

**MEDIA JOGJA BERBAGI BERBASIS WEB DAN RESTFUL WEB SERVICE  
UNTUK SARANA BERBAGI JASA DI WILAYAH YOGYAKARTA**

**TUGAS AKHIR**

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



Disusun Oleh :

Dinizar Lutfia Afrian  
123150155

**PROGRAM STUDI INFORMATIKA  
JURUSAN INFORMATIKA  
FAKULTAS TEKNIK INDUSTRI  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
YOGYAKARTA  
2019**

**HALAMAN PENGESAHAN PEMBIMBING**

**MEDIA JOGJA BERBAGI BERBASIS WEB DAN RESTFUL WEB SERVICE  
UNTUK SARANA BERBAGI JASA DI WILAYAH YOGYAKARTA**



Disusun oleh:

Dinizar Lutfia Afrian  
123150155

Telah diuji dan dinyatakan lulus oleh pembimbing  
pada tanggal: .....

Menyetujui,  
Pembimbing I

Pembimbing II

Hidayatulah Himawan, S.T., M.M., M.eng.  
NIP. 1976 12 24 2005 01 1001

Eka Indarto S.T., M.Eng.  
NIK. 17 2012

Mengetahui,  
Ketua Program Studi

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

**HALAMAN PENGESAHAN PENGUJI**

**MEDIA JOGJA BERBAGI BERBASIS WEB DAN RESTFUL WEB SERVICE  
UNTUK SARANA BERBAGI JASA DI WILAYAH YOGYAKARTA**



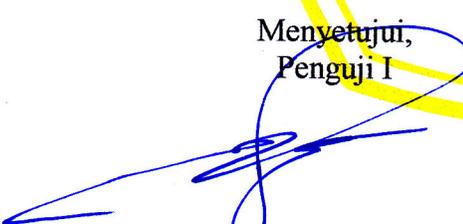
Disusun Oleh :

Dinizar Lutfia Afrian  
123150155

Telah diuji dan dinyatakan lulus pada tanggal 9 Desember 2019

Menyetujui,  
Penguji I

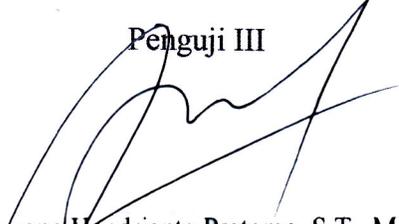
Penguji II

  
Hidayatulah Himawan. S.T.,M.M.,M.eng.  
NIP. 1976 12 24 2005 01 1001

  
Eka Indarto S.T.,M.Eng.  
NIK. 17 2012

Penguji III

Penguji IV

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

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

**SURAT PERNYATAAN  
KARYA ASLI TUGAS AKHIR**

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

Nama : Dinizar Lutfia Afrian  
No Mhs : 123150155

Menyatakan bahwa karya ilmiah saya yang berjudul :

**MEDIA JOGJA BERBAGI BERBASIS WEB DAN RESTFUL WEB SERVICE  
UNTUK SARANA BERBAGI JASA DI WILAYAH YOGYAKARTA**

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

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta  
Pada Tanggal : 10 Desember 2019  
Yang Menyatakan


(Dinizar Lutfia Afrian)

## PERNYATAAN BEBAS PLAGIAT

Saya yang bertanda tangan dibawah ini :

Nama : Dinizar Lutfia Afrian  
No Mhs : 123150155  
Fakultas/Prodi : Teknik Industri/ Informatika

Dengan ini saya menyatakan bahwa judul Tugas Akhir :

**MEDIA JOGJA BERBAGI BERBASIS WEB DAN RESTFUL WEB SERVICE  
UNTUK SARANA BERBAGI JASA DI WILAYAH YOGYAKARTA**

Adalah hasil kerja saya sendiri dan benar bebas dari plagiat kecuali cuplikan serta ringkasan yang terdapat didalamnya telah saya jelaskan sumbernya (sitasi) dengan jelas. Apabila pernyataan ini terbukti tidak benar maka saya bersedia menerima sanksi sesuai peraturan Mendiknas RI No 17 Tahun 2000 dan Peraturan Perundang-undangan yang berlaku.

Demikian surat pernyataan ini saya buat dengan penuh tanggung jawab.

Yogyakarta, 10 Desember 2019

Yang membuat pernyataan

  
  
Dinizar Lut  
NIM 123150155

## **HALAMAN PERSEMBAHAN**

### ***Alhamdulillahirabbil alamin***

*”Puji syukur kehadiran Allah SWT yang telah melimpahkan karunia-Nya sehingga saya bisa menyelesaikan tugas akhir ini. Tugas akhir ini saya persembahkan kepada kedua orang tua saya yang telah mendukung dan memberi kepercayaan kepada saya untuk memilih studi ini. Terimakasih kepada semua yang pernah menjadi proses di kehidupan saya selama di kampus.”*

Saya tidak bangga dengan keberhasilan yang tidak saya rencanakan sebagaimana saya tidak akan menyesal atas kegagalan yang terjadi di ujung usaha maksimal.

-Harun Al Rasyid-

## ABSTRAK

Yogyakarta sebagai daerah yang mengusung konsep *smart city* tentu mengedepankan pemanfaatan teknologi yang maksimal untuk meningkatkan efisiensi pelayanan. Salah satu dimensi yang menjadi fokus pengembangan *smart city* adalah *smart people*. Dari identifikasi *smart people* di daerah Yogyakarta ditemukan problem kota yaitu tingginya angka kemiskinan dan angka pengangguran. Dari permasalahan yang ditemukan penelitian ini akan melakukan pendekatan kepada masyarakat untuk menciptakan media berbagi berbasis web dan menggunakan *web service* sebagai solusinya.

Dengan metodologi penelitian *living lab* dan metodologi pengembangan perangkat lunak *User Center Design* (UCD) yang dalam penerapannya mengedepankan pembuatan perangkat lunak yang mengutamakan pengguna. Oleh karena itu diperlukan pengumpulan data dengan kuisioner sebagai acuan identifikasi kebutuhan pengguna. Selain itu pengumpulan data yang dilakukan adalah identifikasi permasalahan kota mengenai angka kemiskinan dan pengangguran di Yogyakarta. Dari data yang didapat perancangan bertahap dilakukan dari menentukan konteks penggunaan, menentukan kebutuhan pengguna, dan yang terakhir membuat ekosistem pengguna.

Dalam menentukan konteks penggunaan didapat tiga peran pengguna dalam pembuatan yaitu *Application Designer and Programmer, Operation staff, Pengguna*. Kemudian dari hasil kuisioner ditentukan kebutuhan pengguna mengenai aplikasi, dan ekosistemnya. Aplikasi yang sudah jadi kemudian dievaluasi dengan kuisioner dan hasil yang didapat berdasarkan fungsionalitas sistem sesuai dengan apa yang dibutuhkan pengguna dan fungsionalitas aplikasi hampir semua sesuai dengan kebutuhan pengguna.

**Kata Kunci:** *Living Lab, Smart city, User Center Design, Media berbagi*

## KATA PENGANTAR

*Assalamu'alaikum Wr.Wb*

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, hidayah, dan inayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“MEDIA JOGJA BERBAGI BERBASIS WEB DAN RESTFUL WEB SERVICE UNTUK SARANA BERBAGI JASA DI WILAYAH YOGYAKARTA”** serta dapat menyusun Laporan Tugas Akhir ini tepat pada waktunya.

Penyusunan ini merupakan salah satu mata kuliah yang harus dilaksanakan oleh setiap mahasiswa, khususnya mahasiswa Informatika dalam menyelesaikan jenjang Pendidikan S1 Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta. Dalam penulisan skripsi ini tentunya tidak lepas dari kekurangan, baik aspek kualitas maupun aspek kuantitas dari materi penelitian yang disajikan. Semua ini didasarkan dari keterbatasan yang dimiliki penulis.

Laporan ini tidak dapat terselesaikan tanpa bantuan dari berbagai pihak, baik secara langsung maupun tidak langsung. Pada kesempatan ini penulis ingin menyampaikan ungkapan terima kasih kepada:

1. Allah SWT yang senantiasa memberikan petunjuk, kemudahan, rezeki, kesehatan dan kesempatan.
2. Ayah dan Ibu yang telah memberikan doa dan dukungan sehingga pengerjaan skripsi ini bisa berjalan dengan lancar.
3. Bapak Bambang Yuwono, S.T., M.T. selaku Ketua Jurusan Informatika UPN “Veteran” Yogyakarta.
4. Bapak DR. Awang Hendrianto Pratomo, S.T., M.T. selaku Ketua Prodi Informatika UPN “Veteran” Yogyakarta, koordinator tugas akhir.

5. Bapak Hidayatulah Himawan. S.T., M.M., M.eng. selaku dosen pembimbing I atas waktu dan kesabarannya dalam membimbing penulis.
6. Bapak Eka Indarto. S.T., M.eng. selaku dosen pembimbing II atas waktu dan kesabarannya dalam membimbing penulis.
7. Seluruh dosen dan staff tata usaha prodi Informatika yang telah membantu selama menempuh pendidikan.
8. Mas Syamsudin, Mas Fai, Mas Indra yang telah memberikan bantuan selama pengerjaan skripsi.
9. Pandhu, Adit, Aziz, Raul, Affan, Sani, Dion, Ilza, Shalfa, Annisa Reza Dhiya'tulhaq yang telah memberikan doa, dukungan dan bantuan sehingga pengerjaan skripsi ini lancar.
10. Teman-teman Informatika angkatan 2015 yang telah mendukung dan memberikan bantuan selama pengerjaan skripsi.
11. Semua rekan yang telah membantu penyusunan laporan Tugas Akhir ini.

Dengan menyadari bahwa laporan ini masih jauh dari sempurna. Maka dari itu penulis mengharapkan adanya saran, kritik, dan tanggapan yang bersifat membangun dalam upaya pembelajaran lebih lanjut. Akhir kata, semoga laporan ini berguna dan bermanfaat bagi semua pihak, dan penulis sendiri pada khususnya. Semoga Tuhan Yang Maha Esa selalu membimbing kita dalam menyelami dan mengamalkan ilmu-Nya untuk menuju kehidupan yang lebih baik.

***Wassalamu 'alaikum Wr.Wb.***

Yogyakarta, 10 Desember 2019

Penulis

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN PEMBIMBING .....	ii
HALAMAN PENGESAHAN PENGUJI.....	iii
SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR.....	iv
PERNYATAAN BEBAS PLAGIAT .....	v
HALAMAN PERSEMBAHAN .....	vi
ABSTRAK .....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI .....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL .....	xiv
DAFTAR MODUL PROGRAM.....	xv
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	3
1.3 Manfaat Penelitian.....	3
1.4 Tujuan Penelitian.....	4
1.5 Batasan Masalah .....	4
1.6 Metodologi Penelitian .....	4
1.7 Sistematika Penulisan.....	6
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>8</b>
2.1 Smart City.....	8
2.2 Living Lab .....	10
2.3 User Center Designt (UCD) .....	14
2.4 Berbagai .....	16
2.5 Jasa .....	17
2.6 Media.....	18
2.7 Web.....	18
2.8 Web Service.....	19
2.9 SOA (Service-Oriented Architecture) .....	23
2.10 RESTful.....	25
2.11 Integrasi Data.....	28
2.12 JSON dan XML .....	28
2.12.1 JSON.....	29
2.12.2 XML .....	30
2.13 CURL .....	31
2.14 Database Manajemen System (DBMS).....	32
2.15 PHP.....	32
2.16 Javascript .....	32
2.17 HTML.....	33
2.18 CSS.....	33
2.19 Tinjauan Literatur .....	34
<b>BAB III METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM.....</b>	<b>36</b>
3.1 Metodologi Penelitian .....	36

3.2 Smart City .....	38
3.3 Eksplorasi .....	38
3.3.1 Data Kemiskinan dan Pengangguran .....	38
3.3.2 Studi Pustaka .....	40
3.3.3 Kuisoner .....	40
3.4 Inovasi .....	53
3.5 Metodologi Pengembangan Perangkat Lunak .....	54
3.5.1 Memahami, Menentukan Konteks Penggunaan .....	54
3.5.2 Menentukan Kebutuhan Pengguna .....	54
3.5.3 Membuat Ekosistem .....	55
3.6 Membangun Desain (Perancangan Sistem) .....	55
3.6.1 Perancangan Arsitektur Sistem.....	55
3.6.2 Struktur Menu.....	56
3.6.3 Perancangan Model .....	57
3.6.3.1 Usecase Diagram .....	58
3.6.3.2 Activity Diagram .....	58
3.6.3.3 Class Diagram .....	63
3.6.3.4 Sequence Diagram.....	65
3.6.4 Perancangan Basis Data .....	67
3.6.4.1 Perancangan Tabel .....	67
3.6.4.2 Relasi Antar Tabel .....	73
3.6.5 Perancangan Antar Muka .....	74
3.6.5.1 Perancangan Antar Muka Pengguna .....	74
3.6.5.2 Perancangan Antar Muka Admin .....	80
3.7 Perancangan Evaluasi Aplikasi (Evaluate Design) .....	82
<b>BAB IV HASIL, PENGUJIAN DAN PEMBAHASAN.....</b>	<b>83</b>
4.1 Memahami, Menentukan Konteks Penggunaan .....	83
4.2 Menentukan Kebutuhan Pengguna .....	83
4.3 Ekosistem Pengguna.....	84
4.4 Hasil Perancangan Sistem .....	85
4.4.1 Halaman Utama .....	86
4.4.2 Halaman Beranda .....	90
4.4.3 Halaman Profil.....	95
4.4.4 Halaman Komunitas .....	98
4.4.5 Halaman Tentang Aplikasi .....	99
4.5 Web Service .....	100
4.6 Hasil Evaluasi Aplikasi .....	101
<b>BAB V PENUTUP .....</b>	<b>102</b>
5.1 Kesimpulan.....	102
5.2 Saran .....	102
DAFTAR PUSTAKA.....	103

## DAFTAR GAMBAR

Gambar 2.1 Karakteristik Metode Living Lab .....	12
Gambar 2.2 Alur Proses UCD .....	15
Gambar 2.3 Skema Web Service .....	20
Gambar 2.4 Blok Bangunan Web Service .....	21
Gambar 2.5 Arsitektur Web Service .....	23
Gambar 2.6 Ilustrasi SOA .....	25
Gambar 2.7 Arsitektur Restfull Web Service .....	27
Gambar 2.8 Skema Proses Request dan Response .....	27
Gambar 2.9 Format URI untuk Request API .....	28
Gambar 2.10 Contoh Format Data JSON.....	30
Gambar 2.11 Contoh Format Data XML .....	31
Gambar 2.12 Struktur Elemen HTML .....	32
Gambar 2.13 Struktur CSS .....	34
Gambar 3.1 Kerangka Kerja .....	37
Gambar 3.2 Peta Kemiskinan .....	39
Gambar 3.3 TPAK dan TPT menurut Kabupaten/kota di D.I. Yogyakarta .....	39
Gambar 3.4 Hasil Kuisioner Pertanyaan Nomor Satu.....	43
Gambar 3.5 Hasil Kuisioner Pertanyaan Nomor Dua .....	43
Gambar 3.6 Hasil Kuisioner Pertanyaan Nomor Tiga.....	44
Gambar 3.7 Hasil Kuisioner Pertanyaan Nomor Empat.....	44
Gambar 3.8 Hasil Kuisioner Pertanyaan Nomor Lima.....	45
Gambar 3.9 Hasil Kuisioner Pertanyaan Nomor Enam.....	45
Gambar 3.10 Hasil Kuisioner Pertanyaan Nomor Tujuh.....	45
Gambar 3.11 Hasil Kuisioner Pertanyaan Nomor Delapan.....	46
Gambar 3.12 Hasil Kuisioner Pertanyaan Nomor Sembilan.....	46
Gambar 3.13 Hasil Kuisioner Pertanyaan Nomor Sepuluh.....	47
Gambar 3.14 Hasil Kuisioner Pertanyaan Nomor Sebelas.....	47
Gambar 3.15 Hasil Kuisioner Pertanyaan Nomor Dua Belas .....	48
Gambar 3.16 Hasil Kuisioner Pertanyaan Nomor Tiga Belas.....	48
Gambar 3.17 Hasil Kuisioner Pertanyaan Nomor Empat Belas.....	49
Gambar 3.18 Hasil Kuisioner Pertanyaan Nomor Lima Belas.....	49
Gambar 3.19 Hasil Kuisioner Pertanyaan Nomor Enam Belas.....	50
Gambar 3.20 Hasil Kuisioner Pertanyaan Nomor Tujuh Belas.....	50
Gambar 3.21 Hasil Kuisioner Pertanyaan Nomor Delapan Belas.....	51
Gambar 3.22 Hasil Kuisioner Pertanyaan Nomor Sembilan Belas .....	51
Gambar 3.23 Hasil Kuisioner Pertanyaan Nomor Dua Puluh .....	52
Gambar 3.24 Hasil Kuisioner Pertanyaan Nomor Dua Puluh Satu.....	52
Gambar 3.25 Hasil Kuisioner Pertanyaan Nomor Dua Puluh Dua .....	53
Gambar 3.26 Arsitektur Sistem .....	56
Gambar 3.27 Struktur Menu.....	57
Gambar 3.28 Usecase Diagram .....	58
Gambar 3.29 Activity Diagram Berbagi Jasa .....	59
Gambar 3.30 Activity Diagram Merespon Jasa .....	60
Gambar 3.31 Activity Diagram Edit Jasa .....	61
Gambar 3.32 Activity Diagram Hapus Jasa .....	61
Gambar 3.33 Activity Diagram Login .....	62
Gambar 3.34 Activity Diagram Register .....	63

Gambar 3.35 Activity Diagram Logout .....	63
Gambar 3.36 Class Diagram .....	64
Gambar 3.37 Sequence Diagram Pengguna .....	66
Gambar 3.38 Sequence Diagram Admin.....	67
Gambar 3.39 RAT (Relasi Antar Tabel) .....	73
Gambar 3.40 Rancangan Antar Muka Register Pengguna .....	74
Gambar 3.41 Rancangan Antar Muka Login Pengguna .....	75
Gambar 3.42 Rancangan Antar Muka Formulir Postingan .....	76
Gambar 3.43 Rancangan Antar Muka Daftar Postingan .....	76
Gambar 3.44 Rancangan Antar Muka Edit Postingan .....	77
Gambar 3.45 Rancangan Antar Muka Edit Profil .....	78
Gambar 3.46 Rancangan Antar Muka Komunitas.....	79
Gambar 3.47 Rancangan Antar Muka Tentang Kami .....	79
Gambar 3.48 Rancangan Antar Muka Manajemen Postingan .....	80
Gambar 3.49 Rancangan Antar Muka Manajemen Komunitas.....	81
Gambar 3.50 Rancangan Antar Muka Manajemen Pengguna .....	81
Gambar 4.1 Ekosistem Pengguna .....	85
Gambar 4.2 Halaman Utama Login.....	86
Gambar 4.3 Halaman Utama Register .....	87
Gambar 4.4 Daftar Unggahan pada Halaman Utama .....	87
Gambar 4.5 Halaman Beranda untuk Formulir Unggahan .....	90
Gambar 4.6 Halaman Beranda untuk Daftar Unggahan dan Atributnya .....	90
Gambar 4.7 Halaman Utama untuk Notifikasi .....	91
Gambar 4.8 Halaman Utama untuk Edit Unggahan .....	91
Gambar 4.9 Halaman Profil untuk Daftar Unggahan .....	95
Gambar 4.10 Halaman Profil untuk Edit Profil .....	96
Gambar 4.11 Halaman Komunitas .....	98
Gambar 4.12 Halaman tentang Aplikasi .....	99

## DAFTAR TABEL

Tabel 2.1 Perbedaan SOAP dan REST .....	21
Tabel 3.1 Daftar Pertanyaan Kuisoner .....	41
Tabel 3.2 Lanjutan Daftar Pertanyaan Kuisoner .....	42
Tabel 3.3 Daftar Pertanyaan Kuisoner untuk Pemerintah .....	42
Tabel 3.4 Daftar Pertanyaan Kuisoner untuk Komunitas.....	42
Tabel 3.5 Hasil Pertanyaan Kuisoner untuk Pemerintah.....	53
Tabel 3.6 Hasil Pertanyaan Kuisoner untuk Komunitas .....	53
Tabel 3.7 Unggahan.....	68
Tabel 3.8 Pengguna .....	69
Tabel 3.9 Komentar .....	69
Tabel 3.10 Foto .....	70
Tabel 3.11 Komunitas .....	70
Tabel 3.12 Notifikasi .....	71
Tabel 3.13 Kabupaten .....	71
Tabel 3.14 Kecamatan .....	71
Tabel 3.15 Menu .....	72
Tabel 3.16 Submenu .....	72
Tabel 3.17 Kategori Unggahan .....	72
Tabel 3.18 Jumlah Suka .....	73
Tabel 3.19 Admin .....	82
Tabel 3.20 Perancangan Evaluasi .....	84
Tabel 4.1 Kebutuhan Fungsional Pengguna .....	84
Tabel 4.2 Daftar Akses Client Terhadap Server .....	100
Tabel 4.3 Hasil Evaluasi Aplikasi .....	101

## DAFTAR MODUL PROGRAM

Modul Program 4.1 Source code Halaman Utama .....	88
Modul Program 4.2 Lanjutan Source code Halaman Utama .....	89
Modul Program 4.3 Source code Halaman Beranda .....	91
Modul Program 4.4 Lanjutan Source code Halaman Beranda .....	92
Modul Program 4.5 Lanjutan Source code Halaman Beranda .....	93
Modul Program 4.6 Lanjutan Source code Halaman Beranda .....	94
Modul Program 4.7 Source code Halaman Profil .....	96
Modul Program 4.8 Lanjutan Source code Halaman Profil .....	97
Modul Program 4.9 Lanjutan Source code Halaman Profil .....	98
Modul Program 4.10 Source code Halaman Komunitas .....	99
Modul Program 4.11 Source code Halaman Tentang Aplikasi .....	100

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Upaya pemberantasan kemiskinan adalah salah satu prioritas pembangunan nasional yang dilakukan oleh pemerintah Indonesia. Sehubungan dengan hal tersebut, data angka persentase kemiskinan di D.I. Yogyakarta selalu tercatat sebagai yang tertinggi untuk Pulau Jawa menurut Badan Pusat Statistik Provinsi Daerah Istimewa Yogyakarta. Selain itu menurut data dari BPS Provinsi D.I. Yogyakarta Tingkat Pengangguran Terbuka (TPT) D.I Yogyakarta pada Agustus 2018 sebesar 3,35 persen, mengalami peningkatan 0,32 persen poin dibanding TPT Agustus 2017 sebesar 3,02 persen. (BPS, 2014-2018). Dengan melihat data yang ada tentu ada dorongan upaya agar kemiskinan dan tingkat pengangguran di Daerah Istimewa Yogyakarta ini menurun. Salah satu upaya yang mungkin bisa diterapkan adalah dengan meningkatkan kepedulian sosial untuk saling berbagi.

Dalam hal berbagi di Yogyakarta sebagai kota pendidikan yang kaya akan institusi pendidikannya, banyak organisasi kemahasiswaan melakukan bakti sosial sebagai implementasi tridarma perguruan tinggi. Kegiatan bakti sosial seperti berbagi pengetahuan akademis, bersih bersih lingkungan, melakukan cek kesehatan gratis dan lain sebagainya. Dari tahun 2017 hingga Juli 2019 sudah ada 87 kegiatan bakti sosial di wilayah Yogyakarta yang tercatat (TribunJogja, 2019). Tidak hanya dari kalangan mahasiswa tetapi di Yogyakarta juga banyak didapati komunitas dan lembaga kemasyarakatan yang bergerak untuk membantu sesama dari data yang ada di Dinas Sosial Yogyakarta ada 174 lembaga yang terdata (DinasSosialDIY, 2018). Selain itu perkembangan pengguna internet saat ini terus meningkat setiap tahunnya. Hal tersebut dapat dilihat dari data survei yang dilakukan oleh Asosiasi Penyelenggara Internet Indonesia (APJII). Dengan tingginya kesadaran masyarakat Yogyakarta tentang berbagi dan terus meningkatnya pengguna internet,

mendorong adanya suatu potensi besar terciptanya media untuk menjadi wadah bagi terselenggaranya proses berbagi jasa yang efektif dan efisien.

Di wilayah Yogyakarta media untuk kegiatan berbagi sendiri dirasa masih belum fokus dan belum terstruktur karena masih memanfaatkan media sosial. Media sosial yang berkembang sekarang dan pemanfaatannya sebagai sarana untuk bertukar informasi sangatlah tinggi (Fitriani, 2017). Salah satunya media sosial facebook, dimana banyak tercipta grup-grup yang memberikan informasi-informasi seputar masalah sosial. contohnya grup info cegatan jogja, di grup ini setiap harinya memberikan informasi dari berbagi aspek, oleh itu bisa dikatakan belum fokus dalam hal berbagi, hal tersebut membuat diperlukannya media yang benar benar fokus pada proses berbagi jasa atau barang. Untuk mempermudah masyarakat melakukan proses berbagi jasa yang efektif dan efisien diperlukan sistem yang cepat dan fleksibel. Peningkatan dalam elemen-elemen seperti kepercayaan, gotong royong, toleransi, penghargaan, saling memberi dan saling menerima serta kolaborasi sosial adalah salah satu bagian dari *smart people* yang termasuk dalam dimensi untuk menciptakan *smart city* (Insani, 2017). Maka dari itu berbagi juga termasuk salah satu bentuk mewujudkan *smart city*.

Yogyakarta sebagai daerah yang mengusung konsep *smart city* dan mendukung perkembangan ekonomi digital tentu mengedepankan pelayanan masyarakat yang maksimal. *Smart city* ditujukan dalam hal penggunaan informatika dan teknologi perkotaan untuk meningkatkan efisiensi pelayanan dengan mengintegrasikan beberapa teknologi informasi dan komunikasi (ICT) dan solusi *Internet of Things* (IOT) dalam sebuah bentuk yang aman untuk mengelola aset kota (Utomo & Hariadi, 2016) oleh karena itu teknologi *webservice* bisa menjadi teknologi yang bisa diterapkan untuk media berbagi. Menurut (Manes, 2001), *web-service* diartikan sebagai sepotong atau sebagian informasi atau proses yang dapat diakses oleh siapa saja, kapan saja dengan menggunakan piranti apa saja, tidak

terikat dengan sistem operasi atau bahasa pemrograman yang digunakan. Fungsi *web service* disini adalah untuk mempercepat proses informasi yang diunggah dan yang membutuhkannya. Cara kerja *webservis* ini sendiri yang pertama pada bagian aplikasi *server* berguna untuk menyediakan data yang digunakan oleh aplikasi *client*, sedangkan aplikasi *client* berguna untuk meminta data dari aplikasi *server* (Aziz, Wiharto, & Wicaksono, 2013).

Dari pemaparan masalah yang ditemukan dan analisis sementara yang menjadi acuan, maka solusi yang akan dilakukan pada penelitian ini adalah membuat media bernama Jogja Berbagi berbasis web agar lebih terfokus untuk berbagi dan mengefisiensikan proses dengan memanfaatkan teknologi *web-service* dengan sistem *client-server* dan arsitektur *restful*, untuk pertukaran datanya menggunakan format JSON sebagai sarana untuk berbagi jasa, bagi penyedia maupun yang membutuhkan jasa di wilayah Yogyakarta. Kategori jasa yang akan dicantumkan adalah jasa informasi, dan jasa dibidang pendidikan, kesehatan, sosial, budaya, peternakan, pertanian. Bidang bidang tersebut adalah bidang yang umum untuk berbagi jasa di wilayah Yogyakarta. Dengan cara ini diharapkan dapat memberikan kemudahan bagi masyarakat dan yang membutuhkannya.

## **1.2 Rumusan Masalah**

Dari permasalahan informasi berbagi jasa di wilayah Yogyakarta dapat disimpulkan rumusan masalah sebagai berikut :

1. Apakah media jogja berbagi dapat memberikan kemudahan sebagai sarana berbagi di wilayah Yogyakarta sesuai dengan kebutuhan.
2. Bagaimana kesesuaian pertukaran data ketika proses meminta dan respon data pada hubungan antara *client* dan *server*.

## **1.3 Manfaat Penelitian**

Adapun manfaat dari penelitian ini yaitu:

1. Memberikan kemudahan mencari informasi kesempatan berbagi jasa di wilayah Yogyakarta.

#### **1.4 Tujuan Penelitian**

Adapun tujuan dari penelitian ini yaitu :

1. Untuk membangun aplikasi Jogja Berbagi sebagai wadah pertukaran informasi berbagi jasa yang sesuai dengan kebutuhan pengguna.
2. Mengetahui implementasi metode *restful web service* sebagai sarana berbagi antar pengguna di dalam aplikasi Jogja Berbagi.

#### **1.5 Batasan Permasalahan**

Batasan masalah yang ditentukan adalah sebagai berikut :

1. Dilakukan dengan menerapkan pengembangan perangkat lunak UCD.
2. Penelitian ini hanya berfokus untuk membagi jasa dan meminta jasa di wilayah Yogyakarta.

#### **1.6 Metodologi Penelitian**

Metodologi yang digunakan dalam penelitian ini adalah menggunakan *living lab*. *Living lab* adalah ekosistem inovasi yang bersifat terbuka yang berpusat pada pengguna atau warga berdasarkan pendekatan *co-creation* pengguna secara sistematis, mengintegrasikan penelitian dan proses inovasi dalam kehidupan komunitas dan disetting atau dikondisikan sesuai kehidupan nyata atau sehari-hari warga (Indarto, 2019). Tahapan dalam penelitian dengan metode ini adalah sebagai berikut :

##### **1. Eksplorasi**

Dalam hal ini melibatkan semua pemangku kepentingan, terutama komunitas pengguna, pada tahap awal proses penciptaan bersama untuk menemukan skenario, penggunaan, dan perilaku yang muncul melalui skenario langsung di lingkungan nyata atau virtual. Termasuk dalam hal pengumpulan data dan studi pustaka.

## 2. Eksperimentasi

Tahap ini menerapkan artefak teknologi tingkat yang tepat untuk mengalami skenario langsung dengan sejumlah besar pengguna sambil mengumpulkan data yang akan dianalisis dalam konteksnya selama kegiatan evaluasi. Di tahap ini juga dilakukan pembuatan aplikasi jogja berbagi. Metode pengembangan perangkat lunak yang akan dipakai adalah UCD (*User Centered Design*). UCD menempatkan pengguna sebagai pusat dari sebuah proses pengembangan sistem. Teknik, metode, alat, prosedur dan proses yang membantu perancangan sistem interaktif dibangun berdasarkan pengalaman pengguna. UCD adalah menerjemahkan partisipasi dan pengalaman manusia ke dalam rancangan (Jagadish, 2014). Proses dari Metode User Centered Design (UCD) terdapat lima proses menurut (Palilingan, Santoso, & Rahayu, 2017) yaitu :

### a. *Plan the human-centered design process*

Pada Proses ini mengadakan diskusi terhadap orang-orang yang akan mengerjakan proyek, untuk mendapatkan komitmen bahwa proses pembangunan proyek adalah berpusat kepada pengguna atau *user*, Itu berarti bahwa proyek akan memiliki waktu dan tugas untuk melibatkan pengguna atau *user* dalam awal dan akhir proses atau di mana mereka dibutuhkan.

### b. *Understand Specifying the Context of Use*

Dasar dari setiap proses UCD adalah untuk memahami pengguna dari produk yang dimaksudkan dan lingkungan penggunaan mereka. Oleh karena itu, proses UCD khususnya dimulai dengan mengidentifikasi pengguna, Ini juga termasuk semua *stakeholders*, atau pengguna tidak langsung, semua yang berhubungan dengan sistem. Pada tahap ini juga, mengidentifikasi karakteristik pengguna dan kelompok pengguna,

### c. *Specifying the User Requirements*

Pada dasarnya pada tahap ini adalah tahap penggalan informasi atau data untuk mengumpulkan kebutuhan dari pengguna, kemudian setelah informasi/data telah terkumpul, dilakukanlah penataan informasi dari data kebutuhan pengguna tersebut, lalu kebutuhan pengguna digambarkan ke dalam berbagai bentuk atau teknik, seperti narasi, gambar, atau diagram.

### d. *Produce Design Solution*

Pada langkah ini, desain pertama diciptakan. Sketsa, simulasi dan bentuk lain dari prototipe yang digunakan untuk membuat ide-ide terlihat dan memfasilitasi komunikasi yang efisien dengan pengguna. Ketika solusi desain disajikan kepada pengguna, mereka juga harus diperbolehkan untuk melaksanakan tugas-tugas. Umpan balik pengguna yang dikumpulkan harus dimasukkan dalam perbaikan solusi desain.

### e. *Evaluating The Design*

Dalam kegiatan berikutnya, solusi desain yang pada tahap sebelumnya harus dievaluasi. Tujuannya adalah untuk menghasilkan umpan balik untuk lebih meningkatkan produk dan untuk menentukan apakah desain memenuhi kebutuhan pengguna yang ditentukan, tujuan kegunaan dan sesuai dengan pedoman kegunaan umum.

## 3. Evaluasi

Menilai ide-ide baru dan konsep-konsep inovatif serta artefak teknologi terkait dalam situasi kehidupan nyata melalui berbagai dimensi seperti aspek sosial-ergonomi, sosial-kognitif dan sosial-ekonomi; melakukan pengamatan pada potensi adopsi viral konsep baru dan artefak teknologi terkait melalui konfrontasi dengan model nilai pengguna.

### **1.7 Sistematika Penulisan**

Sistematika penulisan pada tugas akhir ini dilakukan dengan pengelompokan materi menjadi beberapa subbab sebagai berikut:

## **Bab I PENDAHULUAN**

Bab I menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian dan pengembangan sistem serta sistematika penulisan Tugas Akhir.

## **Bab II TINJAUAN PUSTAKA**

Bab II berisi tinjauan pustaka yang disusun sebagai tuntunan untuk memecahkan masalah penelitian dan untuk merumuskan hipotesis atau kesimpulan. Tinjauan didapat dari beberapa hasil penelitian terdahulu yang relevan dengan masalah yang diangkat di Tugas Akhir serta referensi studi pustaka untuk memperkuat landasan teori pada penelitian ini.

## **Bab III METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM**

Pada Bab III ini berisi langkah-langkah yang dilakukan dalam pelaksanaan penelitian dan pengembangan sistem. Tahapan penelitian terdiri atas analisis, data yang digunakan dan metode pengumpulan data. Sedangkan untuk tahapan pengembangan sistemnya terdiri atas analisis dan perancangan sistem yang dapat dilakukan dengan metode pengembangan perangkat lunak.

## **Bab IV HASIL, PENGUJIAN, DAN PEMBAHASAN**

Bab IV memuat tentang hasil penelitian dan pembahasan yang sesuai dengan apa yang sudah dikerjakan pada bab sebelumnya. Memuat tampilan sistem yang telah dibuat serta membahas tentang hasil pengujian yang telah dilakukan di sistem.

