop Mie	Kapal Api	4	4	$4/4=1$
12	Pop Mie	Surya 12	4	2	$2/4=0,5$

Setelah ditemukan nilai *confidencenya* maka akan dilakukan penyeleksian aturan yang memenuhi nilai *minimum confidence* sebesar 1. Tabel 3.17 akan menunjukkan aturan-aturan yang memenuhi nilai *minimum confidence*.

Tabel 3. 17 Aturan Yang Memenuhi Minimum Confidence

No	A	B	Confidence
1	Kapal Api, Surya 12	Pop Mie	1
2	Surya 12, Pop Mie	Kapal Api	1
3	Kapal Api	Pop Mie	1
4	Pop Mie	Kapal Api	1

Setelah diperoleh aturan yang memenuhi *minimum confidence* yaitu aturan terkuat, selanjutnya perhitungan *lift ratio* setiap aturan tersebut. Sebelum menghitung nilai *lift ratio*, perlu terlebih dahulu untuk mencari nilai *benchmark confidence*. Untuk memperoleh nilai *benchmark confidence*, dilakukan perhitungan dengan membagi banyaknya kemunculan *consequent* yang dibagi dengan banyaknya data transaksi.

Tabel 3. 18 Nilai Benchmark Confidence Untuk Setiap Aturan Terkuat

No	A	B	Nc	N	Benchmark Confidence
1	Kapal Api, Surya 12	Pop Mie	4	10	0,4

Tabel 3. 19 Nilai Benchmark Confidence Untuk Setiap Aturan Terkuat (lanjutan)

2	Surya 12, Pop Mie	Kapal Api	4	10	0,4
3	Kapal Api	Pop Mie	4	10	0,4
4	Pop Mie	Kapal Api	4	10	0,4

Setelah memperoleh nilai *benchmark confidence*, selanjutnya dilakukan perhitungan nilai *lift ratio* yang dilakukan dengan membagi nilai dari *confidence* dengan *benchmark confidence*.

Tabel 3. 20 Nilai *Lift Ratio* Untuk Setiap Aturan Terkuat

No	A	B	Confidence	Benchmark Confidence	<i>Lift Ratio</i>
1	Kapal Api, Surya 12	Pop Mie	1	0,4	2,5
2	Surya 12, Pop Mie	Kapal Api	1	0,4	2,5
3	Kapal Api	Pop Mie	1	0,4	2,5
4	Pop Mie	Kapal Api	1	0,4	2,5

Dari hasil yang diperoleh pada tabel 3.20 nilai *lift ratio* yang dihasilkan untuk semua *rule* yang berhasil dibentuk memiliki nilai lebih besar dari 1. Hal ini menunjukkan bahwa semua *rule* bersifat kuat dan valid. Langkah-langkah di atas diimplementasikan pada setiap *node* yang ada di *global CFP-Tree* pada gambar 3.16 sehingga dapat disimpulkan bahwa:

1. Jika membeli Kapal Api dan Surya 12 maka juga membeli Pop Mie.
2. Jika membeli Surya 12 dan Pop Mie maka juga membeli Kapal Api.
3. Jika membeli Kapal Api maka juga membeli Pop Mie.
4. Jika membeli Pop Mie maka juga membeli Kapal Api.

BAB IV

HASIL, PENGUJIAN DAN PEMBAHASAN


Dalam bab ini menjelaskan tentang hasil, pengujian dan pembahasan pada aplikasi berdasarkan perancangan yang sudah dijelaskan pada bab sebelumnya. Penjelasan pada bab ini meliputi implementasi, penjelasan program, hasil proses serta pengujian yang dilakukan pada aplikasi penelitian.

4.1 Implementasi Basis Data

Pada tahap ini dibahas mengenai pembuatan basis data dan tabel yang digunakan dan dibutuhkan pada aplikasi. Aplikasi penelitian ini basis data yang digunakan adalah MySQL. MySQL memiliki ukuran basis data yang cukup kecil dan merupakan DBMS Open Source. Sedangkan untuk data yang disimpan di dalam basis data meliputi data products, data transactions, data transactiondetails, data *user*, data rules dan data cfp_tree. Adapun tabel-tabel yang terdapat dalam basis data sebagai berikut :

a. Tabel Products

Tabel products ini berfungsi sebagai media penyimpanan data produk yang dimasukkan oleh admin. Rincian tabel data products diuraikan pada Gambar 4.1.

	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar
<input type="checkbox"/>	1	id 	bigint(20)		UNSIGNED	Tidak	Tidak ada	
<input type="checkbox"/>	2	code	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada	
<input type="checkbox"/>	3	name	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada	
<input type="checkbox"/>	4	price	int(11)			Tidak	Tidak ada	
<input type="checkbox"/>	5	stock	int(10)			Tidak	0	
<input type="checkbox"/>	6	is_active	tinyint(1)			Tidak	1	
<input type="checkbox"/>	7	image	varchar(255)	utf8mb4_unicode_ci		Ya	NULL	
<input type="checkbox"/>	8	created_at	timestamp			Ya	NULL	
<input type="checkbox"/>	9	updated_at	timestamp			Ya	NULL	

Gambar 4. 1 Tabel Products

b. Tabel Transactions

Tabel transactions ini merupakan tabel yang berfungsi sebagai media penyimpanan data transaksi yang dimasukkan oleh admin.. Rincian tabel data transactions diuraikan pada Gambar 4.2

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
<input type="checkbox"/> 1	id 🔑	bigint(20)		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT
<input type="checkbox"/> 2	code	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
<input type="checkbox"/> 3	date	date			Ya	NULL		
<input type="checkbox"/> 4	created_at	timestamp			Ya	NULL		
<input type="checkbox"/> 5	updated_at	timestamp			Ya	NULL		

