

**PENGEMBANGAN APLIKASI INFORMASI TEMPAT PARKIR
TERDEKAT DI PROVINSI DKI JAKARTA MENGGUNAKAN
LOCATION BASED SERVICE (LBS) DAN GOOGLE MAPS API
(STUDI KASUS: UNIT PERPARKIRAN DINAS PERHUBUNGAN DKI
JAKARTA)**

Skripsi



Universitas Islam Negeri
SYARIF HIDAYATULLAH JAKARTA

Oleh:

REVO YULIANGGARA

11160910000017

**PROGAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH
JAKARTA
1443 H/ 2022 M**

PERNYATAAN ORISINALITAS

Dengan ini saya menyatakan bahwa:

1. Skripsi ini merupakan hasil karya asli saya yang diajukan untuk memenuhi salah satu persyaratan memperoleh gelar Strata 1 di UIN Syarif Hidayatullah Jakarta.
2. Semua sumber yang saya gunakan dalam penulisan ini telah saya cantumkan sesuai dengan ketentuan yang berlaku di UIN Syarif Hidayatullah Jakarta.
3. Apabila di kemudian hari terbukti karya ini bukan hasil karya saya atau merupakan hasil jiplak karya orang lain, maka saya bersedia menerima sanksi yang berlaku di UIN Syarif Hidayatullah Jakarta.

Jakarta, April 2022



Revo Yulanggara

11160910000017

PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI

Sebagai civitas akademik UIN Syarif Hidayatullah Jakarta, saya yang bertanda tangan dibawah ini :

Nama : Revo Yulianggara
NIM : 11160910000017
Program Studi : Teknik Informatika
Fakultas : Sains dan Teknologi
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Islam Negeri Syarif Hidayatullah Jakarta Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul :

**“PENGEMBANGAN APLIKASI INFORMASI TEMPAT PARKIR
TERDEKAT DI PROVINSI DKI JAKARTA MENGGUNAKAN
LOCATION BASED SERVICE (LBS) DAN GOOGLE MAPS API (STUDI
KASUS: UNIT PERPARKIRAN DINAS PERHUBUNGAN DKI
JAKARTA)”**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Islam Negeri Syarif Hidayatullah Jakarta berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data, merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jakarta

Pada : April 2022

Yang menyatakan,



(Revo Yulianggara)

Nama : Revo Yulianggara
Program Studi : Teknik Informatika
Judul : Pengembangan Aplikasi Informasi Tempat Parkir Terdekat
Di Provinsi DKI Jakarta Menggunakan *Location Based Service* (LBS) Dan *Google Maps API* (Studi Kasus: Unit
Perparkiran Dinas Perhubungan DKI Jakarta)

ABSTRAK

Semakin bertambahnya jumlah penduduk yang menggunakan transportasi pribadi menyebabkan perlunya jumlah tempat parkir yang memadai. Tetapi, sistem perparkiran yang ada saat ini belum optimal dan masih menggunakan cara konvensional dengan tenaga sumber daya manusia. Untuk sistem informasi tempat parkir (*e-parkir*) yang terdapat pada Dinas Perhubungan DKI Jakarta saat ini sedang dalam tahap perkembangan dengan berbasis *mobile apps* yang dimana sistem ini juga masih memiliki kekurangan, beberapa juga tidak terdapat alternatif titik lokasi parkir lain yang terdekat. Hal tersebut menyebabkan pencarian tempat parkir menjadi tidak efisien. Melihat permasalahan tersebut, para pelaku yang memiliki kebutuhan penting di jalan raya ingin memberikan layanan informasi tempat parkir dengan lebih baik dan efisien bagi pengendara. Tujuan dari penelitian ini adalah dengan mengembangkan sistem aplikasi informasi tempat parkir berbasis *mobile* dengan teknologi *Location Based Service* (LBS) dan *web service* yaitu *Google Maps API*. Penggunaan Android Studio dan bahasa pemrograman PHP juga digunakan untuk memaksimalkan pengembangan sistem berbasis *mobile* ini. Sistem ini dikembangkan menggunakan metode *Rapid Application Development* (RAD) untuk pengembangan *project* yang singkat. Hasil dari penelitian ini adalah sebuah sistem informasi tempat parkir terdekat berbasis android yang dibuat berdasarkan langkah – langkah dari metode RAD. Hasil pengujian sistem yang dilakukan menggunakan pengujian *User Acceptance Testing* (UAT) mendapatkan persentase hasil kesesuaian 100% dan menunjukkan bahwa pengembangan sistem aplikasi informasi tempat parkir terdekat dengan mengimplementasikan LBS dan *Google Maps API* ini dapat menyelesaikan masalah.

Kata Kunci : Tempat Parkir, *mobile*, *Location Based Service*, *Google Maps API*, *Rapid Application Development*, *User Acceptance Testing*.

Daftar Pustaka : 6 Buku, 3 Skripsi, 23 Jurnal dan 6 website.

Jumlah Halaman : 140 Halaman.

Name : Revo Yulianggara
Study of Program : Informatics Engineering
Title : Development of the Nearest Parking Place Information Application in DKI Jakarta Province Using Location Based Service (LBS) and Google Maps API (Case Study: DKI Jakarta Transportation Agency Parking Unit).

ABSTRACT

The growing number of residents who rely on private transportation need a sufficient number of parking places. The current parking system, on the other hand, is inefficient and still relies on human resources. The DKI Jakarta Transportation Agency's parking lot information system is now in the development stage, and it is based on mobile apps. However, this system has disadvantages, and some people do not have other parking options nearby. As a result, looking for a parking spot is inefficient. As a result of these issues, actors with critical needs on the roadway aim to provide better and more effective parking information services to motorists. The goal of this study is to create a mobile-based parking information system using Location Based Service (LBS) technology and a web service, specifically the Google Maps API. To maximize the design of mobile-based information systems, Android Studio and the PHP programming language are employed. For quick project development, this system was created using the Rapid Application Development (RAD) approach. The outcome of this study is an android-based nearby parking information application built using the RAD method's steps. The system testing, which used the User Acceptance Testing (UAT) method, yielded 100 percent results, indicating that the design of the parking lot information application system, which used LBS and the Google Maps API to tackle the problem, was successful.

Keywords : Parking lot, *mobile, Location Based Service, Google Maps API, Rapid Application Development, User Acceptance Testing.*

Bibliography : 6 Books, 3 Theses, 23 Journals and 6 website.

Number of Pages : 140 Pages.

KATA PENGANTAR



Assalamu'alaikum Wr. Wb.

Puji syukur senantiasa dipanjatkan kehadiran Allah SWT yang telah melimpahkan rahmat, hidayat serta nikmat – Nya sehingga penyusunan skripsi ini dapat diselesaikan. Sholawat dan salam senantiasa dihaturkan kepada junjungan kita Baginda Nabi Muhammad SAW beserta keluarganya, para sahabatnya serta umatnya hingga akhir zaman. Penulisan skripsi ini mengambil tema dengan judul:

**PENGEMBANGAN APLIKASI INFORMASI TEMPAT PARKIR
TERDEKAT DI PROVINSI DKI JAKARTA MENGGUNAKAN
LOCATION BASED SERVICE (LBS) DAN GOOGLE MAPS API
(STUDI KASUS: UNIT PERPARKIRAN DINAS PERHUBUNGAN DKI
JAKARTA)**

Penyusunan skripsi ini adalah salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) pada program studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta. Adapun bahan penulisan skripsi ini adalah berdasarkan hasil penelitian, pengembangan aplikasi, wawancara, observasi dan beberapa studi literatur.

Dalam penyusunan skripsi ini, telah banyak bimbingan dan bantuan yang didapatkan dari berbagai pihak sehingga skripsi ini dapat berjalan dengan lancar. Oleh karena itu, penulis ingin mengucapkan banyak terimakasih kepada:

1. Ir. Nashrul Hakiem, S.Si., M.T., Ph.D. selaku Dekan Fakultas Sains dan Teknologi serta selaku dosen pembimbing 1 yang senantiasa sabar dan selalu memberikan bimbingan dan arahan, memberikan nasihat dan saran yang bermanfaat sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
2. Dr. Imam Marzuki Shofi, MT selaku Ketua Program Studi Teknik Informatika dan Bapak Andrew Fiade, M.Kom selaku Sekretaris Program Studi Teknik

Informatika.

3. Ibu Luh Kesuma Wardhani, S.T.,M.T. selaku dosen pembimbing 2 yang senantiasa sabar dan selalu memberikan bimbingan dan arahan, memberikan nasihat dan saran yang bermanfaat sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
4. Ibu Nenny Anggraini, S.Kom., M.T. dan Ibu Nurul Faizah Rozy, MTI selaku dosen penguji 1 dan 2 yang telah memberikan arahan dan saran untuk penulis sehingga dapat menyelesaikan skripsi ini dengan baik.
5. Orang Tua tercinta, Bapak Budi Susanto, S.E dan Almh. Ibunda Siti Solikhah yang selalu mendo'akan serta memberi dukungan moril dan materil sepanjang hidup penulis.
6. Saudara Kandung penulis, Milasari Nurfitria, S.Si yang selalu siap membantu dan memberikan semangat selama penyusunan skripsi.
7. Bapak Syaefudin Zuhri selaku Manajer UP Dishub DKI Jakarta yang senantiasa membimbing dan membantu penulis dalam proses penelitian aplikasi ini.
8. Kepada seluruh staff dan pegawai pada Unit Perparkiran Dinas Perhubungan DKI Jakarta yang sudah membantu penulis dalam melakukan penelitian skripsi.
9. Seluruh dosen dan staff UIN Jakarta, khususnya Fakultas Sains dan Teknologi yang telah memberikan ilmu serta pengalaman yang berharga.
10. Seluruh teman – teman terbaik dari Teknik Informatika 2016, semua teman di kelas TI A dan kepada Sobat TI yang selalu memberikan motivasi dan support kepada penulis.
11. Seluruh pihak yang tidak dapat disebutkan satu persatu yang secara langsung maupun tidak langsung telah membantu dalam menyelesaikan skripsi ini.

Penulisan skripsi ini masih jauh dari kata sempurna. Untuk itu, sangat diperlukan kritik dan saran yang membangun bagi penulis. Akhir kata, semoga laporan skripsi ini dapat bermanfaat bagi penulis dan orang lain.

Wassalamualaikum, Wr. Wb.

Jakarta, Februari 2022

Penulis



Revo Yulianggara

11160910000017



DAFTAR ISI

ABSTRAK.....	v
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	5
1.3 Tujuan Penelitian.....	5
1.4 Manfaat Penelitian	5
1.5 Batasan Masalah	6
1.5.1 Metode.....	6
1.5.2 Proses.....	6
1.5.3 Tools	7
1.6 Metodologi Penelitian.....	7
1.6.1 Metode Pengumpulan Data	7
1.6.2 Metode Pengembangan Sistem.....	8
1.7 Sistematika Penulisan.....	8
BAB II LANDASAN TEORI.....	10
2.1 Pengembangan Sistem.....	10
2.2 Aplikasi.....	11
2.3 Informasi.....	12
2.4 Tempat Parkir.....	13
2.5 Location Based Service (LBS)	14
2.5.1 Komponen Location Based Services (LBS).....	17
2.6 Google Maps API	18
2.7 Android.....	19
2.7.1 Kelebihan Android	20

2.7.2	Kekurangan Android	21
2.8	Rapid Application Development (RAD)	22
2.8.1	Tahapan Pengembangan RAD	23
2.8.2	Kelebihan dan Kekurangan RAD	23
2.9	Android Studio.....	25
2.9.1	Fitur-fitur Android Studio.....	25
2.9.2	Kelebihan dan Kekurangan Android Studio	26
2.10	Postman	27
2.11	Studi Literatur Sejenis	28
BAB III METODOLOGI PENELITIAN		39
3.1	Metode Pengumpulan Data	39
3.1.1	Metode Wawancara	39
3.1.2	Metode Pustaka/Studi Literatur	39
3.1.3	Metode Observasi	40
3.2	Metode Pengembangan Sistem	41
3.2.1	Rapid Application Development (RAD)	41
3.3	Perangkat Penelitian	43
3.4	Kerangka Berpikir	43
3.5	Metode Pengujian	45
BAB IV IMPLEMENTASI		46
4.1	Requirements Planning (Perencanaan Persyaratan)	46
4.1.1	Analisis Sistem Berjalan	46
4.1.2	Kelebihan dan Kekurangan Sistem Berjalan.....	47
4.1.3	Analisis Sistem Usulan	48
4.1.4	Analisis Kebutuhan Sistem	52
4.1.5	Perancangan Antar Muka Sistem	53
4.1.6	Koding Program Aplikasi	53
4.1.7	Testing (Pengujian)	54
4.2	Design Workshop (Workshop Desain RAD).....	54
4.2.1	Design Use Case Diagram Aplikasi	54
4.2.2	Design Mock Up Aplikasi	55
4.2.3	Tahapan Build Location Based Service (LBS).....	61
4.2.4	Tahapan Build Database.....	67
4.2.5	Tahapan Build REST API Dengan PHP	69
4.2.6	Tahapan Build Interface Menggunakan Android Studio	76
4.3	Implementation (Penerapan Pengujian)	112
4.3.1	Black Box Testing.....	112
4.3.2	User Acceptance Testing (UAT)	122

BAB V HASIL DAN PEMBAHASAN.....	126
5.1 Hasil Akhir Sistem	126
5.2 Hasil Pengujian Black Box Testing.....	127
5.3 Hasil Pengujian User Acceptance Testing (UAT)	127
5.4 Hasil Tampilan Antarmuka	128
5.4.1 Halaman Splash Screen	128
5.4.2 Halaman Login User	129
5.4.3 Halaman Home	129
5.4.4 Halaman Menu Wilayah	130
5.4.5 Halaman Menu Peta	131
5.4.6 Halaman Menu Tips&Trik	131
5.4.7 Halaman Login Admin	132
5.4.8 Halaman Dashboard Admin	132
5.4.9 Halaman Menu Wilayah Admin	133
5.4.10 Halaman Menu Parkir Admin	133
5.4.11 Halaman Menu Tips&Trik Admin	134
5.4.12 Halaman Menu Pengaturan Admin	135
BAB VI KESIMPULAN DAN SARAN	136
6.1 Kesimpulan	136
6.2 Saran.....	136
DAFTAR PUSTAKA	138
LAMPIRAN	141
Lampiran Surat Permohonan Wawancara.....	141
Lampiran 3. List Pertanyaan Wawancara 1.....	142
Lampiran 4. List Pertanyaan Wawancara 2.....	143

DAFTAR GAMBAR

Gambar 2. 1 Siklus Informasi	12
Gambar 2. 2 Teknologi Location Based Service (LBS)	15
Gambar 2. 3 Komponen Location Based Service	18
Gambar 2. 4 Fase Pengembangan RAD.....	22
Gambar 3. 1 Kerangka Berpikir	44
Gambar 4. 1 Sistem Berjalan	47
Gambar 4. 2 Sistem Usulan.....	48
Gambar 4. 3 Design Use Case Diagram.....	55
Gambar 4. 4 Rancangan Splash Screen.....	56
Gambar 4. 5 Rancangan Dashboard User	56
Gambar 4. 6 Rancangan Login User.....	57
Gambar 4. 7 Rancangan Menu Wilayah	57
Gambar 4. 8 Rancangan Menu Peta.....	58
Gambar 4. 9 Rancangan Menu Tips&Trik.....	58
Gambar 4. 10 Rancangan Login Admin	59
Gambar 4. 11 Rancangan Dashboard Admin.....	59
Gambar 4. 12 Rancangan Menu Wilayah Admin	59
Gambar 4. 13 Rancangan Menu Parkir Admin.....	60
Gambar 4. 14 Rancangan Menu Tips&Trik Admin.....	60
Gambar 4. 15 Rancangan Menu Pengaturan Admin.....	61
Gambar 4. 16 Tabel User	68
Gambar 4. 17 Tabel Wilayah	68
Gambar 4. 18 Tabel Parkir.....	68
Gambar 4. 19 Tabel Tips.....	69
Gambar 4. 20 Mengaktifkan Apache dan MySQL.....	69
Gambar 4. 21 Konfigurasi Project	70
Gambar 4. 22 Route Perintah pada PHP	70
Gambar 4. 23 Route Perintah get dan post.....	70
Gambar 4. 24 Contoh Running pada API	71
Gambar 4. 25 Request dan Response Parkir API pada Postman	73
Gambar 4. 26 Request dan Response Parkir API pada Postman	74
Gambar 4. 27 Logo Dishub.png.....	77
Gambar 5. 1 Splash Screen Aplikasi.....	128
Gambar 5. 2 Halaman Login User	129
Gambar 5. 3 Halaman Home User	129
Gambar 5. 4 Halaman Menu Wilayah.....	130
Gambar 5. 5 Halaman Menu Peta	131
Gambar 5. 6 Halaman Menu Tips & Trik	131
Gambar 5. 7 Halaman Login Admin.....	132

Gambar 5. 8 Halaman Dashboard Admin	132
Gambar 5. 9 Halaman Menu Wilayah Admin	133
Gambar 5. 10 Halaman Menu Parkir Admin	133
Gambar 5. 11 Halaman Menu Tips&Trik Admin	134
Gambar 5. 12 Halaman Menu Pengaturan Admin	135



DAFTAR TABEL

Tabel 2. 1 Keuntungan dan Kerugian Rapid Application Development (RAD)	24
Tabel 2. 2 Studi Literatur Sejenis.....	28
Tabel 2. 3 Perbandingan Penelitian.....	33
Tabel 2. 4 Komparasi Penelitian dengan Dishub	35
Tabel 3. 1 Perangkat Penelitian (Hardware)	43
Tabel 3. 2 Perangkat Penelitian (Software)	43
Tabel 4. 1 Fitur Admin.....	52
Tabel 4. 2 Fitur User	53
Tabel 4. 3 Deskripsi Api	75
Tabel 4. 4 Pengujian Black Box Login Admin.....	112
Tabel 4. 5 Pengujian Black Box Halaman Utama Admin	113
Tabel 4. 6 Pengujian Black Box Halaman Wilayah (Admin).....	114
Tabel 4. 7 Pengujian Black Box Halaman Parkir (Admin).....	116
Tabel 4. 8 Pengujian Black Box Halaman Tips&Trik (Admin)	117
Tabel 4. 9 Pengujian Black Box Login (User).....	119
Tabel 4. 10 Pengujian Black Box Halaman Utama (User)	120
Tabel 4. 11 Pengujian Black Box Halaman Wilayah (User)	120
Tabel 4. 12 Pengujian Black Box Halaman Peta (User)	121
Tabel 4. 13 Pengujian Black Box Halaman Tips&Trik (User).....	122
Tabel 4. 14 UAT-01A	123
Tabel 4. 15 UAT-02A	124
Tabel 4. 16 UAT-03A	125
Tabel 5. 1 Pengujian User Acceptance Testing	127

BAB I

PENDAHULUAN

1.1 Latar Belakang

Tempat parkir merupakan salah satu sarana yang tidak dapat dipisahkan dari sistem transportasi. Semakin bertambahnya jumlah penduduk yang menggunakan sarana transportasi pribadi untuk melakukan aktivitasnya menyebabkan diperlukan jumlah tempat parkir yang memadai. Lahan parkir sangat dibutuhkan bagi pengunjung pusat perbelanjaan, tempat-tempat wisata atau gedung perkantoran. Banyak pengguna kendaraan pribadi baik roda dua ataupun roda empat yang merasa kesulitan dalam mencari lahan parkir karena minimnya informasi tentang lahan parkir (Nurhayati & Ilmi, 2019).