## **Bab V PENUTUP**

Bab V memuat kesimpulan dan saran berdasarkan sistem yang telah dibuat pada Tugas Akhir ini agar dapat menjadi pertimbangan bagi generasi berikut yang hendak mengembangkan sistem ini guna membuat sistem menjadi lebih baik lagi.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Smart City

Definisi atau pengertian *Smart City* sangat beraneka ragam. Konsep tersebut sudah dikenal dengan populer, namun dalam prakteknya digunakan di berbagai negara dengan istilah yang berbeda-beda dan situasi yang berbeda pula. Ada penggunaan berbagai jenis konsep mengganti *smart* dengan istilah kata sifat lainnya. Menurut Wikipedia, *smart city* adalah visi pembangunan perkotaan untuk mengintegrasikan beberapa teknologi informasi dan komunikasi (ICT) dan *solusi Internet of Things (IOT)* dalam sebuah bentuk yang aman untuk mengelola aset kota. Berdasarkan Wikipedia, definisi dari *Smart City* itu begitu luas mencakup berbagai macam keseluruhan teknologi digital yang dapat meningkatkan kualitas kehidupan, mengurangi biaya dan sumber konsumsi, dan dapat meningkatkan interaksi aktif antara kota dan warganya secara efektif (Utomo & Hariadi, 2016).

Menurut (Utomo & Hariadi, 2016) dalam menggunakan teknologi informasi kota pintar yang didalamnya ada masyarakat, klaster bisnis, aglomerasi perkotaan atau wilayah memiliki tiga karakteristik sebagai berikut:

1. Membuat lebih efisien penggunaan infrastruktur fisik (jalan, lingkungan dibangun dan aset fisik lainnya) melalui intelijen dan data buatan analisis untuk mendukung, pengembangan budaya yang kuat dan sehat ekonomi sosial.
2. Terlibat secara efektif dengan orang-orang lokal dalam pemerintahan lokal dan keputusan dengan menggunakan proses inovasi terbuka dan e-partisipasi, meningkatkan kecerdasan kolektif dari lembaga kota melalui *e-governance*, dengan penekanan pada partisipasi warga dan *co-desain*.

3. Belajar, beradaptasi dan berinovasi dan dengan demikian merespon lebih efektif dan segera untuk mengubah keadaan dengan meningkatkan kecerdasan kota.

Penelitian yang dilakukan oleh (Insani, 2017) yang berjudul Mewujudkan Kota Responsif Melalui *Smart City*. Dalam penelitian tersebut menyebutkan dimensi dimensi dari konsep *smart city*. Ada enam dimensi yang disebutkan, berikut penjelasannya,

#### 1. *Smart Government*

*Smart government* mengacu pada prinsip *Good Governance*. Kunci utama pemerintahan yang cerdas bertujuan untuk mengurangi kesenjangan di tingkat kota, kecamatan dan kelurahan adalah tidak hanya pemeratakan pembangunan fisik di setiap daerah, tetapi juga peningkatan profesionalisme kinerja aparatur yang responsif terhadap kebutuhan masyarakat dengan didukung oleh kecanggihan teknologi.

#### 2. *Smart Economy*

Seperti program pemberdayaan masyarakat melalui UMKM dan koperasi agar mendorong inovasi dan mengantisipasi persaingan usaha. Serta dapat menumbuhkembangkan rasa untuk berwirausaha.

#### 3. *Smart People*

Ditanamkannya nilai-nilai edukasi di dalam masyarakat dapat mendorong kehidupan sosial di perkotaan menjadi kondusif. Diantaranya elemen-elemen seperti kepercayaan, gotong royong, toleransi, penghargaan, saling memberi dan saling menerima serta kolaborasi sosial. Tata nilai ini perlu dipertahankan dalam kehidupan sosial masyarakat.

#### 4. *Smart Mobility*

Berkaitan dengan transportasi dan infrastruktur. Diharapkannya ada transportasi yang terpadu sehingga lebih efisien. Dengan ketersediaan sarana atau prasarana transportasi dan infrastruktur yang memadai, dapat meningkatkan pertumbuhan ekonomi dan kesejahteraan

masyarakat. Pengelolaan infrastruktur kota yang dikembangkan di masa depan merupakan sebuah sistem pengelolaan terpadu dan diorientasikan untuk kepentingan publik.

#### 5. *Smart Environment*

Dilihat dari segi penggunaan bangunan agar tidak berdampak pada kerusakan lingkungan serta cara mengelola sumber daya alamnya. Adanya kerusakan yang berdampak pada menurunnya mutu lingkungan pada dasarnya adalah akibat kelalaian atau kesengajaan yang dilakukan oleh masyarakat dan pemerintah.

#### 6. *Smart Living*

Kualitas hidup masyarakat dapat dilihat dari segi kesehatan dan keamanan pada lingkungannya. Sehingga dapat mewujudkan lingkungan yang kondusif dan berkualitas bagi masyarakatnya.

Salah satu dari pemaparan dimensi-dimensi tersebut ada yang berkaitan dengan masalah berbagi di D.I. Yogyakarta. Yaitu *smart people*, dengan peningkatan dalam elemen-elemen seperti kepercayaan, gotong royong, toleransi, penghargaan, saling memberi dan saling menerima serta kolaborasi sosial akan menjadi salah satu solusi permasalahan kemiskinan dan kepengangguran di D.I. Yogyakarta.

## 2.2 **Living Lab**

Berikut adalah pengertian metode *living lab* menurut beberapa tokoh yang ada pada artikel dari website (<https://u4iot.eu/>). *Living Lab* adalah orkestra proses inovasi terbuka yang berfokus pada penciptaan bersama inovasi dalam konteks dunia nyata dengan melibatkan banyak pemangku kepentingan dengan tujuan untuk menghasilkan nilai berkelanjutan bagi semua pemangku kepentingan yang berfokus terutama pada pengguna akhir. (Anna Ståhlbröst, Botnia Living Lab, 2012). *Living Lab* adalah organisasi multi-stakeholder yang dibentuk untuk melaksanakan proyek-proyek inovasi yang mengikuti prinsip-prinsip inovasi terbuka dan pengguna dan fokus pada eksperimen kehidupan nyata

(Dimitri Schuurman, imec.livinglabs, 2015). *Living Labs* didefinisikan sebagai ekosistem inovasi yang bersifat terbuka yang berpusat pada pengguna/warga berdasarkan pendekatan co-creation pengguna secara sistematis, mengintegrasikan penelitian dan proses inovasi dalam kehidupan komunitas dan disetting/dikondisikan sesuai kehidupan nyata/sehari-hari warga. (Indarto, 2019).

Menurut (Indarto, 2019) *Living Lab* adalah ekosistem inovasi terbuka yang didorong oleh pengguna yang didasarkan pada kemitraan bisnis - warga negara - pemerintah yang memungkinkan pengguna untuk mengambil bagian aktif dalam penelitian, pengembangan, dan proses inovasi:

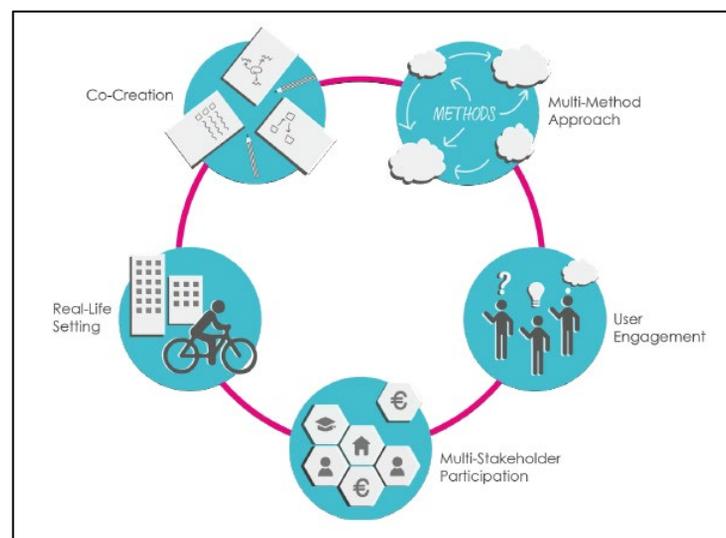
1. Membawa pengguna lebih awal ke dalam proses kreatif untuk lebih baik menemukan perilaku dan pola pengguna yang baru dan muncul;
2. Menjembatani kesenjangan inovasi antara pengembangan teknologi dan penyerapan produk dan layanan baru yang melibatkan semua pemain yang relevan dari jaringan nilai melalui kemitraan antara bisnis, warga negara, dan pemerintah;
3. Memungkinkan penilaian awal implikasi sosial-ekonomi solusi teknologi baru dengan menunjukkan validitas layanan inovatif dan model bisnis.

Manfaat *Living Lab* untuk pengguna dalam peran mereka sebagai warga negara dan masyarakat:

1. Warga diberdayakan untuk mempengaruhi pengembangan layanan dan produk yang melayani kebutuhan nyata, dan untuk bersama-sama berkontribusi pada investasi dan proses yang ditingkatkan melalui partisipasi aktif dalam R & D dan siklus hidup inovasi.
2. Untuk UKM, termasuk wirausaha mikro sebagai penyedia: mengembangkan, memvalidasi dan mengintegrasikan ide-ide baru dan dengan cepat meningkatkan layanan dan produk lokal mereka ke pasar lain.

3. Untuk perusahaan yang lebih besar: membuat proses inovasi lebih efektif dengan bermitra dengan perusahaan lain serta pengguna akhir, yang berakar pada pengalaman pengguna aktif.
4. Untuk pelaku penelitian, ekonomi dan masyarakat: Merangsang kemitraan bisnis-warga-pemerintah sebagai layanan fleksibel dan ekosistem inovasi teknologi; mengintegrasikan inovasi teknologi dan sosial dalam budaya beta yang inovatif.

Living memiliki lima karakteristik berikut penjelasannya.



**Gambar 2.1.** Karakteristik Metode *Living Lab*

### 1. *Multi-method approaches*

Artinya tidak ada satu pun Metodologi *Living Lab* yang berdiri sendiri, tetapi semua *Living Lab* menggabungkan dan menyesuaikan keberbedaan yang berpusat pada pengguna, metodologi penciptaan bersama yang paling sesuai tujuan mereka.

### 2. *User engagement*

Kunci kesuksesan dalam aktivitas apa pun dalam tahapan *Living Lab* sudah melibatkan pengguna di awal proses.

### 3. *Multi-stakeholder participation*

Fokusnya adalah pada pengguna, melibatkan semua yang relevan pemangku kepentingan sangat penting. Ini termasuk semua aktor misalkan perwakilan sektor publik dan swasta, akademisi dan orang-orang.

#### 4. *Real-life setting*

Karakteristik yang sangat spesifik *Living Labs* adalah kegiatan yang dilakukan tempat dalam pengaturan kehidupan nyata untuk mendapatkan menyeluruh ikhtisar konteks.

#### 5. *Co-Creations*

Khususnya dalam teknologi proyek, kegiatan dirancang sebagai eksperimen *top-down*, mendapat manfaat dari pengguna yang terlibat sebagai faktor daripada aktor. Ada pengakuan yang meningkat bahwa ini perlu diubah sehingga pengguna menjadi kontributor dan pencipta yang sama daripada subyek penelitian. Kehidupan pendekatan lab berusaha untuk saling menghargai hasil yang merupakan hasil dari semua pemangku kepentingan terlibat aktif dalam proses dari awal.

Tahapan dari metodologi *Living Lab* menurut artikel dari website (<https://u4iot.eu/>) ada tiga tahapan, berikut penjelasannya.

##### 1. Eksplorasi

Dalam hal ini melibatkan semua pemangku kepentingan, terutama komunitas pengguna, pada tahap awal proses penciptaan bersama untuk menemukan skenario, penggunaan, dan perilaku yang muncul melalui skenario langsung di lingkungan nyata atau virtual. Termasuk dalam hal pengumpulan data dan studi pustaka.

##### 2. Eksperimen

Tahap ini menerapkan artefak teknologi tingkat yang tepat untuk mengalami skenario langsung dengan sejumlah besar pengguna sambil mengumpulkan data yang akan dianalisis dalam konteksnya selama kegiatan evaluasi. Di tahap ini juga dilakukan pembuatan aplikasi

jogja berbagi. Metode pengembangan perangkat lunak yang akan dipakai adalah UCD (*User Centered Design*).

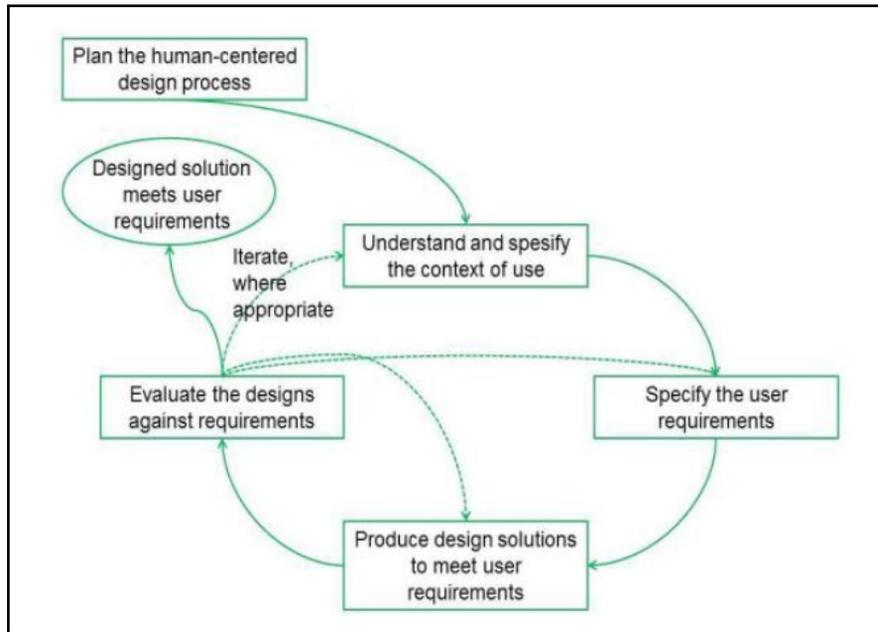
### 3. Evaluasi

Menilai ide-ide baru dan konsep-konsep inovatif serta artefak teknologi terkait dalam situasi kehidupan nyata melalui berbagai dimensi seperti aspek sosial-ergonomi, sosial-kognitif dan sosial-ekonomi; melakukan pengamatan pada potensi adopsi viral konsep baru dan artefak teknologi terkait melalui konfrontasi dengan model nilai pengguna.

### 2.3 User Center Designt (UCD)

Salah satu metode pengembangan perangkat lunak yang akan di pakai pada penelitian ini adalah UCD (*User Center Design*). Menurut (Palilingan, Santoso, & Rahayu, 2017) metode ini adalah paradigma baru dalam pengembangan aplikasi. UCD adalah istilah yang digunakan untuk menggambarkan filosofi perancangan. Konsep dari UCD adalah pengguna sebagai pusat dari proses pengembangan sistem, tujuan sistem, konteks sistem dan lingkungan sistem semua didasarkan dari pengguna atau *user*. UCD adalah proses yang interaktif dimana langkah perancangan dan evaluasi dibuat didalam permulaan proyek sampai implemenasi.

*User Centered Design (UCD)* adalah sebuah filosofi perancangan yang menempatkan pengguna sebagai pusat dari sebuah proses pengembangan sistem. Teknik, metode, alat, prosedur dan proses yang membantu perancangan sistem interaktif dibangun berdasarkan pengalaman pengguna. UCD adalah menerjemahkan partisipasi dan pengalaman manusia ke dalam rancangan (Palilingan, Santoso, & Rahayu, 2017).



**Gambar 2.2.** Alur Proses UCD

Proses dari Metode *User Centered Design* (UCD) terdapat lima proses yaitu :

a. *Plan the human-centered design process*

Pada Proses ini mengadakan diskusi terhadap orang-orang yang akan mengerjakan proyek, untuk mendapatkan komitmen bahwa proses pembangunan proyek adalah berpusat kepada pengguna atau *user*, Itu berarti bahwa proyek akan memiliki waktu dan tugas untuk melibatkan pengguna atau *user* dalam awal dan akhir proses atau di mana mereka dibutuhkan.

b. *Understand Specifying the Context of Use*

Dasar dari setiap proses UCD adalah untuk memahami pengguna dari produk yang dimaksudkan dan lingkungan penggunaan mereka. Oleh karena itu, proses UCD khususnya dimulai dengan mengidentifikasi pengguna, Ini juga termasuk semua *stakeholders*, atau pengguna tidak langsung, semua yang berhubungan dengan sistem. Pada tahap ini juga, mengidentifikasi karakteristik pengguna dan kelompok pengguna,

### c. *Specifying the User Requirements*

Pada dasarnya pada tahap ini adalah tahap penggalan informasi atau data untuk mengumpulkan kebutuhan dari pengguna, kemudian setelah informasi/data telah terkumpul, dilakukanlah penataan informasi dari data kebutuhan pengguna tersebut, lalu kebutuhan pengguna digambarkan ke dalam berbagai bentuk atau teknik, seperti narasi, gambar, atau diagram.

### d. *Produce Design Solution*

Pada langkah ini, desain pertama diciptakan. Sketsa, simulasi dan bentuk lain dari prototipe yang digunakan untuk membuat ide-ide terlihat dan memfasilitasi komunikasi yang efisien dengan pengguna. Ketika solusi desain disajikan kepada pengguna, mereka juga harus diperbolehkan untuk melaksanakan tugas-tugas. Umpan balik pengguna yang dikumpulkan harus dimasukkan dalam perbaikan solusi desain.

### e. *Evaluating The Design*

Dalam kegiatan berikutnya, solusi desain yang pada tahap sebelumnya harus dievaluasi. Tujuannya adalah untuk menghasilkan umpan balik untuk lebih meningkatkan produk dan untuk menentukan apakah desain memenuhi kebutuhan pengguna yang ditentukan, tujuan kegunaan dan sesuai dengan pedoman kegunaan umum.

## **2.4 Berbagi**

Berbagi adalah pemakaian secara bersama atas sumber daya atau ruang. Dalam arti sempit merujuk pada sebuah penggabungan penggunaan secara baik alternatif terbatas atau inheren, dapat kita amati dalam aktivitas manusia atau yang berlaku secara alami. Ketika sebuah organisme memerlukan gizi atau oksigen misalnya, organ ini juga untuk membagi dan mendistribusikan energi yang diambil sebagai pasokan pada bagian tubuh yang memerlukannya sebagaimana sekuntum bunga membagi dan mendistribusikan bibit. (Wikipedia, 2019)

Manusia adalah makhluk sosial, jadi manusia saling membutuhkan satu sama lain, kita membutuhkan orang lain, dan orang lain membutuhkan kita juga, karena hal itu kita harus berbagi dan orang lain akan berbagi kepada kita juga. Berbagi kepada sesama adalah hal penting, karena tanpa berbagi kita sebagai manusia hilang arah dan arti dari makhluk sosial itu sendiri. Kita sebagai makhluk sosial sudah kewajiban kita untuk berbagi, apapun yang dapat kalian bagi, tidak perlu uang, atau sesuatu yang kalian tidak punya, *berbagilah yang kalian punya*, berbagi juga tidak perlu kita mendatangi semua tempat orang yang kesusahan, tetapi berbagi di sekeliling kita, pasti sekeliling kita masih banyak orang yang membutuhkan, tidak hanya finansial, bisa juga dari mental, seperti membagikan Cinta kasih kepada orang lain yang terlihat sangat down, patah semangat, putus asa, bisa juga kita memperhatikan orang tua di sekitar kita yang kurang perhatian, bantu mereka dengan menyebrang jalan, atau membersihkan dan mengangkat sesesuatu yang berat saat mereka mau pindahan, atau sedang dalam kesulitan. (Wikipedia, 2019)

## **2.5 Jasa**

Jasa sering dipandang sebagai suatu fenomena yang rumit. Kata “Jasa” berarti pelayanan pribadi, pelayanan suatu produk. Beberapa pengertian jasa yang lain adalah sebagai berikut:

1. Layanan adalah kegiatan atau serangkaian kegiatan yang terjadi dalam interaksi dengan penghubung atau mesin fisik dan yang memberikan kepuasan konsumen (Lehtinen, 1983).
2. Layanan adalah segala kegiatan yang menguntungkan yang dapat ditawarkan salah satu pihak kepada pihak lain yang pada dasarnya tidak berwujud dan tidak mengakibatkan kepemilikan apa pun. Produksinya mungkin terkait atau tidak dengan produk fisik (Kotler, 1988).

Klasifikasi jasa ada Jasa Bisnis, Jasa Komunikasi, Jasa Konstruksi dan Jasa Teknik, Jasa Distribusi, Jasa Pendidikan, Jasa Lingkungan Hidup, Jasa Keuangan, Jasa Kesehatan

and Jasa Sosial, Jasa Parawisata dan Jasa Perjalanan, Jasa Rekreasi, Budaya, dan Olahraga, Jasa Transportasi, Lain-lain. Karakteristik jasa ada 3 menurut artikel di *academia.edu* oleh (Diantoro, 2011) yaitu,

1. *Intagibility* (tidak berwujud). Jasa tidak dapat dilihat, dirasa, diraba, didengar atau dicium sebelum jasa itu dibeli.
2. *Unstorability* (tidak dapat disimpan). Jasa tidak mengenal persediaan atau penyimpanan dari produk yang telah dihasilkan.
3. *Customization* (kustomisasi). Jasa sering kali didesain khusus untuk memenuhi kebutuhan pelanggan.

## 2.6 Media

Kata “Media” berasal dari bahasa latin yang merupakan bentuk jamak dari “medium”, secara harfiah berarti perantara atau pengantar. *Association for Education and Communication Technology* (AECT), mengartikan kata media sebagai segala bentuk dan saluran yang dipergunakan untuk proses informasi. *National Education Association* (NEA) mendefinisikan media sebagai segala benda yang dapat dimanipulasikan, dilihat, didengar, dibaca atau dibicarakan beserta instrumen yang dipergunakan untuk kegiatan tersebut. Sedangkan (Heinich, dkk, 1982) mengartikan istilah media sebagai “istilah ini merujuk pada apa pun yang membawa informasi antara sumber dan penerima”. (Nurseto, 2011).

## 2.7 WEB

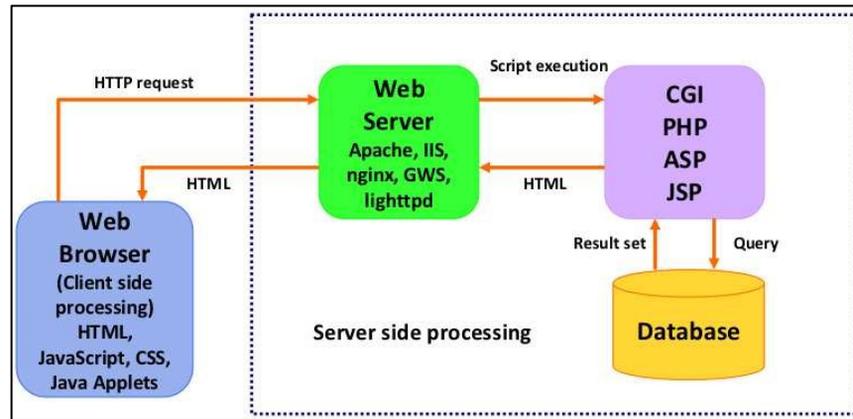
Www atau *world wide web* atau web saja merupakan sebuah sistem yang saling terkait dalam sebuah dokumen yang berformat *hypertext* yang berisi beragam informasi, baik tulisan, gambar, suara, video, dan informasi multimedia lainnya dan dapat diakses melalui sebuah perangkat yang disebut *web browser*. Untuk menterjemahkan dokumen dalam bentuk *hypertext* ke dalam bentuk dokumen yang bisa dipahami, maka *web browser* melalui *web client* akan membaca halaman web yang tersimpan di sebuah *web server*

melalui protocol yang biasa disebut http atau *Hypertext Transfer Protocol*. (Sholikhin & Riasti, 2013).

## 2.8 Web Service

*Web service* merupakan kumpulan suatu layanan berbasis web dengan menggunakan jaringan protokol HTTP (Fielding & Reschke, 2014), layanan tersebut dapat diakses dan dimanfaatkan oleh pengguna dengan bahasa pemrograman, arsitektur dan sistem operasi yang berbeda (*interoperability*) (Fielding & Taylor, 2002). *Web service* harus bersifat *stateless*, sehingga dapat dibaca dan di akses multi-*platform* (Paik dkk, 2017). Arsitektur pada web service dapat dibangun menggunakan *Simple Object Access Protocol* (SOAP) atau *REpresentational State Transfer* (REST), sementara *service* tersebut direpresentasikan dalam bentuk format teks, JSON atau XML (Gottschalk dkk, 2002). Sekalipun mirip dengan *Application Programming Interface* (API) berbasis web, *web service* lebih unggul karena dapat dipanggil dari jarak jauh melalui internet. Pemanggilan *web service* bisa menggunakan bahasa pemrograman apa saja dan dalam platform apa saja, sementara API hanya bisa digunakan dalam *platform* tertentu (Lucky, 2008).

Dalam sistem *web service*, untuk aplikasi web yang bekerja disisi server (*server side*), dibutuhkan adanya web server untuk mengirimkan hasil proses berupa halaman web ke *client*. Dimana halaman web ini dikirim menggunakan protokol http. Web server pulalah yang menerima permintaan dari *client*. Berikut skema dari *web service* menurut Karim (2012) dapat dilihat pada Gambar 2.3.



**Gambar 2.3.** Skema Web Service

Web server juga berfungsi untuk memetakan lokasi-lokasi berkas aplikasi web dalam komputer server agar bisa diakses oleh *client*. Termasuk mengatur apa web bisa diakses dari jaringan global atau hanya jaringan tertentu saja yang diijinkan untuk mengaksesnya. Sedangkan pada sisi *client* merupakan aplikasi dengan *platform* tertentu yang mengakses web server (Karim, 2012).

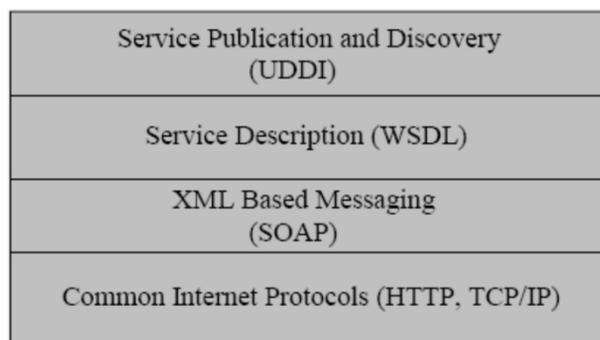
Penelitian yang dilakukan oleh (Barlian & Ermatita, 2017) yang berjudul *A Web of Things dengan RESTful protokol pada Smart City*. Dalam penelitian ini dijelaskan tentang perbedaan antarmuka antara *client* dan server. Pemisahan ini berarti, klien tidak harus mengetahui sistem penyimpanan data internal untuk setiap server, sehingga portabilitas *code* klien semakin tinggi. Begitu pula sebaliknya, server tidak peduli dengan *user interface* atau *user state*, sehingga server bisa lebih sederhana dan lebih *scalable*. Server dan klien juga dapat diganti dan dikembangkan secara independen selama antarmuka tidak diubah.

Hubungan antara *client* dengan server pada Gambar 2.1 menunjukkan adanya arsitektur pada *web service*. Arsitektur *web service* yang sering digunakan sekarang adalah SOAP dan REST. Perbedaan SOAP dan REST dapat dilihat pada Tabel 2.1.

**Tabel 2.1.** Perbedaan SOAP dan REST (Rizal & Rahmatulloh, 2019)

Uraian	SOAP	REST
Protokol komunikasi	HTTP, HTTPS, SMTP, FTP	HTTP, HTTPS
Penggunaan <i>bandwidth</i>	Dalam jumlah request yang banyak, relatif boros <i>bandwidth</i> . Hal ini karena banyaknya markup dalam penulisan format XML	Relatif hemat <i>bandwidth</i> , karena markup-markup ekstra seperti pada XML tidak dipakai
Tren penggunaan	Banyak mulai beralih ke REST, meski masih tetap ada yang mempertahankan, misalnya untuk integrasi aplikasi ke sistem legasi pada sebuah perusahaan.	Mulai populer, banyak dipakai oleh penyedia web servis terkemuka, seperti twitter, yahoo!, flickr, bloglines, technorati, google, amazon, eBay, dsb
Aturan penulisan	Ketat, mengikuti spesifikasi XML (SOAP v1.2)	Tidak ada spesifikasi khusus
Format respon	XML dengan spesifikasi SOAP. Agak sulit bagi kita untuk membaca langsung dan memahaminya.	XML, JSON, atau format plain teks lainnya. Hal ini memudahkan penerima respon membaca dan memahaminya.
<i>Attachment file</i>	Bisa (karena dapat mengembalikan respon dalam format binary)	Tidak bisa
Sifat web service pada umumnya	Tertutup, lebih ditujukan untuk vendor atau perusahaan tertentu	Terbuka, bisa diakses siapa saja
<i>Caching web</i>	Relatif sulit Standar lama (XML, HTTP)	Mudah, karena menggunakan URI
Penggunaan standart	Standar lama (XML, HTTP) dan baru (SOAP) digunakan bersamaan	Standar yang sudah ada, seperti XML dan HTTP
<i>Tools</i> Pengembangan	Banyak, baik komersial maupun <i>opensource</i>	Beberapa, karena tidak begitu dibutuhkan
<i>Tools</i> Manajemen	Perlu, bahkan kadang harganya mahal	Menggunaan <i>tools</i> yang sudah ada disistem jaringan
Ekstensibel	Bisa banyak ekstensi termasuk standar WS	Relatif tidak ekstensibel
Kemudahan implemetasi	Mudah jika sudah mempunyai lingkungan berbasis SOAP	Mudah

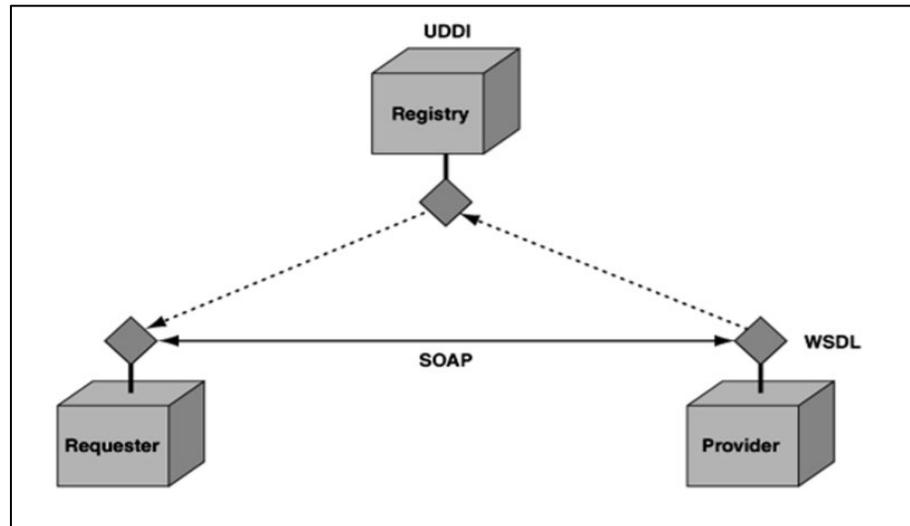
Gambar 2.4 merupakan blok bangunan *web service* yang mana menyediakan fasilitas komunikasi jarak jauh antara dua aplikasi yang merupakan layer arsitektur *web service* (Tidwell, 2001).

**Gambar 2.4.** Blok Bangunan Web Service

- a. Layer 1: protokol internet standar yang digunakan sebagai sarana transportasi adalah HTTP dan TCP/IP.
- b. Layer 2: *Simple Object Access Protocol* (SOAP) atau *REpresentational State Transfer* (REST) berbasis XML dan digunakan untuk pertukaran informasi antar sekelompok layanan.
- c. Layer 3: *Web service Definition Language* (WSDL) digunakan untuk mendiskripsikan atribut layanan.
- d. Layer 4: *Universal Description, Discovery and Integration*, yang mana merupakan direktori pusat untuk deskripsi layanan.

Penelitian yang dilakukan oleh (Hidayat & Ashari, 2013) yang berjudul Penerapan Teknologi Web Service Untuk Integrasi Layanan Puskesmas dan Rumah Sakit. Dalam penelitian ini di jelaskan bahwa konsep arsitektur yang mendasari teknologi Web service adalah *Service Oriented Architecure* (SOA), SOA mendefinisikan tiga peran berbeda yang menunjukkan peran dari masing-masing komponen dalam system, yaitu:

- a. *Service provider*, yaitu suatu entitas yang menyediakan *interface* terhadap sistem yang menjalankan suatu sekumpulan tugas tertentu.
- b. *Service requestor*, yaitu suatu entitas yang meminta/memperoleh (dan menemukan) *software service* dalam rangka meyelesai kan suatu tugas tertentu atau menyediakan solusi bisnis tertentu.
- c. *Service registry*, yaitu entitas yang bertindak sebagai penyimpan (*repository*) suatu *software service* yang dipublikasikan oleh *Service provider*. Gambaran sederhana dari arsitektur *web service* menurut Hidayat dan Ashari (2013) dapat dilihat pada Gambar 2.5.