Gambar 4. 2 Tabel Transactions

c. Tabel Transactiondetails

Tabel transactiondetails ini merupakan tabel yang digunakan sebagai data utama untuk diolah dalam proses asosiasi. Rincian tabel data transactiondetails diuraikan pada Gambar 4.3.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
<input type="checkbox"/> 1	id 🔑	bigint(20)		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT
<input type="checkbox"/> 2	transaction_id 🔑	bigint(20)		UNSIGNED	Tidak	Tidak ada		
<input type="checkbox"/> 3	product_id 🔑	bigint(20)		UNSIGNED	Tidak	Tidak ada		
<input type="checkbox"/> 4	qty	int(11)			Tidak	1		
<input type="checkbox"/> 5	subtotal	int(11)			Tidak	Tidak ada		
<input type="checkbox"/> 6	created_at	timestamp			Ya	NULL		
<input type="checkbox"/> 7	updated_at	timestamp			Ya	NULL		

Gambar 4. 3 Tabel Transactiondetails

d. Tabel Users

Tabel *user* berfungsi sebagai media penyimpanan data *user/admin* aplikasi yang dimasukkan oleh admin. Rincian tabel data *users* diuraikan pada Gambar 4.4.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
<input type="checkbox"/> 1	id 🔑	bigint(20)		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT
<input type="checkbox"/> 2	username 🔑	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
<input type="checkbox"/> 3	email 🔑	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
<input type="checkbox"/> 4	email_verified_at	timestamp			Ya	NULL		
<input type="checkbox"/> 5	password	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
<input type="checkbox"/> 6	remember_token	varchar(100)	utf8mb4_unicode_ci		Ya	NULL		
<input type="checkbox"/> 7	created_at	timestamp			Ya	NULL		
<input type="checkbox"/> 8	updated_at	timestamp			Ya	NULL		

Gambar 4. 4 Tabel Users

e. Tabel Rules

Tabel rules berfungsi sebagai media penyimpanan data hasil asosiasi dengan aturan yang telah ditentukan untuk proses data mining. Rincian tabel data rules diuraikan pada Gambar 4.5.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
<input type="checkbox"/> 1	id 	bigint(20)		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT
<input type="checkbox"/> 2	codes	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
<input type="checkbox"/> 3	support_a	double(8,2)			Tidak	Tidak ada		
<input type="checkbox"/> 4	support_aub	double(8,2)			Tidak	Tidak ada		
<input type="checkbox"/> 5	benchmark_confidence	double(8,2)			Tidak	Tidak ada		
<input type="checkbox"/> 6	confidence	double(8,2)			Tidak	Tidak ada		
<input type="checkbox"/> 7	lift_ratio	double(8,2)			Tidak	Tidak ada		
<input type="checkbox"/> 8	created_at	timestamp			Ya	NULL		
<input type="checkbox"/> 9	updated_at	timestamp			Ya	NULL		

Gambar 4. 5 Tabel Rules

f. Tabel CFP_Tree

Tabel cfp_tree berfungsi sebagai media penyimpanan data *global tree*. Rincian tabel data cfp_tree diuraikan pada Gambar 4.6.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
<input type="checkbox"/> 1	id 	bigint(20)		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT
<input type="checkbox"/> 2	index	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
<input type="checkbox"/> 3	parent	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
<input type="checkbox"/> 4	node	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
<input type="checkbox"/> 5	count	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada		
<input type="checkbox"/> 6	created_at	timestamp			Ya	NULL		
<input type="checkbox"/> 7	updated_at	timestamp			Ya	NULL		

Gambar 4. 6 Tabel CFP_Tree

4.2 Implementasi Proses

Pada implementasi proses akan menjelaskan source code program utama pada aplikasi yang dibuat. Proses utama yang dijelaskan merupakan proses aturan asosiasi dengan menggunakan algoritma *CT-Pro*. Pada proses aturan asosiasi dibutuhkan adanya data transaksi yang akan diproses dalam rentang waktu yang dikehendaki dalam pemrosesannya. Dalam data transaksi tersebut data yang dipakai untuk proses aturan asosiasi ialah id products pada masing-masing data transaksi. Source code tersebut disajikan pada Modul Program 4.1.

```

$products = Produk::get();
$countes = array();
foreach ($products as $product) {
    $countes[] = array(
        "code" => $product->code,
        "frequency" => TransaksiDetail::where('product_id',
$product->id)->count()
    );
}
usort($countes, function($a, $b) {
    return $a['frequency'] > $b['frequency'] ? -1 : 1;
});

$global_item = array();
foreach ($countes as $c) {
    $global_item[$c['code']] = $c['frequency'];
}
usort($countes, function($a, $b) {
    return $a['frequency'] < $b['frequency'] ? -1 : 1;
});
foreach ($countes as $c) {
    $global_item_reverse[$c['code']] = $c['frequency'];
}
usort($countes, function($a, $b) {
    return $a['frequency'] > $b['frequency'] ? -1 : 1;
});

$global_item = array();
foreach ($countes as $c) {
    $global_item[$c['code']] = $c['frequency'];
}
usort($countes, function($a, $b) {
    return $a['frequency'] < $b['frequency'] ? -1 : 1;
});
foreach ($countes as $c) {
    $global_item_reverse[$c['code']] = $c['frequency'];
}
}

```

Modul Program 4. 1 Source Code Pembentukan tabel Global Item

Pada proses di atas data produk sebelumnya diambil dahulu dari tabel products untuk ditampilkan semuanya, selanjutnya untuk nilai frekuensi diambil dari tabel transactiondetails. Setelah diketahui jumlah frekuensi masing-masing *item*, selanjutnya diurutkan berdasarkan *item* yang memiliki nilai frekuensi paling besar dahulu, kemudian yang terkecil. Produk yang sudah diurutkan sekaligus akan menjadi tabel *global item*. Setelah membangun *global item*, selanjutnya mencari *item-item* yang frequent dengan minimal dua *item*. Source code tersebut disajikan pada Modul Program 4.2.

```

$transaksi = Transaksi::with('details', 'details.product')->orderBy('code', 'asc')->get();