Rata-rata sistem perparkiran yang ada saat ini belum optimal dan masih menggunakan cara konvensional dengan tenaga Sumber Daya Manusia (SDM). Dimana hal ini membuat pengendara kendaraan roda dua atau roda empat masih harus mencari lahan parkir dengan upaya sendiri, sehingga pengendara yang akan menempati lahan parkir harus berputar-putar di sekitar tempat tujuan untuk mencari titik keberadaan tempat parkir atau bahkan terpaksa harus memarkir kendaraannya di luar destinasi (Sani et al., 2018).

Permasalahan yang terjadi jika melakukan parkir sembarangan akan mengakibatkan kemacetan, dikarenakan pergerakan kendaraan yang melewati tempat-tempat yang beraktifitas padat dan volume kendaraan yang tinggi. Hal ini terjadi dikarenakan kurangnya informasi keberadaan titik lokasi parkir di sekitar tempat tujuan bagi pengendara. Dengan ini pengendara roda dua atau roda empat memerlukan suatu sistem informasi tentang titik keberadaan tempat parkir terdekat yang terdapat di sekitar destinasi (Venti et al., 2020).

Terlebih di ibukota seperti Provinsi DKI Jakarta yang merupakan daerah dengan pusat kegiatan ekonomi dan bisnis. Keadaan tersebut membuat permasalahan pada pencarian tempat parkir menjadi tidak efisien terhadap waktu dan tenaga. Selain itu, menyebabkan dampak negatif yang cukup besar

bagi pengendara roda dua ataupun roda empat jika memarkir pada lahan parkir liar, karena kendaraan mereka akan berurusan langsung dengan pihak Dinas Perhubungan setempat.

Pada kasus pelanggaran parkir liar yang terjadi selama bulan Januari hingga Maret 2021, tercatat 397 kendaraan baik roda dua ataupun roda empat yang ditertibkan petugas di wilayah Jakarta Selatan. Lokasi rawan parkir liar yang ditertibkan seperti pusat perbelanjaan, gedung perkantoran dan tempat kegiatan yang menjadi pusat keramaian (Gatra, 2021). Berdasarkan data Unit Pengelola Perparkiran Dishub DKI Jakarta, jumlah pelanggaran parkir liar untuk wilayah DKI Jakarta pada periode Desember 2020 sampai Maret 2021 tercatat mencapai 19.261 kendaraan. Sedangkan jumlah pelanggaran pada periode Maret 2021 sampai Agustus 2021 tercatat mencapai 30.059 kendaraan (Dian, 2021).

Pada penelitian ini penulis melakukan wawancara yang bertujuan untuk mendapatkan informasi lebih lengkap mengenai masalah perparkiran yang terjadi di lapangan. Wawancara dilakukan kepada Bapak Syaefudin Zuhri selaku Manajer Perencanaan Unit Perparkiran Dishub DKI Jakarta. Menurutnya, sistem informasi tempat parkir (*e-parkir*) yang terdapat pada Dinas Perhubungan saat ini sedang dalam tahap perkembangan dengan berbasis *mobile apps* dan dibangun antara kerjasama *Vendor* dengan Dishub DKI Jakarta. Namun ada beberapa fitur dari sistem tersebut yang masih memiliki kekurangan, beberapa juga tidak terdapat alternatif pencarian titik lokasi parkir lainnya yang terdekat.

Terdapat beberapa penelitian menggunakan sistem atau teknologi mengenai solusi dari informasi tempat parkir yang telah dilakukan menggunakan *mobile phone* dan sistem berbasis *website*, namun beberapa masih melakukan optimasi pada sistem tersebut. Sebagian besar menggunakan aplikasi dengan platform *android* karena tergolong mudah dalam pengembangan.

Pada penelitian yang berjudul “Sistem Manajemen Parkir Mobil *On Street* dan *Off Street* Berbasis *Location Based Service* (LBS) dan *Google Maps API*”

(Venti et al., 2020), penelitian ini menghasilkan aplikasi berbasis *android* yang berhasil memberikan informasi pencarian rute parkir terdekat. Memiliki fitur pemesanan parkir secara *online* yang dikelola oleh admin di tempat parkir tersebut, fitur pemberitahuan bagi pengendara yang memberikan informasi masa berlaku pemesanan parkir *online*. Tujuan pembuatan sistem ini adalah memberikan kemudahan bagi pengendara mobil untuk mendapatkan tempat parkir terdekat sesuai destinasi, dalam penelitian ini fitur pencarian lokasi parkir terdekat belum berjalan dengan baik sehingga pengendara harus mencari kembali lokasi parkir terdekat lainnya.

Pada penelitian lainnya yang berjudul “Sistem Aplikasi Pencarian Lokasi Parkir Terdekat Menggunakan *Location Based Service* Berbasis Android” (Nurhayati & Ilmi, 2019), penelitian ini menghasilkan aplikasi berbasis *android* yang berhasil memberikan informasi pencarian rute parkir terdekat. Memiliki fitur list data tempat parkir dan kapasitas parkir yang tersedia. Dalam penelitian ini belum terdapat detail data tempat parkir bagi pengendara mengenai harga parkir dan jarak menuju lokasi parkir dari posisi *user*.

Berdasarkan faktor tersebut, dalam penelitian ini penulis mengembangkan sistem aplikasi informasi tempat parkir terdekat berbasis android dengan beberapa keunggulan fitur. Perlu adanya fitur yang dapat memberikan layanan informasi tempat parkir dengan lebih baik dan efisien bagi pengendara. Penulis menambahkan fitur *wilayah* yang dapat memberikan informasi tempat parkir berdasarkan masing-masing wilayah di DKI Jakarta, fitur *peta* yang dapat menampilkan seluruh titik lokasi parkir dengan tampilan maps dan fitur *tips&trik* yang dapat memberikan informasi ataupun *safety* dalam berkendara bagi pengendara.

Teknologi *Location Based Service* (LBS) merupakan suatu layanan yang bereaksi aktif terhadap perubahan entitas posisi sehingga mampu mendeteksi letak objek dan memberikan layanan sesuai dengan letak objek yang telah diketahui tersebut. *Location Based Service* adalah aplikasi yang bergantung pada lokasi tertentu sebagai layanan informasi dengan memanfaatkan teknologi untuk mengetahui posisi sesuatu. Layanan berbasis lokasi tersebut

menggunakan teknologi *Positioning System*, teknologi ini memungkinkan para pengguna dapat memperoleh informasi lokasi sesuai dengan kebutuhannya. (B. Anwar et al., 2017). LBS adalah salah satu bagian dari implementasi *mobile GIS* (*Geographic Information System*) yang lebih cenderung memberikan fungsi terapan sehari-hari seperti menampilkan direktori kota, navigasi kendaraan, pencarian alamat serta jejaring sosial (Mudzakir & Arifudin, 2017).

Dengan menggunakan juga *Google Maps API* sebagai perantara teknologi LBS, LBS tersebut digunakan dengan memanfaatkan *latitude* dan *longitude* sebagai penyedia informasi utama. *API* menjadi penghubung antara *latitude* dan *longitude* yang telah di input pada web admin dengan android dalam bentuk peta digital (maps). Peta digital tersebut dapat membantu dalam visualisasi titik-titik tempat parkir yang berada disekitar destinasi. Fungsi *Google Maps API* sebagai perantara teknologi LBS juga dapat memfasilitasi sistem untuk menyaring tempat parkir berdasarkan jarak terdekat dengan tempat destinasi dengan cara hanya memperlihatkan tempat parkir di dalam jarak yang diinginkan.

Berdasarkan hasil penelitian yang dilakukan oleh (Budiman, 2016), menyatakan bahwasanya dari ketiga sistem layanan navigasi yang dapat digunakan sebagai penentu lokasi, LBS memiliki akurasi yang sangat baik jika dibandingkan dengan menggunakan layanan GPS dan *Cell-Based*. Sehingga berdasarkan hasil yang didapat adalah teknologi LBS merupakan teknologi yang memiliki keunggulan paling baik untuk digunakan dalam menentukan keputusan.

Berdasarkan latar belakang permasalahan di atas, penulis melakukan penelitian yang berjudul **“PENGEMBANGAN APLIKASI INFORMASI TEMPAT PARKIR TERDEKAT DI PROVINSI DKI JAKARTA MENGGUNAKAN *LOCATION BASED SERVICE* (LBS) DAN *GOOGLE MAPS API* (STUDI KASUS: UNIT PERPARKIRAN DINAS PERHUBUNGAN PROVINSI DKI JAKARTA)”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas, maka pokok permasalahan dalam penelitian ini adalah "Bagaimana mengembangkan aplikasi informasi tempat parkir terdekat di Provinsi DKI Jakarta menggunakan *Location Based Service* (LBS) dan *Google Maps API*?"

1.3 Tujuan Penelitian

Berdasarkan uraian diatas, maka tujuan penelitian ini adalah:

Mengembangkan aplikasi informasi tempat parkir terdekat di Provinsi DKI Jakarta menggunakan *Location Based Service* (LBS) dan *Google Maps API*.

1.4 Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah:

a. Bagi Pengguna

1. Memberikan layanan bagi pengendara dalam mencari informasi titik tempat parkir yang terdekat.
2. Dapat memberikan informasi tempat parkir yang tersedia beserta kapasitas dan harga parkir.

b. Bagi Universitas

1. Sebagai bahan referensi atau bahan acuan bagi penelitian berikutnya.
2. Menjadi bahan tambahan ilmu pengetahuan teknologi informasi pada Universitas.
3. Mengukur tingkat kemampuan mahasiswa dalam menerapkan materi yang diperoleh selama masa perkuliahan.

c. Bagi Penulis

1. Merealisasikan suatu aplikasi informasi tempat parkir yang dapat digunakan pengendara.
2. Mengidentifikasi masalah dan memecahkannya dengan membuat sistem aplikasi informasi tempat parkir.

1.5 Batasan Masalah

Berdasarkan uraian rumusan masalah, dalam penelitian ini penulis membatasi permasalahan pada:

1.5.1 Metode

Berikut merupakan metode yang digunakan oleh penulis dalam penelitian ini:

1. Metode pengumpulan data yang digunakan dalam penelitian ini adalah metode studi pustaka/literatur, metode observasi dan metode wawancara.
2. Metode pengujian awal sistem yang digunakan penulis adalah *Black Box Testing* dan *User Acceptance Testing* (UAT).
3. Model pengembangan sistem *Rapid Application Development* (RAD) digunakan sebagai metode pengembangan sistem.

1.5.2 Proses

Berikut adalah proses pengembangan aplikasi informasi tempat parkir yang dibangun, meliputi;

1. Menggunakan sistem berbasis *android* dalam pengembangannya.
2. Pengembangan aplikasi yang akan dibuat menggunakan bahasa pemrograman PHP untuk membangun web admin dan android studio dengan bahasa pemrograman Java sebagai alat untuk membangun *interface*.
3. Aplikasi hanya berfokus pada pencarian titik lokasi parkir, jarak tempat parkir dan detail data parkir dalam pencarian rute parkir.
4. Fitur pendukung lainnya dalam aplikasi ini adalah fitur *wilayah* untuk mencari titik lokasi parkir berdasarkan wilayah masing-masing (DKI Jakarta) dan *tips&trik* dalam berkendara.
5. Narasumber dari penelitian ini adalah Manajer Perencanaan Unit Perparkiran Dinas Perhubungan DKI Jakarta.

1.5.3 Tools

Berikut adalah tools yang digunakan pada penelitian, yaitu :

1. Pengembangan sistem berbasis Android
2. Apps Android: Android Studio (IDE), Bahasa Pemrograman JAVA
3. Web Administrator: XAMPP, Database MySQL, Sublime Text, PHP
4. Microsoft Visio 2016 sebagai media membuat desain UML
5. Figma digunakan sebagai media untuk membuat desain mockup.
6. Postman digunakan sebagai media untuk melakukan uji API.

1.6 Metodologi Penelitian

Metode yang digunakan penulis dalam penulisan dan penelitian dibagi menjadi dua, yaitu metode pengumpulan data dan metode pengembangan sistem. Berikut penjelasan kedua metode tersebut:

1.6.1 Metode Pengumpulan Data

Dalam melakukan analisis data dan penulisan skripsi ini, penulis menggunakan 2 metode pengumpulan data, yaitu:

1. Metode Observasi
Penulis melakukan observasi dengan pengamatan secara langsung ke lapangan untuk mendapatkan informasi mengenai permasalahan yang dialami sistem parkir Dinas Perhubungan dalam mengelola tempat parkir.
2. Metode Pustaka Penulis mengumpulkan informasi yang relevan mengenai penelitian ini yang dapat diperoleh dari buku – buku, karya ilmiah, jurnal, internet, dan sumber – sumber lainnya.
3. Metode Wawancara
Penulis melakukan wawancara kepada pihak Manajer Perencanaan Perparkiran Dinas Perhubungan DKI Jakarta untuk mendapatkan informasi dan data yang terkait dengan penelitian ini.

1.6.2 Metode Pengembangan Sistem

Pada penelitian ini penulis menggunakan *Rapid Application Development* (RAD). Adapun tiga tahap siklus pengembangan *Rapid Application Development* adalah sebagai berikut: (Kendall, 2017).

1. *Requirements Planning* (Perencanaan Persyaratan).
2. *Design Workshop* (Workshop Desain RAD).
3. *Implementation* (Penerapan Pengujian).

1.7 Sistematika Penulisan

Dalam sistematika laporan ini penulis akan melakukan pembahasan dengan membagi ke dalam 6 bab, yaitu:

BAB I : PENDAHULUAN

Bab ini membahas latar belakang, tujuan, rumusan masalah, serta metode penelitian yang mendasari penelitian.

BAB II : LANDASAN TEORI

Bab ini membahas teori-teori yang digunakan dalam mendukung penelitian dan pengembangan sistem.

BAB III: METODOLOGI PENELITIAN

Bab ini membahas secara rinci metodologi yang digunakan dalam penulisan skripsi yang mencakup analisis, merancang serta implementasi sistem.

BAB IV: IMPLEMENTASI

Bab ini memaparkan hasil analisis, perancangan sistem, implementasi, pengujian sistem dan pembahasan terkait penerapan metode yang digunakan dalam penelitian serta akan diuraikan hasil seluruh tahapan pengembangan sistem yang akan dibangun.

BAB V : HASIL DAN PEMBAHASAN

Bab ini akan memaparkan hasil dan pembahasan penelitian yang telah penulis tampilkan pada bab 4.

BAB VI : KESIMPULAN DAN SARAN

Bab terakhir yang berisi kesimpulan dari hasil apa yang telah diuraikan dan dijelaskan dari bab-bab sebelumnya, serta saran yang bermanfaat untuk pengembangan lebih lanjut.



BAB II

LANDASAN TEORI

2.1 Pengembangan Sistem

Pengembangan sistem (*system depelovment*) dapat diartikan yaitu merangkai suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada (Kamus Besar Bahasa Indonesia, 2017). Bahwa “jika ada permasalahan pada suatu sistem, maka sistem yang baru perlu dikembangkan untuk memecahkan permasalahan-permasalahan yang timbul”. Pengembangan sistem perlu dilakukan ketika terjadi kondisi berikut:

1. Adanya permasalahan-permasalahan yang timbul pada sistem yang lama. Permasalahan yang timbul dapat berupa:

- a. Ketidakberesan

Ketidakberesan dalam sistem yang lama menyebabkan sistem yang lama tidak dapat beroperasi sesuai dengan yang diharapkan.

- b. Pertumbuhan Organisasi

Pertumbuhan organisasi yang menyebabkan berubahnya sistem dan harus disusun sistem yang baru.

2. Untuk meraih kesempatan-kesempatan (*opportunities*).

Teknologi informasi telah berkembang dengan cepatnya. Perangkat keras komputer, perangkat lunak dan teknologi komunikasi telah begitu cepat berkembang. Organisasi mulai merasakan bahwa teknologi informasi ini perlu digunakan untuk meningkatkan penyediaan informasi sehingga dapat mendukung Adanya instruksi-instruksi dalam proses pengambilan keputusan yang akan dilakukan oleh manajemen.