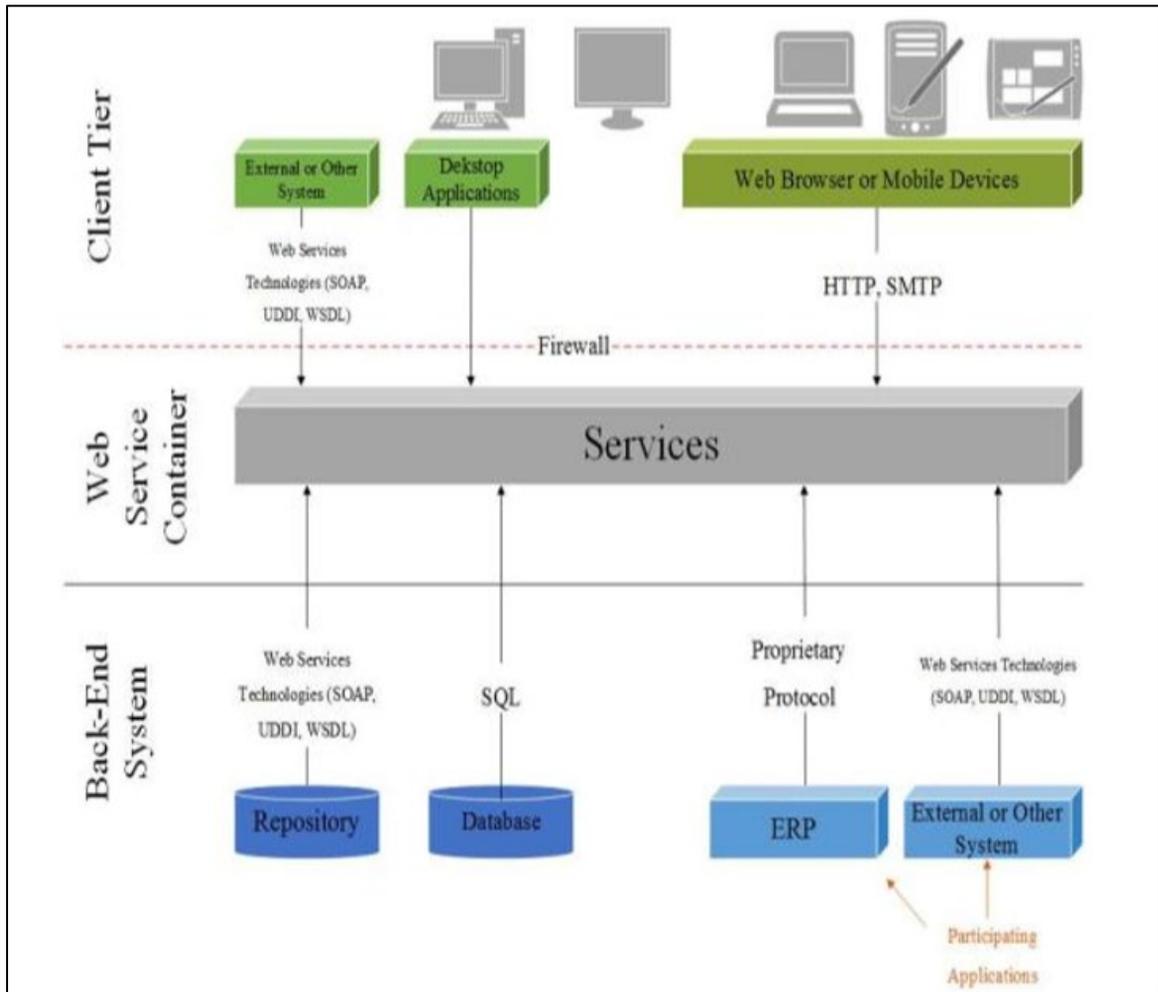
**Gambar 2.5.** Arsitektur *Web Service*

## 2.9 SOA (Service-Oriented Architecture)

Menurut Thomas Erl (2005) *Service Oriented Architecture* (SOA) adalah bagian utama dari service computing platform yang membawa konsep, teknologi, dan tantangan baru baru. Menurut Thomas Erl ada tiga hal penting yang menjadikan sebuah infrastruktur dapat disebut sebagai *Service Oriented Architecture*, yaitu logika bisnis yang dienkapsulasi sebagai *service*, dan proses komunikasi antar *service* dengan menggunakan *message*. Dalam hal ini, *service layer* akan menjembatani hubungan antara *business logic* dan *application logic*. Menurut Kunal Mittal (2007), *Service Oriented Architecture* adalah sebuah kumpulan yang terdiri atas *tools*, teknologi, *framework*, dan *best practice* yang memudahkan implementasi sebuah *service* secara cepat. Proses dalam mengimplementasi SOA menggunakan metodologi yang mengidentifikasi *service* yang dapat dipergunakan kembali (*reusable*) dalam aplikasi dan organisasi suatu perusahaan. Dengan demikian, SOA adalah suatu ide, bukan merupakan teknologi, produk, ataupun standar. Arsitektur SOA difokuskan untuk mengidentifikasi, membangun, mengubah, dan memelihara proses bisnis suatu perusahaan sebagai sekumpulan *service*. Teknologi yang menggunakan SOA digunakan untuk mengurangi kompleksitas dalam membangun sebuah aplikasi atau

*software*. SOA dapat mengantisipasi isu mengenai penggunaan *software* yang terdistribusi, penggunaan *platform* yang berbeda, dan integrasi aplikasi. Menurut Ricardo LaRosa (2008) *Service Oriented Architecture* merupakan sebuah konsep yang sederhana. SOA adalah blueprint yang dapat membantu layanan-layanan dapat beroperasi secara bersama-sama. Layanan yang dimaksud adalah sebuah unit *software* yang mendukung fungsifungsi bisnis agar dapat bekerja secara maksimal.

Dari penelitian yang dilakukan oleh Pakusadewa dan Sudaryanto (2015) dikemukakan bahwa SOA adalah suatu cara mengorganisir perangkat lunak sehingga organisasi dapat dengan cepat merespon perubahan kebutuhan. Teknologi tersebut berdasarkan *service* (layanan), yang terdiri dari unit-unit berdasarkan kebutuhan dari perangkat lunak yang berjalan pada jaringan. *Service* sendiri merupakan komponen umum yang digunakan oleh beberapa sistem aplikasi (*reusable*). *Service* dapat berupa modul program, aplikasi, atau gabungan dari beberapa aplikasi yang berhubungan. SOA merepresentasikan suatu model yang mana fungsi-fungsi dibagi menjadi beberapa unit-unit terpisah yang lebih kecil, yang dapat didistribusikan melalui jaringan dan dapat dikombinasikan dan digunakan secara bersama-sama untuk menciptakan aplikasi. *Service-service* tersebut berkomunikasi satu sama lain dengan cara mengirim data dari satu *service* ke *service* lainnya, atau dengan mengkoordinasikan suatu aktivitas antara dua atau lebih *service*. Sehingga SOA memungkinkan *service* yang *interoperable*, yang berarti *service* tersebut dapat berkomunikasi satu sama lain, meskipun pada implementasinya dibuat dengan bahasa pemrograman yang berbeda atau diakses melalui transport protocol yang berbeda yang memungkinkan pengintegrasian aset-aset sistem aplikasi dari suatu perusahaan. Ilustrasi konsep dari *Service Oriented Architecture* (SOA) dapat dilihat pada Gambar 2.6.



**Gambar 2.6.** Ilustrasi SOA (Thomas Erl, SOA: *principles of service design* 2008)

## 2.10 RESTful

*Representational State Transfer* yang disingkat REST yang merupakan gaya arsitektur untuk penerapan web *service* dalam menerapkan konsep perpindahan antar *state* (Fielding, 2000). REST populer karena kesederhanaannya, sebagai lawan menciptakan standar baru, kerangka kerja dan teknologi. Keuntungan REST adalah interaksi berbasis REST menggunakan *Hypertext Transfer Protocol* (HTTP) internet yang merupakan hal yang sudah umum digunakan. Contoh dari pengaturan ini adalah interaksi berbasis REST semua berkomunikasi status mereka menggunakan kode status HTTP standar. Jadi, 404 berarti sumber daya yang diminta tidak ditemukan, kode 401 berarti permintaan tidak

ditorisasi, kode 200 berarti semuanya OK dan 500 berarti ada kesalahan aplikasi yang tidak dapat dipulihkan pada server (Rizal & Rahmatulloh, 2019).

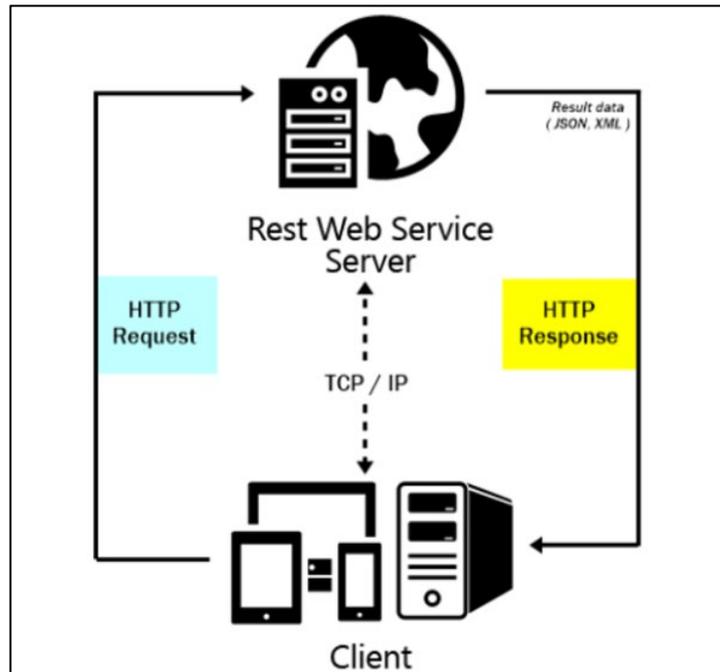
Demikian pula detail seperti enkripsi dan integritas transport data diselesaikan bukan dengan menambahkan kerangka kerja atau teknologi baru, tetapi dengan mengandalkan enkripsi *Secure Sockets Layer* (SSL) dan *Transport LayerSecurity* (TLS) yang terkenal. Jadi, seluruh arsitektur REST dibangun di atas konsep yang sudah dikenal sebagian besar pengembang (Rizal & Rahmatulloh, 2019).

Namun manfaat REST menggunakan konstruk HTTP juga menciptakan batasan. Banyak keterbatasan HTTP juga berubah menjadi kekurangan gaya arsitektur REST. Misalnya, HTTP tidak menyimpan informasi berbasis negara di antara siklus permintaan-respons, yang berarti aplikasi berbasis REST harus tanpa kewarganegaraan dan tugas manajemen negara apa pun harus dilakukan oleh klien (Su & Chiang, 2012).

Metode REST telah dikembangkan dan didasari oleh empat prinsip utama teknologi (Pautasso, 2008), yaitu:

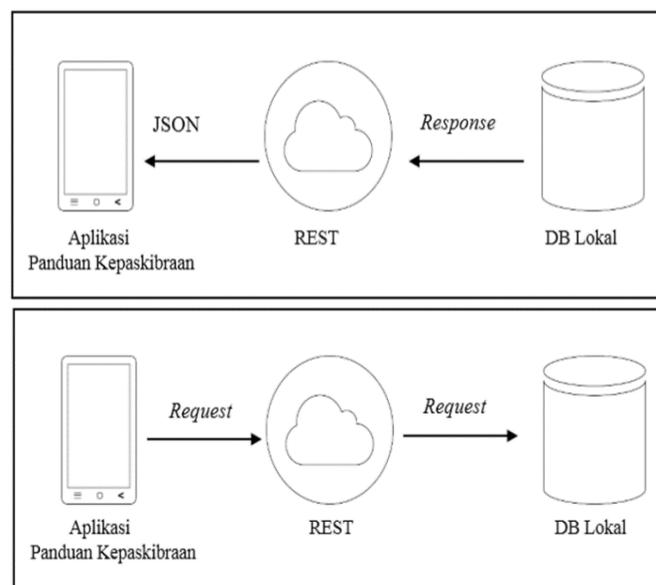
1. *Resource identifier through Uniform Resource Identifier* (URI),
2. *Uniform interface* (sumber daya CRUD menggunakan operasi PUT, GET, POST, dan DELETE),
3. *Self-descriptive messages* (sumber daya tidak terikat sehingga dapat mengakses konten HTML, XML, PDF, JPEG, plain text, meta data, dll),
4. *Stateful interactions through hyperlinks* (bersifat stateless)

Gambar 2.7 menjelaskan tentang alur komunikasi sebuah RESTful *web service*. *Client* akan melakukan *request* terhadap *web service server*, kemudian *server* akan memberikan respon berupa *result* data JSON maupun XML.



**Gambar 2.7.** Arsitektur *Restfull Web Service*

Dari gambar 2.7 bisa dilihat bahwa REST menggunakan arsitektur *client-server* dimana ketika *client* mengirimkan permintaan kepada server, server akan segera memroses permintaan tersebut dan memberikan respon kembali ke *client*. Penelitian yang dilakukan oleh Rulloh dan kawan-kawan (2017) menjelaskan bahwa Skema untuk cara kerja REST dalam penelitiannya tentang pembuatan Aplikasi Panduan Kepaskibraan ditampilkan dalam gambar 2.8 berikut.



**Gambar 2.8.** Skema Proses *Request* dan *Response*

Penelitian yang dilakukan oleh Kurniawan dan kawan-kawan (2013) yang berjudul Implementasi *Rest – API* untuk Portal Akademik UKDW Berbasis Android. Dalam penelitian ini dijelaskan mengenai format URI untuk *request* pada server dalam *restful web service*. Format URI *request* server dalam *restful web service* dapat dilihat pada Gambar 2.9.

```
http://{nama_domain}/{sub_domain}/{nama_fungsi}?{parameter_1}...
&{parameter_n}
```

**Gambar 2.9.** Format URI untuk *Request* API

Keterangan dari Gambar 2.9. untuk format {nama\_domain} adalah nama domain letak API berada. {sub\_domain} adalah folder sub domain letak API berada. {nama\_fungsi} adalah nama fungsi yang akan diakses. Dan format {parameter\_1}...&{parameter\_n} adalah parameter-parameter yang dikirimkan.

## 2.11 Integrasi Data

Integrasi aplikasi *enterprise* adalah pendekatan strategis dalam menggabungkan sistem informasi bersama-sama dan dapat mendukung kemampuan untuk bertukar informasi atau data secara langsung. Integrasi aplikasi merupakan proses integrasi dari beberapa aplikasi yang bertujuan untuk berbagi data dan proses yang dapat dilakukan tanpa merubah aplikasi-aplikasi yang telah ada (Puustjarvi, 2010).

## 2.12 JSON dan XML

JSON dan XML adalah Jenis format serialisasi data yang digunakan dalam *web service*. Salah satu faktor yang mempengaruhi kemampuan format data adalah seberapa cepat *request* itu diproses, sehingga dibutuhkan *request* dengan format pertukaran data yang efisien dan cepat dalam proses parsing tersebut. XML merupakan format pertukaran data yang umum digunakan dalam aplikasi dengan berbasis web. Namun hal ini bukanlah cara yang terbaik. XML memiliki sintaks yang kompleks dan harus diproses menjadi *Document*

*Object Model*. Format pertukaran data selain XML yaitu JSON yang merupakan bagian dari JavaScript sehingga parsing dilakukan oleh JavaScript tersebut dan memiliki sintaks yang lebih sederhana daripada XML. Secara spesifikasi JSON lebih baik daripada XML (Saryanto, Sumarsono, & Retnowati, 2013).

Penelitian yang dilakukan Saryanto dan kawan-kawan (2013) yang berjudul Analisis Perbandingan XML dan JSON Menggunakan *Platform* Aplikasi PHP dengan *Architectural State Transfer* (REST). Penelitian ini membandingkan format data JSON dan XML pada arsitektur *restfull*. Dan ada beberapa segi aspek dalam penelitian ini. Pada penelitian ini berdasarkan hasil pengujian dari ujicoba perhitungan waktu serialisasi data, ujicoba perhitungan ukuran data, ujicoba waktu parsing data yang di lakukan pada jaringan localhost dan jaringan internet menggunakan *web service* REST menunjukkan bahwa selisih waktu dipengaruhi oleh jumlah *record* data, jika jumlah *record* semakin banyak maka selisihpun semakin besar sehingga dapat disimpulkan bahwa JSON lebih cepat pada proses serialisasi data dan proses parsing data daripada XML dan memiliki ukuran data lebih kecil dibandingkan XML.

### **2.12.1 JSON**

JSON, singkatan dari JavaScript Object Notation adalah suatu format ringkas pertukaran data komputer. Formatnya berbasis teks dan terbaca manusia serta digunakan untuk merepresentasikan struktur data sederhana dan larik asosiatif (disebut objek). Format JSON sering digunakan untuk mentransmisikan data terstruktur melalui suatu koneksi jaringan pada suatu proses yang disebut serialisasi. (Kusumawaty, 2012).

JSON adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat oleh komputer. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java,

JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data (Introducing JSON, n.d.).

```

{"list_event":
 [
  {"eventID": "1",
   "judul": "Pengajian Akbar menyambut ramadhan",
   "tanggal": "2014-05-19",
   "jam": "20:00",
   "lokasi": "Masjid Kampus UGM"
 },
  {"eventID": "2",
   "judul": "Bedah buku kesesatan syiah",
   "tanggal": "2014-05-21",
   "jam": "09:00",
   "lokasi": "Masjid Kampus UII "
 }
 ]
}

```

**Gambar 2.10.** Contoh format data JSON (<http://www.candra.web.id/penjelasan-struktur-data-json/>)

*List\_event* yang tertera pada gambar 2.10 adalah Array JSON. Array terdiri dari beberapa object. Dalam contoh gambar mempunyai 2 object *event*. Masing masing object *event* ini mempunyai 5 *field*.

### 2.12.2 XML

XML adalah sebuah bahasa markup yang digunakan untuk mengolah meta data (informasi tentang data) yang menggambarkan struktur dan maksud atau tujuan data yang terdapat dalam dokumen XML, namun bukan menggambarkan format tampilan data tersebut (XML , n.d.).

XML merupakan dasar terbentuknya web service yang digunakan untuk mendeskripsikan data. Pada level paling detail web service secara keseluruhan dibentuk di atas XML. Fungsi utama dari XML adalah komunikasi antar aplikasi, integrasi data dan komunikasi aplikasi eksternal dengan partner luaran. Dengan standarisasi XML, aplikasi-aplikasi yang berbeda dapat mudah berkomunikasi antar satu dengan yang lain. (Wahli, dkk, 2006)

Saat ini XML dapat dikatakan telah menggantikan HTML. Masing-masing dikembangkan untuk tujuan yang berbeda. Kalau HTML digunakan untuk menampilkan informasi dan berfokus pada bagaimana informasi terlihat, XML mendeskripsikan susunan informasi dan berfokus pada informasi itu sendiri. XML terutama dibutuhkan untuk menyusun dan menyajikan informasi dengan format yang tidak mengandung format standar layaknya heading, paragraph, table dan lain sebagainya (Fauziah, 2013).

```

<list_event>
  <event>
    <eventID>1</eventID>
    <judul>Pengajian Akbar menyambut ramadhan</judul>
    <tanggal>2014-05-19</tanggal>
    <jam>20:00</jam>
    <lokasi>Masjid Kampus UGM</lokasi>
  </event>
  <event>
    <eventID>2</eventID>
    <judul>Bedah buku kesesatan syiah</judul>
    <tanggal>2014-05-21</tanggal>
    <jam>09:00</jam>
    <lokasi>Masjid Kampus UII </lokasi>
  </event>
</list_event>

```

**Gambar 2.11.** Contoh format data XML (<http://www.candra.web.id/penjelasan-struktur-data-json/>)

Perbedaan yang paling mendasar adalah ukuran JSON lebih kecil, perhatikan jumlah text dan tag penutup di XML yang membuat ukuran XML menjadi besar. Dari dua format diatas juga terlihat format XML lebih kompleks.

### 2.13 CURL

Merupakan akronim dari fungsi-fungsi pustaka *Client URL*. Kode pustaka ini dipergunakan untuk berkomunikasi melalui Internet menggunakan sembarang protocol di sisi lawan. Kode ini mendukung Gopher, Telnet, dan HTTPS. Untuk dapat menggunakan fungsi ini, konfigurasi instalasi PHP harus menyertakan *flag -with-curl* (Adriana, 2003).

## 2.14 Database Manajemen System (DBMS)

*Database Manajemen System* (DBMS) merupakan paket perangkat lunak yang dirancang khusus untuk membuat, memelihara, mengontrol, menyimpan serta memberikan akses secara terkontrol terhadap data.

DBMS biasanya juga menyediakan bahasa tersendiri yang bisa digunakan untuk berinteraksi dengannya. Namun hampir semua dbms populer menyediakan dukungan pula untuk bahasa SQL (Karim, 2012). Beberapa aplikasi dbms yang populer: MySQL, PostgreSQL, Microsoft SQL Server.

## 2.15 PHP

PHP merupakan salah satu bahasa pemrograman untuk web yang bekerja disisi server. Hampir semua website dinamis yang bisa diakses lewat internet dibangun dengan menggunakan bahasa ini. Sampai saat ini php merupakan satu-satunya bahasa yang menawarkan kemudahan dalam mengembangkan aplikasi internet, terutama dengan dukungan luar biasa ke berbagai database server.

Dalam hal sintaks, sintaks php sangat mirip dengan sintaks bahasa C. Karena memang bahasa ini beserta ekstensionnya juga dibuat dengan menggunakan bahasa C. Sehingga bagi para pengguna bahasa tersebut, akan lebih mudah untuk pindah atau mempelajari php (Karim, 2012).

## 2.16 Javascript

Javascript merupakan bahasa *scripting* yang didisain untuk membuat halaman web menjadi lebih interaktif. Javascript bekerja secara *client-side* (disisi *client*). Sehingga javascript bergantung pada browser yang digunakan untuk menampilkan halaman web yang mengandung javascript.

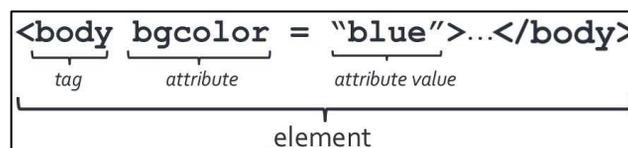
Perlu diperhatikan bahwa Javascript berbeda dengan Java. Keduanya sangat berbeda baik itu dari sisi bahasa maupun dari konsep dan disainnya (Karim, 2012).

## 2.17 HTML

*Hyper Text Markup Language* (html), merupakan bahasa markup yang digunakan untuk membuat halaman web dan menampilkan berbagai informasi didalam sebuah *browser*. Dokumen html (atau halaman web) terdiri dari elemen-elemen yang menginstruksikan *browser* untuk menghasilkan tampilan sesuai dengan yang diinginkan.

Ada dua cara untuk membuat halaman web dengan html yaitu menggunakan web *designer* atau dengan teks editor biasa, dan setiap berkas html disimpan dalam ekstensi .html. Dan perlu diingat bahwa nama berkas pada beberapa sistem operasi akan bersifat *case-sensitive*, sehingga berkas dengan nama dokumen.html berbeda dengan berkas DOKUMEN.html dan berbeda juga dengan berkas Dokumen.html.

Setiap website yang dibuat dengan menggunakan html memiliki dokumen utama yang diberi nama index.html. Sehingga bila ingin membuat website, wajib memiliki berkas ini (Karim, 2012).

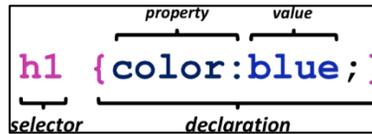


**Gambar 2.12.** Struktur Elemen HTML

## 2.18 CSS

*Cascading Style Sheet* (css), merupakan *style* yang digunakan untuk mengatur tampilan dari halaman web. Dengan menggunakan css, maka untuk menghasilkan tampilan web yang indah dan menarik jauh lebih mudah dan ringkas dibandingkan menggunakan atribut di tiap tag html.

Selain bisa membuat indah, css juga bisa membantu menciptakan web dengan tampilan yang dinamis dan fleksibel. Baik itu untuk tampilan web di browser ataupun ketika halaman web dicetak lewat printer (Karim, 2012).



Gambar 2.13. Struktur CSS

## 2.19 Tinjauan Literatur

Adapun penelitian dan referensi yang menjadi acuan, berikut adalah penelitian dan referensi terdahulu yang berhubungan dengan web dan yang berkaitan dengan berbagi, *smartcity*, *living lab*, *Restful web service*, JSON dan model rekayasa perangkat lunak *UCD*. Berikut *state of the art* dari penelitian ini.

Penelitian berjudul “Implementasi *Rest Web Service* Untuk sales Order *Dansales Tracking* berbasis Mobile” yang dilakukan oleh (Kurniawan, 2014). Penelitian ini berbicara mengenai penerapan *REST service* berformat JSON sebagai *backend* pada penelitian ini mengatakan sistem ini sangat cocok digunakan, karena dengan format JSON yang sehingga pengunduhan data dari *web service* lebih cepat dibandingkan dengan penggunaan dokumen XML yang ukurannya relatif lebih besar.

Penelitian yang dilakukan oleh (Saputri, Fadhli, & Surya, 2017) yang berjudul “Penerapan Metode UCD (*User Centered Design*) pada E-Commerce Putri Intan Shop Berbasis Web”. Pada penelitian ini menjelaskan implementasi dari metode UCD dan pada evaluasinya menggunakan *Usability Testing* dengan *System Usability Scale*. Hasil yang didapat adalah berdasarkan *Usability Testing* dengan *System Usability Scale*, Putri Intan Shop ada pada *range excellent* yaitu sebesar 86,8%. Maka dari itu, *e-commerce* Putri Intan Shop berhasil membangun sistem yang *user-friendly* dengan tingkat *usability* yang tinggi.

Penelitian yang dilakukan oleh (Sutanta & Mustofa, 2012) yang berjudul “Kebutuhan *Web Service* untuk Sinkroni Data Antar Sistem Informasi Dalam *E-Gov* di Pemkab Bantul Yogyakarta” membahas mengenai masalah sinkronisasi data antara 33 aplikasi sistem informasi yang sudah ada dalam *e-Gov* di lingkungan Pemkab Bantul Yogyakarta. Pada

implementasinya menggunakan teknologi *web service* dengan arsitektur *rest* untuk mensinkronkan 33 sistem yang ada. Penelitian ini menyimpulkan bahwa *web-service* merupakan bentuk implementasi konsep interoperabilitas yang dapat menjadi sebuah alternatif solusi untuk proses pertukaran data antar sistem informasi. Dengan kemampuan proses pertukaran data antar sistem informasi, maka dimungkinkan untuk melakukan proses sinkronisasi data di antara antar sistem informasi, termasuk dalam *e-Gov*.

Kemudian ada penelitian berjudul “*A Web of Things* dengan *RESTful* protokol pada *Smart City*” yang dilakukan oleh (Khasoggi & Ermatita, 2017). Membahas mengenai penerapan arsitektur *web-service restful* untuk memanfaatkan prinsip-prinsip kerja web dengan membuat *Web of Things* mudah diakses dan di program sehingga mempermudah akses dan penanganan terhadap perangkat pintar (*smart things*) yang ada pada *Smart City* seperti *Smart Government*, *Smart Digital Citizens*, *Smart Health*, *Smart Farming*, *Smart Energy*, *Smart Transportation*, *Smart Manufacturing* dan perangkat pintar pendukung lainnya sebagai sumber web. Penelitian ini menyimpulkan bahwa peran *Restful* sebagai *middleware* memberikan kemudahan dalam pengelolaan sumber daya perangkat pintar pada *Smart City* karena dapat mengirimkan verba berupa *GET*, *POST*, *PUT*, dan *DELETE*, sesuai dengan permintaan atau perintah yang akan di eksekusi.

Berdasarkan penelusuran terhadap judul Tugas Akhir yang ada di Program Studi Teknik Informatika Fakultas Teknik Industri UPN “Veteran” Yogyakarta, tidak ditemukan judul Tugas Akhir “Media Jogja Berbagi Berbasis Web dan Restfull Web Service untuk Sarana Berbagi Di Wilayah Yogyakarta”. Adapun yang membedakan Tugas Akhir ini dengan penelitian-penelitian sebelumnya adalah, fokus penelitian yang mengarah ke sosial informatika dengan metode pendekatan yang sesuai. Pada penelitian ini menggunakan metode penelitian *Living Lab* dan metode perancangan perangkat lunak menggunakan metode UCD (*User Center Design*).

## BAB III

### METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM

#### 3.1 Metodologi Penelitian

Metodologi penelitian pada bab ini menggunakan metode *living lab*. Pada metode ini ada tiga tahapan yaitu Eksplorasi, Eksperimen, kemudian Evaluasi. Sebelum masuk pada metode *living lab* pada penelitian ini dilakukan identifikasi *smart city*, dimana ditentukannya terlebih dahulu dimensi yang akan masuk dalam penelitian. Kemudian dalam tahap Eksperimen, dilakukan pengembangan perangkat lunak menggunakan metode UCD (*User Center Design*).

Proses analisis masalah yang pertama dilakukan dengan menganalisis data kemiskinan, data pengangguran, dan data Lembaga sosial yang ada di D.I.Yogyakarta. kemudian menentukan dimensi pada *smart city*. Selanjutnya melakukan pengamatan yang terjadi di media sosial dan lingkungan sekitar serta dari penelitian terdahulu. Untuk pengumpulan data pada penelitian ini dilakukan dengan menyebar kuisioner dan pengambilan data Lembaga Sosial dari Dinas Sosial Yogyakarta, dan data angka kemiskinan dan angka pengangguran dari Badan Statistika Yogyakarta. Selain itu pengumpulan data juga dilakukan dengan studi pustaka, membaca literatur-literatur tentang metode dan teknologi yang digunakan pada penelitian ini. Setelah proses analisis masalah dan pengumpulan data selesai maka dilakukan perancangan perangkat lunak menggunakan metode UCD (*User Center Design*) dimana pengguna terlibat langsung untuk pembuatan perangkat lunaknya. Setelah tahap perancangan dan menghasilkan aplikasi, tahap selanjutnya adalah evaluasi dengan menggunakan kuisioner. Untuk kerangka kerjanya sebagai dapat dilihat pada Gambar 3.1.



### 3.2 Smart City

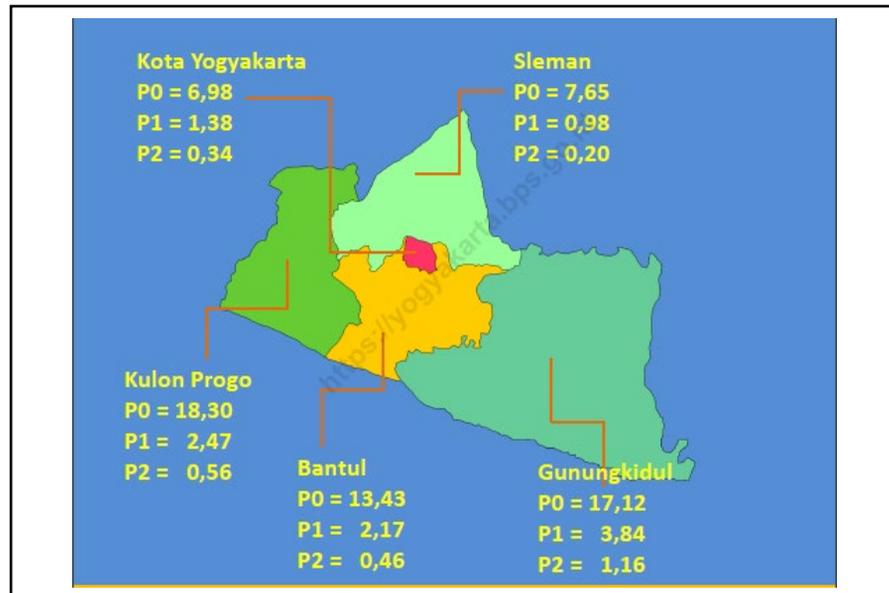
Konsep *Smart City* memiliki enam dimensi yang menjadi fokus penerapannya. Dimensi dimensi tersebut adalah sebagai berikut, *Smart Economy*, *Smart Mobility*, *Smart Environment*, *Smart People*, *Smart Government*, *Smart Living*. Pada penjelasan yang sudah dilakukan pada bab sebelumnya mengenai *smart city* dan dimensinya, pada penelitian ini dipilih dimensi *smart city* yaitu *smart people* karena *smart people* menurut penjelasannya mempertahankan tata nilai sosial termasuk peningkatan dalam elemen-elemen seperti kepercayaan, gotong royong, toleransi, penghargaan, saling memberi dan saling menerima serta kolaborasi sosial. Hal ini berkaitan mengenai permasalahan kota yang ada di Yogyakarta yaitu angka kemiskinan yang tinggi dan angka pengangguran yang tergolong tinggi, dengan fokus ada penelitian ini yaitu kegiatan berbagi.

### 3.3 Eksplorasi

Dari pemaparan permasalahan kota yang terjadi, pada tahapan eksplorasi akan dijelaskan pengambilan data yang menjadi kebutuhan penelitian.

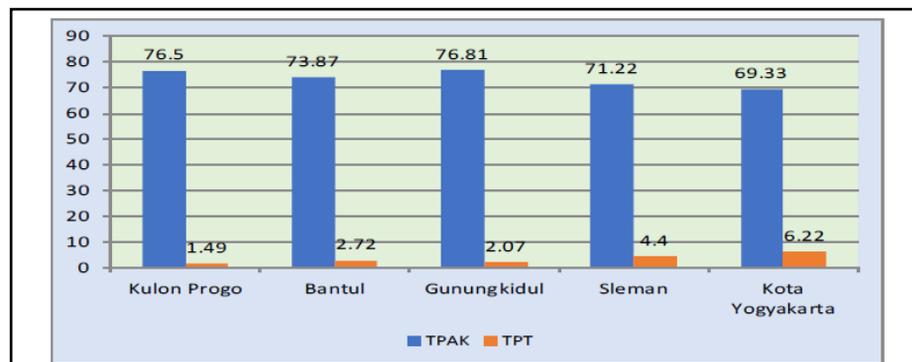
#### 3.3.1 Data Kemiskinan dan Pengangguran

Badan Pusat Statistika Provinsi D.I. Yogyakarta pada tahun 2019 telah merilis sebuah kajian mengenai data kemiskinan yang ada di provinsi D.I. Yogyakarta. Terdapat keterangan indikator-indikator yang biasa digunakan terkait dengan kemiskinan antara lain adalah persentase penduduk miskin (P0), indeks kedalaman kemiskinan (P1) adalah ukuran rata-rata kesenjangan pengeluaran masing-masing penduduk miskin terhadap garis kemiskinan. Dan indeks keparahan kemiskinan (P2) adalah indeks yang memberikan informasi mengenai gambaran penyebaran pengeluaran di antara penduduk miskin. BPS DIY menyebut persentase kemiskinan di D.I. Yogyakarta selalu tercatat sebagai yang tertinggi untuk Pulau Jawa (BPS, 2014 – 2018). Pemetaan wilayah dengan data kemiskinan di D.I. Yogyakarta bisa dilihat pada Gambar 3.2.



**Gambar 3.2.** Peta Kemiskinan Yogyakarta (BPS DIY)

Untuk angka pengangguran sendiri di D.I.Yogyakarta menurut BPS pada tahun 2018 menyatakan Tingkat Pengangguran Terbuka (TPT) sebesar 3,35 persen. Angka tersebut mengalami peningkatan, 0,32 persen poin dibanding TPT Agustus 2017 sebesar 3,02 persen. Angka TPT D.I Yogyakarta pada Agustus 2018 ini lebih rendah dibanding TPT Nasional sebesar 5,34 persen.



**Gambar 3.3.** Tingkat Partisipasi Angkatan Kerja dan Tingkat Pengangguran Terbuka menurut Kabupaten/kota di D.I. Yogyakarta Agustus 2018

Kondisi Tingkat Partisipasi Angkatan Kerja (TPAK) menurut kabupaten/kota tertinggi terjadi pada Kabupaten Gunungkidul sebesar 76,81 persen diikuti Kabupaten Kulon Progo sebesar 76,50 persen dan Kabupaten Bantul sebesar 73,87 persen. Sedangkan TPAK terendah pada Kota Yogyakarta sebesar 69,33 persen. Sedangkan kabupaten/kota dengan TPT tertinggi pada Kota Yogyakarta sebesar 6,22 persen diikuti Kabupaten Sleman

sebesar 4,40 persen dan Kabupaten Bantul 2,72 persen. Kondisi ini menunjukkan semakin besar wilayah perkotaan pada suatu daerah, maka menyebabkan semakin menurunnya penawaran tenaga kerja dan meningkatnya pengangguran yang disebabkan sulitnya untuk masuk ke lapangan kerja di perkotaan. Hal ini karena lapangan kerja di perkotaan lebih banyak pada sektor formal, yang membutuhkan ijazah atau syarat-syarat tertentu.