```

Modul Program 4. 2 Source Code Mecari Frequent Item

```

$list_item_set = array();
$frequent_item_set = array();
$ordered_fis = array();
$frequency_fis = array();
    foreach ($transaksi as $trans) {
        $item_set = array();
        foreach ($trans->details as $detail) {
            $item = array(
                "code" => $detail->product->code,
                "frequency" => $global_item[$detail->product->code],
            );
            if(!in_array($item, $item_set)) {
                $item_set[] = $item;
            }
        }
        $ordered_item_set = $item_set;
        usort($ordered_item_set, function($a, $b) {
            return $a['frequency'] > $b['frequency'] ? -1 : 1;
        });

        if(!in_array($ordered_item_set, $list_item_set)) {
            $list_item_set[] = $ordered_item_set;
            $frequency_fis[] = 1;
            $frequent_item_set[] = $item_set;
            $ordered_fis[] = $ordered_item_set;
        }
    }
    else {
        $index_fis = array_search($ordered_item_set, $ordered_fis);
        $frequency_fis[$index_fis]++;
    }
}
foreach ($frequency_fis as $index_fis => $value) {
    $frequent_item_set[$index_fis]['frequency'] = $value;
}
foreach ($frequency_fis as $index_fis => $value) {
    $ordered_fis[$index_fis]['frequency'] = $value;
}
foreach ($frequent_item_set as $index_fis => $value) {
    if($value['frequency'] < 2) {
        unset($frequent_item_set[$index_fis]);
    }
}
foreach ($ordered_fis as $index_fis => $value) {
    if($value['frequency'] < 2) {
        unset($ordered_fis[$index_fis]);
    }
}
usort($ordered_fis, function($a, $b) {
    return $a['frequency'] > $b['frequency'] ? -1 : 1;
});

```

Modul Program 4. 3 Source Code Mecari Frequent Item (lanjutan)

Proses di atas sekaligus menyeleksi data yang tidak memenuhi nilai *minimum support* yang sudah ditentukan. Di sini nilai *minimum support* secara default *system* adalah 2%.

Setelah data diseleksi berdasarkan nilai *minimum support*, selanjutnya membentuk tabel *global cfp tree*. Source codenya disajikan pada Modul Program 4.4.

```

$this->cfp_tree = array();
    $this->cfp_tree[] = array(
        "parent" => "-",
        "node" => "root",
        "count" => '-'
    );
    // echo json_encode($ordered_fis); die;
    foreach ($ordered_fis as $fis) {
        if(!is_array($fis)) {
            continue;
        }
        // if(count($fis) > 2) {
        for ($i=0; $i < count($fis) - 1; $i++) {
            $prev = $i - 1;
            $prev2 = $i - 2;
            if($prev >= 0) {
                $this->cfp_tree[] = array(
                    "parent" => $prev2 < 0 ? $this-
>findParents(null, $fis[$prev]["code"]) : $this-
>findParents($fis[$prev2]["code"], $fis[$prev]["code"]),
                    "node" => $fis[$i]["code"],
                    "count" => 1
                );
            }
        }
        else {
            $this->addCounter($fis[$i]["code"]);
        }
    }
    // }
}
$this->storeCFPTree();

$rules = $this->aturan_asosiasi($global_item_reverse, $min_sup,
$min_conf);
foreach ($rules as $key => $rule) {
    $codes = explode(" ", $rule['codes']);
    $products = Produk::select(['name', 'code'])->whereIn('code',
$codes)->get();
    $rules[$key]['products'] = $products;
}
$cfp_tree = $this->cfp_tree;
// return json_encode($rules);

return view('pages.algoritma.ctpro', compact('global_item',
'frequent_item_set', 'ordered_fis', 'cfp_tree', 'rules', 'min_sup',
'min_conf'));
}

```

Modul Program 4. 4 Source Code Pembentukan Global CFP-Tree


```

public function findParents($prev2=null, $code=null)
{
    $parent = 0;
    if ($prev2 !== null) {
        foreach ($this->cfp_tree as $index => $branch) {
            $this_parent = intval($this->cfp_tree[$index]["parent"]);
            if($this->cfp_tree[$index]["node"] == $code && $this->cfp_tree[$this_parent]["node"] == $prev2)
            {
                // $this->cfp_tree[$index]["count"]++;
                $parent = $index;
                break;
            }
        }
    } else {
        foreach ($this->cfp_tree as $index => $branch) {
            if($this->cfp_tree[$index]["node"] == $code) {
                // $this->cfp_tree[$index]["count"]++;
                $parent = $index;
                break;
            }
        }
    }
    return $parent;
}

public function addCounter($code)
{
    $found_index = 0;
    foreach ($this->cfp_tree as $index => $branch) {
        if($branch["node"] == $code) {
            $found_index = $index;
            break;
        }
    }
    if($found_index == 0) {
        $this->cfp_tree[] = array(
            "parent" => 0,
            "node" => $code,
            "count" => 1
        );
    } else {
        foreach ($this->cfp_tree as $index => $branch) {
            if($branch["node"] == $code)
            {
                $this->cfp_tree[$index]["count"]++;
                break;
            }
        }
    }
    return null;
}

```

Modul Program 4. 5 *Source Code* Pembentukan Global CFP-Tree (lanjutan)

```

public function storeCFPTree()
{
    CFPTree::query()->truncate();
    foreach ($this->cfp_tree as $index => $branch) {
        $node = new CFPTree;
        $node->index = $index;
        $node->parent = $branch['parent'];
        $node->node = $branch['node'];
        $node->count = $branch['count'];
        $node->save();
    }
    return null;
}

```

Modul Program 4. 6 *Source Code* Pembentukan Global CFP-Tree (lanjutan)

Pada source code di atas adalah proses untuk pembentukan *global cfp tree*. Mula-mula-mula inialisasi tree, kemudian buat node baru untuk setiap *item* yang ada pada tabel *global item*. Selanjutnya megakses setiap *item* yang ada pada *itemset*. Jika *item* yang ada pada *itemset* adalah *currentNode* maka count yang ada pada *currentNode* ditambah satu. Jika *item* tersebut tidak sama dengan *currentNode* maka lakukan penelusuran untuk anak dari *currentNode*. Dan jika *item* tersebut ada pada anak dari *currentNode* maka count yang ada pada anak *currentNode* ditambah satu. Jika tidak buat cabang baru untuk *item* baru dan count pada cabang baru tersebut ditambah satu. Setelah *global cfp tree* terbentuk selanjutnya adalah mencari hasil asosiasinya. Source code disajikan pada Modul Program 4.8.

```

public function aturan_asosiasi($global_item_reverse, $min_sup=0,
$min_conf=0)
{
    $tt = Transaksi::count();
    $pasangan = array();
    $list_pasangan = array();
    $rules = array();
    foreach ($global_item_reverse as $code => $frequency) {
        $associates = array();
        foreach ($this->cfp_tree as $index => $branch) {
            if($branch["node"] == $code) {
                //check parent
                if($branch["parent"] != 0) {
                    $associates[] = $this->cfp_tree[$branch["parent"]]["node"];
                }
                //check children
                foreach ($this->cfp_tree as $index1 => $branch1) {
                    if($branch1["parent"] == $index) {
                        $associates[] = $branch1["node"];
                    }
                }
            }
        }
    }
}