3. Adanya instruksi-instruksi (*directives*).

Penyusunan sistem yang baru dapat juga terjadi karena adanya instruksi-instruksi dari atasan, pimpinan, atau dari luar organisasi, seperti misalnya peraturan pemerintah.

Adapun menurut (Izzaty et al., 2017), pengembangan sistem informasi yang direalisasikan dengan bantuan komputer (*computer information system*) melalui tahapan-tahapan yang dikenal dengan sistem analisis dan desain. Sistem analisis dan desain adalah peningkatan kinerja suatu organisasi dengan tujuan perbaikan prosedur-prosedur dan metode yang lebih baik.

Berdasarkan pengertian pengembangan sistem yang telah diuraikan, proses untuk mengembangkan suatu sistem yang menjadikan sebuah sistem terbaru untuk memecahkan permasalahan baik dengan mengubah sistem yang lama secara keseluruhan ataupun memperbaiki sistem yang telah ada.

2.2 Aplikasi

Aplikasi menurut Dhanta dikutip dari Sanjaya (2017) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, Microsoft Excel. Aplikasi berasal dari kata *application* yang artinya penerapan lamaran penggunaan. Menurut Jogiyanto (2017:4) aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data. Aplikasi secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem/program agar data dapat diolah serta lebih berdaya guna secara optimal.

Jadi, aplikasi merupakan sebuah transformasi dari sebuah permasalahan atau pekerjaan berupa hal yang sulit dipahami menjadi lebih sederhana, mudah dan dapat dimengerti oleh pengguna. Sehingga dengan adanya aplikasi, sebuah permasalahan akan terbantu lebih cepat dan tepat.

Sementara itu, Aplikasi adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur sebuah windows, permainan (*game*) dan sebagainya. (Hakim, 2018)

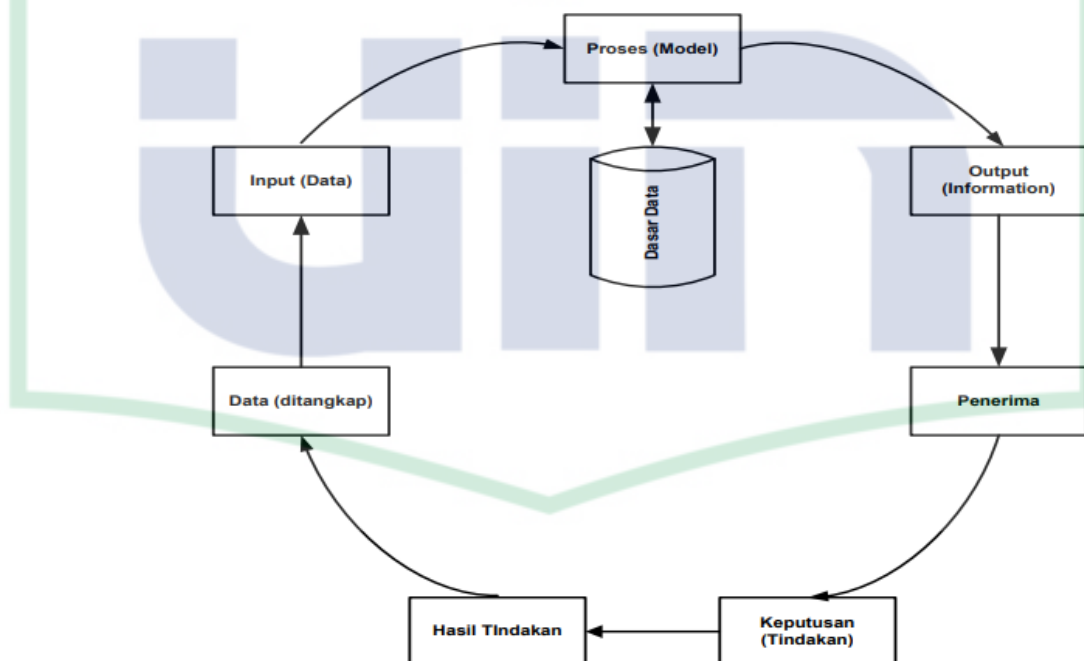
Aplikasi memiliki banyak jenis. Diantaranya aplikasi desktop yang beroperasi secara *offline* dan aplikasi web yang beroperasi secara *online*. Aplikasi web merupakan sebuah aplikasi yang menggunakan teknologi *browser* untuk menjalankan aplikasi dan diakses melalui jaringan komputer, menurut Remick dalam jurnalnya yang dikutip Ramzi (2018).

2.3 Informasi

Informasi merupakan data atau fakta yang telah diproses sedemikian rupa, sehingga berubah bentuknya menjadi informasi. Di samping itu informasi dapat mengurangi ketidakpastian serta mempunyai nilai dalam keputusan karena dengan adanya informasi kita dapat memilih tindakan-tindakan dengan resiko yang paling kecil (Sudjiman, 2020).

Menurut Tata Sutabri (2018:22) pada buku Analisis Sistem Informasi, Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Informasi ibarat darah yang mengalir didalam tubuh suatu organisasi, sehingga informasi ini sangat penting didalam suatu organisasi. Suatu sistem yang kurang mendapatkan informasi akan menjadi luruh, kerdil dan akhirnya berakhir.

Informasi merupakan hal yang sangat penting bagi manajemen dalam pengambilan keputusan. Berikut pengertian informasi menurut beberapa ahli yaitu:



Gambar 2. 1 Siklus Informasi

(Ariati, S.Kom, 2017)

Mulyanto (2017;2) informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya, sedangkan data merupakan sumber informasi yang menggambarkan suatu kejadian nyata. Sedangkan pengertian informasi menurut Kadir (2018;71) data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut.

Berdasarkan uraian, penulis menyimpulkan bahwa informasi adalah data yang telah diproses dengan suatu cara untuk memberikan arti dan memperbaiki pengambilan keputusan.

2.4 Tempat Parkir

Tempat parkir merupakan salah satu sarana yang tidak dapat dipisahkan dari sistem transportasi. Semakin bertambahnya jumlah penduduk yang menggunakan sarana transportasi pribadi untuk melakukan aktivitasnya menyebabkan diperlukan jumlah tempat parkir yang memadai. Permasalahan yang terjadi jika melakukan parkir sembarangan akan mengakibatkan kemacetan dikarenakan pergerakan kendaraan yang melewati tempat-tempat yang beraktifitas tinggi, laju pergerakannya akan terhambat oleh kendaraan yang parkir disembarang tempat, selain itu akan meningkatkan resiko terjadinya kecelakaan. Permasalahan tersebut dikarenakan kurang tahunya pengendara dalam titik keberadaan lokasi parkir di tempat tujuan juga jadi faktor utama pada masalah tersebut karena tidak banyak rambu atau penunjuk arah yang menunjukan lokasi parkir di area tujuan para pengendara (Nurhayati & Ilmi, 2019).

Pada penelitian lainnya, menurut (Venti et al., 2020) Parkir adalah keadaan tidak bergerak suatu kendaraan yang bersifat sementara karena ditinggalkan oleh pengemudinya. Secara hukum dilarang untuk parkir di tengah jalan raya, namun parkir di sisi jalan umumnya diperbolehkan. Fasilitas parkir dibangun bersama-sama dengan kebanyakan gedung, untuk memfasilitasi kendaraan pemakai gedung. Termasuk dalam pengertian parkir adalah setiap kendaraan yang berhenti pada tempat-tempat tertentu baik yang dinyatakan

dengan rambu lalu lintas ataupun tidak, serta tidak semata-mata untuk kepentingan menaikkan, menurunkan orang atau barang.

Parkir memerlukan suatu lahan / tempat parkir yang dapat menampung kendaraan yang ingin diparkirkan seperti tempat parkir di bahu jalan (*on street*) maupun di gedung parkir (*off street*). Maka setiap memarkirkan kendaraan, pengendara harus mencari terlebih dahulu di suatu tempat parkir yang masih tersedia ruang untuk dapat memarkirkan kendaraan pengendara. Parkir merupakan hal penting ketika berpergian kemana-mana, terutama tempat parkir kendaraan dengan semakin banyaknya kendaraan pribadi saat ini, seringkali menyebabkan padatnya tempat parkir dan menyebabkan keterbatasan tempat parkir. Dari sini pengendara diharuskan mencari tempat parkir yang tersedia.

2.5 Location Based Service (LBS)

Teknologi *Location Based Service* (LBS) merupakan suatu layanan yang bereaksi aktif terhadap perubahan entitas posisi sehingga mampu mendeteksi letak objek dan memberikan layanan sesuai dengan letak objek yang telah diketahui tersebut. *Location Based Service* adalah aplikasi yang bergantung pada lokasi tertentu sebagai layanan informasi dengan memanfaatkan teknologi untuk mengetahui posisi sesuatu. Layanan berbasis lokasi menggunakan teknologi *Positioning System*, teknologi ini memungkinkan para pengguna dapat memperoleh informasi lokasi sesuai dengan kebutuhannya. (Anwar et al., 2018).

LBS merupakan salah satu bagian dari implementasi *mobile GIS* (*Geographic Information System*) yang lebih cenderung memberikan fungsi terapan sehari-hari seperti menampilkan direktori kota, navigasi kendaraan, pencarian alamat serta jejaring sosial dibanding fungsionalitas pada teknologi GIS populer untuk Field Based GIS. Metode ini digunakan untuk memberikan layanan informasi lokasi parkir berdasarkan lokasi yang terdekat dengan pengendara. (Mudzakir dan Arifudin, 2017).

Location Based Service (LBS) istilah umum yang digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat

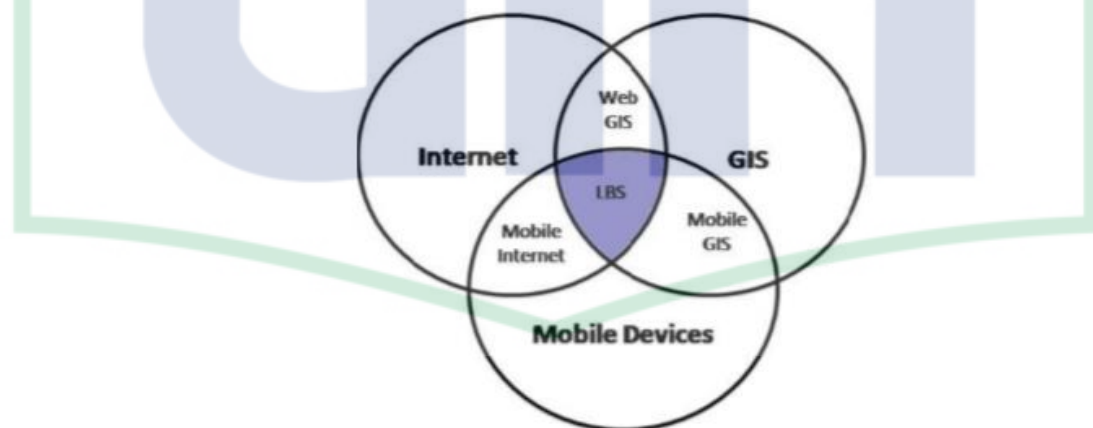
yang kita gunakan. LBS adalah layanan informasi yang dapat diakses melalui *mobile device* dengan menggunakan *mobile network*, yang dilengkapi kemampuan untuk memanfaatkan lokasi dari *mobile device* tersebut. Terdapat dua unsur utama pada LBS, yaitu:

1. *Location Manager (API Maps)*

Menyediakan *tools/source* untuk LBS, *Application Programming Interface (API) Maps* menyediakan fasilitas untuk menampilkan, memanipulasi maps/peta beserta fitur-fitur lainnya seperti tampilan satelit, *street* (jalan), maupun gabungannya. Paket ini berada pada *com.google.android.maps*.

2. *Location Provider (API Location)*

Menyediakan teknologi pencarian lokasi yang digunakan oleh *device/perangkat*. *API Location* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real-time*. *API Location* berada pada paket android yaitu dalam paket *android.location*. Dengan *Location Manager*, kita dapat menentukan lokasi kita saat ini, track gerakan/perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan



Gambar 2. 2 Teknologi Location Based Service (LBS)

(Sumber : <https://www.258822-pemanfaatan-teknologi-location-based-ser-f31b2612.pdf>)

Location Based Service dapat digambarkan sebagai suatu layanan yang berada pada pertemuan tiga teknologi yaitu: *Geographic Information System*, *Internet Service*, dan *Mobile Devices*. Teknologi *Location Based Services* berfokus bagaimana menentukan posisi dari peralatan yang digunakan atau disebut dengan metode *positioning*. Jika berbicara tentang *positioning*, beberapa sistem yang digunakan ini dapat dibedakan dalam tiga jenis, yaitu:

1. Secara Manual

Metode manual adalah cara-cara konvensional yang selama ini sering dilakukan baik yang melalui *yellow pages*, telepon bantuan operator, dan sebagainya. Cara-cara ini tentunya sangat merepotkan dan sudah mulai ditinggalkan. Munculnya internet memberi perspektif baru bagi fasilitas pencarian ini yang lebih luas. Ditambah dengan munculnya komunikasi selular, tentunya mobilitas pun semakin meningkat. Dengan menggabungkan kedua teknologi ini, sudah pasti metode manual akan ditinggalkan.

2. Melalui GPS (*Global Positioning System*)

GPS merupakan sistem navigasi radio di seluruh dunia yang memanfaatkan 24 satelit beserta stasiun bumi. Melalui sistem ini, bumi dibagi menjadi kotak-kotak dengan masing-masing memiliki alamat yang unik sehingga dapat mengidentifikasi dengan tepat setiap lokasi tersebut.

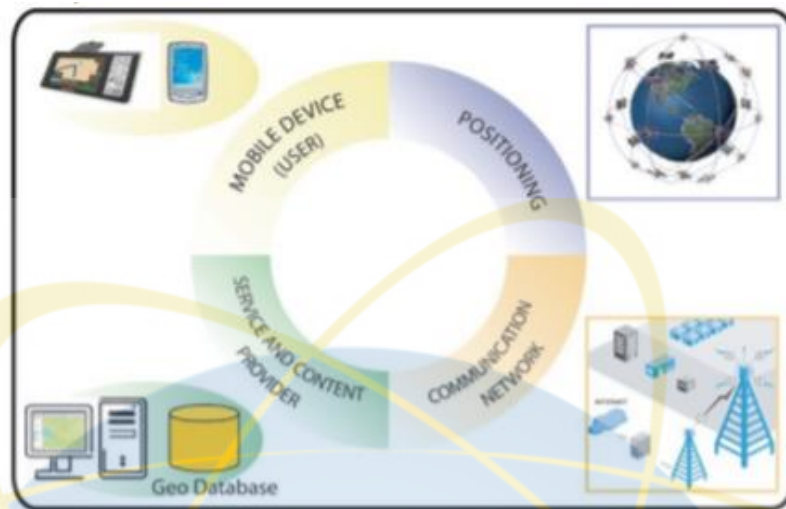
3. Melalui *Cellular Based Station* atau BTS (*Base Transceiver Station*)

Teknologi *Cellular Based Station*, berbasis pada jaringan telekomunikasi selular yang memungkinkan digunakan dalam ruangan atau *indoor*. Sebuah *handphone* dapat ditentukan posisinya saat ini berdasarkan posisi relatif terhadap sebuah *handphone* dalam satu atau lebih *cell tower* terdekat dengan mempertimbangkan sinyal yang digunakan untuk melayani sebuah *handphone*. Dengan menggunakan prinsip triangulasi, posisi *handphone* dapat terdeteksi. *Cellular Based Station* memiliki akurasi yang sangat kurang baik jika dibandingkan dengan menggunakan GPS.

2.5.1 Komponen Location Based Services (LBS)

Dalam menggunakan LBS elemen komponen yang diperlukan antara lain:

1. *Mobile Device*, yaitu sebuah alat yang digunakan untuk meminta informasi yang dibutuhkan. Biasanya perangkat yang memungkinkan yaitu *Mobile Phone*, Laptop, dan perangkat lainnya yang mempunyai fasilitas navigasi.
2. *Communication Network*, adalah jaringan selular yang mengirimkan data pengguna dan permintaan layanan.
3. *Positioning Component*, untuk pengolahan layanan biasanya posisi pengguna harus ditentukan. Posisi pengguna dapat diperoleh menggunakan jaringan komunikasi atau dengan menggunakan *Global Positioning System* (GPS). Selain GPS media pengambil posisi bisa ditentukan oleh *Cell Tower* dan kombinasi antara GPS dan *Cell Tower* (aGPS).
4. *Service and Application Provider*, adalah penyedia layanan pengguna selular yang bertanggung jawab untuk memproses layanan. Provider melakukan proses komputasi seperti menemukan rute perjalanan, informasi tempat terdekat, mencari ke database eksternal seperti *yellow pages* atau google API untuk menghasilkan informasi yang dibutuhkan oleh pengguna.
5. *Data and Content Provider*, yaitu penyedia layanan informasi data yang dapat diminta oleh pengguna. Penyedia layanan tidak selalu menyimpan seluruh data dan informasi yang diolahnya. Karena bisa jadi berbagai macam data dan informasi yang diolah tersebut berasal dari pengembang/pihak ketiga yang memang memiliki otoritas untuk menyimpannya. Sebagai contoh, basis data geografis dan lokasi bisa saja berasal dari badan-badan milik pemerintah atau juga data-data perusahaan/bisnis/industri bisa saja berasal dari *yellow pages*, maupun perusahaan penyedia data lainnya.