### 3.3.2 Studi Pustaka

Pengambilan data pada penelitian ini diperoleh mengambil data komunitas, organisasi sosial dan LKS (Lembaga Kemasyarakatan Sosial) dari Dinas Sosial Daerah Istimewa Yogyakarta. Data ini nantinya akan ditampilkan pada sistem aplikasi jogja berbagi. Kemudian untuk bahan pendukung informasi yang dilakukan pada penelitian ini adalah dengan studi pustaka untuk mempelajari mengenai konsep informatika sosial, metode *living lab*, metode UCD (*User Center Design*), *web service*, arsitektur *restful web-service* dan implementasinya. Tujuannya untuk melakukan studi literatur dari berbagai macam referensi seperti jurnal, buku, dan penelitian serupa. Adapun buku yang menjadi referensi dalam proses pengolahan data kemiskinan adalah *e-book* yang berjudul Statistik Kemiskinan Provinsi Daerah Istimewa Yogyakarta 2019 (BPS DIY, 2019). Kemudian referensi untuk data pengangguran adalah *e-book* yang berjudul Keadaan Ketenagakerjaan Yogyakarta Agustus 2018 (BPS DIY, 2018). Referensi juga diperoleh dari *e-book* berjudul “*Living Lab Methodology Handbook*” oleh (Katariina Malmberg & Ines Vaithine, 2018).

### 3.3.3 Kuisoner

Data kuisoner dengan pertanyaan mengenai kebutuhan masyarakat tentang berbagi dilakukan secara online melalui media sosial *whatsapp* dan *line* dengan bantuan *google form*. Pertanyaan dirancang berdasarkan referensi dari jurnal berjudul “Metode UCD (*User Centered Design*) Untuk Rancangan Kios Informasi Studi Kasus: Rumah Sakit Bersalin XYZ” oleh (Widhiarso, 2007). Berikut daftar pertanyaannya dapat dilihat pada Tabel 3.1.

Tabel 3.1. Daftar Pertanyaan Kuisoner

No.	Pertanyaan	Pilihan Jawaban
1.	Apa jenis kelamin anda?	Laki-laki, Perempuan
2.	Apa pekerjaan anda?	Pelajar, Mahasiswa, Karyawan, Wiraswasta, Pegawai, Pemerintahan, Penganggura
3.	Apakah anda menggunakan internet?	Ya, Tidak
4.	Seberapa sering anda menggunakan internet?	Kurang dai 1 jam, 1-3 jam, Lebih dari 3 jam
5.	Device apa yang paling anda sukai untuk mengakses internet	Komputer/PC, Laptop/Notebook, Handphone/Smartphone
6.	Apakah anda menggunakan media sosial untuk berkomunikasi?	Ya, Tidak
7.	Jika Ya, media sosial apa yang sering anda gunakan untuk berkomunikasi?	Whatsapp, Line, Facebook Messenger, Lainnya
8.	Apakah anda pernah menjadi relawan?	Ya, Tidak
9.	Apakah anda pernah melakukan kegiatan sosial? (baktisosial, belajar bersama anak-anak kurang mampu, cek kesehatan gratis, dll)	Ya, Tidak
10.	Jika Ya, mengapa anda melakukan kegiatan sosial tersebut?	Karena saudara, Karena adanya bencana alam, Karena adanya kegiatan sosial, Lainnya
11.	Apakah anda suka mencari informasi tentang berbagi?	Ya, Tidak
12.	Apabila anda pernah berbagi, dari mana anda mendapat informasi tersebut?	Informasi mulut ke mulut, Media Sosial (cth: Facebook,Whatsapp dan lain sebagainya), Lainnya
13.	Apabila terjadi bencana alam, melalui apa anda berbagi?	Website/Aplikasi Berbagi, Langsung berbagi ke lokasi, Disalurkan melalui komunitas/ lembaga berbagi, Lainnya
14.	Apakah anda lebih suka berbagi secara individu atau dengan kelompok?	Individu, Kelompok
15.	Jika tidak pernah berbagi, kendala apa yang menyebabkan anda tidak berkeinginan untuk berbagi?	Tidak tahu siapa yang ingin dibagi, Tidak ada media yang mendukung, Lainnya
16.	Apakah anda lebih tertarik untuk berbagi jika ada media/sarana yang menarik, atau tidak memperdulikan sarana berbagi tersebut?	Ya, jika media/sarana menarik, Tidak memperdulikan media/sarana
17.	Apakah anda pernah menggunakan aplikasi yang memberikan informasi berbagi?(contoh: Kita bisa, indorelawan)	Ya, Tidak
18.	Jika pernah, seberapa sering anda berbagi setelah melihat informasi dari aplikasi tersebut?	Lebih dari 15 kali, 10-15 kali, 5-9 kali, 1-4 kali, Tidak pernah
19.	Apa yang anda harapkan dari suatu aplikasi berbagi?	Menu yang menarik, Aplikasi yang mudah digunakan, Aplikasi yang cepat, Lainnya
20.	Fitur apa yang perlu ada dari aplikasi berbagi?	Berupa postingan, Kategori, Suka / tidak suka, Komentar, Bisa share gambar, Koneksi Whatsapp, Bisa memberi atau meminta, Lainnya

**Tabel 3.2.** Lanjutan Daftar Pertanyaan Kuisoner

No.	Pertanyaan	Pilihan Jawaban
21.	Apakah anda mempertimbangkan jarak dari lokasi Anda sekarang dengan lokasi anda berbagi?	Ya, Tidak
22.	Apakah anda tertarik dengan aplikasi android atau web yang dapat memberikan informasi tentang berbagi yang terdekat dengan lokasi anda sekarang (D.I.Y Yogyakarta)	Ya, Tidak

Selain kuisoner untuk masyarakat juga terdapat kuisoner untuk pemerintah dan komunitas. Kuisoner ini nanti diperlukan untuk merancang pada tahapan *Use Center Design*. Tabel daftar pertanyaan untuk pemerintah dan komunitas bisa dilihat pada tabel 3.3 dan tabel 3.4.

**Tabel 3.3.** Daftar Pertanyaan Kuisoner untuk Pemerintah

No.	Pertanyaan	Pilihan Jawaban
1.	Apakah diperlukan media online yang berfokus untuk berbagi di wilayah Yogyakarta?	Ya, Tidak
2.	Apakah setuju diterapkannya teknologi web service untuk memaksimalkan pelayanan masyarakat terhadap kegiatan berbagi?	Setuju, Tidak setuju, Lainnya

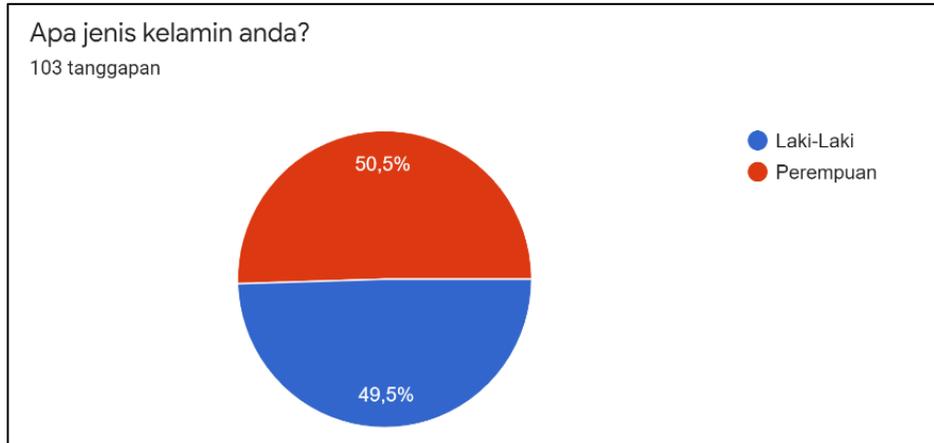
**Tabel 3.4.** Daftar Pertanyaan Kuisoner untuk Komunitas

No.	Pertanyaan	Pilihan Jawaban
1.	Apakah diperlukan media online yang berfokus untuk berbagi di wilayah Yogyakarta?	Ya, Tidak
2.	Fitur apa yang diperlukan komunitas dalam aplikasi berbagi?	(Deskripsi)

Dari pengumpulan data melalui kuisoner yang disebar untuk masyarakat terdapat 103 koresponden. Hasil kuisoner dari masyarakat sebagai pengguna adalah sebagai berikut:

#### 1. Pertanyaan nomor satu

Hasil didapatkan yaitu 50.5% perempuan dan 49.5% laki-laki. Dari 103 tanggapan, frekuensi yang menjawab perempuan sebanyak 52 responden. Sedangkan laki-laki frekuensi yang menjawab sebanyak 51 responden. Penjelasan lebih lanjut bisa dilihat pada gambar 3.4.

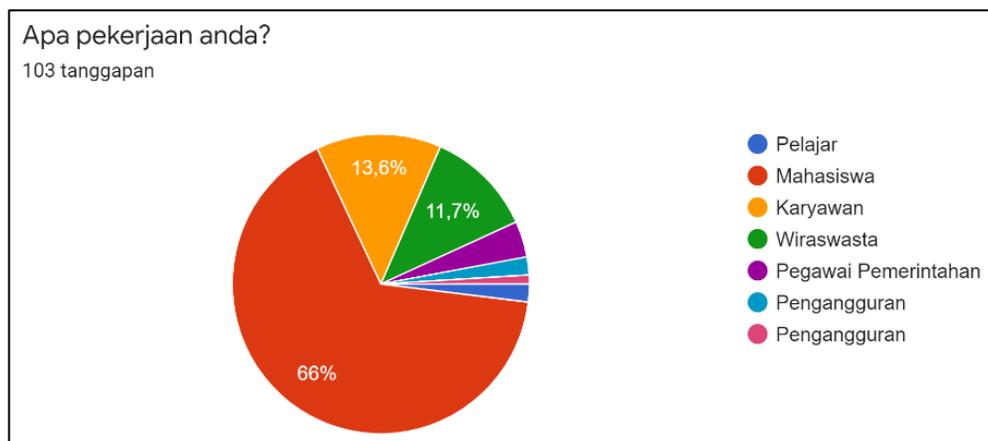


**Gambar 3.4.** Hasil Kuisoner Pertanyaan Nomor Satu

## 2. Pertanyaan nomor dua

Untuk pertanyaan nomor dua didapatkan hasil dari total koresponden 103 dengan data pekerjaan koresponden mahasiswa sebanyak 66%, karyawan sebanyak 13,6%, dan wiraswasta sebanyak 11,7%. Selain itu ada pegawai pemerintah, pelajar dan pengangguran.

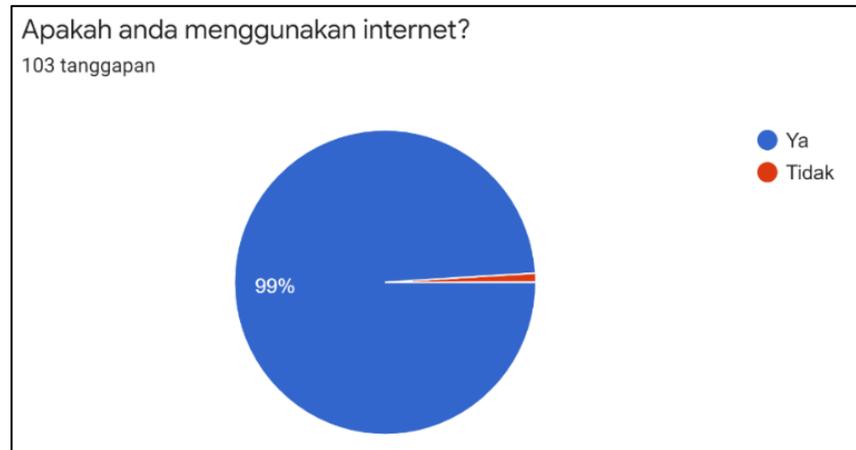
Grafik dari hasil pertanyaan nomor dua bisa dilihat pada gambar 3.5.



**Gambar 3.5.** Hasil Kuisoner Pertanyaan Nomor Dua

## 3. Pertanyaan nomor tiga

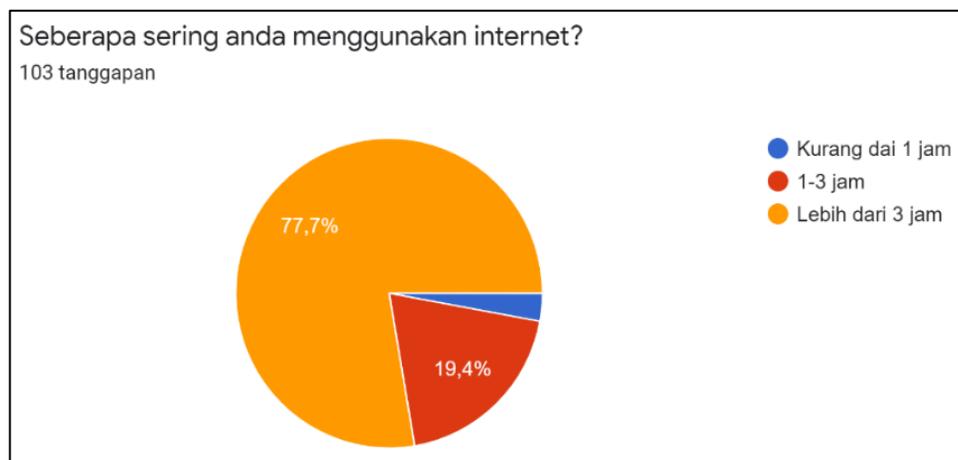
Pertanyaan nomor tiga adalah tentang pengguna internet. Dari gambar 3.6. bisa dilihat grafik menunjukkan 99% koresponden menggunakan internet. Hanya terdapat 1 tanggapan yang tidak menggunakan internet.



**Gambar 3.6.** Hasil Kuisoner Pertanyaan Nomor Tiga

#### 4. Pertanyaan nomor empat

Dari grafik gambar 3.6. ditunjukkan bahwa koresponden lebih banyak menggunakan internet lebih dari 3 jam yaitu sebanyak 77,7%, dan menunjukkan angka 19,4% untuk 1-3 jam. Sedangkan untuk kurang dari 1 jam hanya 2,9% saja.



**Gambar 3.7.** Hasil Kuisoner Nomor Empat

#### 5. Pertanyaan nomor lima

Untuk data yang diperoleh mengenai *device* yang paling disukai oleh pengguna adalah *handphone* atau *smartphone*. Didapatkan angka 70,9% menyukai *handphone* untuk mengakses internet dan 20,4% untuk yang menggunakan laptop. Sedangkan 8,7% menggunakan computer atau PC. Grafik dapat dilihat pada gambar 3.8.



**Gambar 3.8.** Hasil Kuisioner Pertanyaan Nomor Lima

#### 6. Pertanyaan nomor enam

Data yang didapat dari 103 respon mengenai penggunaan media sosial untuk berkomunikasi adalah 97,1% menjawab iya, dan sisanya menjawab tidak.



**Gambar 3.9.** Hasil Kuisioner Pertanyaan Nomor Enam

#### 7. Pertanyaan nomor tujuh

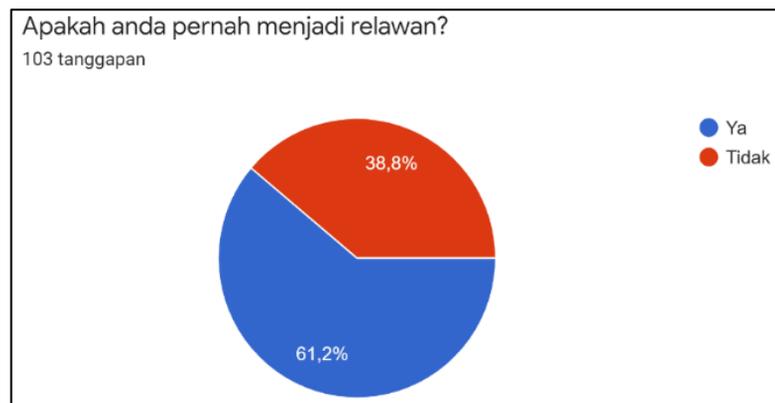
Dari grafik pada gambar 3.10 menunjukkan bahwa aplikasi *whatsapp* adalah media sosial yang paling digunakan oleh pengguna. Diurutan ke dua ada aplikasi *Line*. Dari 102 tanggapan, 79,4% memilih *whatsapp* sebagai media sosial yang sering digunakan. Untuk media sosial *Line* sebanyak 15,7% dari keseluruhan tanggapan.



**Gambar 3.10.** Hasil Kuisioner Pertanyaan Nomor Tujuh

#### 8. Pertanyaan nomor delapan

Dari 103 tanggapan, sebanyak 61,2% pengguna menyatakan pernah menjadi relawan. Dan 38,8% menanggapi tidak pernah menjadi relawan. Grafik dari hasil pertanyaan nomor tiga belas dapat dilihat pada gambar 3.11.



**Gambar 3.11.** Hasil Kuisoner Pertanyaan Nomor Delapan

#### 9. Pertanyaan nomor sembilan

Data yang diperoleh dari pertanyaan nomor empat belas mengenai keikutsertaan dalam kegiatan sosial adalah 85,4% pengguna pernah mengikuti kegiatan sosial dari 103 tanggapan yang diperoleh. Dan didapat 14,6% tidak pernah mengikuti kegiatan sosial.

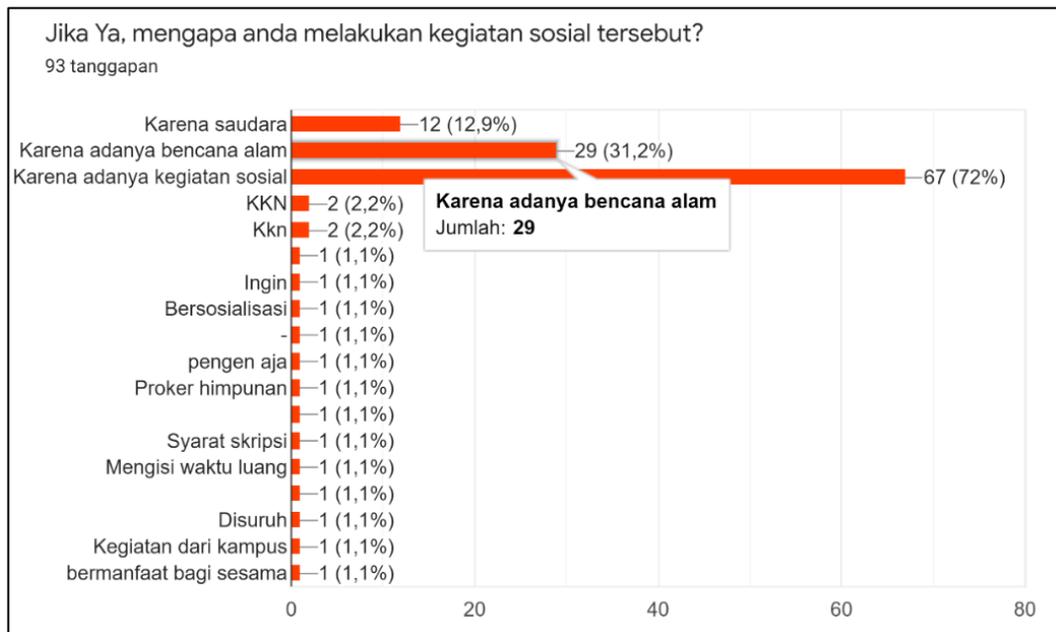


**Gambar 3.12.** Hasil Kuisoner Pertanyaan Nomor Sembilan

#### 10. Pertanyaan nomor sepuluh

Data yang diperoleh dari pertanyaan nomor lima belas adalah bahwa alasan pengguna melakukan kegiatan sosial adalah karena ada kegiatan sosial yang diadakan. Kemudian alasan ke dua yang memiliki presentase dibawahnya yaitu karena ada bencana alam dan

karena saudara. Selebihnya alasannya karena ada KKN, bersosialisasi, dan lain sebagainya. Pada gambar 3.13 terdapat rincian presentasi detail dari masing masing alasan yang menjadi tanggapan dari 93 responden.



**Gambar 3.13.** Hasil Kuisioner Pertanyaan Nomor Sepuluh

#### 11. Pertanyaan nomor sebelas

Gambar 3.14 menunjukkan hasil dari pertanyaan tentang informasi berbagi yang diperoleh oleh pengguna. Dari 103 tanggapan 64% menjawab suka mencari informasi berbagi. Bisa disimpulkan bahwa potensi kegiatan berbagi sangat tinggi.

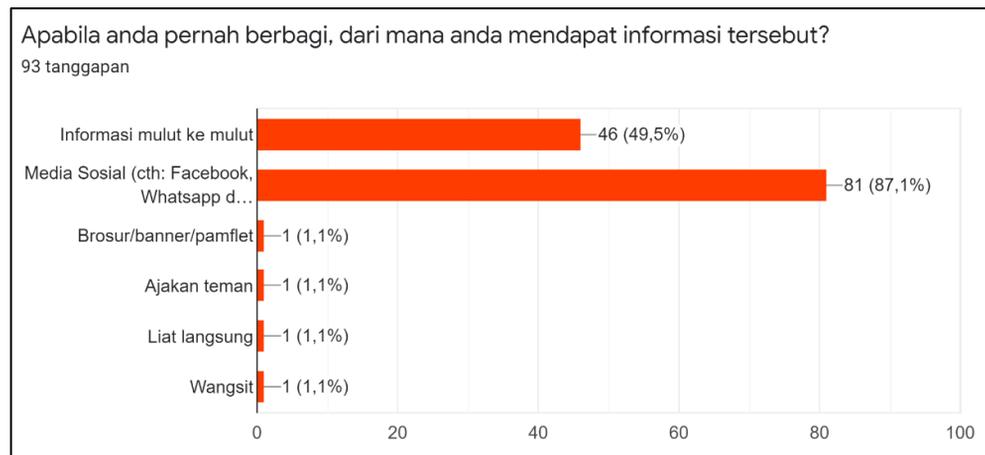


**Gambar 3.14.** Hasil Kuisioner Pertanyaan Nomor Sebelas

#### 12. Pertanyaan nomor dua belas

Pertanyaan nomor tujuhbelas adalah pertanyaan mengenai informasi yang didapat dari kegiatan berbagi. Tanggapan yang diperoleh adalah 93 respon. Dari gambar 3.15

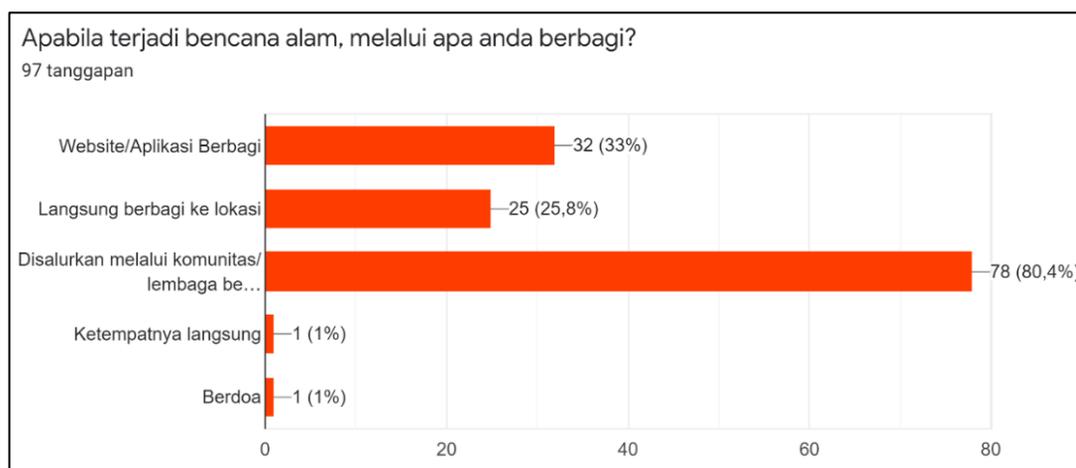
diketahui bahwa pengguna mendapat informasi tentang berbagi paling banyak lewat media sosial. Detail presentase dari masing-masing tanggapan dapat dilihat pada gambar 3.15.



**Gambar 3.15.** Hasil Kuisioner Pertanyaan Nomor Tujuhbelas

### 13. Pertanyaan nomor tiga belas

Apabila terjadi bencana alam dari hasil yang didapat presentase paling banyak yang menjadi pilihan responden adalah menyalurkan bantuan lewat komunitas atau lembaga. Hal tersebut dapat dibuktikan dari gambar 3.16 yang menunjukkan bahwa sebanyak 78 tanggapan kebanyakan akan menyalurkan bantuan lewat komunitas atau lembaga. Sedangkan sebanyak 33% pengguna memilih lewat aplikasi, dan langsung berbagi ke lokasi dipilih pengguna sebanyak 25%. Detail dari masing-masing jawaban dapat dilihat dari gambar 3.16.



**Gambar 3.16.** Hasil Kuisioner Pertanyaan Nomor Tiga belas

#### 14. Pertanyaan nomor empat belas

Pertanyaan nomor empat belas adalah tentang kesukaan berbagi dengan cara individu atau dengan kelompok. Dari gambar 3.17 mayoritas pengguna lebih senang berbagi secara kelompok. Dari 101 tanggapan sebanyak 66,3% memilih berbagi secara kelompok. Dan 33,7% memilih berbagi secara individu.



**Gambar 3.17.** Hasil Kuisoner Pertanyaan Nomor Empat belas

#### 15. Pertanyaan nomor lima belas

Pertanyaan nomor lima belas menghasilkan grafik yang ditunjukkan pada gambar 3.18, grafik menunjukkan bahwa kendala paling banyak untuk tidak berbagi menurut pengguna adalah informasi mengenai siapa yang membutuhkan bantuannya. Bisa dilihat pada gambar 3.18. sebanyak 46 respon menjawab tidak tau siapa yang ingin dibantu. Dan sebanyak 45 menjawab tidak ada media yang mendukung. Selebihnya pengguna takut salah sasaran.



**Gambar 3.18.** Hasil Kuisoner Pertanyaan Nomor Lima belas

16. Pertanyaan nomor enam belas

Gambar 3.19. menunjukkan bahwa pengguna akan lebih tertarik berbagi ketika ada media pendukung yang menarik. Dari 103 tanggapan sebanyak 77,7% setuju bahwa harus ada media yang mendukung dan menarik agar lebih mudah melakukan kegiatan berbagi. Dan sebanyak 22,3% tidak memperdulikan media.



**Gambar 3.19.** Hasil Kuisoner Pertanyaan Nomor Enam belas

17. Pertanyaan nomor tujuh belas

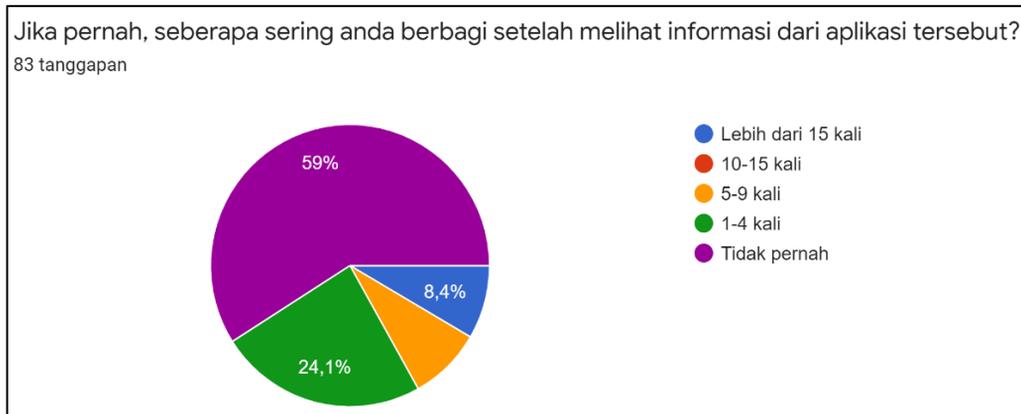
Pertanyaan nomor dua puluh dua yang berkaitan dengan penggunaan aplikasi yang memberi informasi berbagi. Dari gambar 3.20. menjelaskan bahwa 67% pengguna tidak pernah melakukan kegiatan berbagi melalui aplikasi. Dan 33% lainnya pernah.



**Gambar 3.20.** Hasil Kuisoner Pertanyaan Nomor Tujuh belas

18. Pertanyaan nomor delapan belas

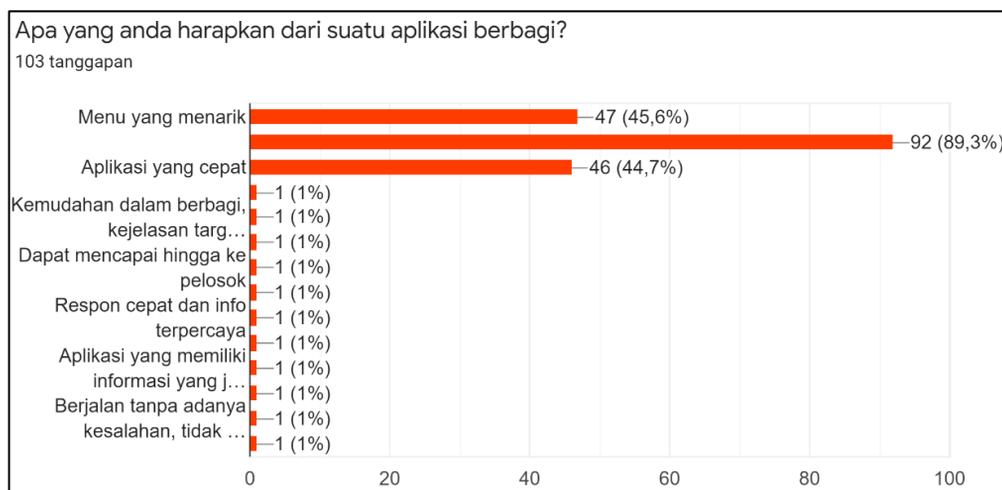
Untuk data mengenai seberapa sering pengguna melakukan kegiatan berbagi setelah melihat informasi dari aplikasi, gambar 3.26 menunjukkan bahwa 59% pengguna tidak pernah berbagi setelah mengetahui informasinya, dan sisanya pernah.



**Gambar 3.21.** Hasil Kuisoner Pertanyaan Nomor Delapan belas

19. Pertanyaan nomor sembilan belas

Hasil yang diperoleh dari pertanyaan nomor sembilan belas adalah tentang aplikasi yang diharapkan. Dari gambar 3.22 dapat dilihat bahwa yang diharapkan pengguna untuk aplikasi berbagi adalah aplikasi yang mudah digunakan, menu yang menarik, dan aplikasi yang cepat. Detail dari hasil kuisoner sebagai berikut:

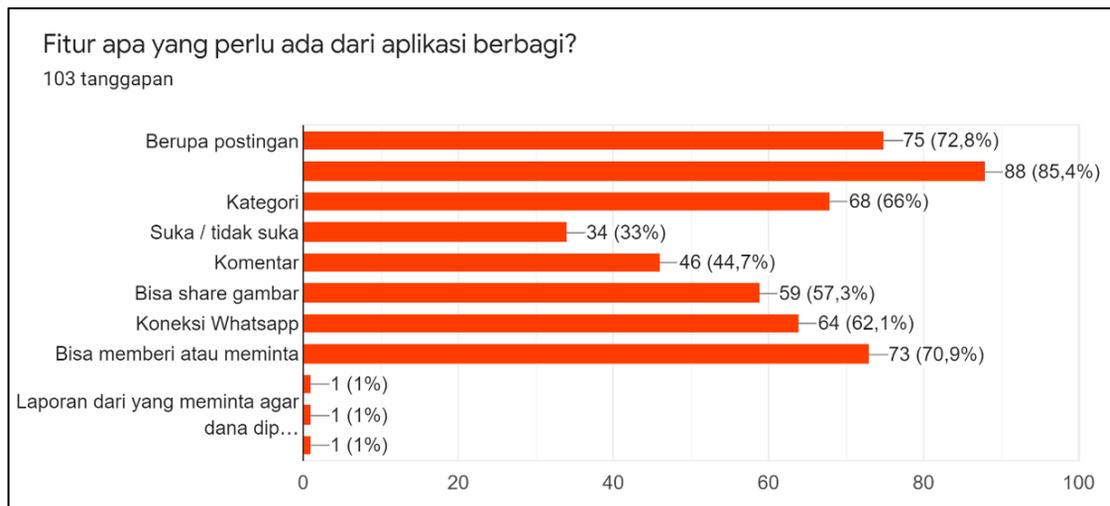


**Gambar 3.22.** Hasil Kuisoner Pertanyaan Nomor Sembilan belas

20. Pertanyaan nomor dua puluh

Pada pertanyaan dua puluh lima menunjukkan keperluan fitur yang diinginkan oleh pengguna. Pada grafik fitur keterangan kategori menjadi fitur yang paling diinginkan. Kemudian fitur yang lain yaitu berupa postingan, bisa memberi atau meminta, koneksi

dengan aplikasi whatsapp, bisa share gambar kemudian fitur komentar, dan fitur suka atau tidak suka. Detail dari hasilnya bisa dilihat pada gambar 3.28.



**Gambar 3.23.** Hasil Kuisoner Pertanyaan Nomor Duapuluh

#### 21. Pertanyaan nomor dua puluh satu

Dari grafik pada gambar 3.24. menunjukkan bahwa pengguna mempertimbangkan jarak lokasi untuk melakukan kegiatan berbagi. Dari 103 tanggapan, 68% menyatakan mempertimbangkan jarak, 32%nya tidak mempertimbangkan jarak.



**Gambar 3.24.** Hasil Kuisoner Pertanyaan Nomor Duapuluh satu

#### 22. Pertanyaan nomor dua puluh dua

Gambar 3.25 adalah grafik data yang menunjukkan data ketertarikan pengguna terhadap media aplikasi untuk berbagi yang terdekat dengan lokasi. Dari 103 tanggapan 98,1% menjawab ya. Dan sisanya tidak. Ini menunjukkan bahwa perlunya media aplikasi untuk menunjang kegiatan berbagi di wilayah Yogyakarta.



**Gambar 3.25.** Hasil Kuisoner Pertanyaan Nomor Duapuluh dua

Kemudian untuk hasil yang didapat dari kuisoner untuk pemerintah dan komunitas adalah sebagai berikut:

**Tabel 3.5.** Hasil Kuisoner untuk Pemerintah

No.	Pertanyaan	Jawaban
1.	Apakah diperlukan media online yang berfokus untuk berbagi di wilayah Yogyakarta?	Dari 5 koresponden semua menjawab iya
2.	Apakah setuju diterapkannya teknologi web service untuk memaksimalkan pelayanan masyarakat terhadap kegiatan berbagi?	Dari 5 koresponden 4 menjawab setuju dan 1 tidak

**Tabel 3.6.** Hasil Kuisoner untuk Komunitas

No.	Pertanyaan	Jawaban
1.	Apakah diperlukan media online yang berfokus untuk berbagi di wilayah Yogyakarta?	Dari 3 komunitas semua menjawab iya
2.	Fitur apa yang diperlukan komunitas dalam aplikasi berbagi?	Dari 3 komunitas bisa disimpulkan adalah fitur yang bisa memberikan informasi dari komunitas untuk pengguna.