```

Modul Program 4. 7 *Source Code* Mencari Hasil Asosiasi

```

    }
    }
    $cek_asosiasi[] = array(
        "code" => $code,
        "associates" => $associates
    );
}

foreach ($cek_asosiasi as $cek) {
    if(count($cek["associates"]) == 0) {
        continue;
    }
    foreach ($cek["associates"] as $associate) {
        $codes = array($cek["code"], $associate);
        sort($codes);
        if(!in_array($codes, $list_pasangan)) {
            $list_pasangan[] = $codes;
        }
    }
}

foreach ($list_pasangan as $pasangan) {
    $count_a = Transaksi::where(function ($query) use
($pasangan) {
        $query->whereHas('details', function($q)
use ($pasangan){
            $q->whereHas('product', function($p)
use ($pasangan){
                $p->where('code', $pasangan[0]);
            });
        });
    }->count();

    $count_b = Transaksi::where(function ($query) use
($pasangan) {
        $query->whereHas('details', function($q)
use ($pasangan){
            $q->whereHas('product', function($p)
use ($pasangan){
                $p->where('code', $pasangan[1]);
            });
        });
    }->count();

    $count_aub = Transaksi::where(function ($query) use ($pasangan){
        foreach ($pasangan as $code) {
            $query->whereHas('details',
function($q) use ($code){
                $q->whereHas('product',
function($p) use ($code){
                    $p->where('code', $code);
                });
            });
        }
    }->count();
}

```

Modul Program 4. 8 Source Code Mencari Hasil Asosiasi (lanjutan)

```

        $support_a = ($count_a / $tt)*100;
        $benchmark_confidence = $count_b / $tt;
        $support_aub = ($count_aub / $tt)*100;
        $confidence = $count_a == 0 ? 0 : ($count_aub/$count_a)*100;
        $lift_ratio = $benchmark_confidence == 0 ? 0 : $confidence /
$benchmark_confidence;

        if($support_aub >= $min_sup && $confidence >= $min_conf) {
            $rules[] = array(
                "codes"=> implode(" ", $pasangan),
                "support_a" => $support_a,
                "support_aub" => $support_aub,
                "count_b" => $count_b,
                "benchmark_confidence" => $benchmark_confidence,
                "confidence" => $confidence,
                "lift_ratio" => $lift_ratio
            );
        }
    }
    Rule::query()->truncate();
    foreach ($rules as $rule) {
        $data = new Rule;
        $data->codes = $rule['codes'];
        $data->support_a = $rule['support_a'];
        $data->support_aub = $rule['support_aub'];
        $data->benchmark_confidence = $rule['benchmark_confidence'];
        $data->confidence = $rule['confidence'];
        $data->lift_ratio = $rule['lift_ratio'];
        $data->save();
    }
    return $rules;
}

```

Modul Program 4. 9 Source Code Mencari Hasil Asosiasi (lanjutan)

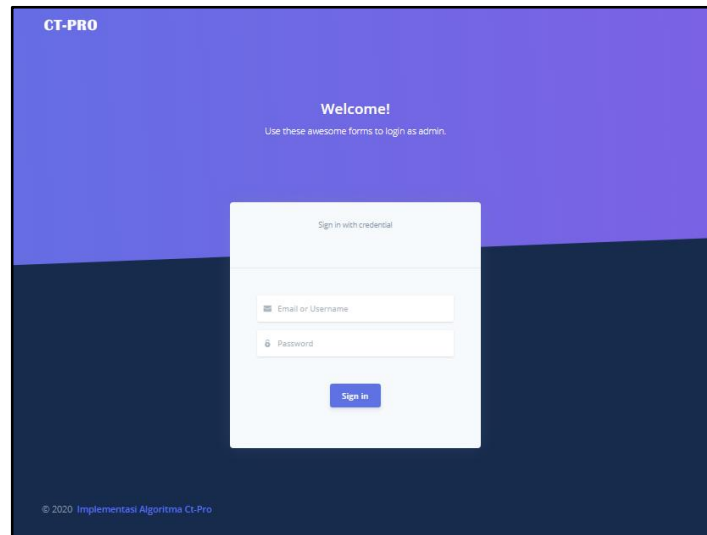
Untuk perhitungan asosiasinya, menggunakan konsep bottom up pada tabel *global item*, dengan dimulai dari *item* yang memiliki frekuensi paling kecil atau dengan kata lain *item* dengan index paling bawah. Dari proses tersebut akan terbentuk local cfp tree, setelah local cfp tree didapatkan, langkah selanjutnya adalah membentuk aturan yang akan diperoleh dari proses *subset frequent item*. Dari hasil *subset* tersebut dilakukan perhitungan nilai *support*, *confidence* dan *lift ratio*. Setelah hasil perhitungan ditemukan, selanjutnya hasil akan ditampilkan pada halaman CT-PRO.