Gambar 2. 3 Komponen Location Based Service

(Sumber : <https://www.258822-pemanfaatan-teknologi-location-based-ser-f31b2612.pdf>)

2.6 Google Maps API

Google Maps Application Programming Interface (API) merupakan suatu fitur aplikasi yang dikeluarkan oleh google untuk memfasilitasi pengguna yang ingin mengintegrasikan Google Maps ke dalam website masing-masing dengan menampilkan data *point* milik sendiri. Dengan menggunakan *Google Maps API*, Google Maps dapat di-embed pada website eksternal. Agar aplikasi Google Maps dapat muncul di website tertentu, diperlukan adanya API key. API key merupakan kode unik yang digenerasikan oleh google untuk suatu website tertentu, agar server Google Maps dapat mengenali, namun untuk *Google Maps API* versi 3 sudah tidak membutuhkan API key, sehingga mempermudah untuk menggunakan *Google Maps API*, dan pada versi ini juga terdapat beberapa perbedaan sintaks dari kode yang digunakan. (Venti et al., 2020).

Menurut (Rachmat, 2017) *Google Maps API* adalah suatu *library* yang berbentuk Javascript. Cara membuat Google Maps untuk ditampilkan pada suatu *web* atau *mobile* membutuhkan pengetahuan mengenai HTML serta Javascript, serta koneksi internet yang sangat stabil. Dengan menggunakan

Google Maps API, dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga programmer dapat fokus dengan data – data yang ditampilkan.

Dengan *Google Maps API*, sistem dapat menampilkan posisi *latitude* and *longitude* dari posisi tempat parkir dalam bentuk peta digital. Peta digital tersebut dapat membantu dalam visualisasi tempat-tempat parkir yang berada disekitar lokasi destinasi. *Google Maps API* juga dapat memfasilitasi sistem untuk menyaring tempat parkir yang terdapat dalam radius terdekat dengan tempat destinasi dengan cara hanya memperlihatkan tempat parkir di dalam radius saja dan yang masih tersedia tempat parkirnya.

API atau *Application Programming Interface* merupakan suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Dengan adanya API ini, maka memudahkan *programmer* untuk “membongkar” suatu *software* yang kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan programmer menggunakan sistem *function*. Proses ini dikelola melalui *operating system*. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi. Bahasa pemrograman yang digunakan oleh Google Maps yang terdiri dari *HTML*, *AJAX* serta *XML*, memungkinkan untuk menampilkan peta Google Maps di website lain (Literate & In, 2020).

2.7 Android

Android merupakan sistem informasi berbasis *Linux* yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet (2017 : 1) (Yudhanto & Wijayanto, 2017). Pernyataan tersebut sejalan dari (Späth, 2018) yang menyatakan bahwa Android merupakan sistem operasi yang didasarkan pada kernel *Linux* yang dirancang khusus sehingga menyediakan semua saluran komunikasi tingkat rendah (2018 : 1). Penggunaan teknologi Android memiliki keunggulan tersendiri yaitu untuk proses pengembangan

sebuah aplikasi yang tidak memerlukan akun tertentu (Hakim, Harefa, & Widodo, 2019).

2.7.1 Kelebihan Android

Android memiliki kelebihan yang membuatnya semakin diminati oleh penggunanya (Satyaputra, 2017). Berikut merupakan kelebihan dari android:

1. *Hardware*, didukung secara perangkat keras. Android dapat ditemukan pada banyak *smartphone* yang berasal dari produsen berbeda.
2. *Software*, memungkinkan untuk menjalankan aplikasi *software* pada saat yang sama.
3. *User Friendly*, yang dimaksudkan disini adalah sistem Android sangat mudah untuk dijalankan. Sama halnya pada sistem operasi Windows yang ada pada komputer. Bagi pengguna yang belum terlalu terbiasa dengan *smartphone*, pengguna hanya membutuhkan waktu sebentar saja untuk bisa mempelajari sistem Android.
4. *User* akan sangat mudah mendapatkan beragam notifikasi dari *smartphone* miliknya. Untuk mendapatkannya, *user* bisa mengatur akun yang dimiliki seperti *SMS*, *Email*, *Voice Dial*, dan lainnya.
5. Keunggulan lainnya terdapat dari segi tampilan sistem Android yang menarik dan tidak kalah baiknya dengan iOS (*Apple*). Hal ini dikarenakan dari awal, Android memang mengusung konsep dan teknologi iOS hanya saja Android merupakan versi murah dari iOS
6. Sistem operasi ini memang memiliki konsep *open source* yang mana pengguna dapat bebas mengembangkan sistem Android versi miliknya sendiri. Sehingga akan banyak sekali *custom ROM* yang bisa digunakan.
7. Tersedia beragam pilihan aplikasi menarik, bahkan hingga jutaan aplikasi. Dari mulai aplikasi gratis hingga aplikasi berbayar. Pengguna dapat mendownloadnya langsung di *Google Playstore*

yang tersedia pada smartphone miliknya.

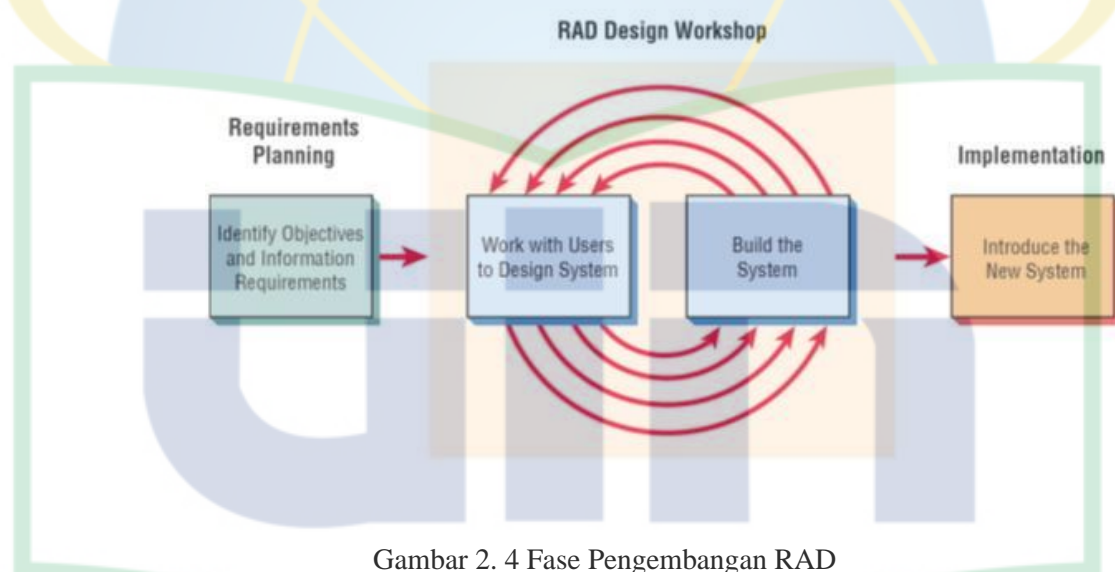
2.7.2 Kekurangan Android

Selain kelebihan, Android juga memiliki kekurangan, yaitu:

1. *Update system* yang kurang efektif. Sistem Android memang seringkali mengalami peningkatan versi yang ditawarkan kepada penggunanya. Namun untuk *update system* Android ini bukanlah hal yang mudah. Pengguna diharuskan untuk menunggu masing – masing *vendor* merilis resmi *update* terbaru dari sistem Android tersebut. Hal tersebut menyebabkan hal yang kurang menyenangkan bagi penggunanya.
2. Baterai yang cepat habis, mungkin hal ini sering dialami oleh pengguna sistem Android sekarang. Apalagi jika sering menyalakan paket data serta menggunakan *widget* dan aplikasi yang berjalan terlalu berlebihan sehingga menyebabkan daya baterai berkurang dengan cepat.
3. Ketersediaan aplikasi, *Google* tidak menyaring aplikasi yang ditambahkan pada *market* Android sehingga *user* dapat menemukan aplikasi yang kualitasnya dibawah standar.
4. Sering mengalami koneksi lambat atau lag, hal ini biasanya berkaitan dengan spesifikasi dari masing – masing perangkat seluler. Namun meskipun begitu, jika sistem Android memang tidak bersahabat dengan aplikasi – aplikasi yang dimiliki tentunya akan berdampak pada lambatnya penggunaan *smartphone*. Hal ini dikaitkan dengan *RAM* atau *Processor* yang memang kurang memadai. Sehingga akan lebih baik jika menyimpan aplikasi yang memang sesuai dengan *smartphone* yang dimiliki.

2.8 Rapid Application Development (RAD)

RAD merupakan model proses perangkat lunak yang menekankan pada daur pengembangan hidup yang singkat. RAD merupakan versi adaptasi cepat dari model *waterfall*, dengan menggunakan pendekatan konstruksi komponen (Putri & Effendi, 2018). RAD merupakan gabungan dari bermacam-macam teknik terstruktur dengan teknik *prototyping* dan teknik pengembangan *joint application* untuk mempercepat pengembangan sistem/aplikasi (Kendall, 2017). Dari definisi konsep RAD ini, dapat dilihat bahwa pengembangan aplikasi dengan menggunakan metode RAD dapat dilakukan dalam waktu yang relatif lebih cepat. Sesuai dengan metodologi RAD berikut ini adalah tahap-tahap pengembangan aplikasi dari tiap-tiap fase :



Gambar 2. 4 Fase Pengembangan RAD

(Sumber: <https://adikristanto.net/rapid-application-development-rad/>)

Menurut Whitten & Bentley, *Rapid Application Development* (RAD) adalah sebuah strategi pengembangan sistem yang menekankan kecepatan dalam pengembangan melalui keterlibatan pengguna dalam pembangunan secara cepat, iteratif, dan incremental dari suatu serangkaian prototype dari suatu sistem yang dapat berkembang menjadi suatu sistem akhir atau versi tertentu.

2.8.1 Tahapan Pengembangan RAD

Tahapan RAD terdiri dari 3 tahap yang terstruktur dan saling bergantung disetiap tahap (Kendall, 2017), yaitu :

1. *Requirements Planning* (Perencanaan Persyaratan).
 - a) Pengguna dan analisis bertemu untuk mengidentifikasi tujuan dari aplikasi atau sistem.
 - b) Berorientasi pada pemecahan masalah bisnis.
2. *Design Workshop* (Workshop Desain RAD).
 - a) Fase desain dan menyempurnakan.
 - b) Gunakan kelompok pendukung keputusan sistem untuk membantu pengguna setuju pada desain.
 - c) *Programmer* dan analis membangun dan menunjukkan tampilan visual desain dan alur kerja pengguna.
 - d) Pengguna menanggapi prototipe kerja aktual.
 - e) Analis menyempurnakan modul dirancang berdasarkan tanggapan pengguna.
3. *Implementation* (Penerapan).
 - a) Sebagai sistem yang baru dibangun, sistem baru atau parsial diuji dan diperkenalkan kepada organisasi.
 - b) Ketika membuat sistem baru, tidak perlu untuk menjalankan sistem yang lama secara paralel.

2.8.2 Kelebihan dan Kekurangan RAD

Menurut Whitten & Bentley, *Rapid Application Development* (RAD) memberikan beberapa keuntungan dalam pemanfaatannya dalam membangun *prototype website* namun juga memiliki kekurangan yang dapat menyebabkan kerugian. Berikut adalah penjabaran lebih lanjut mengenai keuntungan dan kerugian *Rapid Application Development* dalam pemanfaatannya.

Tabel 2. 1 Keuntungan dan Kerugian Rapid Application Development (RAD)

Keuntungan	Kerugian
<ol style="list-style-type: none"> 1. Sangat berguna untuk proyek dimana kebutuhan sistem yang akan dikembangkan belum sepenuhnya tepat atau tidak menentu. 2. Mendorong antusias <i>end-user</i> untuk berpartisipasi dalam proyek tersebut. 3. Proyek memiliki visibilitas dan dukungan yang tinggi yang dikarenakan keterlibatan pengguna. 4. Solusi berbasis <i>software</i> lebih cepat dibandingkan solusi berbasis bisnis model. 5. Kesalahan dan kelalaian dapat diketahui lebih cepat pada <i>prototype</i> ini dibandingkan pada <i>system model</i>. 6. Uji coba adalah suatu keharusan dari suatu produk yang mendasari dari pendekatan <i>prototype</i>. 7. Pendekatan secara iterative lebih wajar karena perubahan adalah factor yang diharapkan dalam proses pengembangannya. 	<ol style="list-style-type: none"> 1. Beberapa isu yang mengatakan RAD lebih membutuhkan banyak tenaga, dukungan, dan pengembangan yang menyebabkan meningkatnya biaya yang diperlukan. 2. RAD dapat menyelesaikan masalah yang keliru jika masalah tersebut diabaikan dan disingkat. 3. <i>Prototype</i> berbasis RAD dapat memungkinkan berkurangnya semangat dari <i>analysts</i> dibandingkan yang lainnya. 4. Terkadang para <i>stakeholder</i> melihat <i>prototype</i> sebagai pembuangan tenaga dan waktu. 5. Menekankan pada kecepatan penyelesaian proyek dapat berdampak buruk bagi kualitas produk.

(Sumber : Whitten & Bentley, 2016, p100)

2.9 Android Studio

Menurut (Makiolor, Sinsuw, & Najoan, 2017) “Android Studio adalah sebuah IDE (*Integrated Development Environment*) yang bisa digunakan untuk pengembangan aplikasi Android, dan dikembangkan oleh google”. Android Studio adalah hasil pengembangan dari Eclipse IDE, terdapat fitur-fitur baru dalam pengembangan tersebut diantaranya adalah menggunakan *Gradle-based build system* yang fleksibel, dapat membangun *multiple* APK (format berkas), *Template support* untuk *Google Service* dan berbagai macam tipe perangkat dan layout editor yang lebih bagus (Rohman et al., 2019). Sistem informasi berbasis android saat ini sudah menjadi suatu kebutuhan yang penting, terlebih lagi dengan bantuan internet, wisatawan tidak perlu menunggu terlalu lama untuk mendapatkan pelayanan informasi dan lokasi pada tempat yang di tuju. Untuk mendukung hal tersebut maka dibutuhkan sebuah aplikasi untuk membantu pelayanan informasi dan petunjuk arah secara online dengan android yang sebelumnya dilakukan secara manual, dengan adanya aplikasi ini diharapkan mempermudah wisatawan untuk pergi ketempat yang dituju. (Rusdina et al., 2020)

Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi android dan bersifat *open source* atau gratis. Android studio awalnya dimulai dengan 0.1 pada bulan Mei 2013, kemudian dibuat versi beta 0.8 yang dirilis pada bulan Juni 2014. Yang paling stabil dirilis pada bulan Desember 2014, dimulai dari versi 1.0. Berbasiskan *JetBrainns IntelliJ IDEA*, Studio didesain khusus *Android Development* yang kini sudah bisa di *download* untuk *Windows*, *Mac OS X*, dan *Linux*. Android studio menggantikan *Eclipse* sebagai IDE resmi untuk mengembangkan aplikasi android. (Zaningsih, 2018).

2.9.1 Fitur-fitur Android Studio

Android Studio adalah lingkungan pengembangan terpadu *Integrated Development Environment* (IDE) untuk mengembangkan aplikasi android, selain merupakan editor kode *IntelliJ* dan alat pengembangan yang berdaya guna. Android studio menawarkan fitur

yang lebih banyak untuk meningkatkan produktifitas dalam pembuatan aplikasi android. Diantaranya:

1. *Instant Run*, adalah fitur baru Android Studio yang memungkinkan untuk melakukan *compile* dan *run* program menjadi lebih cepat. Pada *Instant Run*, android studio akan melakukan *compile* dan *run* program lebih lama pada saat pertama kali, tetapi untuk kedua kali dan seterusnya akan lebih cepat. Hal ini dikarenakan pada saat eksekusi untuk kedua kalinya, Android Studio hanya mengaplikasikan perubahan dari hasil *file* APK sebelumnya
2. *Code Editor* Pintar, Android Studio merupakan code editor yang pintar. Hal ini dikarenakan Android Studio memiliki fitur *auto completion* yaitu menampilkan saran kode yang kita kaitkan. Selain itu IDE juga memiliki kemampuan analisa kode yang baik serta dapat melakukan *refactoring*. Dengan ini tentunya pembuatan program menjadi lebih produktif.
3. Sistem *Build* yang Handal dan Fleksibel, pada fitur ini para pembuat program dimudahkan dalam melakukan *build* aplikasi android kekurangan dan kelebihan Android Studio. Apabila hendak melakukan *compile* dan menjalankan aplikasi, *file* APK sudah otomatis terbentuk sehingga para pembuat program tidak perlu repot-repot untuk *build* aplikasi kembali.
4. Dengan Android Studio, dapat membuat aplikasi yang diperuntukkan bagi perangkat *handphone* atau *smartphone*. Selain itu Android Studio dapat membuat aplikasi berjalan di *Smartwatch*, *Tablet*, Android TV dan Android Auto.

2.9.2 Kelebihan dan Kekurangan Android Studio

Menurut pengembangan Android Studio, kelebihan dari software ini dilengkapi dengan *intelligent code* editor yang mampu mengolah dan menganalisis kode secara lengkap yang menjadikan *developer* semakin produktif. Selain itu, Android Studio dilengkapi dengan *Code Templates*

dan *Github integration* yang memudahkan *developer* Android dalam mengembangkan aplikasi mereka dari sampel kode yang disediakan ataupun mengimpornya dari *Github*. Android Studio dilengkapi dengan emulator yang mencakup semua *devices*, baik ukuran maupun bentuk. Keunggulan tersebut memudahkan *developer* untuk melihat hasil project mereka dari berbagai *device* yang ada.

Kekurangan dalam *software* ini yaitu, tidak banyak kelemahan yang bisa ditemukan untuk produk-produk *Google*, termasuk *software* Android Studio. Namun ukuran *file* Android Studio untuk *windows* yang direkomendasikan sangat besar (hampir 1 GB) yang seringkali menjadi masalah saat mendownload *software* ini.