### 3.4 Inovasi

Dari pemaparan problem kota yang ditemukan dan analisis yang menjadi acuan, maka inovasi yang akan dilakukan pada penelitian ini adalah membuat media bernama Jogja Berbagi berbasis web. Jogja Berbagi adalah media yang memfasilitasi masyarakat khususnya daerah Yogyakarta yang ingin berbagi. Permasalahan kota mengenai tingginya angka kemiskinan dan pengangguran ini harapannya bisa diperkecil dengan cara saling membantu dan berbagi kepada yang membutuhkan. Pada pembuatan media Jogja Berbagi ada beberapa teknologi yang akan digunakan adalah bahasa pemrograman menggunakan

PHP, HTML, Javascript. Menggunakan *software* Sublime 3. *Data base* yang diolah adalah MySql. Koneksi *web service* menggunakan *restful* format *Javascript Object Notation* (JSON), *framework* yang digunakan *CodeIgneter 4.0 MVC*, kemudian menggunakan Web Browser Google Chrome, dan menggunakan aplikasi Postman untuk menjalankan atau mengecek fungsi *web server API*nya.

### **3.5 Metodologi Pengembangan Perangkat Lunak**

Pada tahap pengembangan sistem metode yang digunakan adalah UCD (*User Centered Design*). UCD menempatkan pengguna sebagai pusat dari sebuah proses pengembangan sistem. Teknik, metode, alat, prosedur dan proses yang membantu perancangan sistem interaktif dibangun berdasarkan pengalaman pengguna. UCD adalah menerjemahkan partisipasi dan pengalaman manusia ke dalam rancangan (Jagadish, 2014).

#### **3.5.1 Memahami, Menentukan Konteks Penggunaan**

Mengidentifikasi orang yang akan menggunakan produk. Ini akan menjelaskan untuk apa dan dalam kondisi seperti apa mereka akan menggunakan produk. Dasar dari setiap proses UCD adalah untuk memahami pengguna dari produk yang dimaksudkan dan lingkungan penggunaan mereka. Oleh karena itu, proses UCD khususnya dimulai dengan mengidentifikasi pengguna, Ini juga termasuk semua *stakeholders*, atau pengguna tidak langsung, semua yang berhubungan dengan sistem. Pada tahap ini juga, kita mengidentifikasi karakteristik pengguna dan kelompok pengguna.

#### **3.5.2 Menentukan Kebutuhan Pengguna**

Pada dasarnya pada tahap ini adalah tahap penggalian informasi atau data untuk mengumpulkan kebutuhan dari pengguna, kemudian setelah informasi atau data telah terkumpul, dilakukanlah penataan informasi dari data kebutuhan pengguna tersebut, lalu kebutuhan pengguna digambarkan ke dalam berbagai bentuk atau teknik.

### 3.5.3 Membuat Ekosistem Pengguna

Ekosistem pengguna juga bisa diartikan peta hubungan. Pada tahap ini mengidentifikasi orang yang akan saling berhubungan. Ini akan menjelaskan hubungan antar pengguna yang terlibat secara langsung maupun tidak. Lingkungan pengguna dianalisis dari kebutuhan pengguna. Oleh karena itu, proses UCD khususnya dimulai dengan mengidentifikasi pengguna, Ini juga termasuk semua *stakeholders*, atau pengguna tidak langsung, semua yang berhubungan dengan sistem. Pada tahap ini juga, kita mengidentifikasi karakteristik pengguna dan kelompok pengguna.

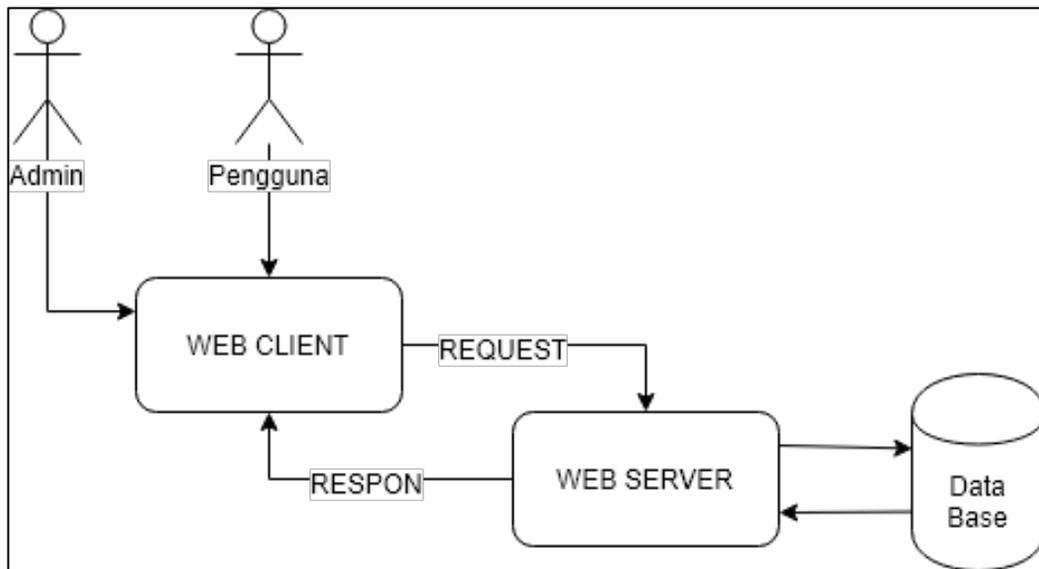
### 3.6 Membangun Desain (Perancangan Sistem)

Perancangan sistem ini adalah Membangun desain sebagai solusi dari produk yang sedang dianalisis. Pada langkah ini, pembuatan alur informasi, perancangan antar muka digunakan untuk membuat ide-ide terlihat dan memfasilitasi komunikasi yang efisien dengan pengguna. Ketika solusi desain disajikan kepada pengguna, mereka juga harus diperbolehkan untuk melaksanakan tugas-tugas. Umpan balik pengguna yang dikumpulkan harus dimasukkan dalam perbaikan solusi desain. Ini harus iterasi terus sampai tujuan desain telah dipenuhi. Tetapi pada penelitian ini akan hanya satu kali iterasi.

#### 3.6.1 Perancangan Arsitektur Sistem

Perancangan aplikasi jogja berbagi ini menggunakan *framework CodeIgneter* yang mengusung konsep MVC (*Model, View dan Controller*). Untuk menerapkan web service maka diperlukan *web client* dan *web server* yang saling terhubung (Aziz, Wiharto, & Wicaksono, 2013). Karena menggunakan konsep MVC maka koneksi antara *web client* dengan *web server* berada di *controller* masing-masing dengan menggunakan *restful JSON*. Tetapi meskipun menggunakan konsep MVC, dalam penerapan rancangan ada yang diabaikan, karena *web client* hanya terfokus pada *view* dan *controller* saja, sedangkan *web server* terfokus pada *controller* dan *models*. Jadi pengguna dan admin hanya akan

berinteraksi dengan *web client*, dan permintaan atau pemasukan data akan diteruskan oleh *web client* ke *web server* untuk disimpan di database. Arsitektur sistem dapat dilihat pada Gambar 3.26.

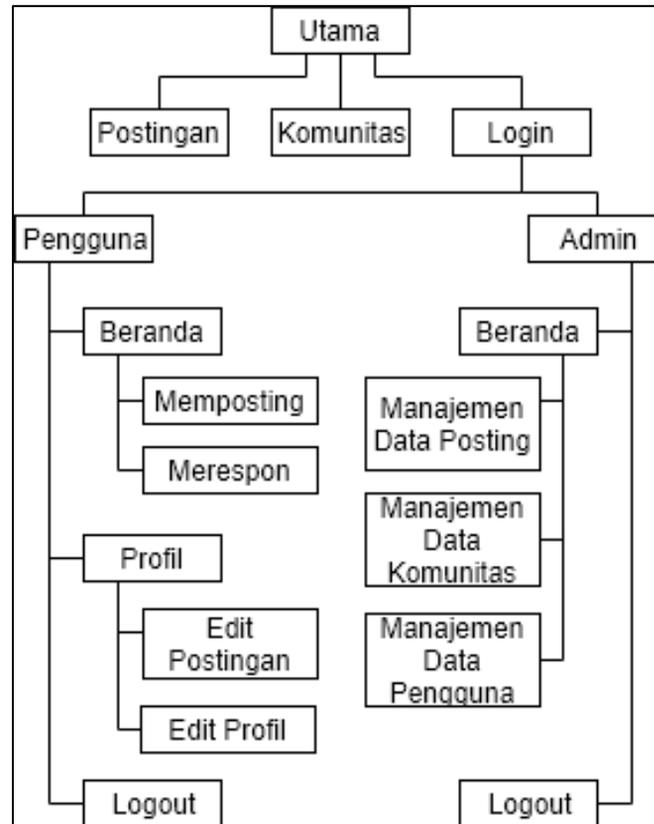


**Gambar 3.26.** Arsitektur Sistem

Perancangan sistem pada jogja berbagi ini ada dua tipe pengguna, yang pertama adalah pengguna terdaftar dan yang kedua adalah admin. Kedua pengguna ini hanya berhubungan dengan *web client* saja. Apapun perintah dari pengguna nantinya akan diproses terlebih dahulu oleh *web client*. Setelah perintah itu sesuai maka *web client* akan meneruskannya ke *web server*. Dalam *web server* mengidentifikasi perintah tersebut untuk mengelola *database*. Ketika perintah berhasil diidentifikasi maka *web server* merespon *web client* sesuai *request*. *Web client* akan menunjukkan hasil respon tadi ke pengguna.

### 3.6.2 Struktur Menu

Struktur menu yang dimiliki pengguna dan admin tentu berbeda. Pada pengguna mempunyai fitur *login*, *logout*, *posting*, edit profil, dan komentar. Sedangkan pada admin lebih fokus untuk manajemen sistem seperti manajemen pengguna, manajemen postingan, manajemen komunitas. Pada rancangan struktur menu digambarkan menu utama pada aplikasi beserta sub-menu yang ada dibawahnya gambar rancangan struktur menu aplikasi dapat dilihat pada Gambar 3.27.



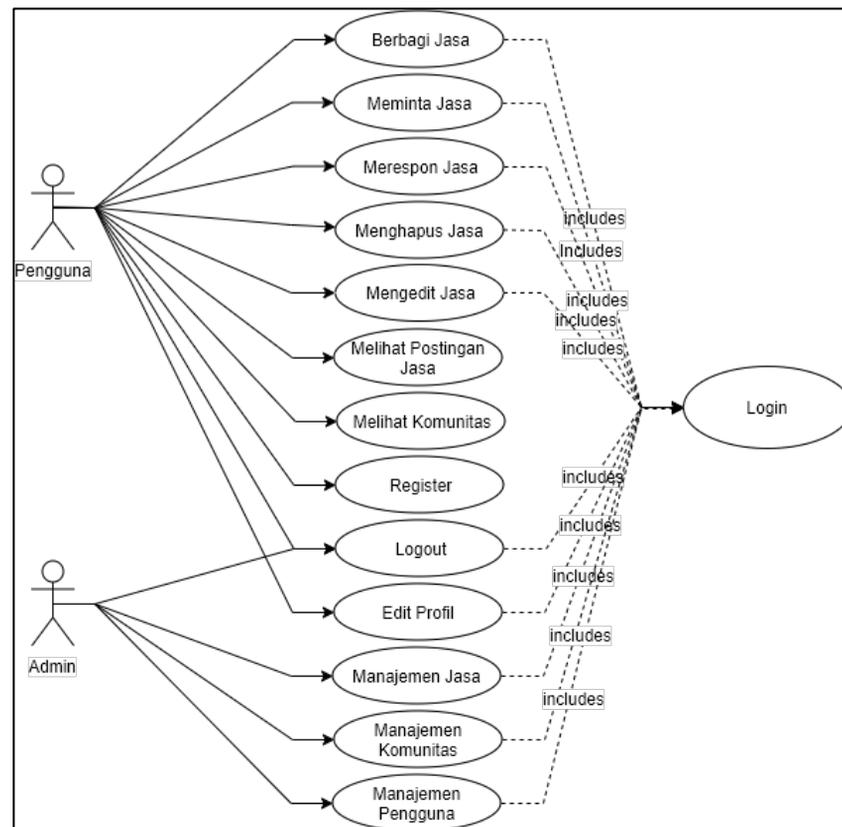
Gambar 3.27. Struktur Menu

Utama adalah halaman utama ketika mengakses aplikasi jogja berbagi. Di halaman Utama pengguna yang belum terdaftar hanya bisa melihat postingan dan komunitas saja. Ketika pengguna atau admin sudah melakukan login maka pengguna bisa memposting atau merespon informasi. Begitu juga bisa mengedit profilnya. Sedangkan admin bisa memanajemen pengguna, data komunitas, dan postingan.

### 3.6.3 Perancangan Model

Perancangan model ini adalah implementasi tahapan pemodelan data (*Data Modelling*) dan pemodelan proses (*Process Modelling*) pada metode RAD untuk memahami alur informasi dan proses yang terjadi dalam sistem Jogja Berbagi. Pada perancangannya menerapkan UML (*Unified Modeling Language*). Terdiri dari *Usecase diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

### 3.6.3.1 Usecase Diagram



**Gambar 3.28.** *Usecase Diagram*

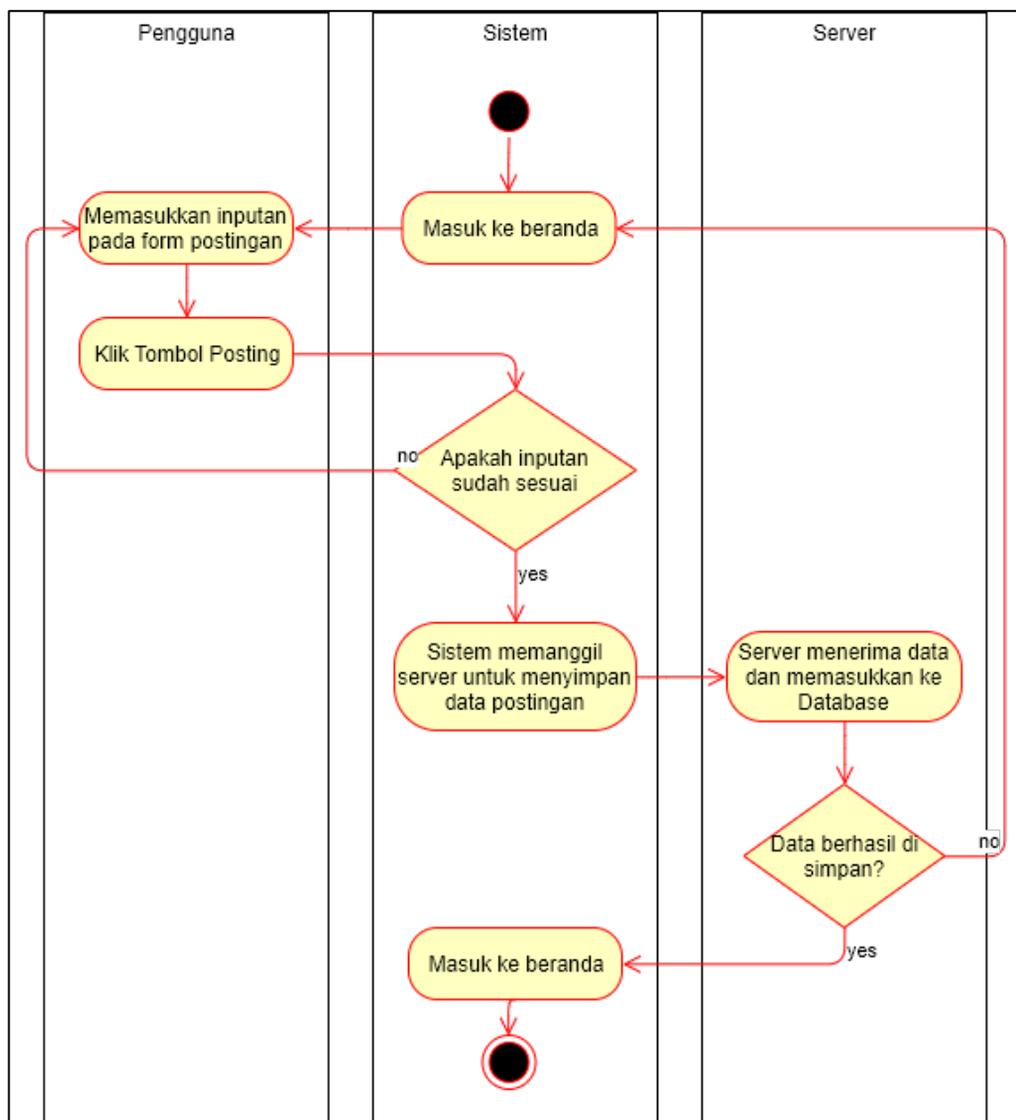
Ada dua aktor pada *usecase* diagram Gambar 3.28. keduanya memiliki fitur yang berbeda. Pada aktor pengguna memiliki fitur untuk bisa membagi unggahan dimana pengguna bisa memberi jasa atau meminta jasa. Kemudian pengguna bisa merespon unggahan pengguna lain dengan cara komentar atau melalui aplikasi *whatsapp*. Pengguna juga bisa mengedit unggahan dan profilnya. Semua fitur tersebut *include login*. Artinya pengguna harus login terlebih dahulu, kalau tidak maka hanya bisa melihat unggahan pengguna lain dan melihat komunitas saja. Kemudian untuk aktor admin fitur yang tersaji adalah selain harus login, admin bisa memanajemen unggahan yang diunggah pengguna selain itu admin bisa memanajemen pengguna. Serta memanajemen komunitas atau LKS.

### 3.6.3.2 Activity Diagram

Perancangan *activity* diagram menggambarkan aktifitas pengguna maupun admin yang ada di usecase diagram.

### 1. Activity diagram berbagi jasa

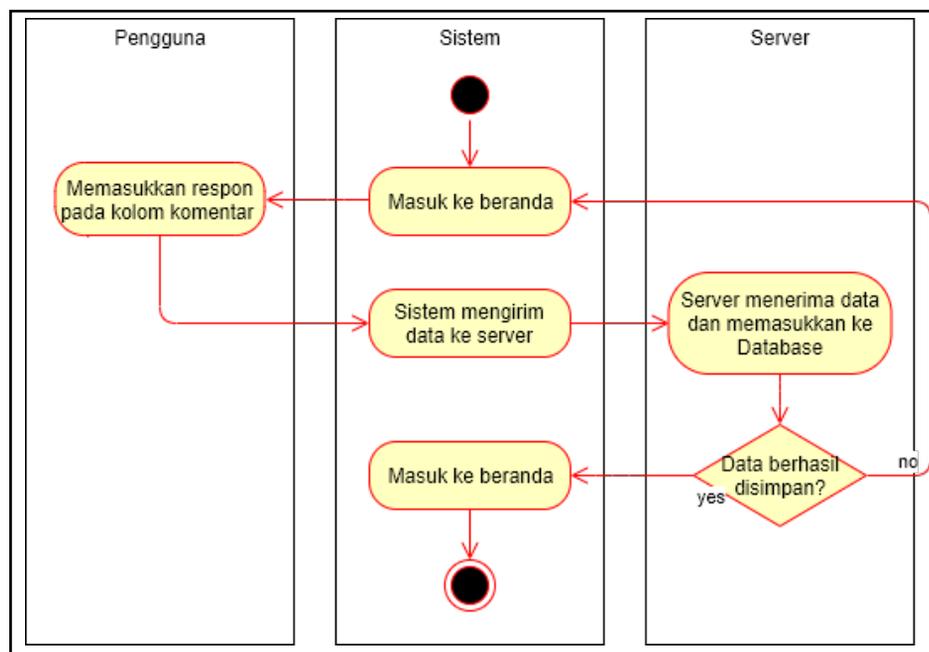
*Activity* diagram berbagi jasa adalah alur ketika pengguna melakukan unggah postingan. Setelah pengguna login, maka akan mengarah ke halaman beranda. Di halaman beranda pengguna memasukkan data postingan seperti judul keterangan, lokasi dan lain sebagainya. Setelah itu inputan diverivikasi oleh sistem apakah sesuai atau belum dengan kolom inputan. Kalau sesuai maka sistem mengirim data ke server untuk disimpan. Server menerima data kemudian meneruskan ke database. Dari data base melakukan *return* apakah data berhasil disimpan atau tidak. Jika tidak maka kembali ke halaman beranda. Activity diagram bebagi bisa dilihat di Gambar 3.29.



**Gambar 3.29.** Activity Diagram Berbagi Jasa

## 2. Activity diagram merespon postingan

Pada activity diagram ini dijelaskan alur ketika pengguna merespon postingan pengguna lain. Tentu tahap ini dilakukan ketika pengguna sudah *login* terlebih dahulu. Dalam tahap merespon pengguna bisa memilih untuk merespon lewat kolom komentar atau lewat whatsapp. Ketika pengguna memilih kolom komentar prosesnya pertama adalah pengguna mengisi komentar pada kolom setelah selesai data komentar akan dikirim ke server dan mengirim ke database untuk disimpan dan database melakukan *return* apakah data berhasil diimpen atau tidak, ketika tidak maka kembali ke halaman beranda. Sedangkan ketika pengguna memilih menggunakan whatsapp maka akan diarahkan ke halaman *whatsapp* web disitu pengguna bisa merespon dengan chat whatsapp. Gambar 3.30. akan memper jelas alur untuk merespon postingan.

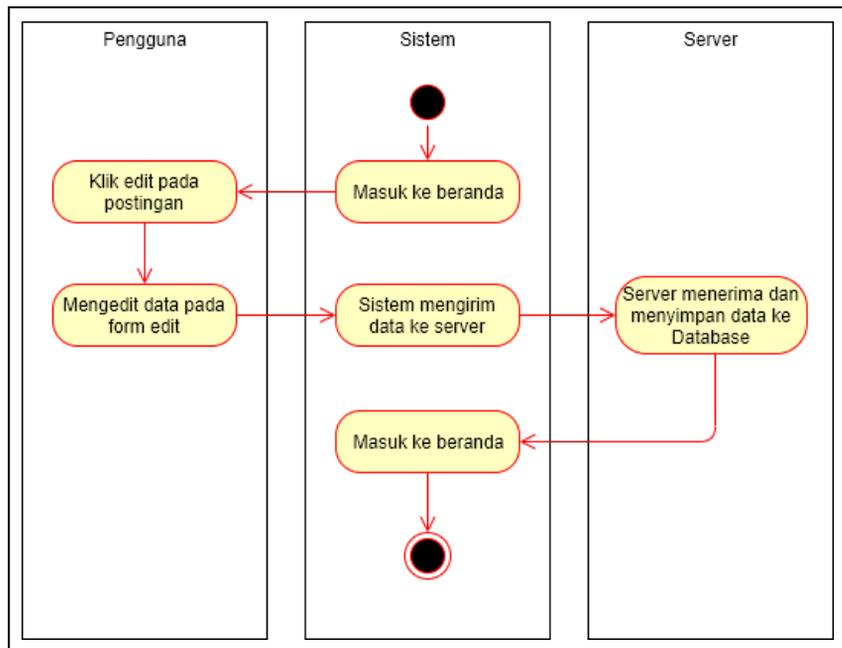


**Gambar 3.30.** Activity Diagram Merespon Jasa

## 3. Activity diagram edit jasa

Diagram ini akan menggambarkan alur ketika pengguna mengedit jasa. Pengguna memilih postingan yang bersangkutan dan klik edit. Sistem akan menampilkan *popup* edit postingan. Pengguna mengedit postingannya. Setelah selesai data postingan dikirim ke

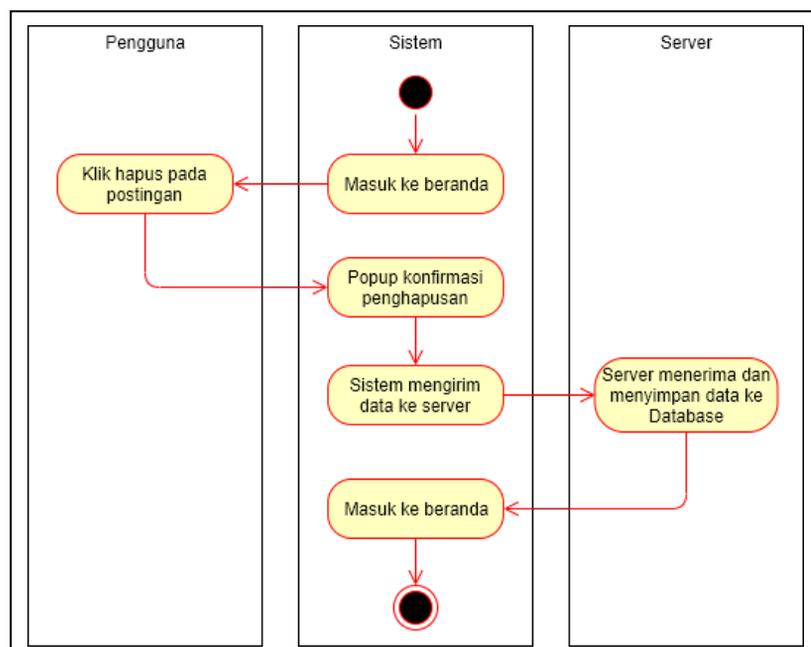
server. Server menerima dan meneruskan ke database untuk disimpan. Gambar alur *activity* diagramnya bisa dilihat pada gambar 3.31.



**Gambar 3.31.** Activity Diagram Edit Jasa

#### 4. Activity diagram hapus jasa

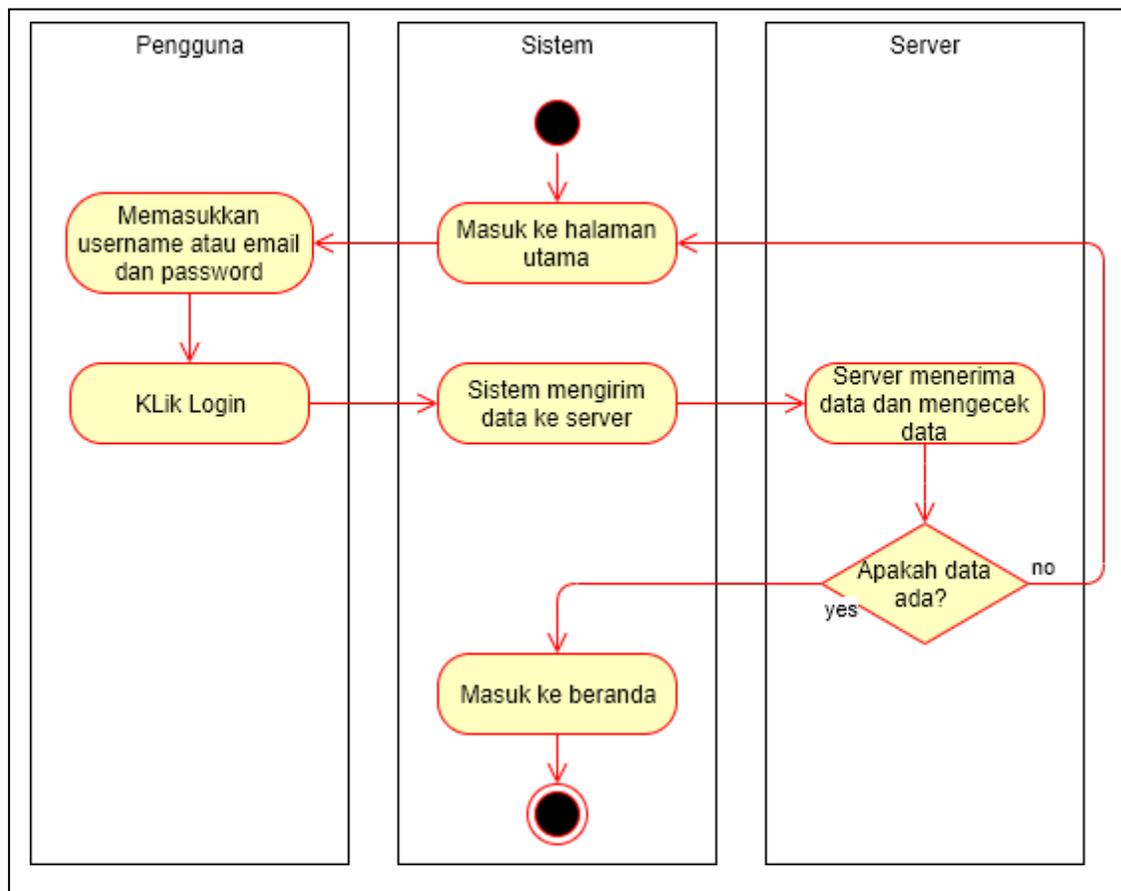
Proses untuk menghapus postingan ketika pengguna mengklik hapus pada postingan sistem akan mengambil id postingan tersebut kemudian dikirim ke server. Pada server meneruskan data id postingan untuk dihapus di database.



**Gambar 3.32.** Activity Diagram Hapus Jasa

### 5. Activity diagram login

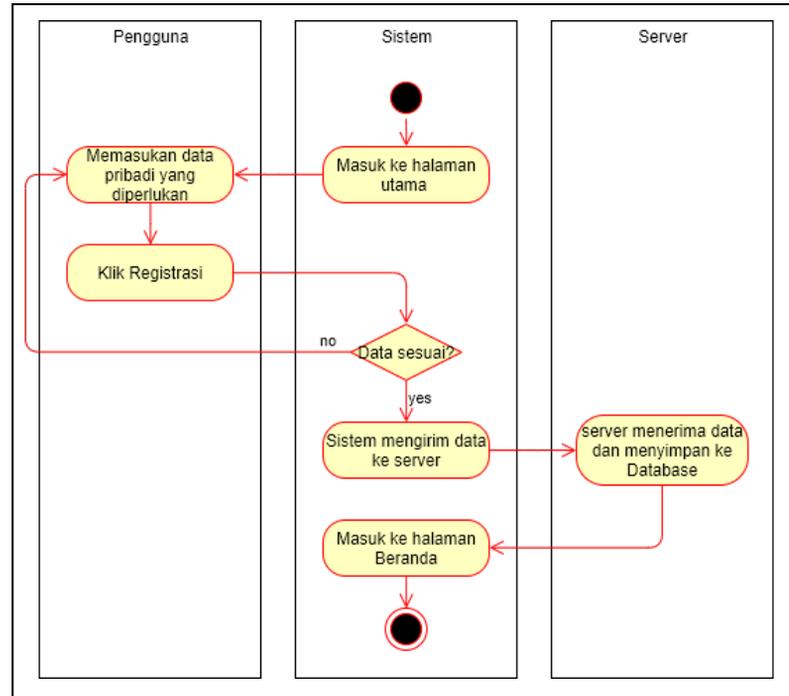
Activity diagram untuk login dimulai dari halaman utama. pengguna mengisi formulir login kemudian sistem akan memverifikasi apakah inputan sudah sesuai atau belum. Jika sudah sesuai maka sistem akan mengirim inputan ke server untuk mengecek apakah username atau email dan password pengguna ada atau tidak. Jika ada kemudian ke halaman beranda. Untuk lebih jelas bisa dilihat di gambar 3.33.



**Gambar 3.33.** Activity Diagram Login

### 6. Activity diagram registrasi

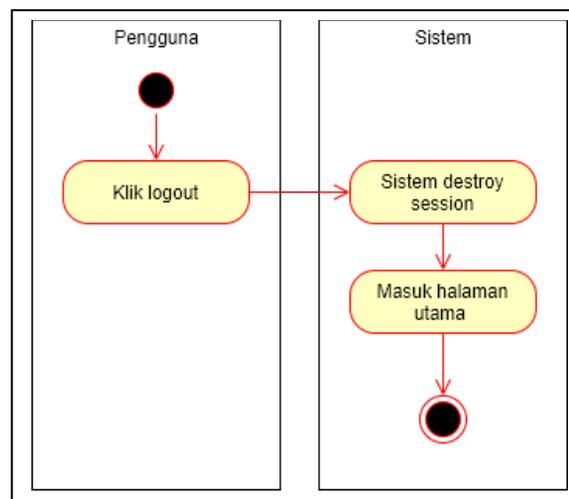
Proses registrasi dimulai dari pengguna yang belum terdaftar mengisi formulir di halaman utama. Setelah selesai sistem akan memverifikasi data inputan apakah sudah sesuai atau belum. Jika sudah maka data akan dikirim ke server untuk disimpan ke database. Jika data berhasil disimpan maka akan langsung masuk ke halaman beranda tanpa harus login dahulu.



**Gambar 3.34.** Activity Diagram Register

#### 7. Activity diagram logout

Proses logout dimulai ketika pengguna mengklik logout, setelah itu sistem akan melakukan penghancuran session agar tidak bisa kembali ke halaman tanpa login. Gambar 3.35. akan memerjelasnya.

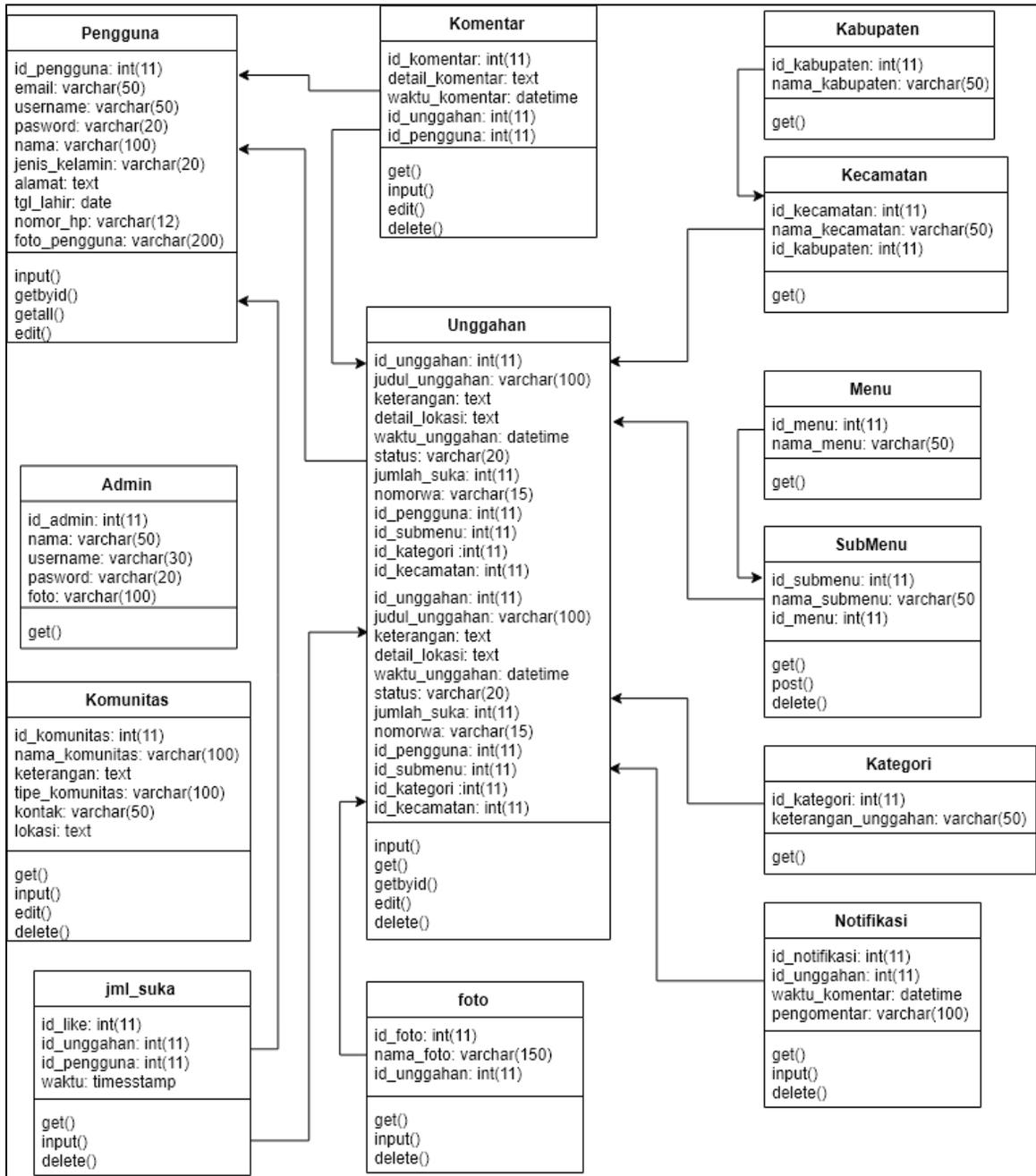


**Gambar 3.35.** Activity Diagram Logout

#### 3.6.3.3 Class Diagram

Class diagram merupakan model statis yang menggambarkan struktur dan deskripsi kelas dan hubungannya antar kelas. Kelas diagram terdiri dari nama kelas, atribut dan

operasi atau method yang digunakan. Didalam aplikasi jogja berbagi perancangan kelas yang di rancang yaitu terdiri dari kelas unggahan, kelas pengguna, kelas admin, kelas kategori, kelas komentar, kelas kabupaten, kelas kecamatan, kelas menu, kelas sub menu, kelas notifikasi, kelas foto, kelas komunitas. Gambar 3.36 akan memperjelas perancangan class diagram pada sistem ini.