4.3 Implementasi Interface

Implementasi interface dibangun berdasarkan pada rancangan interface dan kebutuhan sistem. Adapun implementasi interface untuk pengguna yang adalah sebagai berikut :

a. Halaman Login

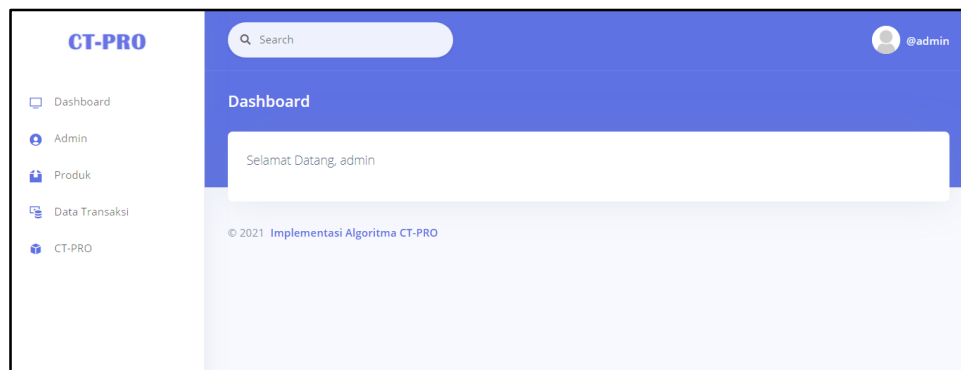
Pada halaman *login* digunakan untuk masuk ke halaman utama aplikasi dan halaman lainnya dengan memasukkan *username* dan password. Implementasi halaman *login* diuraikan pada Gambar 4.7.



Gambar 4. 7 Halaman Login

b. Halaman Dashboard

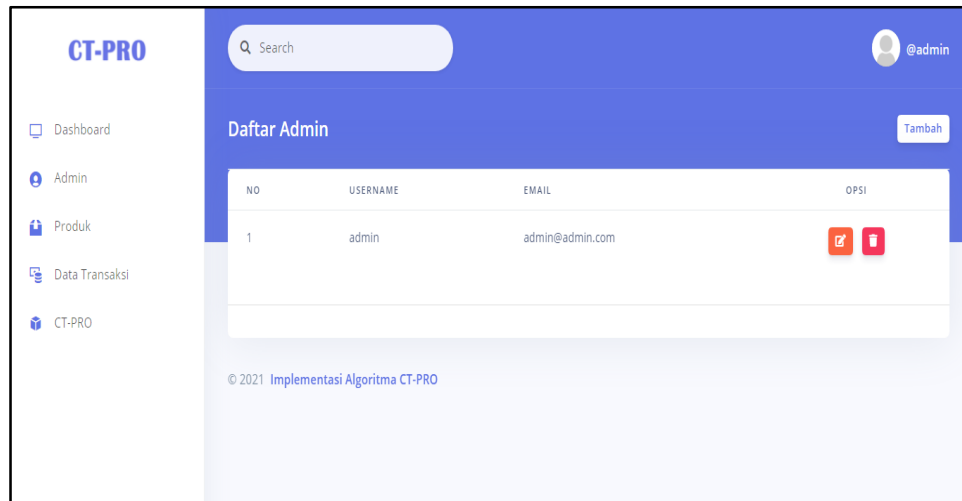
Pada halaman dashboard merupakan ketika pengguna berhasil masuk dari halaman *login*. Halaman dashboard hanya berisikan tampilan ucapan selamat datang. Implementasi halaman dashboard diuraikan pada Gambar 4.8.



Gambar 4. 8 Halaman Dashboard

c. Halaman Admin

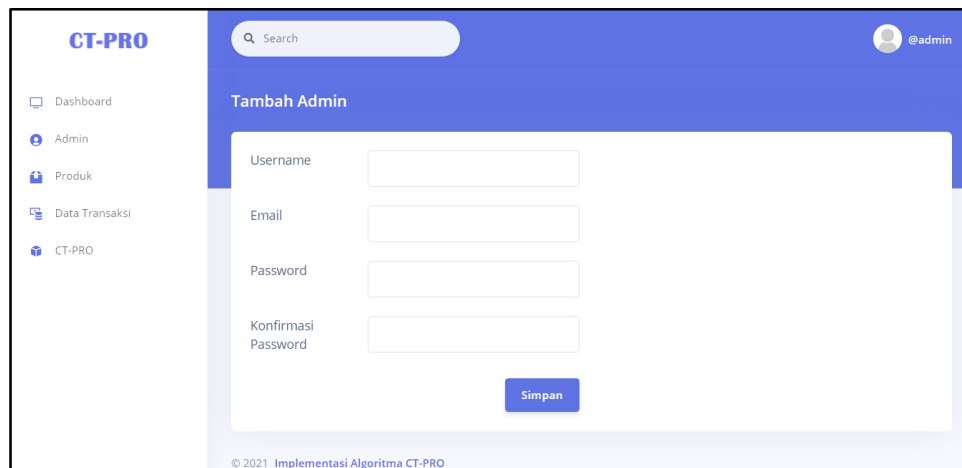
Pada halaman admin merupakan halaman yang berisi semua data *user/admin* yang sudah dimasukkan dan tersimpan di dalam basis data. Implementasi halaman admin diuraikan pada Gambar 4.9.



Gambar 4. 9 Halaman Admin

d. Halaman Tambah Admin/*User*

Pada halaman tambah admin/*user* merupakan halaman yang berisi form tambah admin/*user* yang akan disimpan di dalam basis data. Implementasi halaman tambah admin/*user* diuraikan pada Gambar 4.10.



Gambar 4. 10 Halaman Tambah Admin/*User*

e. Halaman *Edit Admin/User*

Pada halaman *edit admin/user* merupakan halaman yang berisi form *edit* data admin/user sesuai data yang dipilih yang akan disimpan di dalam basis data. Implementasi halaman *edit admin/user* diuraikan pada Gambar 4.11.

Gambar 4. 11 Halaman *Edit Admin/User*

f. Halaman Produk

Pada halaman produk merupakan halaman yang berisi semua data produk yang sudah dimasukkan dan tersimpan di dalam basis data. Implementasi halaman produk diuraikan pada Gambar 4.12.

NO	KODE	NAMA	STATUS	OPS
1	P-226	GG Signature Mild	Aktif	
2	P-227	Lodjie Hijau	Aktif	
3	P-228	Wismlak	Aktif	
4	P-229	Sampoerna Kretek	Aktif	
5	P-230	Dji Sam Soe	Aktif	
6	P-231	Aroma Kretek	Aktif	

Gambar 4. 12 Halaman Produk

g. Halaman Tambah Produk

Pada halaman tambah produk merupakan halaman yang berisi form tambah produk yang akan disimpan di dalam basis data. Implementasi halaman tambah produk diuraikan pada Gambar 4.13.