2.10 Postman

Postman merupakan sebuah REST client berbasis web yang tersedia dalam bentuk ekstensi pada Google Chrome. Postman mempunyai tampilan antarmuka (*user interface*) yang baik dan juga lengkap. Postman mempunyai fitur seperti *design*, *build*, *test* dan *documentation* API (Rachmat, 2017). Sedangkan dalam penelitian lain, postman merupakan *tool* bagi para *developer* yang berfokus pada pembuatan API, fungsi utama postman ini adalah sebagai GUI API Caller namun sekarang postman juga menyediakan fitur lain, yaitu Sharing Collection API for Documentation, Testing API, Realtime Collaboration Team, Monitoring API dan Integration (Priyadi & Wellem Sagay, 2019).

Postman adalah aplikasi untuk unit test dalam segi koneksi pada *backend* aplikasi tersebut, aplikasi ini untuk mencoba hasil respon yang diberikan oleh *backend* melalui API, hasil respon tersebut berbentuk JSON, dimana nanti akan dipakai pada aplikasi. (Priyadi & Wellem Sagay, 2019).

2.11 Studi Literatur Sejenis

Penulis menggunakan literatur sejenis yang sudah ada sebelumnya sebagai referensi. Hal ini bertujuan agar penelitian ini dapat dilakukan dengan benar, serta faktor – faktor yang mempengaruhi kualitas penelitian ini dengan penelitian sebelumnya berupa teknik tertentu yang digunakan dapat diadopsi pada penelitian ini. Dengan demikian penulis merangkum literatur yang berkaitan dengan penelitian ini sebagai berikut:

Tabel 2. 2 Studi Literatur Sejenis

No	Judul (Penulis & Tahun)	Masalah / Rumusan Masalah	Metode, Pengujian, Tools, & Proses	Kelebihan	Kekurangan
1.	IOT Based Prototype for Parking Monitoring and Management System Commanded by Mobile App Durga Devi TJB, Dr. A.Subramani, Dr. Vijender Kumar Solanki (2017)	Perlunya prototipe yang akan memantau ketersediaan ruang parkir melalui teknologi <i>Internet of Things (IoT)</i> yang muncul dan mengotomatiskan sistem ticketing dari ponsel pintar dengan sistem <i>m-Ticketing</i> .	Metode & Pengujian : Server Cloud Tools : - Codeigniter - PHP Proses : - Proses pengumpulan data - menganalisis kebutuhan. - Perancangan sistem dengan membuat use case diagram, class diagram	Sistem Server berbasis <i>cloud</i> biasanya dapat berupa server elastis Amazon yang mudah ditingkatkan dan diturunkan berdasarkan kecepatan pembuatan data	- <i>Tools</i> yang digunakan tidak dijelaskan secara detail. - Proses pengambilan data masih belum dijelaskan. - Tidak dijelaskan proses pengujian aplikasi. - Perlu adanya perbaikan tampilan.

			- Pembuatan program		
2.	<p>Aplikasi Pencarian Parkir DKI Jakarta Berbasis Android Menggunakan <i>RESTFUL API</i></p> <p>Muhammad Adam Dzulqarnain, Muhammad Luthfi Aziz, Muhammad Fauzi Rachman (2019)</p>	Memudahkan pencarian parkir di Jakarta berupa aplikasi berbasis android menggunakan <i>RESTFUL API</i>	<p>Metode/Pengujian :</p> <p><i>Scrum</i></p> <p>Tools :</p> <ul style="list-style-type: none"> - Codeigniter - PHP <p>Proses :</p> <ul style="list-style-type: none"> - 	<p>Aplikasi sudah berbasis <i>mobile</i> sehingga dapat digunakan dimana saja dan kapan saja. Aplikasi sudah mengimplementasikan <i>webservice</i> (REST API) sehingga penggunaannya dapat bersifat fleksibel karena selalu terupdate data masukan baru.</p>	<ul style="list-style-type: none"> - <i>Tools</i> yang digunakan tidak dijelaskan secara detail. - Proses pengambilan data masih belum dijelaskan. - Proses pengembangan sistem lebih menekankan pada REST API dan tidak dijelaskan pengembangan sistem pada mobile nya.
3.	<p>Sistem Aplikasi Pencarian Lokasi Parkir Terdekat Menggunakan Location Based Service Berbasis Android</p> <p>S Nurhayati, Egi Nur Ilmi (2019)</p>	Perlunya sistem yang memudahkan pengendara dalam mencari lokasi parkir terdekat dan lokasi parkir yang kosong.	<p>Metode & Pengujian:</p> <p><i>Location Based Service</i> (LBS)</p> <p>Tools :</p> <ul style="list-style-type: none"> - PHP - Database MySQL - UML (Unified Modeling Language) <p>Proses :</p>	<p>Sistem sudah menggunakan basis <i>mobile apps</i> yang berguna untuk memudahkan para pengendara mencari informasi tempat parkir. Aplikasi sudah mengimplementasikan <i>Location Based Service</i> (LBS) sehingga lebih efektif terhadap penggunaanya.</p>	<ul style="list-style-type: none"> - Perlu adanya fitur pemberitahuan untuk user - Data tempat parkir yang digunakan berdasarkan sampel peneliti sendiri. - <i>Tools</i> yang digunakan tidak dijelaskan secara detail.

			<ul style="list-style-type: none"> - Proses pengumpulan data - menganalisis kebutuhan. - Perancangan sistem dengan membuat use case diagram, class diagram. - Pembuatan program 		
4.	<p>An Intelligent Indoor Location-Based Smartphone Parking Service</p> <p>Jingbin Liu, Ruizhi Chen, Yuwei Chen, Ling Pei, Liang Chen (2020)</p>	<p>Mengembangkan dan mendemonstrasikan layanan parkir <i>smartphone</i> berbasis lokasi dalam ruangan (<i>iParkir</i>) di fasilitas parkir nyata dari pusat perbelanjaan.</p>	<p>Metode & Pengujian: <i>Location Based Service</i> (LBS)</p> <p>Tools :</p> <ul style="list-style-type: none"> - Android Studio - Google Maps API - Hardware (NOKIA N8) - Software (Qt SDK & Qt Creator) <p>Proses :</p> <ul style="list-style-type: none"> - Proses pengumpulan data - menganalisis kebutuhan. - Pembuatan program 	<p>Sistem sudah menggunakan basis <i>mobile apps</i> yang berguna untuk memudahkan para pengendara mencari informasi tempat parkir.</p> <p>Aplikasi mengimplementasikan RSSI WLAN yang tergolong hemat biaya dan dapat dioperasikan bersama dengan layanan komunikasi</p>	<p>Proses pengumpulan data nya kurang dijelaskan.</p> <ul style="list-style-type: none"> - Metode pengembangan sistemnya tidak dijelaskan. - <i>Tools</i> yang digunakan tidak dijelaskan secara rinci. - Tidak dijelaskan proses pengujian aplikasi.

5.	<p>Sistem Manajemen Parkir Mobil <i>On Street</i> dan <i>Off Street</i> Berbasis <i>Location Based Service</i> (LBS) dan <i>Google Maps API</i></p> <p>Cornelius Venti, Yus Sholva, Rudy Dwi Nyoto (2020)</p>	<p>Perlunya suatu informasi tentang ketersediaan tempat parkir yang tersedia di sekitar destinasi sebelum sampai di destinasi. Maka sistem manajemen parkir sangat diperlukan untuk menyediakan informasi tentang tempat parkir yang tersedia sekitar destinasi.</p>	<p>Metode & Pengujian: <i>Location Based Service</i> (LBS)</p> <p>Tools :</p> <ul style="list-style-type: none"> - PHP - Database MySQL - Laravel Framework - UML (Unified Modeling Language) <p>Proses :</p> <ul style="list-style-type: none"> - Proses pengumpulan data - menganalisis kebutuhan. - Perancangan sistem dengan membuat use case diagram, class diagram. - Pembuatan program. 	<p>Sistem ini memiliki fitur-fitur yang cukup baik, contoh: booking parkir, top up saldo. Pencarian rute terdekat dengan teknologi LBS pun cukup akurat. Sistem ini berbasis website untuk memudahkan pengguna dalam mencari tempat parkir.</p>	<ul style="list-style-type: none"> - Belum maksimalnya fitur booking dan top up saldo - Belum adanya pemberitahuan yang cukup jelas dalam fitur <i>notifications</i> - Data tempat parkir yang digunakan berdasarkan sampel peneliti sendiri, - Aplikasi peneliti hanya ditujukan untuk kendaraan roda 4.
6.	<p>Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terbaik Pada <i>Mobile E-Parking</i> Berbasis Sistem Informasi Geografis</p>	<p>Diperlukan algoritma untuk mengetahui rute terpendek antara tempat awal dan tempat tujuan. Mengakses informasi yang cepat salah satunya adalah dengan menggunakan</p>	<p>Metode & Pengujian: <i>Prototyping</i> (Prototipe)</p> <p>Tools :</p> <ul style="list-style-type: none"> - Android Studio - Database MySQL - UML (Unified Modeling Language) 	<p>Sistem sudah menggunakan basis <i>mobile apps</i> yang berguna untuk memudahkan para pengendara mencari informasi tempat parkir. Sistem juga menyediakan navigasi rute terbaik dengan menerapkan Algoritma</p>	<ul style="list-style-type: none"> - Proses pengambilan data masih belum dijelaskan. - Tidak dijelaskan proses detail pengujian aplikasi.

	Cahyo Prianto, Muh. Kusnadi (2018)	sistem informasi geografis berbasis mobile dengan memanfaatkan fitur GPS untuk mendeteksi keberadaan pengguna	<ul style="list-style-type: none"> - Algoritma Dijkstra Proses : <ul style="list-style-type: none"> - Proses analisis kebutuhan - Perancangan sistem dengan membuat use case diagram, flowmap. - Pembuatan program. 	Dijkstra pada fitur Google Maps	<ul style="list-style-type: none"> - Perlu adanya fitur pemberitahuan untuk user
.7.	<p>Sistem Cerdas Reservasi dan Pemantauan Parkir pada Lokasi Kampus Berbasis Konsep Internet of Things</p> <p>Dody Ichwana, Ratna Aisuwarya, Styviandra Ardopa, Indah Purnama (2018)</p>	<p>Perlunya sistem online berupa aplikasi pencarian lahan parkir kosong di wilayah kampus yang dapat membantu pengendara menemukan lahan parkir kosong secara cepat, aman dan mudah. Dengan adanya sistem IoT, dapat dibentuk sebuah sistem untuk mengontrol ketersediaan tempat parkir yang masih tersedia dan mengetahui lokasinya</p>	Metode & Pengujian: <i>Prototyping</i> (Prototipe) Tools : <ul style="list-style-type: none"> - NFC - Sensor Ultrasonik - Arduino Mega - Android Studio - Database MySQL - XAMPP Proses : <ul style="list-style-type: none"> - Proses analisis kebutuhan data - Perancangan sistem dengan membuat use case diagram, diagram activity. - Pembuatan program. 	<p>Sistem ini berhasil membangun sistem reservasi (<i>booking</i>) dan pemantauan parkir pada lokasi kampus melalui aplikasi yang berjalan di sistem operasi Android dengan menerapkan konsep IoT. Sistem dapat mengetahui lokasi parkir yang kosong dan mampu melakukan reservasi (<i>booking</i>) untuk lokasi parkir kosong tersebut menggunakan <i>mobile apps</i> serta dapat melakukan pemantauan parkir di lokasi kampus.</p>	<ul style="list-style-type: none"> - Proses pengambilan data masih belum dijelaskan. - Tidak dijelaskan proses detail pengujian aplikasi. - Perlu adanya fitur pemberitahuan untuk user

Tabel 2. 3 Perbandingan Penelitian

No	Nama Peneliti/ Tahun	Judul Penelitian	Studi Kasus			Data Tempat Parkir		LBS	Metode RAD	API
			Tidak Ada	Pemerintahan	Perorangan/ Swasta	Data Real	Data Sampel			
1.	Durga Devi TJB, Dr. A.Subramani, Dr. Vijender Kumar Solanki (2017)	IOT Based Prototype for Parking Monitoring and Management System Commanded by Mobile App	√	-	-	-	√	-	-	√
2.	Muhammad Adam Dzulqarnain, Muhammad Luthfi Aziz, Muhammad Fauzi Rachman (2019)	Aplikasi Pencarian Parkir DKI Jakarta Berbasis Android Menggunakan RESTFUL API	-	-	√	-	√	-	-	√
3.	S Nurhayati, Egi Nur Ilmi (2019)	Sistem Aplikasi Pencarian Lokasi Parkir Terdekat Menggunakan Location Based Service Berbasis Android	-	-	√	-	√	√	-	√
4.	Jingbin Liu, Ruizhi Chen, Yuwei Chen, Ling Pei, Liang Chen (2020)	An Intelligent Indoor Location-Based Smartphone Parking Service	√					√		-
5.	Cornelius Venti, Yus Sholva, Rudy Dwi Nyoto (2020)	Sistem Manajemen Parkir Mobil On Street dan Off Street Berbasis Location Based	√	-	-	-	√	√	-	√

		Service (LBS) dan Google Maps API								
6.	Cahyo Prianto, Muh. Kusnadi (2018)	Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terbaik Pada Mobile E-Parking Berbasis Sistem Informasi Geografis	√	-	√	-	√	-	-	√
7.	Dody Ichwana, Ratna Aisuwarya, Styviandra Ardopa, Indah Purnama (2018)	Sistem Cerdas Reservasi dan Pemantauan Parkir pada Lokasi Kampus Berbasis Konsep Internet of Things	√	-	√	√	-	-	-	√
8.	Revo Yulianggara (2022)	Perancangan Aplikasi Informasi Tempat Parkir Terdekat Di Provinsi DKI Jakarta Menggunakan Location Based Service (LBS) Dan Google Maps API (Studi Kasus: Unit Perpustakaan Dinas Perhubungan DKI Jakarta)	-	√	-	√	-	√	√	√

Tabel 2. 4 Komparasi Penelitian dengan Dishub

No	Nama Peneliti/ Tahun	Judul Penelitian	Menu Fitur			Metode & Tools	Kelebihan	Kekurangan
			Fitur pencarian lokasi parkir berdasarkan wilayah di DKI Jakarta	Fitur pencarian lokasi parkir melalui peta digital (maps)	Fitur Tips&trik untuk informasi dan safety dalam berkendara			
1.	Unit Perparkiran Dinas Perhubungan Provinsi DKI Jakarta (2021)	Sistem Informasi Tempat Parkir (<i>e-parkir</i>) Di Wilayah Provinsi DKI Jakarta by Aspeparindo	-	√	-	Metode: <i>Scrum</i> Tools : <ul style="list-style-type: none"> - Flutter: Dart - PHP: Framework Laravel Proses : <ul style="list-style-type: none"> - Proses diawali dengan melakukan komunikasi dengan pihak tertentu 	<ul style="list-style-type: none"> - Sistem informasi tempat parkir yang dimiliki oleh Dishub dapat mencari informasi titik lokasi parkir terdekat melalui tampilan maps - Sistem tersebut menggunakan data tempat parkir yang terdaftar secara legalitas dalam Dinas Perhubungan 	<ul style="list-style-type: none"> - Sistem tersebut belum menggunakan <i>login/register</i> dalam penggunaannya - Sistem tersebut belum memiliki fitur pendukung lainnya yang dapat memudahkan user seperti fitur: <i>wilayah</i> dan <i>tips&trik</i> - Sistem tersebut belum dapat melakukan

					<ul style="list-style-type: none">- Pengumpulan data di lapangan dan analisis kebutuhan- Melakukan pembangunan sistem- Melakukan pengkodean sistem.- Pengujian sistem aplikasi.	<ul style="list-style-type: none">- Sistem tersebut menggunakan algoritma dijkstra	<p>pemantauan ketersediaan tempat parkir yang dicari secara <i>real time</i></p> <ul style="list-style-type: none">- Sistem tersebut belum menggunakan teknologi LBS dan <i>Google Maps API</i> dalam implementasinya	
2.	Revo Yuliangara (2022)	<p>Pengembangan Aplikasi Informasi Tempat Parkir Terdekat Di Provinsi DKI Jakarta Menggunakan <i>Location Based Service</i> (LBS) dan <i>Google Maps API</i> (Studi Kasus: Unit Perparkiran Dinas Perhubungan Provinsi DKI Jakarta)</p>	√	√	√	<p>Metode: <i>Rapid Application Development</i> (RAD)</p> <p>Tools :</p> <ul style="list-style-type: none">- Android Studio: Java- XAMPP: PHP <p>Proses :</p> <ul style="list-style-type: none">- Pengumpulan data di lapangan- menganalisis kebutuhan sistem.	<ul style="list-style-type: none">- Sistem informasi tempat parkir yang dikembangkan dapat mencari informasi titik lokasi parkir terdekat melalui tampilan maps- Sistem tersebut sudah menggunakan <i>login/register</i> untuk user dalam penggunaannya- Sistem tersebut memiliki fitur	<ul style="list-style-type: none">- Pencarian titik lokasi parkir terdekat belum dapat dicari berdasarkan jarak yang diinginkan oleh <i>user</i>, sebagai contoh pencarian titik lokasi parkir terdekat dalam jarak 200m dari posisi <i>user</i>.- Sistem tersebut belum dapat melakukan pemantauan

					<ul style="list-style-type: none"> - Perancangan design dan pembuatan program - Pengujian sistem aplikasi 	<p>pendukung lainnya, yaitu fitur: <i>wilayah, peta dan tips&trik</i></p> <ul style="list-style-type: none"> - Sistem tersebut menggunakan data tempat parkir yang terdaftar secara legalitas dalam Dinas Perhubungan - Sistem tersebut sudah menggunakan teknologi LBS dan Google Maps API 	<p>ketersediaan tempat parkir yang dicari secara <i>real time</i></p>
--	--	--	--	--	---	---	---

Berdasarkan penelitian sejenis yang telah dijelaskan dalam tabel 2.2 dan tabel 2.3. Kemudian tabel perbandingan sistem informasi tempat parkir terdekat (*e-parkir*) milik Dishub DKI Jakarta dengan sistem yang dikembangkan oleh penulis dalam tabel 2.4. Penelitian ini dapat memberikan kemudahan untuk user dalam mengatasi pencarian titik lokasi parkir terdekat. Beberapa keunggulan dalam penelitian ini adalah sebagai berikut:

1. Aplikasi ini berbasis *Mobile Apps* yang berarti sudah fleksibel.

2. Parameter yang digunakan dalam penelitian ini adalah jarak lokasi parkir dari posisi *user* dan informasi detail tempat parkir (kapasitas roda 2 ataupun roda empat).
3. Dapat digunakan dimana saja dengan menggunakan *smartphone* yang sudah dimiliki oleh developer Dinas Perhubungan dan seluruh pegawai pada UP Dinas Perhubungan Provinsi DKI Jakarta.
4. Aplikasi ini sudah mengimplementasikan teknologi *Location Based Service* (LBS) dan *web service* yaitu REST API.
5. Aplikasi ini dapat menampilkan informasi detail data tempat parkir di wilayah DKI Jakarta.
6. Aplikasi ini memberikan *tips&trik* berupa informasi ataupun *safety* dalam berkendara yang berguna untuk pengguna.
7. Penulis memilih metode pengujian *Black Box Testing* dan *User Acceptance Testing* (UAT) untuk mengetahui fungsi dari sistem yang telah dibangun sudah sesuai dengan rancangan. Kemudian mengetahui penilaian masing-masing responden terhadap fungsi dari pengujian awal sistem.