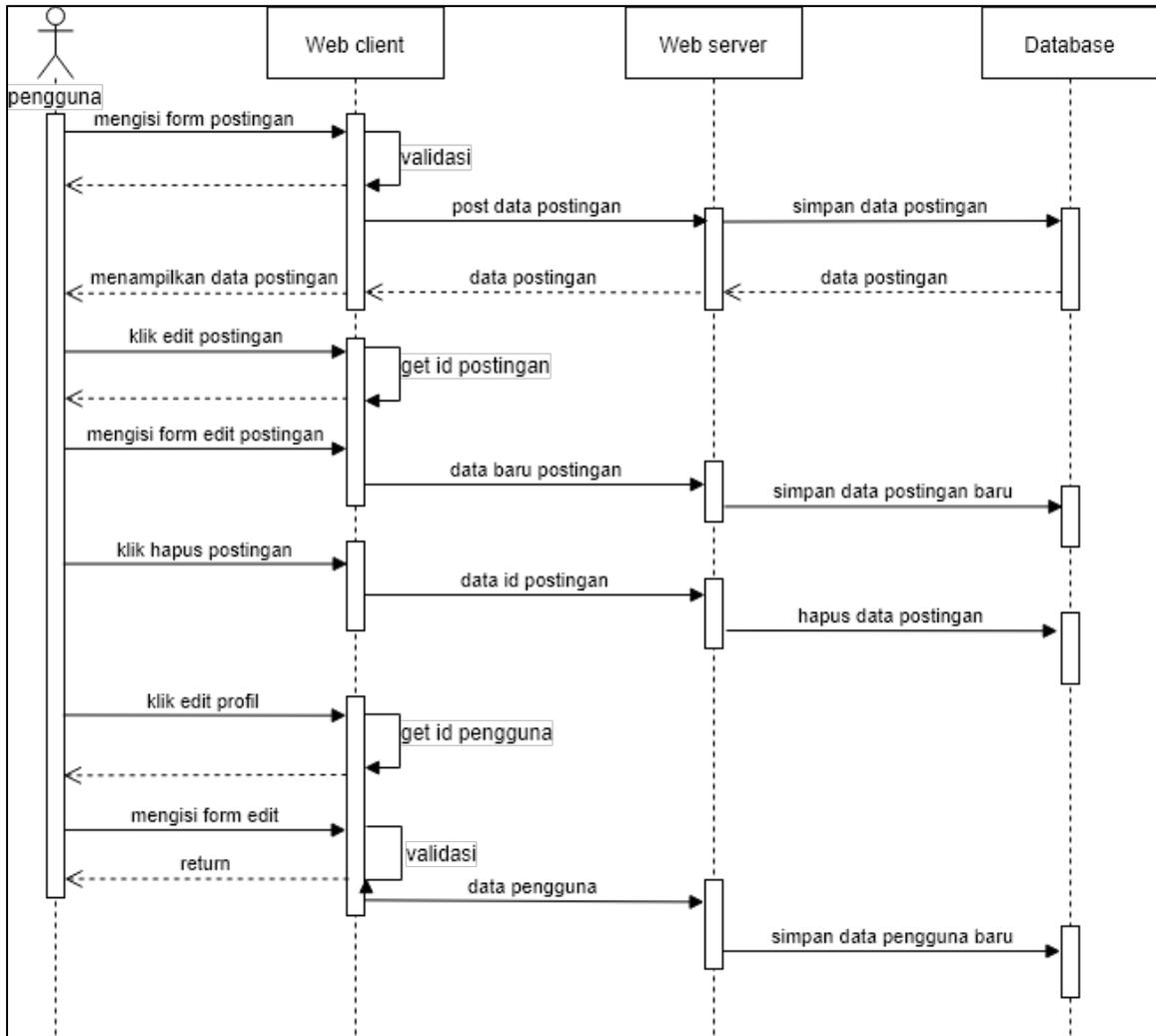


Gambar 3.36. Class Diagram

Pada setiap kelas yang dirancang memiliki atribut yang berkaitan dengan kelas tersebut dan pada kelas memiliki *method input()* yang berfungsi untuk memasukan data ke dalam *database*. Untuk *method get()* berfungsi untuk mengambil semua data yang ada dan untuk *getbyid()* berfungsi untuk mengambil data berdasarkan id data. Untuk *edit()* berfungsi untuk mengedit data dan *delete()* untuk menghapus data. Pada setiap kelas saling berhubungan sesuai kebutuhan kelasnya. Pada kelas pengguna berhubungan dengan kelas komentar, unggahan dan *jml\_suka* karena setiap pengguna nantinya bisa mengunggah, mengomentari, dan menyukai postingan pengguna lain. Dan untuk kelas yang menampung data wilayah kelas kabupaten hanya berkaitan dengan kelas kecamatan karena satu kabupaten memiliki banyak kecamatan. Begitu juga dengan kelas menu dan submenu.

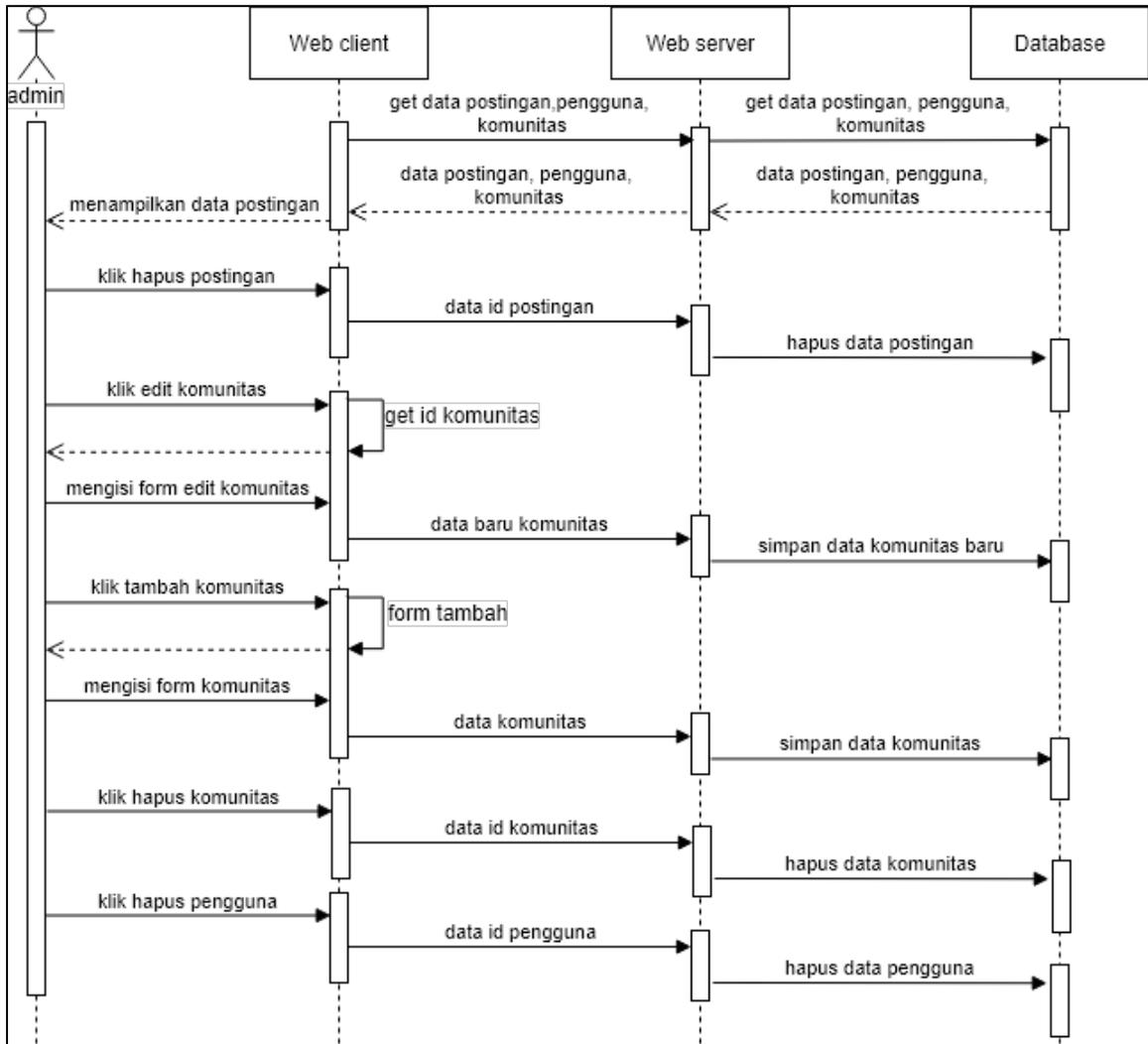
#### 3.6.3.4 Sequence Diagram

Pembuatan *sequence* diagram bertujuan untuk menggambarkan skenario atau langkah-langkah yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan *output* tertentu. Pada *sequence* diagram untuk pengguna memiliki hak akses untuk memposting, merespon, edit, hapus dan mengedit profil. Ketika pengguna memposting proses yang terjadi pertama mengisi formulir pada halaman *web client*, pada *web client* mengecek apakah inputan pada formulir sudah sesuai atau belum, kalau sudah *web client* meneruskan data masukan tadi ke *web server*. dalam *web server* menerimanya kemudian disimpan ke *database*. Begitu juga untuk edit, hapus yang dilakukan oleh pengguna semua proses akan berhubungan dengan *web client* terlebih dahulu, kemudian data yang dikirim adalah data id yang bersangkutan, dikirim ke *web server* untuk disimpan di *database*. *Sequence* diagram apa saja yang dapat dilakukan pengguna dan admin dapat dilihat pada gambar 3.37. dan gambar 3.38.



**Gambar 3.37.** *Sequence Diagram Pengguna*

Dalam *squnce* diagram yang terjadi pada admin, data yang diolah tidak ada inputan, jadi data yang dikelola admin adalah data yang sudah ada kecuali untuk data komunitas. Untuk proses yang terjadi yaitu admin bisa menghapus data pengguna, komunitas dan postingan. Untuk proses edit dan hapus data yang dibawa ke *web client* adalah data id yang bersangkutan. Data tersebut dikirim ke *web server* untuk diedit atau dihapus. Untuk penambahan data komunitas pertama admin mengisi data untuk komunitas yang baru kemudian data akan dikirim ke server untuk disimpan. *Sequence* diagram apa saja yang dapat dilakukan admin dapat dilihat pada gambar 3.38.



Gambar 3.38. Sequence Diagram Admin

### 3.6.4 Perancangan Basis Data

Dalam perancangan basis data ada tiga poin yang akan dibuat yaitu perancangan perancangan tabel dan relasi antar tabel.

#### 3.6.4.1 Perancangan Tabel

Penampungan data yang dibutuhkan sistem ditampung di dalam tabel yang dirancang dan terstruktur. Ada tiga belas tabel dengan struktur masing masing.

##### 1. Tabel Unggahan

Tabel unggahan menampung data unggahan yang diunggah oleh pengguna. Pada tabel tersebut terdiri dari id unggahan sebagai id dari data yang ada dan menjadi *primary key*,

judul unggahan sebagai judul berbaginya, keterangan sebagai keterangan yang ada pada postingan, detail lokasi adalah lokasi jasa yang diposting, waktu unggahan adalah waktu dimana pengguna memposting, status disini sebagai data apakah unggahan terpenuhi atau belum, jumlah suka sebagai data jumlah yang menyukai pada unggahan terkait, nomor wa adalah data nomor wa yang disertakan pada unggahan, kemudian untuk id pengguna, id submenu, id kategori id kecamatan adalah *foreign key* pada tabel ini untuk mengaitkan dengan tabel yang bersangkutan.

**Tabel 3.7.** Unggahan

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Constraint</b>	<b>Keterangan</b>
Id unggahan	Int (11)	<i>Not Null, Primary Key</i>	Id unggahan
judul unggahan	varchar(100)	<i>Not Null</i>	Judul unggahan
keterangan	text	<i>Not Null</i>	Keterangan unggahan
detail lokasi	text	<i>Not Null</i>	Detail lokasi
waktu_unggahan	datetime	<i>Not Null</i>	Waktu ketika mengunggah
status	varchar(20)	<i>Not Null</i>	Terpenuhi atau belum
jumlah suka	int(11)	<i>Not Null</i>	Jumlah suka
nomorwa	varchar(15)	<i>Not Null</i>	Nomor whatsapp
id pengguna	int(11)	<i>Not Null, Foreign key</i>	Id pengguna
id submenu	int(11)	<i>Not Null, Foreign key</i>	Id submenu
id kategori	int(11)	<i>Not Null, Foreign key</i>	Id kategori
id kecamatan	int(11)	<i>Not Null, Foreign key</i>	Id kecamatan

Pada tabel 3.7 id pengguna adalah data id pengguna yang berkaitan dengan data pengguna pada tabel pengguna. Dan id submenu adalah data id submenu yang berisi data kategori bidang seperti pendidikan, sosila dan lain sebagainya. Tabel submenu yang tersambung pada tabel unggahan ini juga bersambung dengan tabel menu yang berisi data jasa dan barang.

## 2. Tabel Pengguna

Tabel pengguna menampung data pengguna yang terdaftar di sistem. Ketika pengguna melakukan registrasi maka data yang sudah di masukkan akan disimpan pada tabel ini. Pada tabel ini terdapat atribut id pengguna sebagai id pengguna pada sistem ini. Email sebagai data email pengguna. *Username* adalah nama yang digunakan ketika masuk ke sistem. Password adalah atribut untuk *login* ke sistem. Nama adalah nama dari pengguna.

Jenis kelamin berisi data laki-laki atau perempuan dari pengguna. Alamat berisi data alamat pengguna. Tgl lahir adalah data tanggal lahir dari pengguna. Nomor hp sebagai data nomor yang dicantumkan pengguna. Dan atribut yang terakhir adalah foto pengguna yang menyimpan foto profil dari pengguna.

**Tabel 3.8.** Pengguna

Nama Field	Tipe Data	Constraint	Keterangan
id_pengguna	int(11)	<i>Not Null, Primary Key</i>	Id pengguna
email	varchar(50)	<i>Not Null</i>	Email pengguna
username	varchar(50)	<i>Not Null</i>	Username pengguna
password	varchar(20)	<i>Not Null</i>	Password pengguna
nama	varchar(100)	<i>Not Null</i>	Nama pengguna
jenis_kelamin	varchar(20)	<i>Not Null</i>	Jenis kelamin
alamat	text	<i>Not Null</i>	Alamat pengguna
tgl_lahir	date	<i>Not Null</i>	Tanggal lahir
nomor_hp	varchar(12)	<i>Not Null</i>	Nomor HP
foto_pengguna	varchar(200)	<i>Not Null</i>	Foto pengguna

Pada tabel pengguna terdapat *primary key* yaitu pada atribut id pengguna, yang nantinya akan berhubungan dengan tabel, unggahan, komentar, dan jml suka. Dan tidak ada *foreign key* pada tabel ini.

### 3. Tabel Komentar

Tabel komentar berisi data komentar yang dibuat oleh pengguna. Pada tabel ini memiliki lima atribut yaitu id komentar sebagai penyimpan id dari komentar yang ada. Detail komentar adalah isi dari komentar pengguna. Waktu komentar berisi waktu saat pengguna melakukan komentar. Id unggahan adalah id unggahan yang bersangkutan dengan komentar. Id pengguna adalah id pengguna yang melakukan komentar.

**Tabel 3.9.** Komentar

Nama Field	Tipe Data	Constraint	Keterangan
id_komentar	int(11)	<i>Not Null, Primary Key</i>	Id komentar
detail_komentar	text	<i>Not Null</i>	Detail komentar
waktu_komentar	datetime	<i>Not Null</i>	Waktu ketika membuat komentar
id_unggah	int(11)	<i>Not Null, Foreign key</i>	Id unggahan
id_pengguna	int(11)	<i>Not Null, Foreign key</i>	Id pengguna

Terdapat *primary key* pada atribut id komentar yang nantinya menjadi data unik. Dan ada dua *foreign key* yaitu pada atribut id unggahan yang berkaitan dengan tabel unggahan dan id pengguna yang berkaitan dengan tabel pengguna.

#### 4. Tabel Foto

Tabel foto berisi data foto yang diunggah bersamaan dengan unggahan oleh pengguna. Tabel ini berisi tiga atribut yaitu id foto sebagai id yang dimiliki oleh data foto. Kemudian nama foto adalah nama data foto. Id unggahan adalah id unggahan yang bersangkutan dengan data foto.

**Tabel 3.10.** Foto

Nama Field	Tipe Data	Constraint	Keterangan
id foto	int(11)	<i>Not Null, Primary Key</i>	Id foto
nama foto	varchar(150)	<i>Not Null</i>	Nama file foto
id unggahan	int(11)	<i>Not Null, Foreign key</i>	Id unggahan

Pada tabel foto memiliki *primary key* pada id foto, dan *foreign key* pada id unggahan yang saling berkaitan.

#### 5. Tabel Komunitas

Tabel komunitas berisi data komunitas yang berhubungan dengan berbagi jasa di wilayah Yogyakarta. Atribut dari tabel komunitas disesuaikan dengan data yang diperoleh dari dinas sosial Yogyakarta. Id komunitas berisi id dari komunitas yang ada. Nama komunitas berisi data nama dari komunitas. Keterangan berisi data deskripsi dari komunitas tersebut. Dan tipe komunitas berisi bidang yang dijalankan oleh komunitas. Atribut kontak berisi kontak yang bisa dihubungi. Dan alamat berisi data alamat komunitas.

**Tabel 3.11.** Komunitas

Nama Field	Tipe Data	Constraint	Keterangan
<i>id komunitas</i>	int(11)	<i>Not Null, Primary Key</i>	Id komunitas
nama komunitas	varchar(100)	<i>Not Null</i>	Nama komunitas
keterangan	text	<i>Not Null</i>	Penjelasan komunitas
tipe_komunitas	varchar(100)	<i>Not Null</i>	Tipe atau jenis komunitas
kontak	varchar(50)	<i>Not Null</i>	Kontak komunitas
alamat	text	<i>Not Null</i>	Alamat komunitas

## 6. Tabel Notifikasi

Tabel notifikasi berisi data notifikasi ketika unggahan pengguna yang bersangkutan dikomentari. Atribut yang dimiliki adalah id notifikasi sebagai id yang dimiliki kemudian id unggahan sebagai id unggahan yang berkaitan, waktu komentar adalah waktu dimana pengguna berkomentar. Dan pengomentor berisi nama pengguna yang berkomentar.

**Tabel 3.12.** Notifikasi

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Constraint</b>	<b>Keterangan</b>
id notifikasi	int(11)	<i>Not Null, Primary Key</i>	Id notifikasi
id unggahan	int(11)	<i>Not Null, Foreign key</i>	Id unggahan
waktu_komentar	datetime	<i>Not Null</i>	Waktu ketika membuat komentar
pengomentor	varchar(100)	<i>Not Null</i>	Nama dari pengomentor

## 7. Tabel Kabupaten

Tabel kabupaten berisi data kabupaten di wilayah Yogyakarta. Memiliki dua atribut yaitu id kabupaten yang berisi id data dan nama kabupaten yang berisi data nama kabupaten seluruh Yogyakarta.

**Tabel 3.13.** Kabupaten

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Constraint</b>	<b>Keterangan</b>
id kabupaten	int(11)	<i>Not Null, Primary Key</i>	Id kabupaten
Nama_kabupaten	varchar(50)	<i>Not Null</i>	Nama kabupaten

## 8. Tabel Kecamatan

Tabel kecamatan berisi data kecamatan yang ada di wilayah Yogyakarta. Atribut yang ada adalah id kecamatan untuk data uniknya. Kemudian nama kecamatan. Dan id kabupaten untuk berkaitan dengan tabel kabupaten.

**Tabel 3.14.** Kecamatan

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Constraint</b>	<b>Keterangan</b>
id kecamatan	int(11)	<i>Not Null, Primary Key</i>	Id kecamatan
nama kecamatan	varchar(50)	<i>Not Null</i>	Nama kecamatan
id kabupaten	int(11)	<i>Not Null, Foreign key</i>	Id kabupaten

## 9. Tabel Menu

Tabel menu berisi kategori unggahan yaitu jasa atau barang. Tabel ini memiliki dua atribut yaitu id menu dan nama menu. Id menu berisi data id sebagai data yang unik. Dan nama adalah nama data menu yang berisi jasa atau barang.

**Tabel 3.15.** Menu

Nama Field	Tipe Data	Constraint	Keterangan
id_menu	int(11)	<i>Not Null, Primary Key</i>	Id menu
nama_menu	varchar(50)	<i>Not Null</i>	Nama menu

## 10. Tabel Submenu

Tabel submenu adalah tabel yang berisi dari kategori unggahan berdasarkan menu yaitu jasa atau barang. Dalam tabel ini memiliki tiga atribut yaitu id submenu, nama submenu, dan id menu. Data pada tabel ini berisi kategori dari menu, misal pendidikan, sosial, lingkungan dan lain sebagainya yang berhubungan dengan tabel menu.

**Tabel 3.16.** Submenu

Nama Field	Tipe Data	Constraint	Keterangan
id_submenu	int(11)	<i>Not Null, Primary Key</i>	Id submenu
nama_submenu	varchar(50)	<i>Not Null</i>	Nama submenu
id_menu	int(11)	<i>Not Null, Foreign key</i>	Id menu

## 11. Tabel Kategori\_Unggahan

Tabel ini berisi data tentang keterangan unggahan apakah memberi atau meminta. Berisi atribut id kategori dan keterangan unggahan yang memiliki data unggahan dengan kategori memberi atau meminta.

**Tabel 3.17.**Kategori Unggahan

Nama Field	Tipe Data	Constraint	Keterangan
id_kategori	int(11)	<i>Not Null, Primary Key</i>	Id kategori
Keterangan_unggah	varchar(50)	<i>Not Null</i>	Nama keterangan

## 12. Tabel Jml\_suka

Tabel jml suka adalah tabel yang berisi data jumlah suka pada unggahan. Memiliki empat atribut yaitu id like sebagai id data, id unggahan sebagai id unggahan yang berkaitan,



Pada sistem ini memiliki beberapa tabel yang saling berhubungan. Hubungan tersebut menunjukkan adanya hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi antar tabel dapat dilihat pada Gambar 3.39. pada tabel unggahan memiliki hubungan dengan tabel jml suka, komentar, foto, kecamatan, pengguna, dan sub menu. Pada tabel pengguna memiliki hubungan dengan tabel jml suka, komentar, dan unggahan, kemudian tabel admin dan komunitas tidak ada relasi karena tidak ada proses yang saling berkaitan dengan tabel lain.

### 3.6.5 Perancangan Antar Muka

Rancangan antar muka aplikasi ada dua bagian utama yaitu perancangan antar muka pengguna dan admin. Pada perancangan antar muka pengguna memiliki fitur yang berbeda dengan admin. Antar muka pada pengguna lebih fokus pada fitur untuk membagi, merespon unggahan. Sedangkan admin lebih fokus untuk manajemen sistem.

#### 3.6.5.1 Perancangan Antar Muka Pengguna

##### 1. Halaman antar muka untuk register dan login

Pada halaman ini adalah halaman utama sebelum pengguna masuk ke beranda. Halama register dan login berada pada satu halaman yaitu halaman utama. Halaman utama dapat dilihat pada Gambar 3.40. dan Gambar 3.41.

The image shows a web interface for 'Jogja Berbagi' with the tagline 'Berbagi Barang dan Jasa'. The page features a navigation menu with 'Komunitas' and 'Tentang Kami'. A central message reads 'Selamat Datang di Jogja Berbagi, Ayo Berbagi !'. To the right is a registration form with the following fields and options:

- Buttons: Register (highlighted in red), Login
- Input fields: NAMA, USERNAME, BUAT PASSWORD, KETIK ULANG PASSWORD, EMAIL, TANGGAL LAHIR, ALAMAT
- Form elements: JENIS KELAMIN (radio buttons for LAKI-LAKI and PEREMPUAN), UNGGAH FOTO PROFIL
- Submit button: REGISTER (highlighted in red)

**Gambar 3.40.** Rancangan Antar Muka Register Pengguna

Halaman untuk registrasi terdapat logo dari aplikasi, navigasi tentang komunitas yang nantinya menuju daftar komunitas, kemudian tentang kami adalah halaman yang berisi deskripsi aplikasi. Pada formulir registrasi ada sembilan kolom yang harus diisi oleh pengguna yang ingin mendaftar. Kolom nama berisi data nama pengguna, kolom *username* berisi nama yang digunakan saat diaplikasi, kolom buat *password* adalah untuk membuat data *password* yang nantinya dipakai untuk *login*. Kemudian ada kolom email, jenis kelamin, tanggal lahir pengguna, alamat dan foto. Ketika salah satu kolom tidak diisi atau tidak sesuai maka akan ada pemberitahuan agar pengguna mengisi dengan benar.

**Gambar 3.41.** Rancangan Antar Muka Login Pengguna

Perancangan antar muka untuk halaman *login* tidak jauh berbeda dengan registrasi karena terdapat pada satu bagian. Pada kolom *login* ada *username* atau email dan *password* untuk diisi agar bisa menuju halaman beranda.

## 2. Halaman antar muka untuk beranda

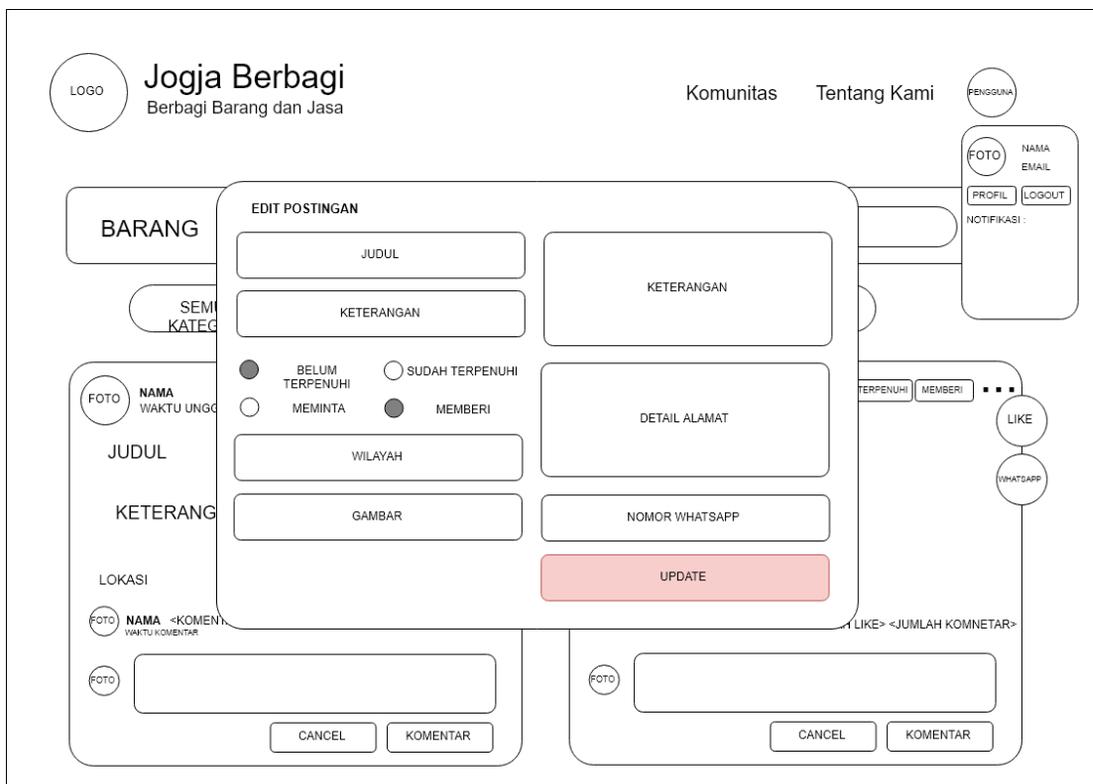
Halaman beranda adalah halaman ketika pengguna sudah melakukan *register* atau *login*. Didalam halaman beranda berisi formulir untuk mengunggah postingan, daftar postingan dari pengguna lain dan notifikasi. Dapat dilihat pada Gambar 3.42.

**Gambar 3.42.** Rancangan Antar Muka Formulir Postingan

Pada gambar 3.43. terdapat logo aplikasi, formulir untuk membagi unggahan yang terdiri dari judul, kategori, keterangan, detail alamat, wilayah, keterangan membagi atau dibagi, nomor whatapp dan gambar pendukung dan tombol untuk memposting. Selain itu ada navigasi untuk profil di situ terdapat foto profil, tombol edit profil, tombol logout dan dibawahnya ada daftar notifikasi yang dimiliki.

**Gambar 3.43.** Rancangan Antar Muka Daftar Unggahan

Rancangan antar muka daftar unggahan memiliki tombol untuk memilih unggahan yang termasuk barang atau jasa. Dan ketika salah satu dipilih juga ada pilihan kategori seperti pendidikan, sosial, dan lain sebagainya. Ada juga fitur untuk mencari unggahan. Pada bagian unggahan terdapat foto pengguna yang mengunggahnya, disertai nama dan waktu unggahannya. Kemudian ada keterangan sudah terpenuhi atau belum, dan keterangan meminta atau memberi. Untuk navigasi edit dan hapus unggahan terdapat pada tombol titik tiga. Sistem menyukai unggahan terdapat tombol *like* untuk menyukai unggahan pada samping bagian. Dalam antarmuka ini juga terdapat kolom komentar untuk merespon unggahan oleh pengguna lain, selain itu juga ada tombol whatsapp untuk merespon unggahan melalui whatsapp. Untuk bagian unggahannya sendiri terdapat judul, keterangan kemudian gambar ketika disertakan gambar, di bawahnya ada alamat yang dicantumkan oleh pengguna.



**Gambar 3.44.** Rancangan Antar Muka Edit Unggahan

Antar muka untuk edit unggahan berupa *popup* yang berisi kolom-kolom dan *radio button*. Kolom-kolom tersebut berisi data sesuai unggahan yang diedit oleh pengguna. Dan juga terdapat tombol untuk mengupdate.

### 3. Halaman untuk profil

Pada halaman ini berisi halaman untuk mengedit profil pengguna.

**Gambar 3.45.** Rancangan Antar Muka Edit Profil

Pada rancangan antar muka edit profil terdapat dua navigasi yaitu *timeline* dan edit profil namun akan dijelaskan rancangan antar muka yang edit profil saja. Pada halaman edit profil terdapat formulir dengan kolom-kolom yang sudah terisi data pengguna yang bersangkutan. Data dari pengguna diperoleh ketika pengguna melakukan registrasi. Selain kolom-kolom juga ada dua tombol reset dan simpan.

### 4. Halaman untuk komunitas

Rancangan antar muka untuk halaman komunitas ada fitur mencari komunitas dan dibawahnya ada daftar komunitas. Daftar komunitas dilengkapi dengan nama komunitas, tipe komunitas, kontak kemudian alamat komunitas. Halaman untuk melihat daftar komunitas dapat dilihat pada gambar 3.46.



**Gambar 3.46.** Rancangan Antar Muka Komunitas

## 5. Halaman untuk tentang kami

Pada halaman ini berisi tentang aplikasi jogja berbagi.

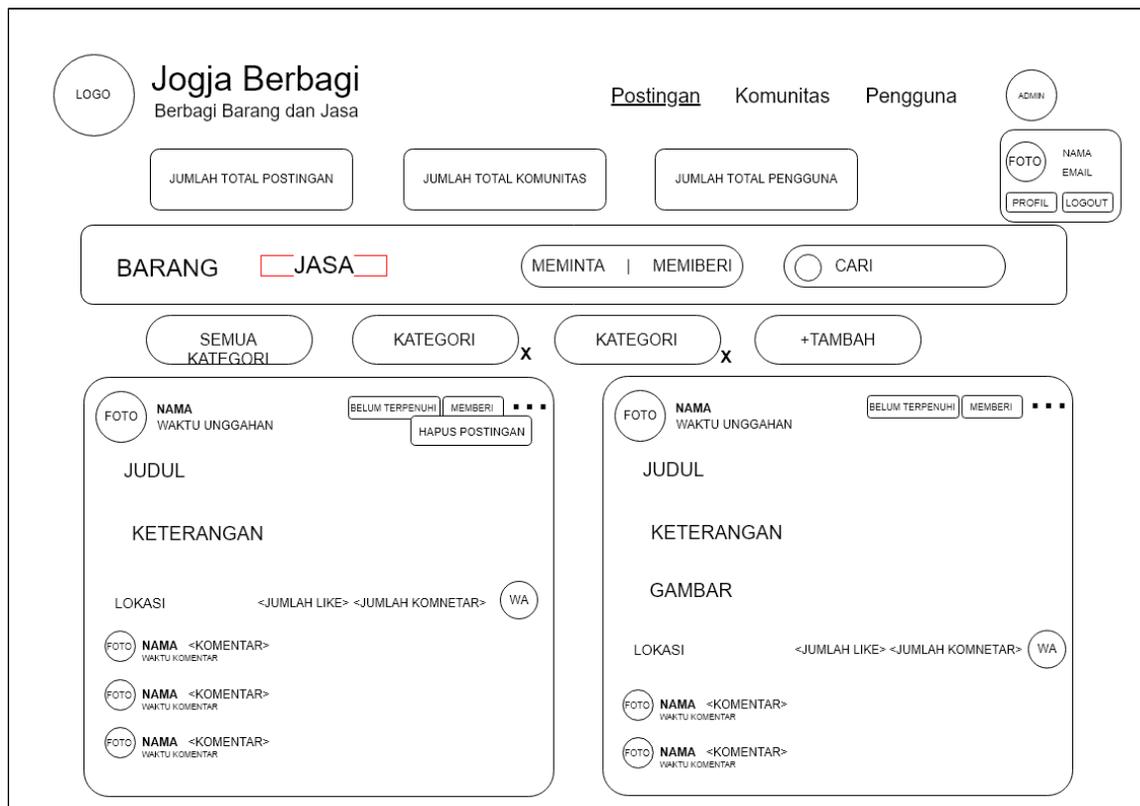


**Gambar 3.47.** Rancangan Antar Muka Tentang Kami

Rancangan antar muka untuk halaman tentang kami berisi logo aplikasi kemudian deskripsi dari aplikasi jogja berbagi dengan dilengkapi gambar.