Gambar 4. 13 Halaman Tambah Produk

h. Halaman *Edit Produk*

Pada halaman *edit* produk merupakan halaman yang berisi form *edit* data produk sesuai data yang dipilih yang akan disimpan di dalam basis data. Implementasi halaman *edit* produk diuraikan pada Gambar 4.14.

Gambar 4. 14 Halaman *Edit Produk*

i. Halaman Transaksi

Pada halaman transaksi merupakan halaman yang berisi semua data transaksi yang sudah dimasukkan dan tersimpan di dalam basis data. Implementasi halaman produk diuraikan pada Gambar 4.15.

NO	NO TRANSAKSI	DATE	HARGA	STATUS
1	INV/202109/00734	2021-09-06	Rp. 15.500	Info Edit Delete
2	INV/202104/21016	2021-03-31	Rp. 37.000	Info Edit Delete
3	INV/202104/21334	2021-03-31	Rp. 27.500	Info Edit Delete
4	INV/202104/21412	2021-03-31	Rp. 115.500	Info Edit Delete

Gambar 4. 15 Halaman Transaksi

j. Halaman Tambah Transaksi

Pada halaman tambah transaksi merupakan halaman yang berisi form tambah transaksi yang akan disimpan di dalam basis data. Implementasi halaman tambah transaksi diuraikan pada Gambar 4.16.

Gambar 4. 16 Halaman Tambah Transaksi

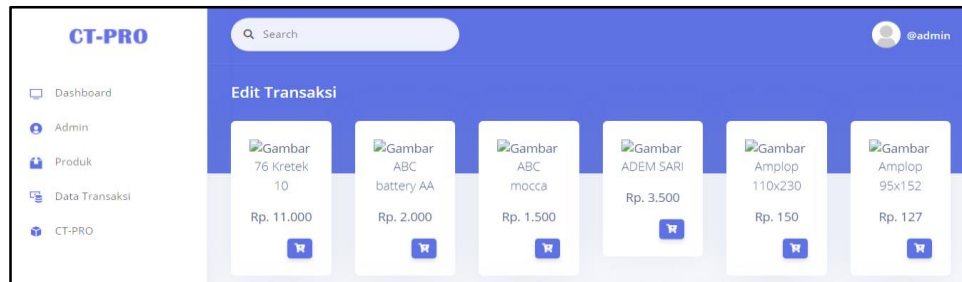
k. Lanjutan Halaman Tambah Transaksi

Lanjutan halaman tambah transaksi adalah lanjutan dari Gambar 4.16 yang akan diuraikan pada Gambar 4.17.

Gambar 4. 17 Lanjutan Halaman Tambah Transaksi

l. Halaman *Edit* Transaksi

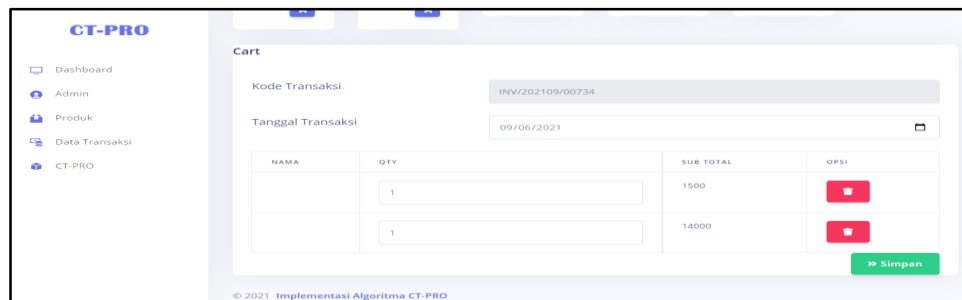
Pada halaman *edit* transaksi merupakan halaman yang berisi form *edit* data transaksi sesuai data yang dipilih yang akan disimpan di dalam basis data. Implementasi halaman *edit* transaksi diuraikan pada Gambar 4.18.



Gambar 4. 18 Halaman *Edit* Transaksi

m. Lanjutan Halaman *Edit* Transaksi

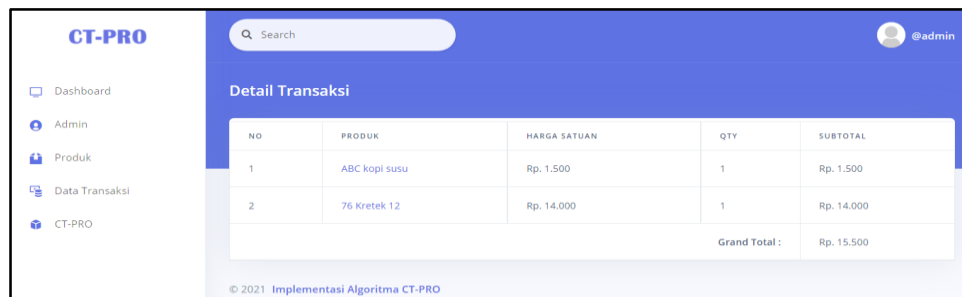
Lanjutan halaman *edit* transaksi adalah lanjutan dari Gambar 4.18 yang akan diuraikan pada Gambar 4.19.



Gambar 4. 19 Lanjutan Halaman *Edit* Transaksi

n. Halaman Detail Transaksi

Pada halaman detail transaksi merupakan halaman yang berisi detail data pada suatu data transaksi yang disimpan di dalam basis data. Implementasi halaman detail transaksi diuraikan pada Gambar 4.20.