BAB III

METODOLOGI PENELITIAN

3.1 Metode Pengumpulan Data

Metode pengumpulan data digunakan untuk mencari dan mengumpulkan data yang terkait dengan penelitian seperti dasar teori, metodologi penulisan, metodologi proses, dan acuan penelitian sejenis. Dalam penelitian ini, metode pengumpulan data yang dilakukan adalah wawancara, studi pustaka dan studi literatur.

3.1.1 Metode Wawancara

Wawancara dilakukan kepada seorang narasumber yang berprofesi sebagai Manajer Perencanaan Perparkiran Dishub DKI Jakarta. Narasumber tersebut bernama Bapak Syaefudin Zuhri yang sudah bekerja selama 13 tahun di Unit Perparkiran Dinas Perhubungan DKI Jakarta. Wawancara dilakukan dengan mengajukan pertanyaan mengenai hal – hal yang berkaitan dengan sistem informasi tempat parkir (*e-parkir*) di UP Dishub DKI Jakarta saat ini. Wawancara juga bertujuan untuk mengetahui permasalahan yang dimiliki oleh Dinas Perhubungan dalam mengatasi perparkiran di jalan raya. Secara detail, untuk pertanyaan wawancara bisa dilihat pada lampiran.

3.1.2 Metode Pustaka/Studi Literatur

Tahapan pengumpulan data dengan cara studi pustaka, penulis mencari referensi-referensi yang berkaitan dengan penelitian. Pencarian referensi dilakukan secara online melalui internet, seperti portal berita dan *ebook (because pandemic of covid)*. Setelah mendapatkan referensi tersebut, kemudian dipilih berbagai informasi yang dibutuhkan dalam penelitian ini.

Studi literatur sejenis merupakan kegiatan mencari literatur yang mempunyai persamaan atau keterkaitan dengan penelitian yang sedang dilakukan. Literatur sejenis yang didapatkan berupa skripsi, jurnal, dan produk sejenis yang kemudian ditelaah dan dibuat perbandingan sehingga

penelitian ini dapat menjadi pelengkap atau penyempurnaan dari penelitian-penelitian yang sudah dilakukan sebelumnya.

3.1.3 Metode Observasi

Observasi menurut Adler, menyebutkan bahwa observasi merupakan salah satu dasar fundamental dari semua metode pengumpulan data dalam penelitian kualitatif, khususnya menyangkut ilmu – ilmu sosial dan perilaku manusia. Menurut Werner dan Schoepfle (1987 : 257) observasi juga dipahami sebagai “andalan perusahaan etnografi”, yaitu proses pengamatan sistematis dari aktivitas manusia dan pengaturan fisik dimana kegiatan tersebut berlangsung secara terus menerus dari lokus aktivitas bersifat alami untuk menghasilkan fakta (Rahmat, 2021).

Metode ini dilakukan penulis dengan cara pengamatan langsung ke lapangan mengenai sistem parkir pada UP Dinas Perhubungan DKI Jakarta. Penulis melakukan observasi dalam kurun waktu kurang lebih 1 (satu) bulan, dimana hal yang diamati adalah pengembangan sistem informasi tempat parkir yang sedang dilakukan untuk *e-parkir* milik Dinas Perhubungan. Dari pengamatan secara langsung, penulis menemukan beberapa hal yang masih menjadi kekurangan mendasar. Lalu, penulis mengajukan untuk melakukan pengembangan sistem dari yang sudah ada untuk kemudian diterapkan dengan beberapa keunggulan fitur. Untuk mengetahui kekurangan tersebut maka dilakukan pengamatan pada:

Tanggal : 5 Juli 2021 – 28 Juli 2021

Tempat : Bagian Perencanaan Unit Perparkiran Dinas Perhubungan
DKI Jakarta

Alamat : Jl. Perintis Kemerdekaan No.1, RT.1/RW.15, Kayu Putih, Kec.
Pulo Gadung, Kota Jakarta Timur, Daerah Khusus Ibukota
Jakarta 13210

3.2 Metode Pengembangan Sistem

Dalam pengembangan sistem ini, penulis menggunakan metode pengembangan sistem dengan metode *Rapid Application Development* (RAD). Alasan penulis menggunakan metode ini karena cocok dalam pengembangan sebuah sistem dalam waktu pengerjaan yang singkat.

Terdapat 3 tahapan pengembangan RAD (Kendall, 2017), yaitu:

1. *Requirements Planning* (Perencanaan Persyaratan).
2. *Design Workshop* (Workshop Desain RAD).
3. *Implementation* (Penerapan Pengujian).

3.2.1 Rapid Application Development (RAD)

Pada penelitian ini penulis menggunakan metode pengembangan sistem *Rapid Application Development* (RAD) untuk mengembangkan aplikasi informasi tempat parkir di Provinsi DKI Jakarta.

RAD adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus pengembangan sistem konvensional antara perancangan dan penerapan suatu sistem informasi (Kendall, 2017).

Menurut Presmann yang dikutip oleh Sanjaya (2017), dalam bukunya yang berjudul “Rekayasa Perangkat Lunak: Pendekatan Praktisi”, mengatakan bahwa RAD adalah proses model perangkat lunak inkremental yang menekankan siklus pengembangan yang singkat. Model RAD adalah sebuah adaptasi dari model *waterfall*, dimana perkembangan pesat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen. Jika tiap-tiap kebutuhan dan batasan ruang lingkup proyek telah diketahui dengan baik, maka proses RAD memungkinkan tim pengembang untuk menciptakan sebuah “sistem berfungsi penuh” dalam jangka waktu yang sangat singkat.

3.2.1.1 Requirements Planning (Perencanaan Persyaratan)

Pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan tersebut (Kendall, 2017).

Pada tahapan ini, penulis melakukan perencanaan dengan pihak Dinas Perhubungan terkait kebutuhan aplikasi atau sistem yang dapat membantu para pengendara di lapangan. Kemudian mendiskusikannya dengan pakar atau ahli dibidang penelitian penulis (diskusi dilakukan dengan dosen pembimbing skripsi) untuk menganalisis masalah lebih lanjut dan menyimpulkan sebuah solusi yang telah teruji serta manfaat penelitian kedepannya.

3.2.1.2 Design Workshop (Workshop Desain RAD)

Merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Penganalisis dan *programmer* bekerja membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna. *Workshop* desain ini dapat dilakukan selama beberapa hari tergantung dari ukuran aplikasi yang akan dikembangkan (Kendall, 2017).

Pada tahapan ini, penulis merancang apa yang bisa digambarkan sebagai desain lalu menunjukkan tampilan visual desain dan alur kerja pengguna. Kemudian pihak Dinas Perhubungan merespon dan menanggapi prototipe yang ada. Selanjutnya penulis memperbaiki modul-modul yang dirancang berdasarkan respon tersebut.

3.2.1.3 Implementation (Penerapan Pengujian)

Penganalisis bekerja dengan para pengguna secara intens selama *workshop* dan merancang aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek ini disetujui dan sistem dibangun, sistem baru atau bagian dari sistem di ujicoba dan kemudian diperkenalkan kepada perusahaan.

Pada tahapan ini, sistem yang baru dibangun oleh penulis diuji dan diperkenalkan kepada pihak Dinas Perhubungan dengan monitoring secara berkala. Ketika diperlukan pengembangan sistem, maka yang dikembangkan hanya yang dibutuhkan oleh sistem tersebut.

3.3 Perangkat Penelitian

Dalam penelitian ini, penulis menggunakan perangkat keras (*Hardware*) dan perangkat lunak (*Software*) sebagai berikut:

1. Perangkat Keras (*Hardware*):

Tabel 3. 1 Perangkat Penelitian (Hardware)

No.	Perangkat Keras	Spesifikasi
1.	Device	HP EliteBook 8470p
2.	Processor	2,6 GHz 4-Core Intel Core i5
3.	VGA	Intel UHD Graphics 4000
4.	RAM	4 GB

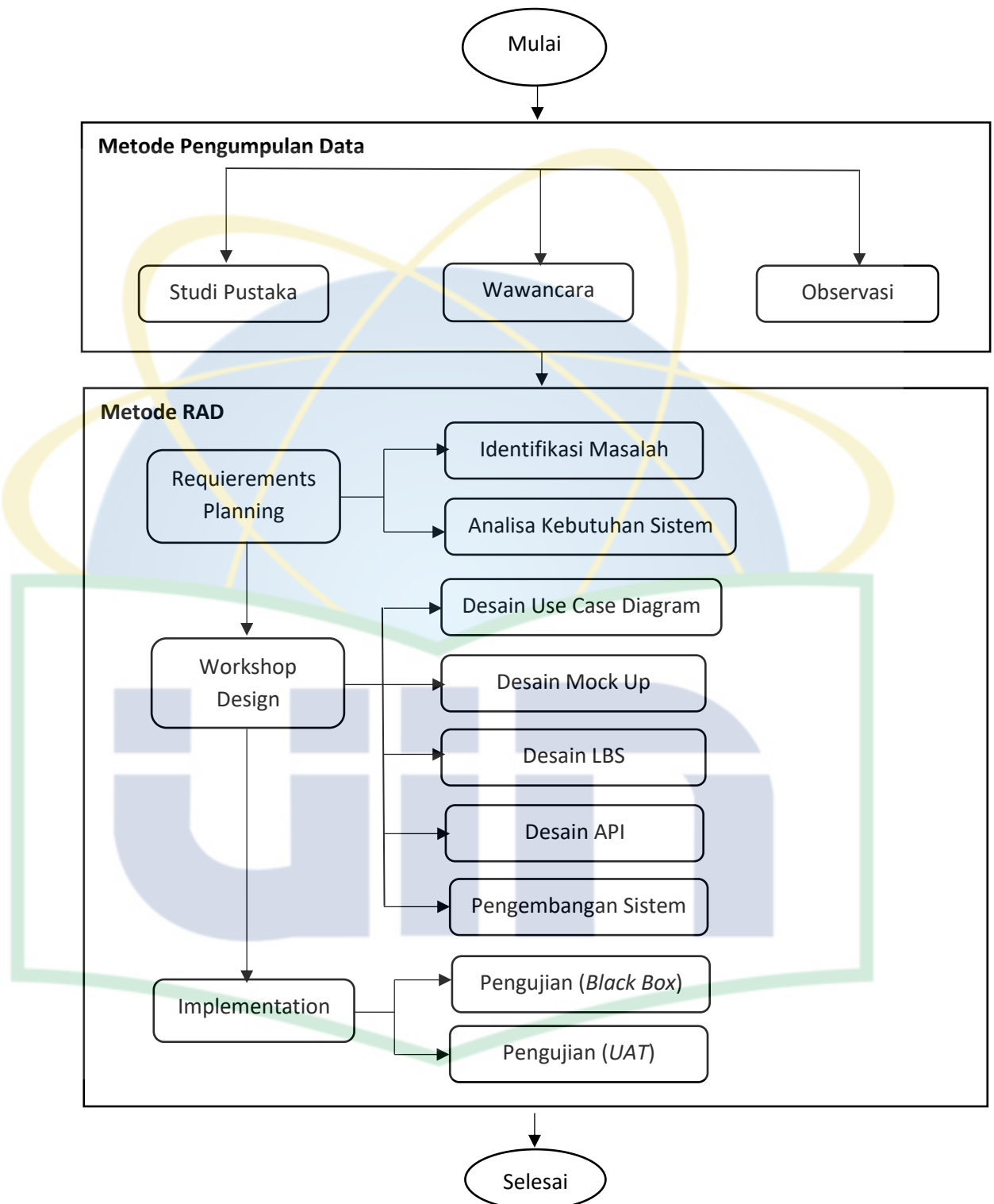
2. Perangkat Lunak (*Software*):

Tabel 3. 2 Perangkat Penelitian (Software)

No.	Perangkat Lunak	Spesifikasi
1.	Sistem Operasi	Windows 10
2.	Bahasa Pemrograman	PHP, Java
3.	Tools	a. Android Studio b. Sublime Text, Notepad++ c. XAMPP d. Postman

3.4 Kerangka Berpikir

Berdasarkan rumusan masalah yang dibuat pada pembahasan sebelumnya, tahapan kerangka berfikir dibuat untuk memperjelas apa saja yang menjadi metode penelitian.



Gambar 3. 1 Kerangka Berpikir

3.5 Metode Pengujian

Pada tahap ini penulis melakukan pengujian sistem menggunakan metode *black box testing* untuk mengetahui masing-masing setiap fungsi sistem yang di bangun telah sesuai dengan rancangan yang dibuat. Kemudian melakukan pengujian sistem aplikasi menggunakan metode *user acceptance testing* (UAT) terhadap Developer Dinas Perhubungan dan 5 responden (Manajer UP Dinas Perhubungan dan 4 kepala *staff*) di bidangnya masing-masing yang berkaitan dengan perparkiran di DKI Jakarta, serta 5 responden (*user*) aplikasi untuk mengetahui penilaian dan respon terhadap sistem aplikasi yang telah dibuat.

Teknik pemilihan responden ini dilakukan dengan melakukan wawancara terhadap Developer Dishub, Manajer UP Dinas Perhubungan dan 4 kepala *staff* yang berada dalam *layer* 2 dan *layer* 3 pada struktur jabatan di Unit Perparkiran Dinas Perhubungan DKI Jakarta. Lalu, melakukan wawancara terhadap 5 responden (*user*) sebagai pengendara kendaraan roda 2 ataupun roda 4.



BAB IV

IMPLEMENTASI

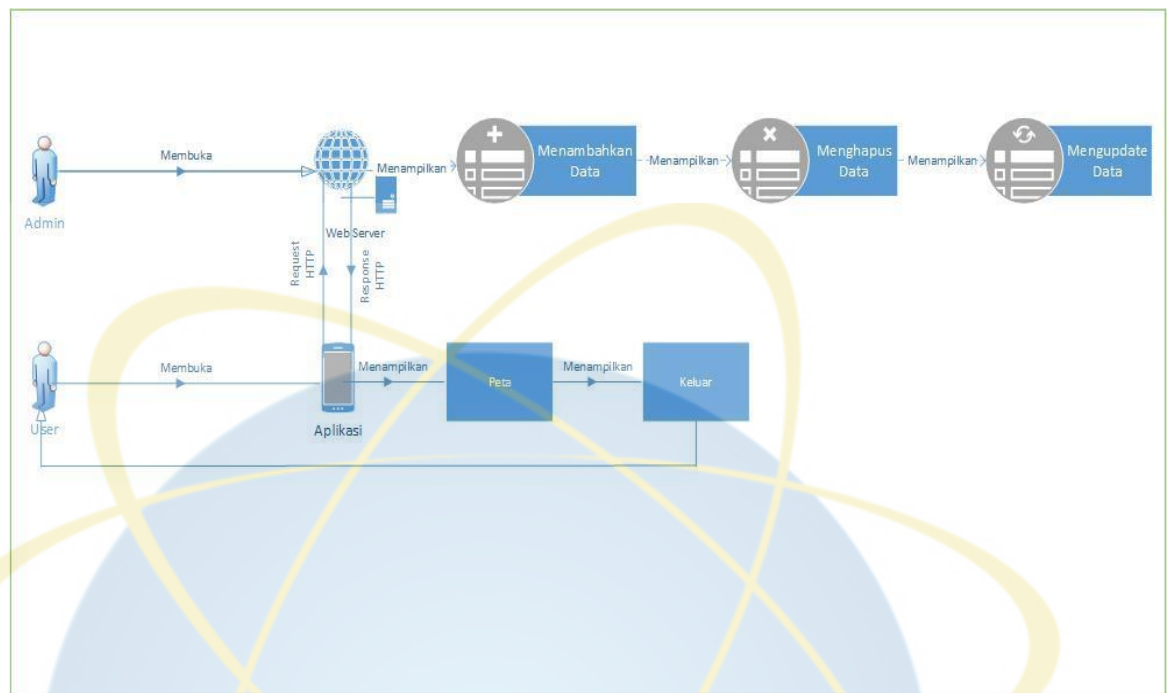
Penulis menggunakan metode pengembangan sistem dengan RAD (*Research And Development*) agar dapat membantu semua pihak yang berkepentingan untuk memahami jalannya pengembangan sistem sehingga dapat sesuai dengan perencanaan hingga ke tahapan implementasi serta mempersingkat waktu yang diperlukan dalam siklus pengembangan antara perancangan dan penerapan suatu sistem informasi. Terdapat tiga proses dalam penggunaan RAD ini, yaitu *Requirements Planning* (melakukan perencanaan dengan pihak Dinas Perhubungan terkait kebutuhan aplikasi atau sistem yang dapat membantu para pengendara, *Design Workshop* (merancang apa yang bisa digambarkan sebagai desain lalu menunjukkan tampilan visual desain dan alur kerja pengguna), dan *Implementation* (sistem yang baru dibangun diuji dan diperkenalkan kepada pihak Dinas Perhubungan dengan monitoring secara berkala).