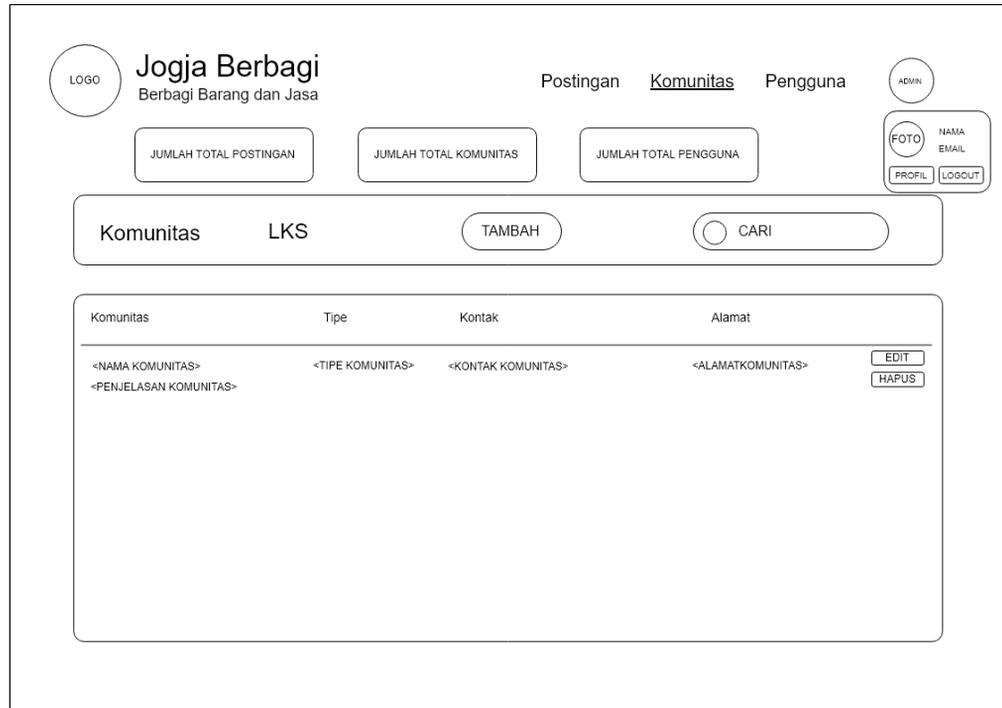
### 3.6.5.2 Perancangan Antar Muka Admin

Perancangan antar muka admin terdiri dari halaman untuk manajemen postingan, komunitas dan pengguna. Gambar 3.48., gambar 3.49., dan gambar 3.50. akan memperjelas rancangannya.



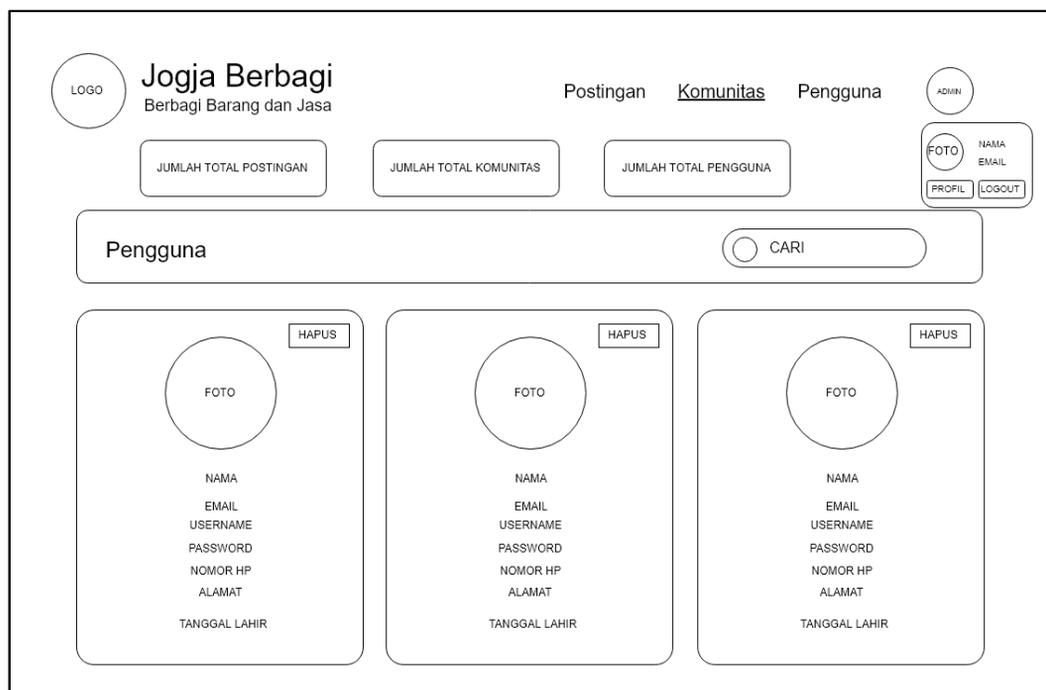
**Gambar 3.48.** Rancangan Antar Muka Manajemen Unggahan

Dalam perancangan antar muka admin memiliki tampilan untuk mengetahui jumlah unggahan, jumlah pengguna, dan total jumlah komunitas. Untuk manajemen unggahan admin memiliki tampilan sama dengan pengguna, yang membedakan adalah ada fitur menambah dan menghapus kategori dan admin memiliki fitur hapus unggahan. Dan untuk navigasi profil terdapat tombol *logout*.



**Gambar 3.49.** Rancangan Antar Muka Manajemen Komunitas

Rancangan antar muka untuk memanajemen komunitas terdapat fitur tambah komunitas. Ketika menambah komunitas akan muncul *popup* untuk menambah komunitas oleh admin. Selain itu pada rancangan ini terdapat tombol edit dan hapus untuk setiap daftar komunitas yang ada.



**Gambar 3.50.** Rancangan Antar Muka Manajemen Pengguna

Perancangan antar muka untuk halaman manajemen pengguna memiliki fitur pencarian dan daftar pengguna terdaftar. Dalam daftar pengguna terdapat foto, nama dan data-data pengguna kecuali password. Dan untuk fitur adminnya terdapat tombol hapus pengguna.

### 3.7 Perancangan Evaluasi Aplikasi (Evaluate Design)

Pada tahap ini solusi desain yang pada tahap sebelumnya harus dievaluasi.. Tujuannya adalah untuk menghasilkan umpan balik untuk lebih meningkatkan produk dan untuk menentukan apakah desain memenuhi kebutuhan pengguna yang ditentukan, tujuan kegunaan dan sesuai dengan pedoman kegunaan umum. Evaluasi aplikasi akan dilakukan dengan menggunakan kuisioner kepada pengguna. Untuk rancangan pertanyaan kuisioner bisa dilihat pada tabel 3.20.

**Tabel 3.20.** Rancangan Evaluasi Aplikasi

No	Pertanyaan	Ya	Tidak
<b>Fungsionalitas Sistem</b>			
1.	Sistem dapat menampilkan informasi berbagi berupa unggahan beserta atributnya?		
2.	Sistem dapat memasukkan data unggahan kemudian ditampilkan?		
3.	Sistem dapat menghapus unggahan?		
4.	Sistem dapat mengedit data unggahan?		
5.	Sistem dapat memverifikasi dan memasukkan data profil saat registrasi?		
6.	Sistem dapat memverifikasi data saat <i>login</i> ?		
7.	Sistem dapat mengedit profil?		
8.	Sistem dapat memasukkan data komentar kemudian ditampilkan?		
9.	Sistem dapat mengedit komentar?		
10.	Sistem dapat menghapus komentar?		
11.	Sistem dapat terhubung dengan aplikasi Whatsapp?		
12.	Sistem dapat menampilkan informasi komunitas?		
<b>Fungsionalitas Aplikasi</b>			
13.	Tampilan halaman utama sesuai dan nyaman?		
14.	Tampilan halaman beranda sesuai dan nyaman?		
15.	Tampilan halaman profil sesuai dan nyaman?		
16.	Paduan warna sesuai dan nyaman?		
17.	Letak kontrol navigasi sesuai dan nyaman?		
18.	Apakah Aplikasi mudah digunakan untuk sarana berbagi?		

## BAB IV

### HASIL, PENGUJIAN DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai implementasi dari apa yang sudah dirancang pada bab sebelumnya. Sistem akan dioperasikan pada keadaan yang sebenarnya. Pada bab ini juga akan mengetahui apakah dapat menghasilkan tujuan yang diinginkan berdasarkan pada analisis dan perancangan yang telah dilakukan. Pembahasan akan dibagi berdasarkan tahapan perancangan.

#### 4.1 Memahami, Menentukan Konteks Penggunaan

Tahap ini merupakan dasar dari setiap metode UCD, yaitu untuk memahami siapa pengguna dari produk yang dimaksudkan dan lingkungan penggunaan mereka. Termasuk juga mengidentifikasi *stakeholders*, atau siapa saja yang terlibat secara langsung atau tidak langsung dalam proses pengembangan sistem atau aplikasi. Hasil dari tahap ini sebagai berikut:

1. *Application Designer and Programmer* sebagai yang menganalisis kebutuhan sistem, merancang sistem dan membangun sistem.
2. *Operation staff*, adalah *staff* atau pegawai dari Pemerintahan atau yang mengelola aplikasi, yang bertanggung jawab untuk mengoperasikan sistem atau aplikasi pada bagian admin.
3. Pengguna aplikasi yaitu masyarakat Yogyakarta.

#### 4.2 Menentukan Kebutuhan Pengguna

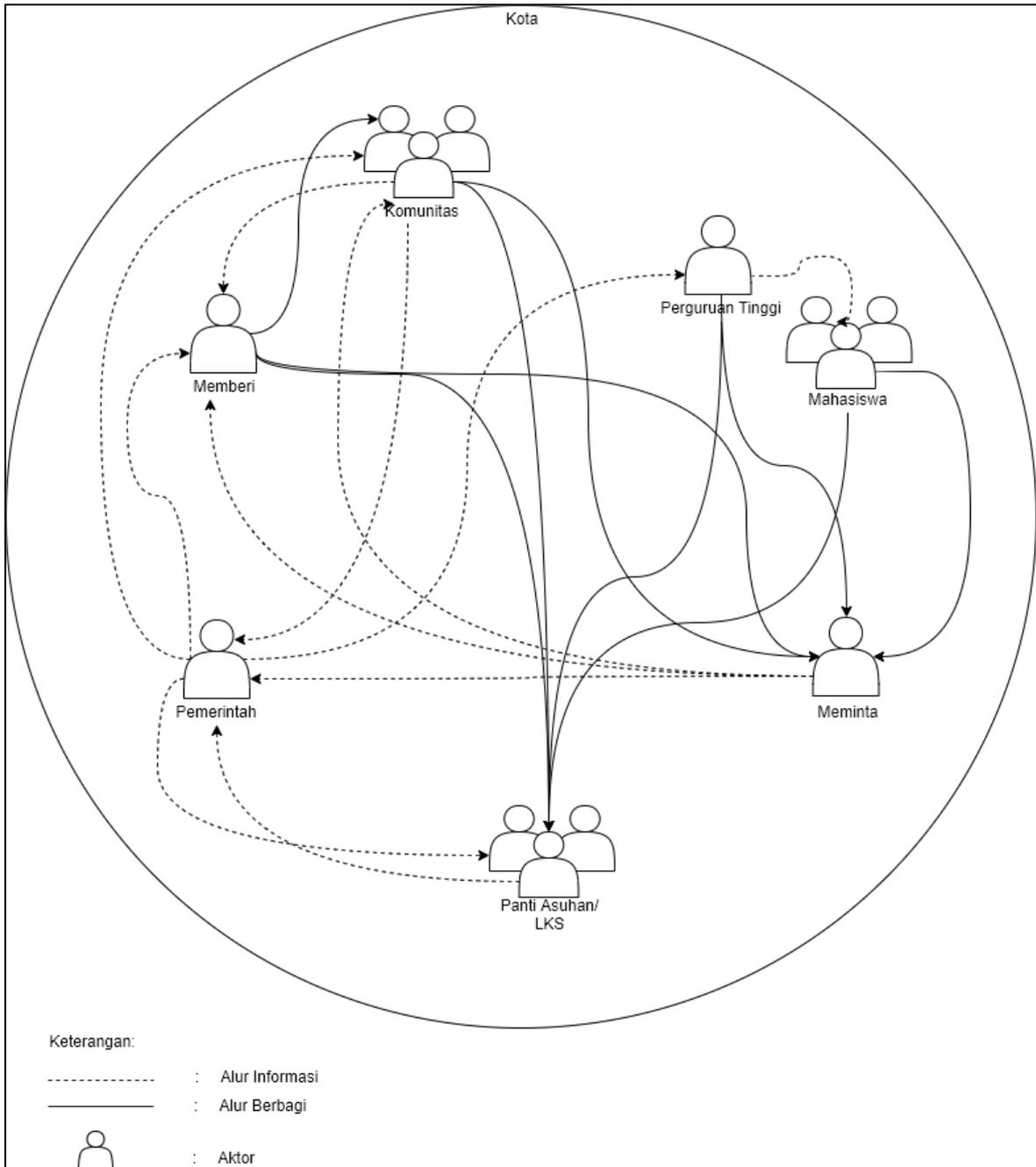
Pada tahap ini adalah tahap penggalian informasi atau data untuk mengumpulkan kebutuhan dari pengguna, kemudian setelah informasi atau data telah terkumpul, dilakukanlah penataan informasi dari data kebutuhan pengguna tersebut. Dari data hasil kuisioner maka ditentukan kebutuhan fungsional pengguna sebagai berikut:

**Tabel 4.1. Kebutuhan Fungsional Pengguna**

No.	Nama Fungsi	Keterangan
1.	Unggahan	Memberikan informasi berbagi berbentuk unggahan
2.	Meminta atau memberi	Fitur untuk bisa memberi atau meminta barang maupun jasa.
3.	Kategori	Keterangan informasi dalam kategori berbagi
4.	Keterangan sudah terpenuhi atau belum	Keterangan informasi sebagai informasi yang sudah terpenuhi atau belum
5.	Komentar	Fitur merespon unggahan dengan komentar
6.	Koneksi dengan aplikasi whatsapp	Fitur merespon unggahan dengan aplikasi whatsapp
7.	Suka tidak suka	Fitur untuk menyukai atau tidak sebuah unggahan
8.	Keterangan wilayah	Keterangan wilayah yang ada pada informasi berbagi
9.	Upload gambar	Fitur menambahkan gambar pada unggahan
10.	Edit unggahan	Menampilkan form untuk mengedit unggahan
11.	Edit profil	Menampilkan form untuk mengedit profil
12.	Komunitas	Memberikan informasi komunitas yang aktif dalam berbagi

### 4.3 Ekosistem Pengguna

Tahapan pembuatan ekosistem yang merancang alur dari interaksi aplikasi berbagi pada penelitian ini. Dalam perancangan ini terdapat pengguna yang memberi, pengguna yang memina, ada komunitas, pemerintah, perguruan tinggi dan mahasiswa, dan ada panti asuhan atau LKS (Lembaga Kesejahteraan Sosial). Ekosistem yang dirancang ruang cangkupnya adlah kota, ekosistem ini menjelaskan tentang alur informasi dan alur berbagi yang terjadi. Alur informasi yang terjadi terletak antara hubungan pemerintah ke pengguna yang memeberi, komunitas, panti, pengguna yang meminta, dan kepada perguruan tinggi. Selain itu pengguna meminta dan panti juga memberi informasi ke pemerintah. Kemudian alur berbagi terjadi pada hubungan pengguna memberi ke pengguna meminta, komunitas, panti. Selain itu komunitas juga berbagi ke pengguna meminta dan panti. Untuk perguruan tinggi dan mahasiswa berbagi ke pengguna meminta dan panti juga. Penjelasan lebih detail bisa dilihat pada gambar 4.1.



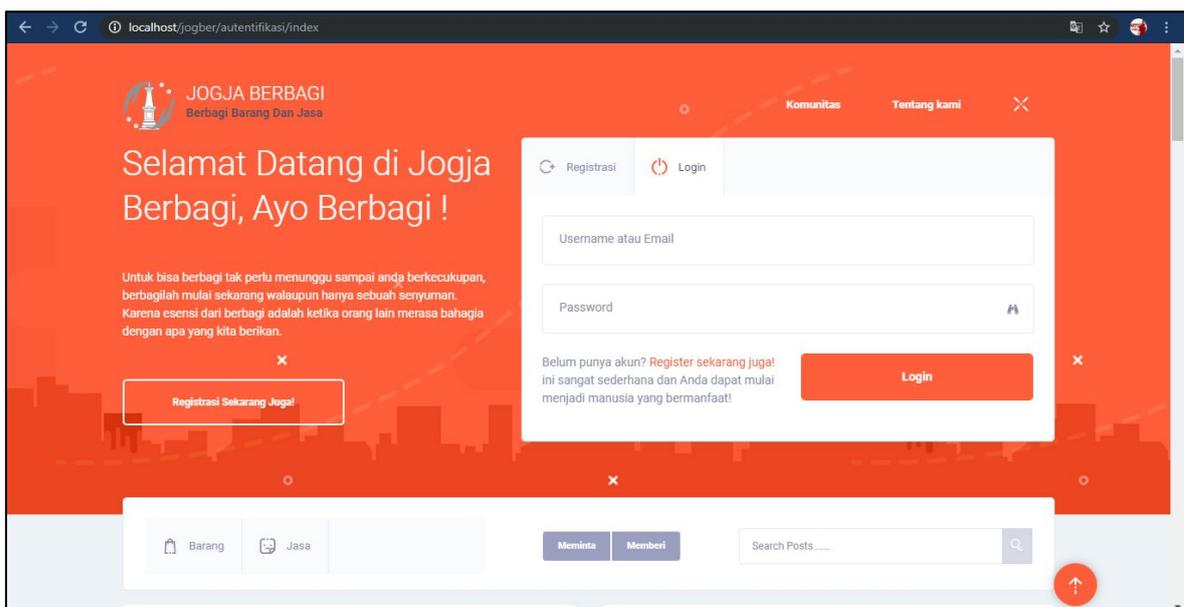
**Gambar 4.1.** Ekosistem Pengguna

#### 4.4 Hasil Perancangan Sistem

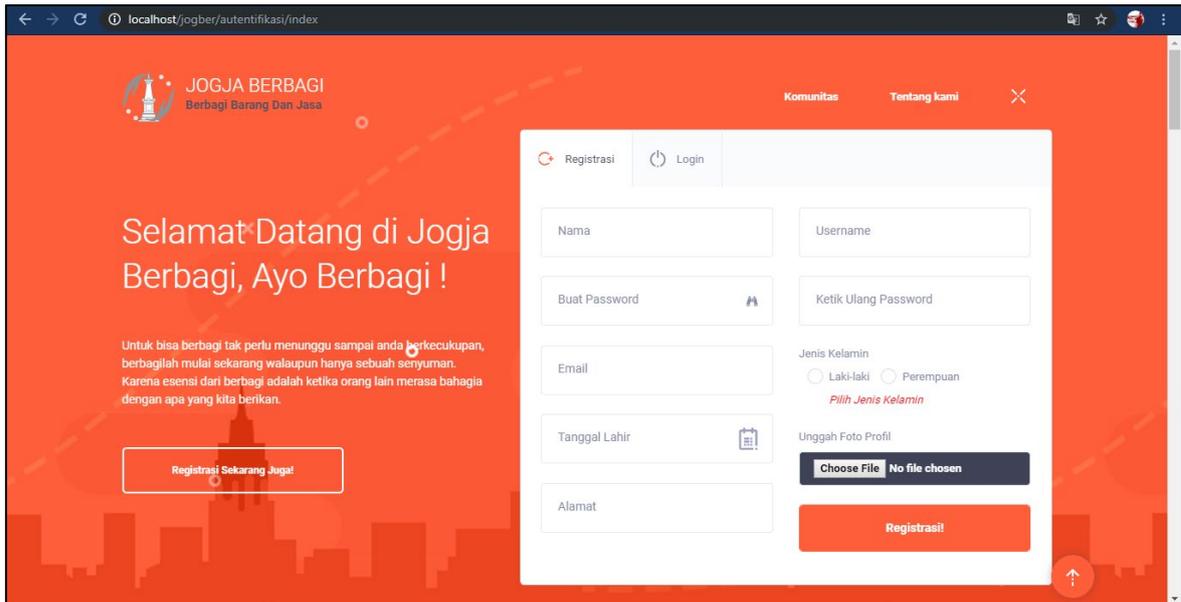
Dari analisis dan perancangan sebelumnya akan disajikan hasil berupa gambar tampilan dari program yang dibuat serta modul-modul program pembentuk dari sisi *client*. Hanya beberapa menu yang akan dibahas dari hak akses pengguna. Berikut adalah rinciannya.

#### 4.4.1 Halaman Utama

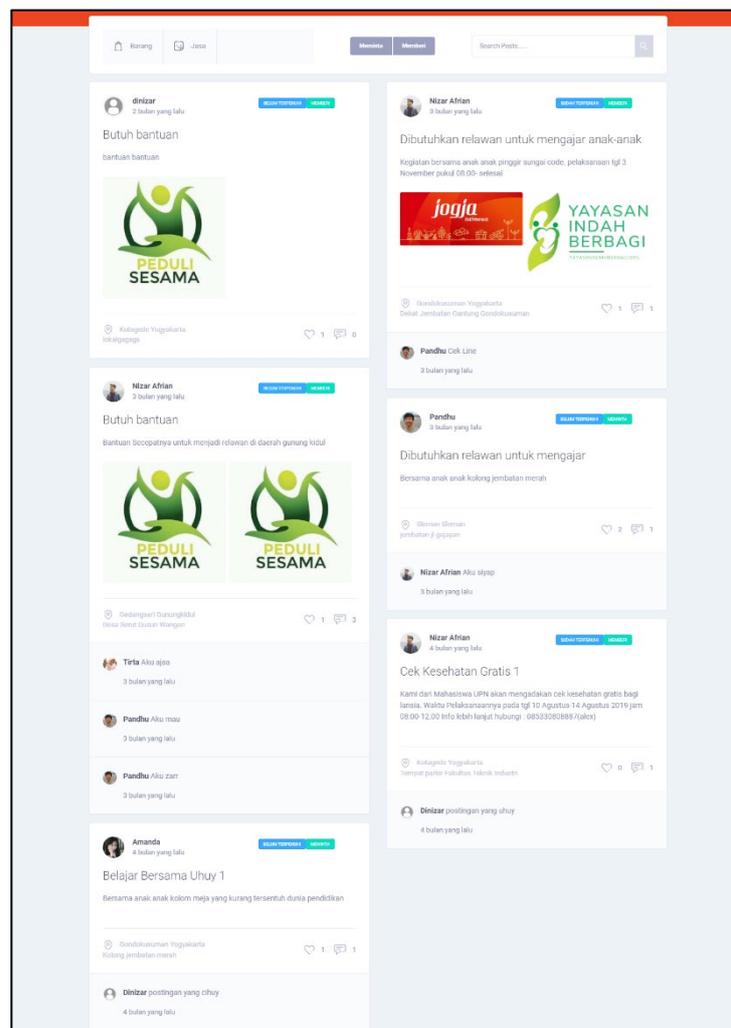
Halaman utama adalah halaman awal ketika pengguna mengakses web Jogja Berbagi. Halaman utama terdiri dari halaman yang berisi fitur *login* dan register untuk pengguna. Selain itu juga ada daftar unggahan dan tombol komunitas, tentang kami. Tetapi untuk daftar unggahan hanya bisa dilihat saja, tidak bisa di edit maupun dihapus. Pada tampilan dibuat sesuai perancangan pada bab sebelumnya. Formulir pada kolom *login* terdiri dari kolom *username* atau email, kemudian dibawahnya ada kolom *password*. Pada aplikasi ini dapat melakukan verifikasi terhadap data login yang dimasukkan sehingga pengguna tidak bisa *login* ketika belum menjadi pengguna terdaftar. Pada formulir registrasi terdapat sembilan kolom yang harus diisi. Masing masing kolom terdapat verifikasi masukkan oleh sistem agar data yang dimasukkan sesuai. Untuk daftar unggahan pada halaman ini hanya bisa dilihat, artinya pengguna harus melakukan login terlebih dahulu untuk memanajemen unggahannya. Selain itu daftar unggahan pada halaman ini juga tidak bisa dikomen, disukai, dan direspon dengan *whatsapp*. Hasil dari perancangan halaman utama dapat dilihat pada gambar 4.2, gambar 4.3, dan gambar 4.4.



Gambar 4.2. Halaman Utama Login



Gambar 4.3. Halaman Utama Register



Gambar 4.4. Daftar Unggahan pada Halaman Utama

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Autentifikasi extends CI_Controller
{
    public function __construct(){
        parent::__construct();
        $this->API = 'http://103.195.90.35:2700/jogber_api/index.php/api/';
        /*$this->API = 'http://localhost/jogber_api/index.php/api/';*/
    }

    public function index(){
        $data['sub_menu_barang'] = json_decode($this->curl->simple_get($this->API.'menu_submenu/menu?id_menu=1'));
        $data['sub_menu_jasa'] = json_decode($this->curl->simple_get($this->API.'menu_submenu/menu?id_menu=2'));
        $data['unggahan'] = json_decode($this->curl->simple_get($this->API.'unggahan'));
        foreach ($data['unggahan'] as $key=>$value){
            $data['unggahan'][$key]->komentar= json_decode($this->curl->simple_get($this->API.'unggahan/komentar?id unggahan='.$value->id unggahan));
            $data['unggahan'][$key]->jml_suka= json_decode($this->curl->simple_get($this->API.'Jml_suka?id unggahan='.$value->id unggahan));
            $data['unggahan'][$key]->gambar= json_decode($this->curl->simple_get($this->API.'foto unggahan?id unggahan='.$value->id unggahan));
        }
        $this->load->view('autentifikasi/v_index',$data);
    }

    public function register(){
        $tgl_lahir = $_POST['datetimepicker'];
        $tgl_lahir = date('Y/m/d', strtotime($tgl_lahir));
        $config['upload_path'] = './Gambar/'; // folder upload
        $config['allowed_types']= 'jpg|png'; // jenis file
        $config['max_size'] = 2500; //2.5 mb maksimal ukuran
        $config['max_width'] = 1000;
        $config['max_height'] = 1000;
        $this->load->library('upload', $config);
        if($this->upload->do_upload('foto-profil')){
            $file = $this->upload->data();
            $foto = $file['file_name'];
        }else{
            $foto = 'user.png';
        }
        $data = array('nama' => $this->input->post('nama'),
            'email' => $this->input->post('email'),
            'username' => $this->input->post('username'),
            'password' => $this->input->post('password'),
            'jenis_kelamin' => $this->input->post('optionsRadios'),
            'tgl_lahir' => $tgl_lahir,
            'alamat' => $this->input->post('alamat'),
            'foto_pengguna' => $foto,
        );
    }
}

```

```

        $id=json_decode($this->curl->simple_post($this->API.'pengguna',
        $data, array(CURLOPT_BUFFERSIZE => 10)));
        $data_session = array(
            'id_pengguna' => $id,
            'nama' => $data['nama'],
            'email' => $data['email'],
            'foto' => $data['foto_pengguna'],
            'status' => "login"
        );
        $this->session->set_userdata($data_session);
        redirect(base_url('berbagi/index'));
    }

    public function cek_username(){
        $cek_username = array('username' => $this->input->post('username'));
        $getUsername = json_decode($this->curl->simple_post($this-
        >API.'Pengguna/cek_username', $cek_username, array(CURLOPT_BUFFERSIZE
        => 10)));
        if($getUsername == 0){
            echo 'true';
        }
        else{
            echo 'false';
        }
    }

    public function login(){
        $emailusername = $this->input->post('emailusername');
        $password = $this->input->post('password');
        $where = array('emailusername' => $emailusername,
            'password' => $password
        );
        $data = json_decode($this->curl->simple_post($this-
        >API.'pengguna/cek_login', $where, array(CURLOPT_BUFFERSIZE => 10)));
        if($data){
            $data_session = array('id_pengguna' => $data->id_pengguna,
                'nama' => $data->nama,
                'email' => $data->email,
                'foto' => $data->foto_pengguna,
                'status' => "login"
            );
            $this->session->set_userdata($data_session);
            redirect(base_url('berbagi/index'));

        } else {
            $this->session->set_flashdata('error','Gagal Login, Coba
            lagi');
            redirect(base_url('autentifikasi/index'));
        }
    }

    public function logout()
    {
        $this->session->sess_destroy();
        redirect(base_url('autentifikasi/index'));
    }
}

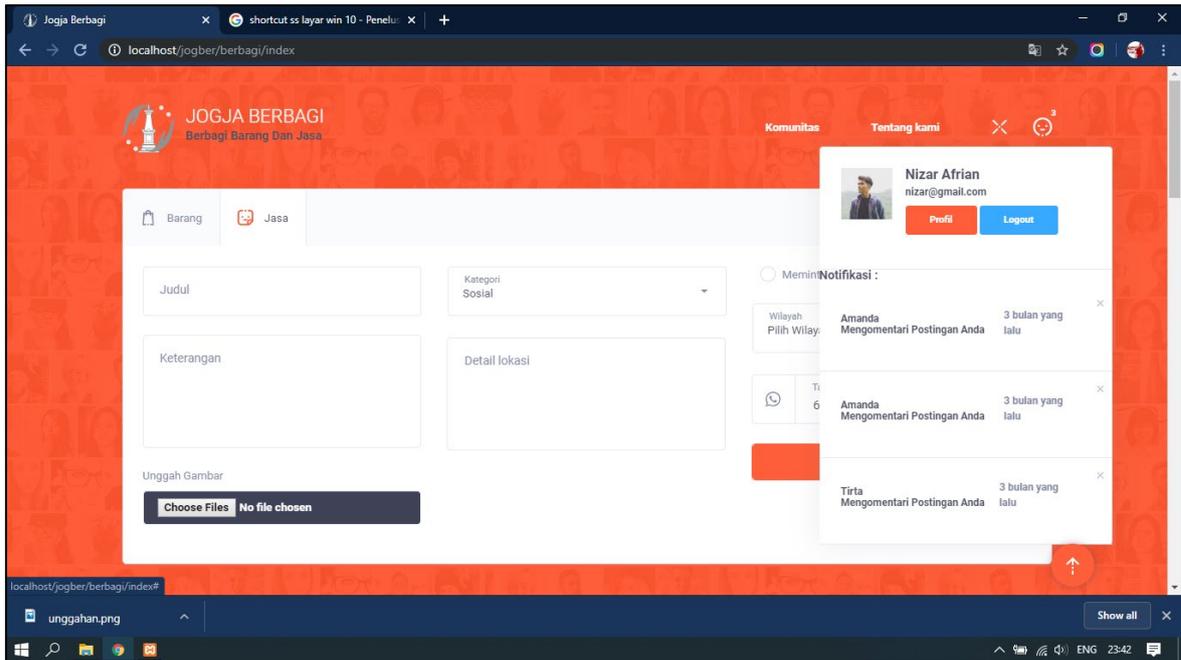
```

#### 4.4.2 Halaman Beranda

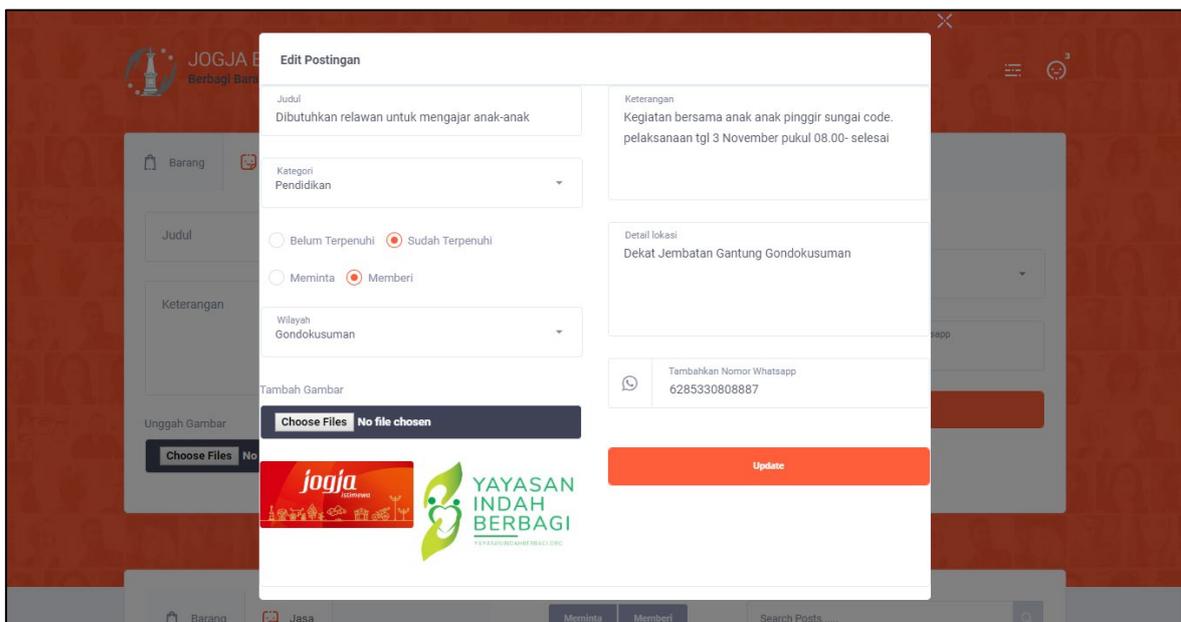
Halaman beranda berisi konten formulir untuk mengunggah unggahan, kemudian notifikasi, dan dibawahnya adalah navigasi unggahan, dan daftar unggahan dari pengguna. Dalam halaman beranda daftar unggahan dapat diedit atau dihapus. Berikut adalah tampilannya.

Gambar 4.5. Halaman Beranda untuk Formulir Unggahan

Gambar 4.6. Halaman Beranda untuk Daftar Unggahan dan Atributnya



**Gambar 4.7.** Halaman Utama untuk Notifikasi



**Gambar 4.8.** Halaman Utama untuk Edit Unggahan

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Berbagi extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        if($this->session->userdata('status') != "login"){
            redirect(base_url('autentifikasi/index'));
        }
        $this->API = 'http://103.195.90.35:2700/jogber_api/index.php/api/'
    }
}
```