Gambar 4. 20 Halaman Detail Transaksi

o. Halaman *CT-Pro*

Pada halaman *ct-pro* merupakan halaman yang berisi hasil dari proses hitung data yang memenuhi *minimum support* dan *minimum confidence* yang kemudian menjadi *global item* yang diurutkan secara descending serta hasil. Implementasi halaman *ct-pro* diuraikan pada Gambar 4.21

Gambar 4. 21 Halaman *CT-Pro*

4.4 Pengujian Sistem

Pada tahap pengujian sistem merupakan tahap yang dilakukan setelah tahap implementasi selesai. Tujuan pada tahap pengujian ini adalah untuk menjamin apakah aplikasi atau sistem yang dibangun sesuai dengan kebutuhan fungsional aplikasi sehingga tujuan yang diinginkan tercapai. Pengujian yang dipakai adalah *black box testing*. *Black box testing* merupakan suatu pengujian yang dilakukan untuk mengamati hasil dari eksekusi pada aplikasi tersebut. Pengamatan hasil melalui data uji dan memeriksa fungsional dari aplikasi itu sendiri. Hasil pengujian diuraikan pada Tabel 4.1.

Tabel 4. 1 Hasil *Black Box Testing*

Fitur Aplikasi	Status
Login	Terpenuhi
Halama Dashboard	Terpenuhi
Tampil Data Admin	Terpenuhi
Tambah Admin	Terpenuhi
Edit Data Admin	Terpenuhi
Hapus Data Admin	Terpenuhi
Tampil Data Produk	Terpenuhi
Tambah Data Produk	Terpenuhi
Edit Data Produk	Terpenuhi
Hapus Data Produk	Terpenuhi
Import Data Produk	Terpenuhi
Export Data Produk	Terpenuhi
Tampil Data Transaksi	Terpenuhi
Tambah Data Transaksi	Terpenuhi
Edit Data Transaksi	Terpenuhi

Tabel 4. 2 Hasil *Black Box Testing*

Fitur Aplikasi	Status
Hapus Data Transaksi	Terpenuhi
Import Data Transaksi	Terpenuhi
Export Data Transaksi	Terpenuhi
Tampil Detail Transaksi	Terpenuhi
Tampil Hasil Proses Asosiasi	Terpenuhi

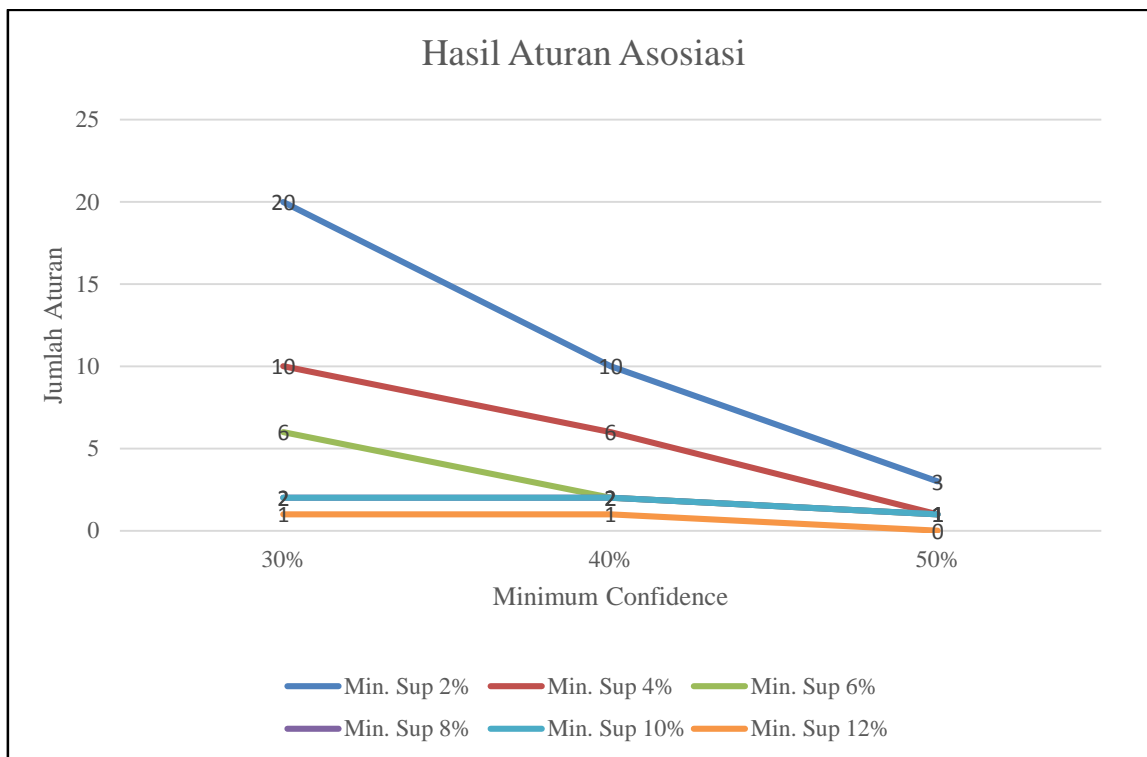
4.5 Hasil dan Pembahasan

Hasil aturan asosiasi yang dianalisa berasal dari data transaksi selama 3 bulan yaitu bulan Januari – Maret 2021. Data transaksi tersebut diproses untuk memperoleh hasil aturan asosiasi dengan memasukkan nilai *minimum support* dan nilai *minimum confidence* yang beragam. Berikut Tabel 4.2 merupakan hasil dari aturan asosiasi yang diproses :

Tabel 4. 3 Hasil Dari Aturan Asosiasi Bulan Januari-Maret 2021

Minimum Support (%)	Minimum Confidence (%)	Jumlah Aturan
2	30	20
2	40	10
2	50	3
4	30	10
4	40	6
4	50	1
6	30	6
6	40	3
6	50	1
8	30	2
8	40	2
8	50	1
10	30	2
10	40	2
10	50	1
12	30	1
12	40	1
12	50	0

Grafik hasil penghitungan jumlah aturan dari proses aturan asosiasi pada bulan Januari – Maret 2021 diuraikan pada Gambar 4.22.



Gambar 4. 22 Grafik Proses Aturan Asosiasi Bulan Januari-Maret 2021

Analisis hasil dari proses aturan asosiasi dari data pada bulan April 2019 adalah sebagai berikut :

1. Hasil proses aturan asosiasi dengan memasukkan nilai *minimum support* 2 - 12 dan *minimum confidence* 30% - 50% maka terbentuk aturan asosiasi maksimal 20 aturan dan minimal 0(nol) adanya aturan.
2. Hasil aturan asosiasi yang memiliki kemunculan terbaik yaitu produk segitiga terigu → telur 1kg dengan nilai *support* 11,27%, nilai *confidence* 53,01% dan *lift ratio* 1,79%.