4.1 Requirements Planning (Perencanaan Persyaratan)

Pada tahapan ini, penulis melakukan perencanaan dengan pihak Dinas Perhubungan terkait kebutuhan aplikasi atau sistem yang dapat membantu para *user*. Kemudian menentukan fitur yang nantinya akan dibangun sesuai dengan prioritasnya, pada proses ini akan dijelaskan secara rinci persyaratan atau fitur yang diperoleh.

4.1.1 Analisis Sistem Berjalan

Sebelum mengembangkan aplikasi ini, proses pencarian informasi tempat parkir terdekat masih terbilang konvensional, berikut gambaran dari sistem yang telah berjalan berdasarkan wawancara dan observasi.



Gambar 4. 1 Sistem Berjalan

Pada sistem berjalan ini, *user* (pengendara) memerlukan proses yang kurang efisien, yaitu user mengakses aplikasi pencarian lokasi parkir terdekat milik Dinas Perhubungan dan mendapatkan beberapa list tempat parkir. Aplikasi yang tersedia belum sepenuhnya dapat menampilkan data detail mengenai tempat parkir tersebut, baik dari kapasitas maupun jarak menuju lokasi. *Web server* yang dikelola oleh admin masih minim dalam menginput data tempat parkir sehingga terkadang user tidak menemukan pilihan lain dalam mencari tempat parkir terdekat yang diperlukan. Proses tersebut menyebabkan kurang efisiennya proses pencarian lokasi parkir terdekat dikarenakan masih minimnya data tempat parkir beserta detailnya yang tersedia.

4.1.2 Kelebihan dan Kekurangan Sistem Berjalan

Setelah melakukan analisis terhadap sistem yang sedang berjalan tersebut, ditemukan kelebihan dan kekurangan yang akan dijadikan sebagai bahan evaluasi sistem didalam perancangan sistem tersebut, yaitu:

1. Kelebihan
 - a. Sistem informasi tempat parkir yang dimiliki oleh Dishub dapat mencari informasi titik lokasi parkir terdekat melalui tampilan

maps.

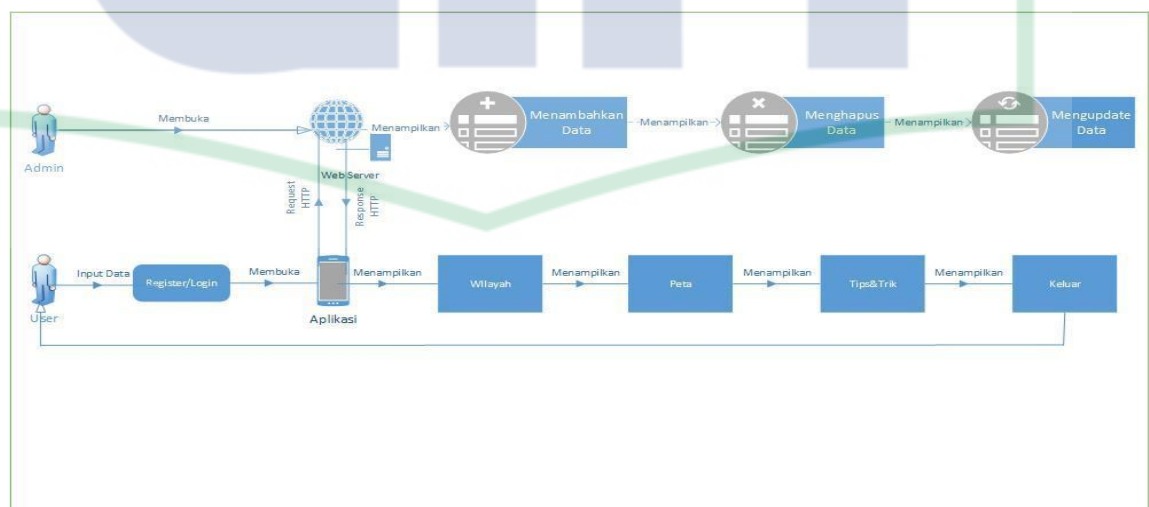
- b. Sistem tersebut menggunakan data tempat parkir yang terdaftar secara legalitas dalam Dinas Perhubungan.
- c. Sistem tersebut menggunakan algoritma dijkstra.

2. Kekurangan

- a. Sistem tersebut belum menggunakan *login/register* dalam penggunaannya.
- b. Sistem tersebut belum memiliki fitur pendukung lainnya yang dapat memudahkan user seperti fitur: *wilayah* dan *tips&trik*.
- c. Sistem tersebut belum dapat melakukan pemantauan ketersediaan tempat parkir yang dicari secara *real time*.
- d. Sistem tersebut belum menggunakan teknologi LBS dan Google Maps API dalam implementasinya.
- e. *Web server* yang dikelola oleh admin masih minim dalam menginput data tempat parkir, hal tersebut mengakibatkan user tidak menemukan pilihan lain dalam mencari tempat parkir terdekat yang diperlukan.

4.1.3 Analisis Sistem Usulan

Berikut ini merupakan skema dari sistem usulan pencarian tempat parkir terdekat yang digambarkan penulis sebagai berikut:



Gambar 4. 2 Sistem Usulan

Setelah adanya sistem usulan ini, proses yang dijalankan yaitu *user* dapat melihat beberapa list titik lokasi parkir beserta data detailnya. Lalu, *user* menemukan tempat parkir terdekat berdasarkan wilayah nya masing-masing (DKI Jakarta) dan dapat melihat beberapa *tips&trik* dari fitur yang tersedia.

Dalam proses perancangan sistem usulan ini akan dijelaskan dan diuraikan tentang proses pencarian tempat parkir terdekat dari admin pada aplikasi sampai proses user mengakses aplikasi tersebut.

1. Untuk mengakses kepada user administrator, user admin melakukan *input username* dan *password* pada aplikasi untuk melakukan *update* dan input data parkir yang ada. Apabila *username* dan *password* tidak cocok maka sistem akan menampilkan pesan bahwa *username* atau *password* salah dan admin tersebut tidak bisa mengakses aplikasi sistem tersebut.
2. Apabila *username* dan *password* benar maka admin langsung masuk ke dalam dashboard admin yang berisi menu input data tempat parkir, data tempat parkir berdasarkan wilayah, menu *tips&trik* untuk pengendara.
3. Admin dapat melakukan input data tempat parkir dengan memasukkan *content* yang ada mengenai tempat parkir dan menyimpannya.
4. Setelah proses input data tempat parkir selesai, maka data-data tersebut akan disimpan pada *database*. Kemudian data-data yang terdapat pada sistem tersebut dapat ditampilkan pada aplikasi untuk dilihat oleh user dan admin pada menu *Peta* dan *Wilayah*.
5. Admin dapat melihat data tempat parkir yang sudah terdaftar pada *database* sistem di menu *Peta*, pada menu ini admin dapat mencari secara langsung mengenai data tempat parkir yang ingin di *edit* atau *delete* (apabila terjadi kesalahan).
6. Pada opsi *edit*, admin dapat melakukan perubahan data mengenai tempat parkir yang terdaftar pada sistem dengan cara mengubah *content* yang ingin diubah admin (apabila terjadi kesalahan).

7. Pada opsi *delete*, admin dapat menghapus data tempat parkir yang terdaftar pada sistem (apabila terjadi kesalahan).
8. Pada menu *Wilayah*, admin dapat melihat kategori tempat parkir berdasarkan wilayah dan jumlahnya sesuai yang dikategorikan. Terdapat lima kategori wilayah dalam menu ini, yaitu Jakarta Pusat (Data tempat parkir yang berada di wilayah Jakarta Pusat), Jakarta Utara (Data tempat parkir yang berada di wilayah Jakarta Utara), Jakarta Barat (Data tempat parkir yang berada di wilayah Jakarta Barat), Jakarta Selatan (Data tempat parkir yang berada di wilayah Jakarta Selatan), dan Jakarta Timur (Data tempat parkir yang berada di wilayah Jakarta Timur).
9. Ketika user ingin mengakses aplikasi, terdapat tiga menu yang dapat diakses oleh user. Yaitu menu *Peta*, menu *Wilayah*, dan menu *Tips&trik*.
10. Ketika user mengakses menu *Peta*, user dapat mencari lokasi tempat parkir terdekat yang sesuai dengan posisi atau lokasi yang dituju.
11. Pada menu *Wilayah*, user dapat melihat kategori tempat parkir berdasarkan wilayah dan jumlahnya sesuai yang dikategorikan. Terdapat lima kategori wilayah dalam menu ini, yaitu Jakarta Pusat (Data tempat parkir yang berada di wilayah Jakarta Pusat), Jakarta Utara (Data tempat parkir yang berada di wilayah Jakarta Utara), Jakarta Barat (Data tempat parkir yang berada di wilayah Jakarta Barat), Jakarta Selatan (Data tempat parkir yang berada di wilayah Jakarta Selatan), dan Jakarta Timur (Data tempat parkir yang berada di wilayah Jakarta Timur).
12. Pada menu *Tips&Trik*, user dapat melihat beberapa list mengenai informasi dan *safety* dalam berkendara.

Untuk menggambarkan skema kerja dari implementasi LBS dengan Google Maps API, anggaplah teknologi LBS akan mencari informasi mengenai lokasi restoran yang berada di sekitar posisi pengguna:

1. Anggaplah sekarang fungsi pencarian telah diaktifkan, posisi pengguna sebenarnya dari perangkat mobile diperoleh dari *Positioning Service*. Hal ini dapat dilakukan baik oleh perangkat menggunakan GPS sendiri atau layanan posisi jaringan yang berasal dari *provider (Cell Tower)*. Setelah itu perangkat *mobile* pengguna mengirimkan permintaan informasi, yang berisi tujuan untuk mencari dan mengirimkan posisi melalui jaringan komunikasi ke *gateway* telekomunikasi.
2. *Gateway* memiliki tugas untuk bertukar pesan di antara jaringan komunikasi selular dan internet. Oleh karena itu dia mengetahui alamat web dari beberapa aplikasi server dan rute permintaan ke spesifik server tertentu. *Gateway* akan menyimpan juga informasi tentang perangkat mobile yang telah meminta informasi.
3. Aplikasi server membaca permintaan dan mengaktifkan layanan yang terkait.
4. Kemudian, *service* menganalisis lagi pesan dan memutuskan mana informasi tambahan selain kriteria pencarian (restoran + padang) dan posisi pengguna diperlukan untuk menjawab permintaan pengguna. Dalam kasus ini *service* akan menemukan bahwa pengguna membutuhkan informasi tentang restoran dari database *yellow pages* pada wilayah tertentu dan kemudian *service* tersebut akan meminta penyedia data untuk memberikan data tersebut.
5. Selanjutnya *service* akan menemukan informasi tentang jalan, jarak dan cara yang diperlukan untuk memeriksa apakah restoran dapat dicapai.
6. Sekarang semua informasi *service* akan melakukan *buffer* spasial dan *query routing* untuk mendapatkan beberapa restoran terdekat. Setelah menghitung daftar restoran terdekat, hasil dikirim kembali ke pengguna melalui internet, *gateway* dan jaringan *mobile*.
7. Kemudian, informasi mengenai restoran akan disampaikan kepada pengguna dalam bentuk peta digital (maps).

4.1.4 Analisis Kebutuhan Sistem

Pada tahapan analisis kebutuhan sistem ini, penulis melakukan wawancara dan observasi langsung ke bagian perencanaan Unit Perparkiran Dinas Perhubungan DKI Jakarta untuk mengetahui masalah dan kebutuhan yang diperlukan oleh sistem. Dari hasil wawancara dan observasi tersebut, didapatkan perbedaan yang menjadi dasar penulis meneliti mengenai pencarian informasi tempat parkir terdekat ini. Perbedaan pencarian informasi tempat parkir terdekat yang akan dibuat ini dengan yang sudah berjalan yaitu sudah cukup fleksibel dan mudah, sehingga *user* dapat mengetahui informasi tempat parkir terdekat dengan menggunakan beberapa fitur dan menu yang tersedia.

4.1.4.1 Fitur untuk Admin

Fitur untuk admin dibuat dengan bahasa pengguna secara umum yang difungsikan agar mudah dimengerti oleh orang bisnis dan orang teknis (penulis). Salah satu hal yang paling penting dengan adanya fitur admin ini ialah persyaratan untuk mencapai harapan dari pengembangan *software*.

Tabel 4. 1 Fitur Admin

No.	Fitur Admin
1.	<i>Login</i> ke dalam sistem dengan memasukkan <i>username</i> dan <i>password</i> yang sesuai.
2.	Dapat melakukan input data tempat parkir berdasarkan wilayah setelah melakukan login ke sistem.
3.	Dapat melihat keseluruhan data tempat parkir yang telah di input web admin.
4.	Dapat melakukan input notifikasi untuk tips & trik berkendara di dalam web admin.
5.	Dapat melakukan edit data tempat parkir di dalam web admin.
6.	Dapat melakukan delete data tempat parkir di dalam web admin.
7.	Dapat melakukan pencarian data tempat parkir dengan fitur <i>pencarian</i> di dalam web admin.

8.	Dapat melakukan input data <i>username</i> dan <i>password</i> baru di dalam web admin.
----	---

4.1.4.2 Fitur untuk User

Fitur untuk user dibuat dengan bahasa pengguna secara umum yang difungsikan agar mudah dimengerti oleh orang bisnis dan orang teknis (penulis). Salah satu hal yang paling penting dengan adanya fitur user ini ialah persyaratan untuk mencapai harapan dari pengembangan *software*.

Tabel 4. 2 Fitur User

No.	Fitur User
1.	<i>Login</i> ke dalam sistem dengan memasukkan <i>username</i> dan <i>password</i> yang sesuai.
2.	Dapat mencari informasi mengenai data detail tempat parkir.
3.	Dapat melakukan pencarian data tempat parkir dengan melihat dalam menu <i>peta</i> atau berdasarkan wilayahnya.
4.	Dapat melihat informasi dalam berkendara dalam menu <i>tips&trik</i> .
5.	Dapat melakukan input data <i>username</i> dan <i>password</i> baru di dalam aplikasi.

4.1.5 Perancangan Antar Muka Sistem

Penulis merancang *user interface* yang berfungsi sebagai gambaran aplikasi yang akan dibuat.

4.1.6 Koding Program Aplikasi

Pada tahap ini penulis menuliskan kode program untuk membuat aplikasi informasi tempat parkir terdekat berbasis android. Dalam proses pengkodean, penulis melakukan pengkodean program menggunakan bahasa pemrograman Java dan PHP, membangun *interface* menggunakan android studio, dan melakukan uji API dengan menggunakan Postman.

4.1.7 Testing (Pengujian)

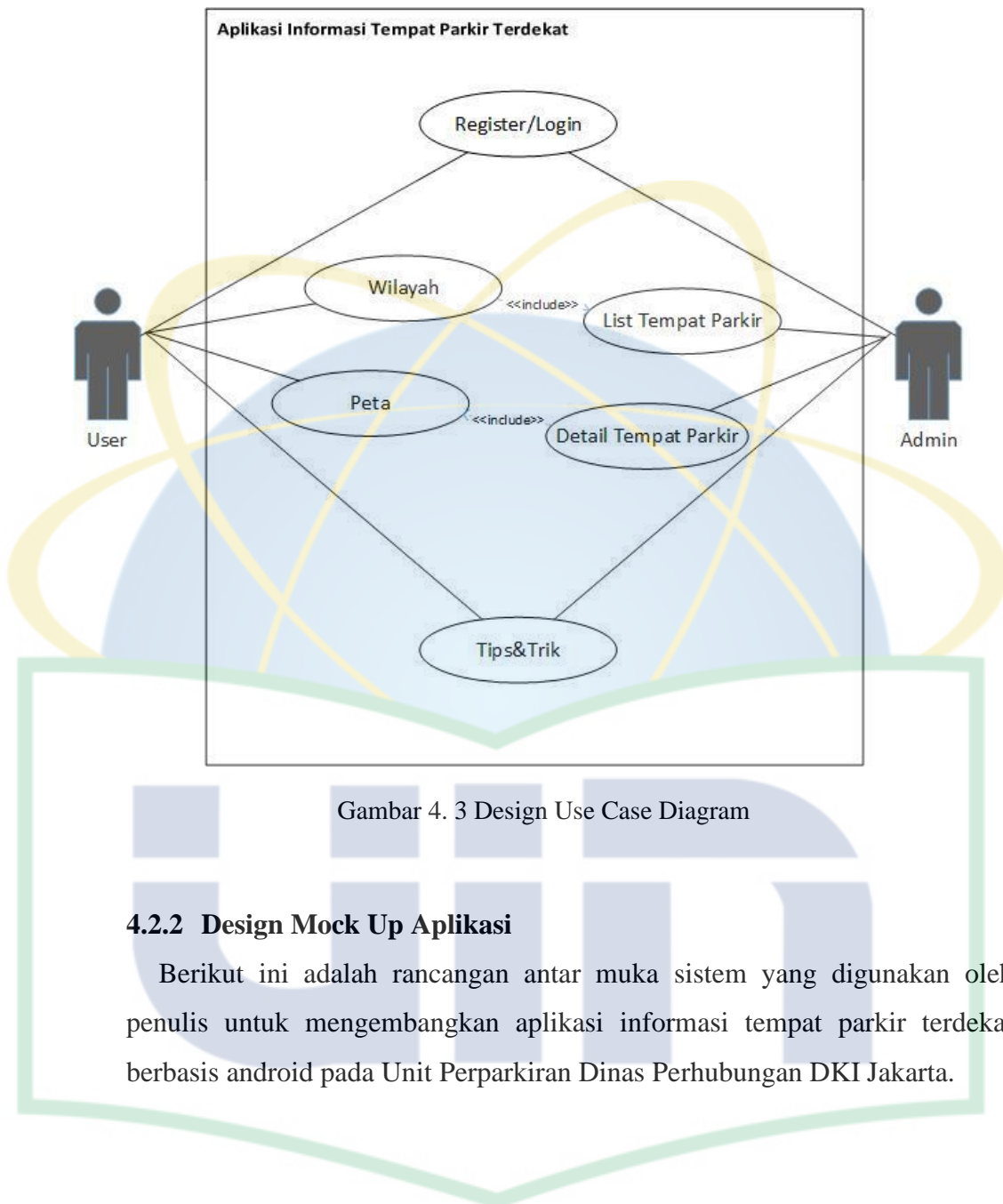
Pada tahap ini penulis melakukan pengujian menggunakan metode *black box testing* untuk mengetahui apakah setiap fungsi sistem yang di bangun telah sesuai dengan rancangan yang dibuat, dan melakukan pengujian *user acceptance testing* untuk mengetahui penilaian dan respon stakeholder terhadap sistem yang dibuat.