**Modul Program 4.3.** Source code Halaman Beranda

```

public function index(){
    $data['notifikasi']= json_decode($this->curl-
>simple_get($this->API.'Notifikasi'));
    $data['kategori_unggah'] = json_decode($this->curl-
>simple_get($this->API.'Kategori'));
    $data['sub_menu_barang'] = json_decode($this->curl-
>simple_get($this->API.'Menu_submenu/menu?id_menu=1'));
    $data['sub_menu_jasa'] = json_decode($this->curl-
>simple_get($this->API.'Menu_submenu/menu?id_menu=2'));
    $id_pengguna=$this->session->userdata('id_pengguna');
    $jumlah_suka= json_decode($this->curl->simple_get($this-
>API.'Jml_suka?id_pengguna='.$id_pengguna));
    foreach ($jumlah_suka as $key => $value) {
        $data['jumlah_suka'][$value->id_unggah]= $value-
>id_unggah;}
    $data['wilayah'] = json_decode($this->curl->simple_get($this-
>API.'Wilayah/kecamatan'));
    $data['unggah'] = json_decode($this->curl->simple_get($this-
>API.'Unggahan'));
    foreach ($data['unggah'] as $key=>$value) {
        $data['unggah'][$key]->komentar= json_decode($this->curl-
>simple_get($this->API.'Unggahan/komentar?id_unggah='.$value-
>id_unggah));
        $data['unggah'][$key]->jml_suka= json_decode($this->curl-
>simple_get($this->API.'Jml_suka?id_unggah='.$value->id_unggah));
        $data['unggah'][$key]->gambar= json_decode($this->curl-
>simple_get($this->API.'Foto_unggah?id_unggah='.$value-
>id_unggah));}
    $this->load->view('berbagi/v_index', $data);}

public function insert_jasa(){
    $data = array('judul_unggah' => $this->input->post('judul'),
    'keterangan' => $this->input->post('keterangan'),
    'detail_lokasi' => $this->input->post('lokasi'),
    'waktu_unggah' => date("Y-m-d H:i:s"),
    'status' => 'Belum Terpenuhi',
    'nomorwa' => $this->input->post('nomorwa'),
    'id_pengguna'=>$this->session->userdata('id_pengguna'),
    'id_submenu' => $this->input->post('sub_menu'),
    'id_kategori'=> $this->input->post('optionsRadios'),
    'id_kecamatan'=> $this->input->post('kecamatan'));
    $id_unggah = json_decode($this->curl->simple_post($this-
>API.'Unggahan', $data, array(CURLOPT_BUFFERSIZE => 10));
    $data = array();
    if($this->input->post('submit-post') &&
!empty($_FILES['files']['name'])){
        $filesCount = count($_FILES['files']['name']);
        for($i = 0; $i < $filesCount; $i++){
            $_FILES['file']['name'] = $_FILES['files']['name'][$i];
            $_FILES['file']['type'] = $_FILES['files']['type'][$i];
            $_FILES['file']['tmp_name']=$_FILES['files']['tmp_name'][$i];
            $_FILES['file']['error'] = $_FILES['files']['error'][$i];
            $_FILES['file']['size'] = $_FILES['files']['size'][$i];
            $uploadPath = './Gambar/unggah/';
            $config['upload_path'] = $uploadPath;
            $config['allowed_types'] = 'jpg|jpeg|png';
            $this->load->library('upload', $config);
            $this->upload->initialize($config);

```

```

if($this->upload->do_upload('file')){
    $fileData = $this->upload->data();
    $uploadData[$i]['nama_foto'] = $fileData['file_name'];
    $uploadData[$i]['id_unggah'] = $id_unggah;
if(!empty($uploadData)){
json_decode($this->curl->simple_post($this-
>API.'Foto_unggah',$uploadData[$i], array(CURLOPT_BUFFERSIZE =>
10)));}}}
    redirect(base_url('berbagi/index'));}

public function update_post($id_unggah){
    $data = array('id_unggah' => $id_unggah,
        'judul_unggah' => $this->input->post('judul'),
        'keterangan' => $this->input->post('keterangan'),
        'detail_lokasi' => $this->input->post('lokasi'),
        'judul_unggah' => $this->input->post('judul'),
        'waktu_unggah' => $this->input->post('waktu'),
        'status' => $this->input->post('optionsRadios-status'),
        'nomorwa' => $this->input->post('nomorwa'),
        'id_pengguna' => $this->session->userdata('id_pengguna'),
        'id_submenu' => $this->input->post('sub_menu'),
        'id_kategori'=> $this->input->post('optionsRadios-keterangan'),
        'id_kecamatan'=> $this->input->post('kecamatan'));
    json_decode($this->curl->simple_put($this->API.'Unggahan', $data,
array(CURLOPT_BUFFERSIZE => 10)));
    $data = array();
    if($this->input->post('submit-post') &&
!empty($_FILES['files']['name'])){
        $filesCount = count($_FILES['files']['name']);
        for($i = 0; $i < $filesCount; $i++){
            $_FILES['file']['name'] = $_FILES['files']['name'][$i];
            $_FILES['file']['type'] = $_FILES['files']['type'][$i];
            $_FILES['file']['tmp_name'] = $_FILES['files']['tmp_name'][$i];
            $_FILES['file']['error'] = $_FILES['files']['error'][$i];
            $_FILES['file']['size'] = $_FILES['files']['size'][$i];
            $uploadPath = './Gambar/unggah/';
            $config['upload_path'] = $uploadPath;
            $config['allowed_types'] = 'jpg|jpeg|png';
            $this->load->library('upload', $config);
            $this->upload->initialize($config);
            if($this->upload->do_upload('file')){
                $fileData = $this->upload->data();
                $uploadData[$i]['nama_foto'] = $fileData['file_name'];
                $uploadData[$i]['id_unggah'] = $id_unggah;
                if(!empty($uploadData)){
                    json_decode($this->curl->simple_post($this-
>API.'Foto_unggah',$uploadData[$i], array(CURLOPT_BUFFERSIZE =>
10)));}}}
            redirect(base_url('berbagi/index'));
        }
    }

public function tambah_jml_suka(){
    $data = array('id_pengguna' => $this->session-
>userdata('id_pengguna'),
        'id_unggah' => $this->input->post('id'));
    $jml=json_decode($this->curl->simple_post($this->API.'Jml_suka',
    $data, array(CURLOPT_BUFFERSIZE => 10)));
    echo json_encode($jml);}

```

**Modul Program 4.5. Lanjutan Source code Halaman Beranda**

```

public function kurang_jml_suka(){
    $data=array('id_pengguna' => $this->session->userdata('id_pengguna'),
        'id_unggah' => $this->input->post('id'));
    $jml=json_decode($this->curl->simple_delete($this-
>API.'Jml_suka', $data, array(CURLOPT_BUFFERSIZE => 10)));
    echo json_encode($jml);}

public function insert_komen(){
    $data=array('detail_komentar'=>$this->input->post('detail_komentar'),
        'waktu_komentar' => date("Y-m-d H:i:s"),
        'id_unggah' => $this->input->post('id_unggah'),
        'id_pengguna' => $this->session->userdata('id_pengguna'));
    json_decode($this->curl->simple_post($this-
>API.'unggah/insert_komen', $data, array(CURLOPT_BUFFERSIZE =>
10)));
    if ($this->input->post('id_pengguna')!=$this->session-
>userdata('id_pengguna')){
        $data_notif = array(
            'id_unggah' => $this->input->post('id_unggah'),
            'waktu_komentar' => date("Y-m-d H:i:s"),
            'pengomentar' => $this->session->userdata('nama'));
        json_decode($this->curl->simple_post($this->API.'notifikasi',
        $data_notif, array(CURLOPT_BUFFERSIZE => 10)));}
    echo 1;}

public function delete_post(){
    $id_unggah=$this->input->post('id');
    $this->curl->simple_delete($this->API.'Unggahan',
    array('id'=>$id_unggah), array(CURLOPT_BUFFERSIZE => 10));
    echo 1;}

public function delete_foto(){
    $id_foto=$this->input->post('id');
    if($this->curl->simple_delete($this->API.'Foto_unggah',
    array('id'=>$id_foto), array(CURLOPT_BUFFERSIZE => 10))){
    echo 1;}else{
    echo 1;}}

public function delete_komen(){
    $id_komentar=$this->input->post('id');
    $this->curl->simple_delete($this->API.'Unggahan/del_komen',
    array('id'=>$id_komentar), array(CURLOPT_BUFFERSIZE => 10));
    echo 1;}

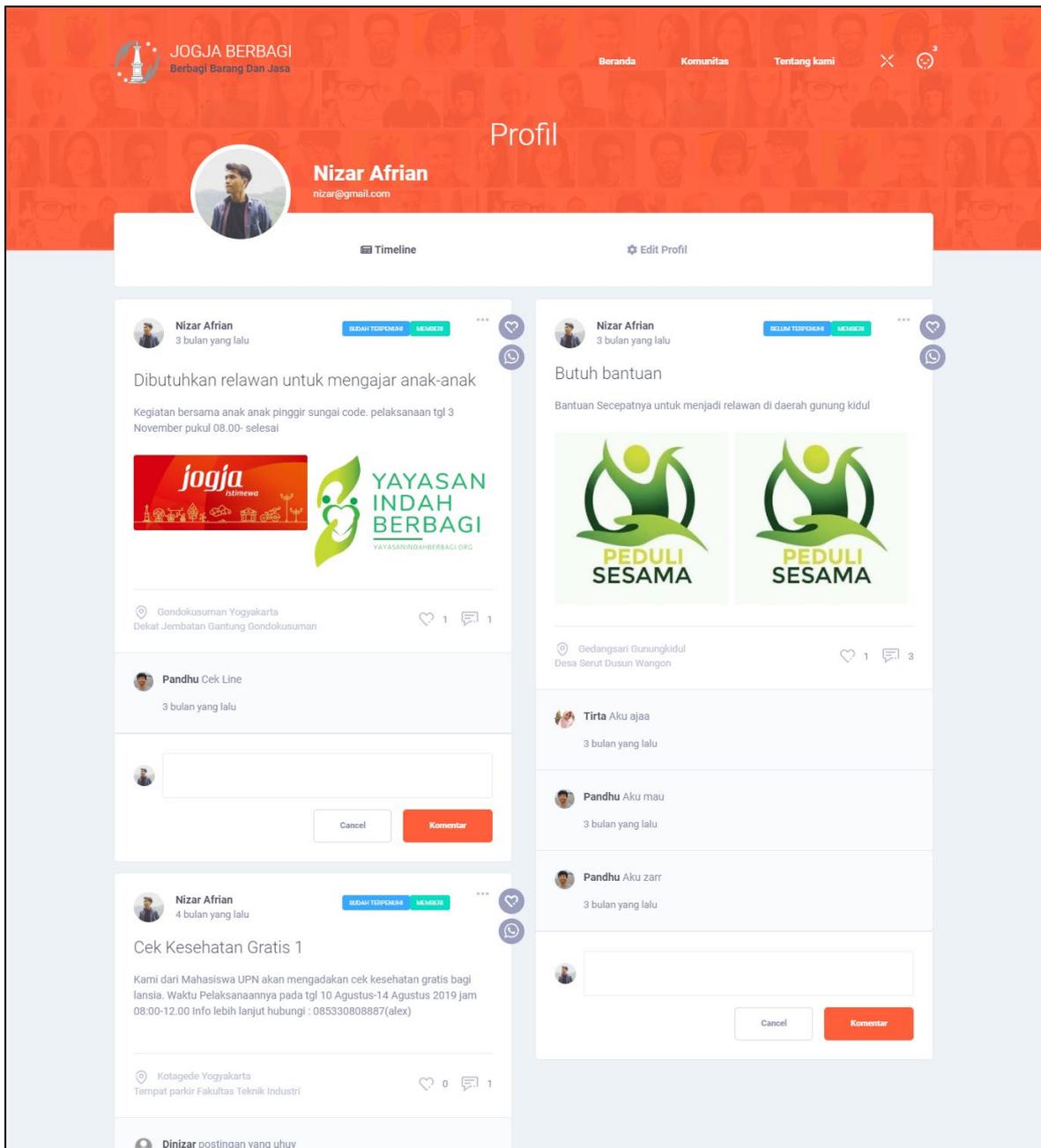
public function update_komentar(){
    $data = array( 'id_komentar' => $this->input->post('id'),
        'detail_komentar' => $this->input->post('detail_komentar'),
        'waktu_komentar' => $this->input->post('waktu_komentar'),
        'id_unggah' => $this->input->post('id_unggah'),
        'id_pengguna' => $this->input->post('id_pengguna'));
    $this->curl->simple_put($this->API.'unggah/edit_komen', $data,
    array(CURLOPT_BUFFERSIZE => 10));echo 1;}

public function delete_notifikasi(){
    $id_notifikasi=$this->input->post('id');
    $delete = $this->curl->simple_delete($this->API.'Notifikasi',
    array('id'=>$id_notifikasi), array(CURLOPT_BUFFERSIZE => 10));
    echo 1;}}

```

### 4.4.3 Halaman Profil

Pada halaman profil terdapat dua halaman yaitu halaman *timeline* dan halaman edit profil. Pada halaman timeline berisi daftar unggahan yang diunggah oleh pengguna. Sedangkan halaman edit profil berisi tentang formulir untuk mengedit profil. Berikut adalah tampilannya.



**Gambar 4.9.** Halaman Profil untuk Daftar Unggahan

**Gambar 4.10.** Halaman Profil untuk Edit Profil

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Akun extends CI_Controller{
public function __construct(){
    parent::__construct();
    if($this->session->userdata('status') != "login"){
        redirect(base_url('autentifikasi/index'));
    }
    $this->API = 'http://103.195.90.35:2700/jogber_api/index.php/api/';
public function profil_timeline(){
    $data['kategori_unggahan']=json_decode($this->curl->simple_get($this->API.'Kategori'));
    $data['notifikasi']=json_decode($this->curl->simple_get($this->API.'notifikasi'));
    $id_pengguna=$this->session->userdata('id_pengguna');
    $jumlah_suka= json_decode($this->curl->simple_get($this->API.'Jml_suka?id_pengguna='.$id_pengguna));
    foreach ($jumlah_suka as $key => $value) {
        $data['jumlah_suka'][$value->id_unggahan]= $value->id_unggahan;}
    $data['unggahan'] = json_decode($this->curl->simple_get($this->API.'unggahan'));
    foreach ($data['unggahan'] as $key=>$value) {
        $data['unggahan'][$key]->komentar= json_decode($this->curl->simple_get($this->API.'unggahan/komentar?id_unggahan='.$value->id_unggahan));
        $data['unggahan'][$key]->jml_suka= json_decode($this->curl->simple_get($this->API.'Jml_suka?id_unggahan='.$value->id_unggahan));
    }
}
}

```

**Modul Program 4.7.** Source code Halaman Profil

```

$data['unggahannya'][$key]->gambar= json_decode($this->curl->
>simple_get($this->API.'foto_unggahannya?id_unggahannya='.$value->
id_unggahannya));}
    $this->load->view('profil/v_profil_timeline', $data);}

public function profil_edit(){
    $data['notifikasi']= json_decode($this->curl->simple_get($this->
    >API.'notifikasi'));
    $where = $this->session->userdata("id_pengguna");
    $data['pengguna'] = json_decode($this->curl->simple_get($this->
    >API.'pengguna?id_pengguna='.$where));
    $data['unggahannya'] = json_decode($this->curl->simple_get($this->
    >API.'unggahannya'));
    foreach ($data['unggahannya'] as $key=>$value) {
        $data['unggahannya'][$key]->komentar= json_decode($this->curl->
        >simple_get($this->API.'unggahannya/komentar?id_unggahannya='.$value->
        id_unggahannya));
        $data['unggahannya'][$key]->jml_suka= json_decode($this->curl->
        >simple_get($this->API.'Jml_suka?id_unggahannya='.$value->id_unggahannya));
        $data['unggahannya'][$key]->gambar= json_decode($this->curl->
        >simple_get($this->API.'foto_unggahannya?id_unggahannya='.$value->
        id_unggahannya));}
    $this->load->view('profil/v_profil_edit', $data);}

public function update_profil(){
    $tgl_lahir = $_POST['datetimepicker'];
    $tgl_lahir = date('Y/m/d', strtotime($tgl_lahir));
    $config['upload_path']= './Gambar/'; // folder upload
    $config['allowed_types']      = 'jpg|jpeg|png'; // jenis file
    $config['max_size']           = 2500; //2.5 mb maksimal ukuran
    $config['max_width']          = 1000;
    $config['max_height']         = 1000;
    $this->load->library('upload', $config);
    $this->upload->do_upload('foto-profil');
    $file = $this->upload->data();
    $foto = $file['file_name'];
    $data = array('id_pengguna'=> $this->session->userdata("id_pengguna"),
        'nama' => $this->input->post('nama'),
        'email' => $this->input->post('email'),
        'username' => $this->input->post('username'),
        'password' => $this->input->post('password'),
        'jenis_kelamin' => $this->input->post('optionsRadios'),
        'tgl_lahir' => $tgl_lahir,
        'alamat' => $this->input->post('alamat'),
        'foto_pengguna' => $foto,);
    $this->curl->simple_put($this->API.'pengguna', $data,
    array(CURLOPT_BUFFERSIZE => 10));
    $where=$this->session->userdata("id_pengguna");
    $data_user = json_decode($this->curl->simple_get($this->
    >API.'pengguna?id_pengguna='.$where));
    $data_session = array('id_pengguna' => $data_user->id_pengguna, 'nama'
    => $data_user->nama, 'email' => $data_user->email, 'foto' => $data_user->
    foto_pengguna, 'status' => "login");
    $this->session->set_userdata($data_session);
    redirect(base_url('akun/profil_timeline'));
}

```

#### Modul Program 4.8. Lanjutan Source code Halaman Profil

```

public function hapus_notif()
{
    $id_notifikasi=$this->input->post('id');
    if($this->m_notifikasi->del_notifikasi($id_notifikasi)){
        echo 1;
    }else{
        echo 1;
    }
}
}
}

```

#### Modul Program 4.9. Lanjutan *Source code* Halaman Profil

#### 4.4.4 Halaman Komunitas

Halaman komunitas adalah halaman yang memberikan informasi komunitas di Yogyakarta yang bergerak untuk membantu sesama. Informasi yang diberikan pada halaman ini terdiri dari nama komunitas, deskripsi komunitas, lokasi, bidang, dan kontak yang bisa dihubungi. Hasil perancangan dari halaman komunitas dapat dilihat pada gambar 4.11.

Komunitas	Tipe	Kontak	Alamat
<b>Atap Senja</b> Atap senja adalah komunitas yang bergerak dibidang Pendidikan anak-anak bantaran kali Winongo dan kali Code	Jasa Pendidikan	IG : ataprenjajogja Line : @aau2310i	
<b>Akademi Berbagi Yogyakarta</b> Akademi berbagi adalah kelas belajar gratis tentang berbagai hal : menulis, fotografi, jurnalistik, komunikasi, dan seterusnya secara gratis.	Jasa Pendidikan dan Pelatihan	Twitter : @akademiberbagi FB : Akademi Berbagi	Perumahan Griya Permai Jl. Tampomas Raya No.4, Kwarasan, Nogotirto, Kec. Gamping, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55992
<b>Jogja Menyala</b> Jogja Menyala adalah organisasi yang berada di bawah Indonesia Mengajar. Komunitas ini mengupayakan berdirinya perpustakaan desa, gizi dan kesehatan anak-anak, dan sebagainya	Jasa Pendidikan, Sosial, Kesehatan	Twitter : @PenyalaJogja FB: Jogja Menyala Tel	TPA Jampi Stress, Gg. Melati No 585, Buriarjo, Jelis, Yogyakarta, Daerah Istimewa Yogyakarta (55231)
<b>Rumah Kagem</b> Rumah belajar kreatif kaki gunung merapi (kagem) Yogyakarta adalah lembaga non-profit yang bergerak di berbagai kegiatan : sosial, Pendidikan, pemberdayaan ekonomi masyarakat.	Jasa Pendidikan, Sosial, Ekonomi	Blog : kagemjogja.org Email : yrbkkagem@yahoo.	Perumahan Mandala I kav 1-S Jalan Kallurang km.10, Jetis Baran, Ngaglik, Sleman, Yogyakarta 55581
<b>Yayasan Dunia Damai Museum Anak Kolong Tangga</b> Adalah organisasi sosial non-profit yang bergerak di bidang seni, budaya, dan Pendidikan bagi anak-anak.	Jasa Pendidikan Seni	Email : untak_semma_anak@yahoo.com Telepon : 0	Gedung Taman Budaya Lantai 2 Jl. Shwedani no 1 Yogyakarta Sekretariat : Jl Tirtodipuran no 26

**Gambar 4.11.** Halaman Komunitas

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Komunitas extends CI_Controller{
public function __construct(){
    parent::__construct();
    $this->API = 'http://localhost/jogber_api/index.php/api/';}
public function komunitas(){
    $data['notifikasi']= json_decode($this->curl->simple_get($this->API.'Notifikasi'));
    $data['komunitas']= json_decode($this->curl->simple_get($this->API.'Komunitas'));
    $this->load->view('v_komunitas', $data);}
}

```

**Modul Program 4.10.** *Source code* Halaman Komunitas

#### 4.4.5 Halaman Tentang Aplikasi

Halaman tentang aplikasi adalah halaman yang memberikan informasi tentang jogja berbagi. Didalam halaman ini terdiri dari deskripsi, logo, kemudian kata-kata inspiratif untuk berbagi. Hasil dari perancangan halaman ini dapat dilihat pada gambar 4.12.



**Gambar 4.12.** Halaman Tentang Kami

```

<?php defined('BASEPATH') OR exit('No direct script access allowed');
class Tentangkami extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->API = 'http://localhost/jogber_api/index.php/api/';}
    public function tentangkami(){
        $data['notifikasi']= json_decode($this->curl->simple_get($this->API.'notifikasi'));
        $this->load->view('v_tentang_kami', $data);}}

```

**Modul Program 4.11.** *Source code* Halaman Tentang Kami

## 4.5 Web Service

Sesuai rancangan arsitektur pada bab sebelumnya, pembuatan aplikasi Jogja Berbagi menggunakan teknologi *web service*. Dimana terdapat dua sisi yaitu sisi *client* dan sisi *server*. Pada sisi *client* akan mengakses, dan menerima respon dari server. Dan pada sisi server bertugas menerima permintaan *client* dan mengirim respon balik. Karena pada pembuatan aplikasi ini menggunakan konsep MVC (*Model, View, Controller*), maka pembahasan akan dilakukan pada *controller client* dan hubungannya dengan *controller server*, beserta fungsi yang terkait. Detail hubungan antara *controller client* dan server dapat dilihat pada tabel 4.2.

**Tabel 4.2.** Daftar Akses *Client* Terhadap Server

No.	<i>Controller Client</i>	Fungsi	Method dan Parameter	<i>Controller Server</i>	Fungsi
1.	Akun	profil_timeline()	GET()	Kategori	index_get()
			GET(id_pengguna)	Jml_suka	index_get()
			GET(id_unggahan)	Unggahan	index_get() komentar_get()
		edit_profil()	GET(id_pengguna)	Pengguna	index_get()
		update_profil()	PUT(data update)	Pengguna	index_put()
2.	Autentifikasi	regiter()	POST(data pengguna)	Pengguna	index_post()
		cek_username()	POST(username)	Pengguna	cek_username_post()
		login()	POST(username, email, password)	Pengguna	cek_login_post()
3.	Berbagi	index()	GET()	Notifikasi	index_get()
			GET()	Kategori	index_get()
			GET()	Wilayah	kecamatan_get()
			GET()	Unggahan	index_get()
			GET(id_unggahan)	Unggahan	komentar_get()
			GET(id_pengguna)	Jml_suka	index_get()
		insert_jasa()	POST(data unggahan)	Unggahan	index_post()
		update_unggahan()	PUT(data unggahan)	Unggahan	index_put()
		delete_unggahan()	DELETE(id unggahan)	Unggahan	index_delete()
		insert_komen()	POST(data komentar)	Unggahan	insert_komen_post()
		edit_komen()	PUT(id komentar)	Unggahan	edit_komen_put()
delete_komentar()	DELETE(id komentar)	Unggahan	del_komen_delete()		
4.	Komunitas	komunitas()	GET()	Komunitas	index_get()

#### 4.6 Hasil Evaluasi

Tahap ini merupakan tahap evaluasi terhadap perancangan yang telah disesuaikan dengan kebutuhan pengguna. Evaluasi ini dilakukan untuk mengetahui sejauh mana perancangan yang dihasilkan sesuai dengan kebutuhan pengguna dengan teknik kuesioner. Hasil dari kuisoner diperoleh 30 koresponden, detail hasil bisa dilihat pada table 4.3.

**Tabel 4.3.** Hasil Evaluasi Aplikasi

No	Pertanyaan	Ya	Tidak
<b>Fungsionalitas Sistem</b>			
1.	Sistem dapat menampilkan informasi berbagi berupa unggahan beserta atributnya?	30	
2.	Sistem dapat memasukkan data unggahan kemudian ditampilkan?	30	
3.	Sistem dapat menghapus unggahan?	30	
4.	Sistem dapat mengedit data unggahan?	30	
5.	Sistem dapat memverifikasi dan memasukkan data profil saat registrasi?	30	
6.	Sistem dapat memverifikasi data saat <i>login</i> ?	30	
7.	Sistem dapat mengedit profil?	30	
8.	Sistem dapat memasukkan data komentar kemudian ditampilkan?	30	
9.	Sistem dapat mengedit komentar?	30	
10.	Sistem dapat menghapus komentar?	30	
11.	Sistem dapat terhubung dengan aplikasi Whatsapp?	30	
12.	Sistem dapat menampilkan informasi komunitas?	30	
<b>Fungsionalitas Aplikasi</b>			
13.	Tampilan halaman utama sesuai dan nyaman?	29	1
14.	Tampilan halaman beranda sesuai dan nyaman?	30	
15.	Tampilan halaman profil sesuai dan nyaman?	28	2
16.	Paduan warna sesuai dan nyaman?	30	
17.	Letak kontrol navigasi sesuai dan nyaman?	30	
18.	Apakah Aplikasi mudah digunakan untuk sarana berbagi?	30	

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil penelitian dan pembahasan tugas akhir yang telah dilakukan, maka dapat ditarik kesimpulan bahwa penerapan metode *living lab* dan *user center design* (UCD) pada pembuatan aplikasi jogja berbagi, adalah sebagai berikut:

1. Dari evaluasi yang dilakukan dengan kuisioner hasil yang didapat menunjukkan aplikasi yang dibuat memudahkan pengguna dan sesuai dengan kebutuhan pengguna untuk melakukan kegiatan berbagi jasa.
2. Proses permintaan data dan pengiriman data antara *client* dan server sudah sesuai, dilihat dari hasil evaluasi aplikasi mengenai fungsionalitas sistem yang berhasil menampilkan dan mengirim data.

#### **5.2 Saran**

Pada penelitian ini masih ada kekurangan dan dapat dikembangkan lebih lanjut agar menjadi lebih baik. Adapun saran untuk sebagai acuan atau pertimbangan adalah berikut:

1. Menambah fitur baru atau menjadikan sistem lebih kompleks.
2. Perlunya tindakan lanjutan sebagai tolak ukur yang lebih detail mengenai kesesuaian aplikasi dengan kebutuhan pengguna.
3. Pengembangan terhadap fitur komunitas yang lebih kompleks.

## DAFTAR PUSTAKA

- Adriana, D. (2003). Pengenalan Pemrograman Pengenalan Pemrograman E-Commerce E-Commerce dengan PHP dan dengan PHP dan MySQL. *Kuliah Umum Ilmu Komputer*, 9.
- Aziz, A., Wiharto, & Wicaksono, B. (2013). Pemanfaatan Web Service Moodle Berbasis REST-JSON untuk Membangun Moodle Online Learning Extension berbasis Android. *ITSMART*.
- Badan Pusat Statistika DIY (2019). Statistik Kemiskinan Provinsi Daerah Istimewa Yogyakarta 2019, Keadaan Ketenagakerjaan Yogyakarta Agustus 2018.
- Barlian, K., & Ermatita. (2017). A Web of Things dengan RESTful protokol pada Smart City. *ARS*.
- Diantoro, F. (2011). *Manajemen Pemasaran Jasa*. Retrieved from Academia: [https://www.academia.edu/31671697/Manajemen\\_Pemasaran\\_Jasa](https://www.academia.edu/31671697/Manajemen_Pemasaran_Jasa)
- DinasSosialDIY. (2018). *Data LKS dan Komunitas*. Yogyakarta.
- Fauziah, Y. (2013). Aplikasi Iklan Baris Online menggunakan Arsitektur REST Web Service. *Telematika*.
- Fielding, R. dan Reschke, J. (2014), “*Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*,” *Internet Engineering Task Force (IETF)*.
- Fielding, R., T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*, University of California, Irvine.
- Fielding, R., T., dan Taylor, R., N. (2002), “*Principled design of the modern Web architecture*,” *ACM Transactions on Internet Technology (TOIT)*, vol. 2, no. 2, pp. 115-150.
- Fitriani, Y. (2017). Analisis Pemanfaatan Berbagai Media Sosial Sebagai Sarana Penyebaran Informasi Bagi Masyarakat. *Paradigma*.
- Gottschalk, S., K., Graham, Kreger, H., dan Snell, J. (2002). “*Introduction to Web services architecture*,” *IBM SYSTEMS JOURNAL*, vol. 41, no. No. 2.
- Hidayat, R., & Ashari, A. (2013). Penerapan Teknologi Web Service Untuk Integrasi Layanan Puskesmas dan Rumah Sakit. *Berkala MIPA*.
- Heinich, Robert, Molenda, M., & Russel, J. D. (1982). *Instructional Media: and the New Technology of Instruction*.
- Indarto, E. (2019). *Tantangan metode Pengembangan metode inovasi memberikan dampak ekonomi dan sosial yang luas dan cepat*.
- Insani, P. A. (2017). MEWUJUDKAN KOTA RESPONSIF MELALUI SMART CITY. *PUBLISIA*.

- Introducing JSON*. (n.d.). Retrieved from JSON: json.org (accessed 9 Agustus 2019).
- Karim, W. S. (2012). PEMROGRAMAN WEB.
- Khasoggi, B., & Ermatita. (2017). A Web of Things dengan RESTful protokol pada Smart City . *ARS*.
- Kreger, H. (2001). *Web-Services Conceptual Architecture*. IMB software Group USA.
- Kurniawan, E. (2014). IMPLEMENTASI REST WEB SERVICE UNTUKSALES ORDER DANSALES TRACKINGBERBASIS MOBILE. *EKSIS*.
- Kurniawan, Y. K., Oslan, Y., & Kristanto, H. (2013). IMPLEMENTASI REST - API UNTUK PORTAL AKADEMIK UKDWBERBASIS ANDROID. *EKSIS*.
- Kusumawaty, A. (2012). APLIKASI PEMESANAN MAKANAN PADA RESTORAN .
- Lucky (2008). *XML Web services: Aplikasi Desktop, Internet & Handphone*, Jasakom, Jakarta.
- Luis, F., C., Kurt, C. (2004). *An Introduction to the Web Services Architecture and Its Specifications. Web Services Technical Articles. Microsoft Corporation*.
- Manes, A.T. (2001). *Introduction to web Service*. <<http://www.systinet.com>>.
- Malmberg, K., Vaittine, I., (2018). *Living Lab Methodology Handbook* (<https://u4iot.eu/Livinglabmethodology>)
- Nugroho, A., Mustofa, K. (2014, Mey). PERBANDINGAN ANTARA “BIG” WEB SERVICE DENGAN RESTFUL WEB SERVICE UNTUK INTEGRASI DATA BERFORMAT GML. *JURNAL INFORMATIKA*, 11, 8-16.
- Nurseto, T. (2011). MEMBUAT MEDIA PEMBELAJARAN YANG MENARIK . 20.
- Paik, H.-y., Lemos, A. L., Barukh, M. C., Benatallah, B., & Natarajan, A. (2017). *Web Service Implementation and Composition Techniques*, Springer. Switzerland: Springer.
- Palilingan, K. Y., Santoso, A. J., & Rahayu, F. L. (2017). Perancangan Aplikasi Pencarian Tempat Wisata Berbasis Lokasi Menggunakan Metode User Centered Design (UCD).
- Putra, S. A. (2018). *Perbedaan Api dengan Web Service*. Retrieved from Garuda Cyber Indonesia: <https://garudacyber.co.id/artikel/284-perbedaan-api-dengan-web-service>
- Pautasso, C., (2008), *REST vs SOAP Making the Right Architectural Decision*, *SOA Symposium, Amsterdam*.
- Puustjärvi, Puustjärvi, L. (2010). “*Application Integration and Semantic Integration in Electronic Prescription Systems*,” *International Journal of Computer Science Issues*, vol. 7, no. 1, pp. 1-8.

- Rizal, R., & Rahmatulloh, A. (2019). RESTful Web Service untuk Integrasi Sistem Akademik dan Perpustakaan Universitas Perjuangan . *JIF*, 55.
- Rulloh, A., Mahmudah, D. E., & Herman, K. (2017). Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android. *Teknikom*.
- Saputri, I. S., Fadhli, M., & Surya, I. (2017). *Penerapan Metode UCD (User Centered Design) pada E-Commerce Putri Intan*. JURNAL NASIONAL TEKNOLOGI DAN SISTEM INFORMASI.
- Saryanto, Sumarsono, & Retnowati, N. D. (2013). COMPARATIVE ANALYSIS OF XML AND JSON USING PHP APPLICATION PLATFORM WITH REPRESENTATIONAL STATE TRANSFER ( REST ) ARCHITECTURA.
- Sholikhin, A., & Riasti, B. K. (2013). PEMBANGUNAN SISTEM INFORMASI INVENTARISASI SEKOLAH PADA DINAS PENDIDIKAN KABUPATEN REMBANG BERBASIS WEB. *IJNS*.
- Su, C., J. dan Chiang, C., Y. (2012). “*Enabling successful Collaboration 2.0: A REST-based Web Service and Web 2.0 technology oriented information platform for collaborative product development*,” *Computers in Industry*, vol. 63.
- Sutanta, E., & Mustofa, K. (2012). KEBUTUHAN WEB SERVICE UNTUK SINKRONISASI DATA ANTAR SISTEM INFORMASI DALAM E-GOV DI PEMKAB BANTUL YOGYAKARTA . *STMIK BANDUNG* .
- Tidwell, D. (2001). *Web Services: The Web’s next Revolution*. <http://www6.software.ibm.com/developerworks/education/wsbasics/wsbasics-a4.pdf>.
- TribunJogja*. (2019, Juli). Retrieved from *TribunJogja*: <http://www.jogja.tribunnews.com/Baktisosial>
- Utomo, C. E., & Hariadi, M. (2016). Strategi Pembangunan Smart City dan Tantangannya bagi Masyarakat Kota. *Jurnal Strategi dan Bisnis*.
- Wahli, U., Burroughs, O., Cline, O., Tung, L., (2006). *Service Handbook for Web Sphere Application Server 6.1*, <http://www.redbooks.ibm.com/redbooks/pdfs/>
- XML* . (n.d.). Retrieved from XML: xml.com (accessed 9 Agustus 2019).