BAB V

PENUTUP

5.1 Kesimpulan

Hasil aturan asosiasi dengan menggunakan algoritma *CT-Pro* dari penelitian ini, maka diperoleh kesimpulan sebagai berikut:

1. Pada penelitian ini dapat menghasilkan aturan asosiasi dari kombinasi produk pada periode waktu Januari-Maret 2021.
2. Analisis aturan asosiasi dengan algoritma *CT-Pro* dapat menghasilkan aturan asosiasi untuk kombinasi penjualan pada periode bulan tertentu, serta mengetahui produk yang memiliki keterikatan tinggi.
3. Saat pengujian berdasarkan data selama 3 bulan, dilakukan sebanyak 6 kali pengujian dengan *minimum support* dan *minimum confidence* mulai dari nilai default system sampai dengan tidak ditemukan lagi aturan yang muncul. Hingga akhirnya didapat batas maksimal penentuan *minimum support* 12% dan *minimum confidence* 40%, dan dari 6 kali pengujian tersebut yang mempunyai nilai kemunculan terbaik yaitu segitiga terigu → telur 1kg dengan nilai *support* 11,27%, nilai *confidence* 53,01% dan *lift ratio* 1,79%.
4. Algoritma *CT-Pro* akan menghasilkan aturan asosiasi pola konsumen jika *minimum support* dan *minimum confidence* yang ditentukan terpenuhi. Semakin tinggi nilai *minimum support* dan *minimum confidence* yang ditentukan maka semakin sedikit hasil asosiasi *item* yang ditemukan, sedangkan semakin kecil *minimum support* dan *minimum confidence* maka semakin banyak hasil asosiasi *itemnya*.

5.2 Saran

Adapun saran-saran yang dapat digunakan untuk mengembangkan penelitian ini yaitu:

1. Untuk pengembangan penelitian ini lebih lanjut disarankan untuk membuat hasil aturan asosiasi yang mempunyai lebih dari 2 *itemset*.
2. Hasil aturan asosiasi yang terbentuk dapat dikembangkan lagi untuk memberikan manfaat yang lebih lagi dan dapat dikembangkan menjadi basis pengetahuan sehingga penelitian dan aplikasi menjadi lebih bagus lagi.
3. Pengembangan dapat dilakukan dengan penerapan pada tempat selain toko agar aplikasi terus berkembang.

DAFTAR PUSTAKA

- Anggraeni, Rizky Mei. 2014. "Perbandingan Algoritma *Apriori* Dan Algoritma *FP-Growth* Untuk Rekomendasi Pada Transaksi Peminjaman Buku Di Perpustakaan Universitas Dian Nuswantoro." *Teknik Informatika* 1–6.
- Astuti, Femi Dwi. 2015. "Panduan Praktikum Algoritma Dan Pemrograman 1."
- Atmaja, Eduardus Hardika Sandy, Risky Simaremare, and Paulina Heruningsih Prima Rosa. 2019. "Application of *CT-Pro* Algorithm For Crime Analysis." *Conference SENATIK STT Adisutjipto Yogyakarta* 5:435–44. doi: 10.28989/senatik.v5i0.360.
- Dewantara, Heru, Purnomo Budi Santosa, and Nasir Widha Setyanto. 2013. "Perancangan Aplikasi *Data Mining* Dengan Algoritma *Apriori* Untuk Frekuensi Analisis Keranjang Belanja Pada Data Transaksi Penjualan (Studi Kasus Di Swalayan Kpri Universitas Brawijaya) Designing Applications *Data Mining* With The *Apriori* Algorithm To Freq." *Jurnal Rekayasa Dan Manajemen Sistem Industri* 1(3):415–26.
- Fatihatul, Fathimah, Atje Setiawan, and Rudi Rosadi. 2016. "Asosiasi *Data Mining* Menggunakan Algoritma *FP-Growth* Untuk *Market Basket Analysis*." *Jatinangor: Universitas Padjadjaran* 1–8.
- Gunadi, Goldie. 2012. "Penerapan Metode *Data Mining Market Basket Analysis* Terhadap Data Penjualan Produk Buku Dengan Menggunakan Algoritma *Apriori* Dan *Frequent Pattern Growth (FP-Growth)* : Studi Kasus Percetakan PT. Gramedia." *Jurnal TELEMATIKA MKOM* 4(1).
- Pendekatan, Dengan, Dinamika Sistem, and System Dynamics. 2014. "B a b I." (1979):13–17.
- Priyana, Frismadani Anggita, and Acun Kardianawati. 2019. "*Data Mining* Asosiasi Untuk Menentukan Cross-Selling Produk Menggunakan Algoritma *Frequent Patern-Growth* Pada Koperasi Karyawan PT . Phapros Semarang." *Ilmu Komputer*.
- Purba, Berta Rohana Br, Nelly Astuti Hasibuan, Guidio L. Ginting, and Suginam Suginam. 2018. "Implementasi Algoritma *Apriori* Untuk Mencari Relasi Pada Transaksi Pembelian Alat-Alat Kesehatan." *JURIKOM (Jurnal Riset Komputer)* 5(3):269–77.
- Rohmania Nurlaili Putri, Sihwi Sari Widya, Sulistyo Meiyanto Eko. n.d. "Implementasi Algoritma *Ct-Pro* Untuk Menemukan Pola Pada Data Siswa Sma (Studi Kasus : Madrasah Aliyah Negeri (Man) Karanganon Klaten)." 392–98.
- Satrio Agung W, Ari Kusyanti, Mahendra Data. 2011. "*Database Entity Relationship Diagram*." *Materi Kuliah* 2–7.
- Ummah, Khoirul. 2015. "Sistem Rekomendasi Pembelajaran Pada E-Learning Menggunakan Algoritma *CT-PRO*." *Jurnal Teknologi Technoscientia* 7(2):144–49.