4.2 Design Workshop (Workshop Desain RAD)

Pada tahapan ini, penulis merancang apa yang bisa digambarkan sebagai desain lalu menunjukkan tampilan visual desain dan alur kerja pengguna. Kemudian penulis melakukan penulisan kode dengan mengimplementasikan bahasa pemrograman Java dan PHP, kemudian Android Studio dan *sublime text* sebagai *tools* yang berfungsi untuk mengimplementasikan desain yang telah dibuat.

4.2.1 Design Use Case Diagram Aplikasi

Berikut ini adalah rancangan use case diagram yang digambarkan oleh penulis untuk mengembangkan aplikasi informasi tempat parkir terdekat berbasis android pada Unit Perparkiran Dinas Perhubungan DKI Jakarta.



Gambar 4. 3 Design Use Case Diagram

4.2.2 Design Mock Up Aplikasi

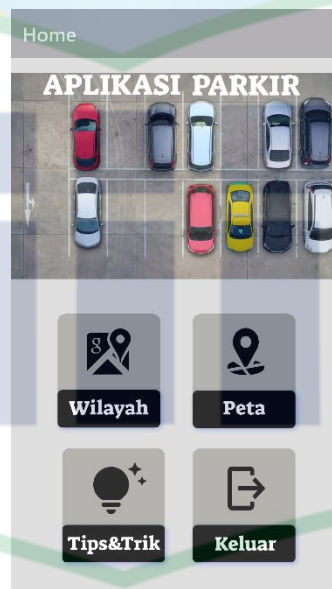
Berikut ini adalah rancangan antar muka sistem yang digunakan oleh penulis untuk mengembangkan aplikasi informasi tempat parkir terdekat berbasis android pada Unit Perparkiran Dinas Perhubungan DKI Jakarta.

4.2.2.1 Rancangan Splash Scree



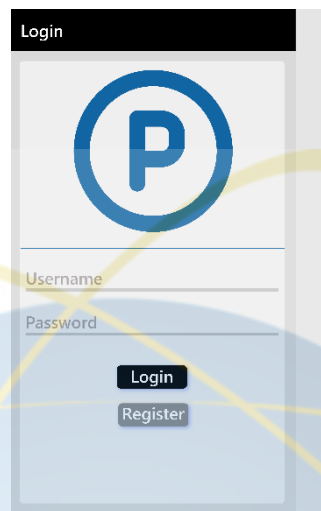
Gambar 4. 4 Rancangan Splash Screen

4.2.2.2 Rancangan Dashboard User



Gambar 4. 5 Rancangan Dashboard User

4.2.2.3 Rancangan Login User



Login

P

Username

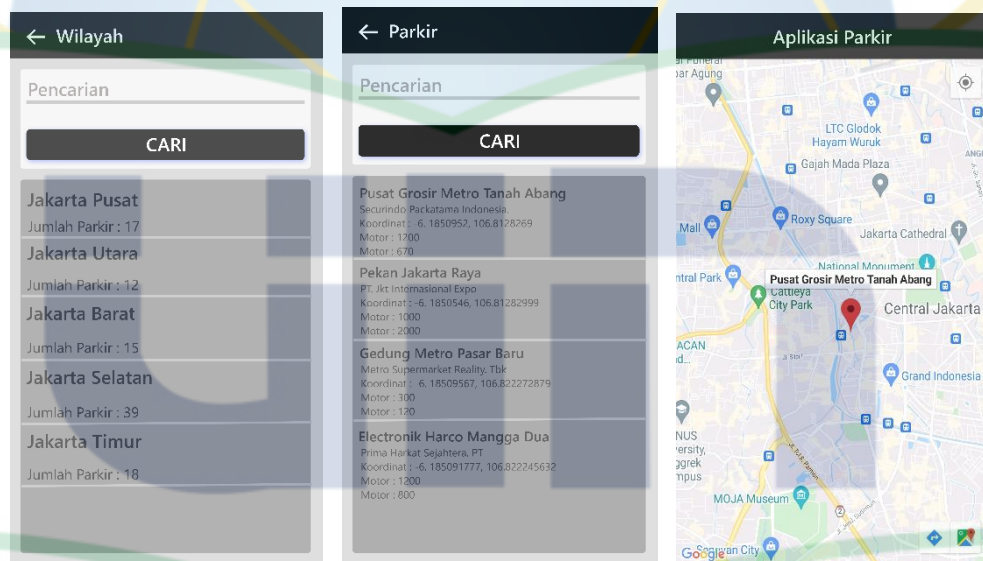
Password

Login

Register

Gambar 4. 6 Rancangan Login User

4.2.2.4 Rancangan Menu Wilayah



← Wilayah

Pencarian

CARI

Jakarta Pusat	Jumlah Parkir : 17
Jakarta Utara	Jumlah Parkir : 12
Jakarta Barat	Jumlah Parkir : 15
Jakarta Selatan	Jumlah Parkir : 39
Jakarta Timur	Jumlah Parkir : 18

← Parkir

Pencarian

CARI

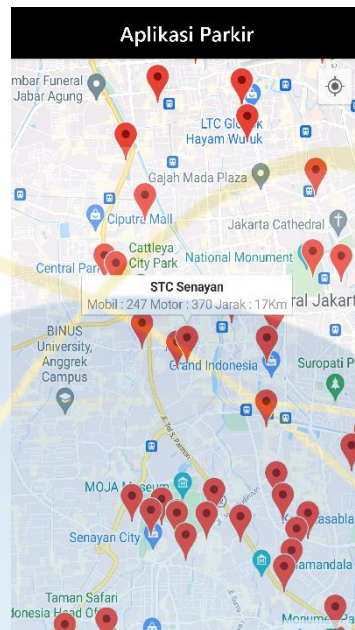
Pusat Grosir Metro Tanah Abang	Securindo Parkatama Indonesia	Koordinat : - 6.1850952, 106.8178269	Motor : 1200
Pekan Jakarta Raya	PT. Jkt Internasional Expo	Koordinat : - 6.1850546, 106.8128299	Motor : 1000
Gedung Metro Pasar Baru	Metro Supermarket Reality. Tbk	Koordinat : - 6.18509567, 106.822272879	Motor : 300
Elektronik Harco Mangga Dua	Prima Harkat Sejahtera. PT	Koordinat : - 6.185091777, 106.822245632	Motor : 1200

Aplikasi Parkir

Map showing various locations in Jakarta, including LTC Glodok, Hayam Wuruk, Gajah Mada Plaza, Roxy Square, Jakarta Cathedral, National Monument, Central Jakarta, Grand Indonesia, NUS, MOJA Museum, and Soekarno City.

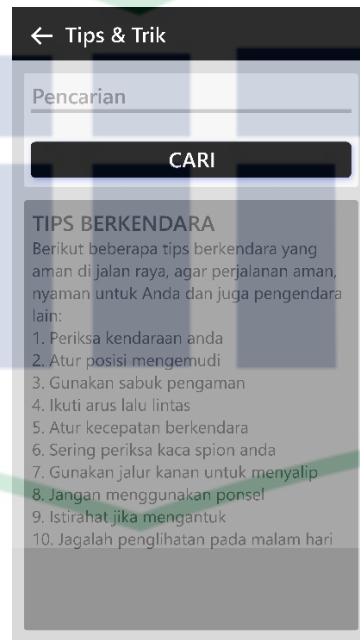
Gambar 4. 7 Rancangan Menu Wilayah

4.2.2.5 Rancangan Menu Peta



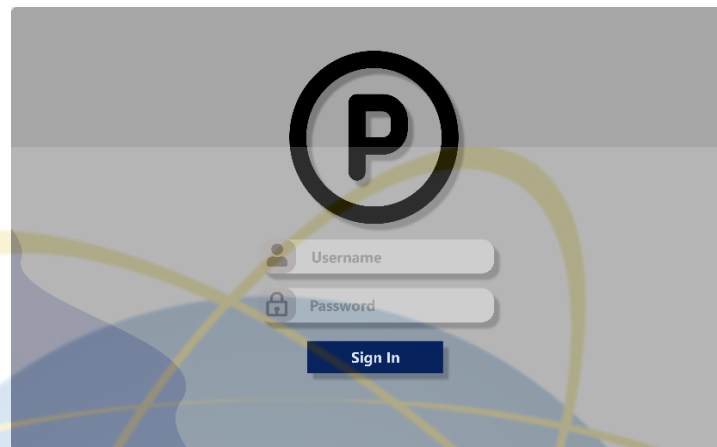
Gambar 4. 8 Rancangan Menu Peta

4.2.2.6 Rancangan Menu Tips&Trik



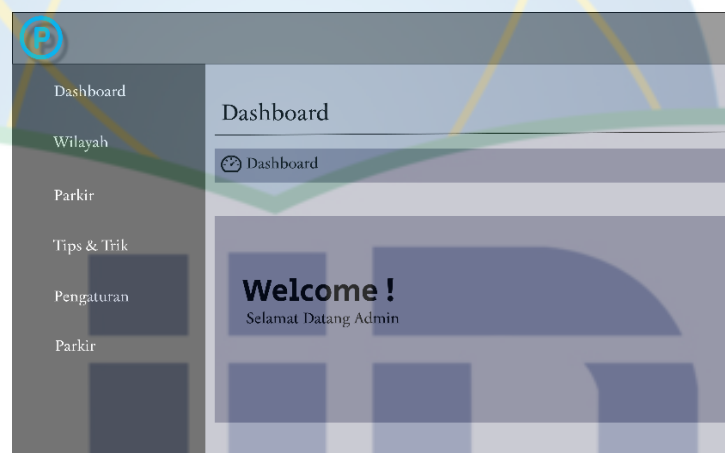
Gambar 4. 9 Rancangan Menu Tips&Trik

4.2.2.7 Rancangan Login Admin



Gambar 4. 10 Rancangan Login Admin

4.2.2.8 Rancangan Dashboard Admin



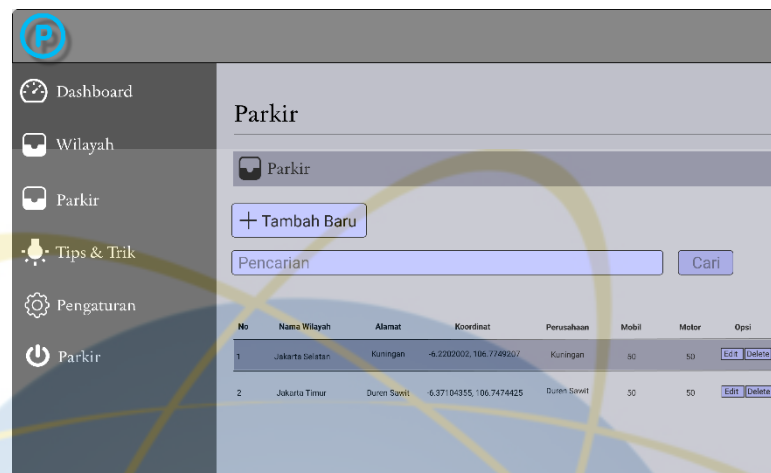
Gambar 4. 11 Rancangan Dashboard Admin

4.2.2.9 Rancangan Menu Wilayah Admin



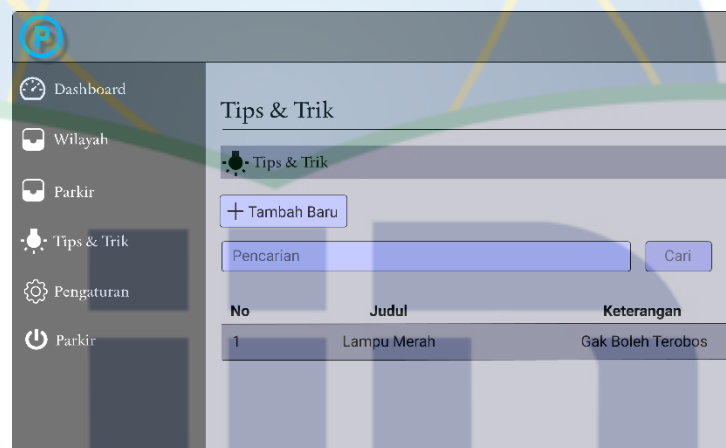
Gambar 4. 12 Rancangan Menu Wilayah Admin

4.2.2.10 Rancangan Menu Parkir Admin



Gambar 4. 13 Rancangan Menu Parkir Admin

4.2.2.11 Rancangan Menu Tips&Trik Admin



Gambar 4. 14 Rancangan Menu Tips&Trik Admin

4.2.2.12 Rancangan Menu Pengaturan Admin

Gambar 4. 15 Rancangan Menu Pengaturan Admin

4.2.3 Tahapan Build Location Based Service (LBS)

Pada tahap ini, penulis mengembangkan sistem aplikasi informasi tempat parkir terdekat berbasis android dengan menggunakan teknik *Location Based Service* (LBS). LBS digunakan untuk menerapkan pencarian tempat parkir terdekat dengan memanfaatkan *latitude* dan *longitude* sebagai penyedia informasi utama. *Latitude* dan *longitude* yang telah di input pada web admin akan dihubungkan melalui *API Google Maps* dengan android sebagai perantara. Berikut tahapan untuk membangun LBS seperti dibawah ini:

4.2.3.1 Perancangan LBS Pada Web Server (Admin)

Pada tahapan ini, penulis memasukkan *latitude* dan *longitude* dari setiap data tempat parkir pada *web admin* (halaman parkir). *Latitude* dan *longitude* yang di input nantinya akan menghasilkan data tempat parkir berupa *list-list* poin pada maps android, kemudian dihubungkan melalui *API google maps* sebagai perantara. Berikut merupakan *code php* pada halaman parkir.


```
<div class="form-group">

    Lat : <input type="text" class="form-control" name="lat"
required oninvalid="this.setCustomValidity('Kolom Lat Tidak
Boleh Kosong')" oninput="setCustomValidity('')">

</div>

<div class="form-group">

    Lang : <input type="text" class="form-control"
name="lng" required oninvalid="this.setCustomValidity('Kolom
Lang Tidak Boleh Kosong')" oninput="setCustomValidity('')">

</div>
```

```
<?php

include "../config/koneksi.php";

$query = mysqli_query($con,"SELECT * FROM parkir");

$json = array();

while($row = mysqli_fetch_assoc($query)){

    $json[] = $row;    }

echo json_encode($json);

mysqli_close($con);

?>
```

4.2.3.2 Perancangan LBS Pada Android (User)

Pada tahap ini, penulis membangun *latitude* dan *longitude* yang telah di input pada *web admin* untuk dihubungkan melalui *API google maps* yang kemudian menghasilkan data tempat parkir berupa *list-list* poin pada maps android yang dapat membantu *user* mencari tempat parkir terdekat (halaman peta). Berikut merupakan *code* pada halaman peta.

```
@Override
public class Peta extends AppCompatActivity implements
    OnMapReadyCallback,
        LocationListener, GoogleApiClient.ConnectionCallbacks,
        GoogleApiClient.OnConnectionFailedListener {

    GoogleMap mMap;
    GoogleMap gMap;
    MarkerOptions markerOptions = new MarkerOptions();
    CameraPosition cameraPosition;
    LatLng center, latLng;
    String title, mobil, motor;
    Marker mCurrLocationMarker;
    GoogleApiClient mGoogleApiClient;
    Location mLastLocation;
    LocationRequest mLocationRequest;
    String koordinat, koorlat, koorlong;
    Double latt, lngg;

    public void onMapReady(GoogleMap googleMap) {
        gMap = googleMap;
        checkLocationPermission();
        lat = getIntent().getStringExtra("lat");

        lng = getIntent().getStringExtra("lng");
        nama = getIntent().getStringExtra("alamat");
        // Mengarahkan ke monas
        double dLat = Double.parseDouble(lat);
        double dLng = Double.parseDouble(lng);
        Value = 10;
        center = new LatLng(dLat, dLng);
        cameraPosition = new
        CameraPosition.Builder().target(center).zoom(10).build();
        gMap.addMarker(new
        MarkerOptions().position(center).title(nama));

        gMap.animateCamera(CameraUpdateFactory.newLatLngZoom(center, Value));
        /* if (lat.equals("123") && lng.equals("123")) {
        }
        else{
            //double dLat = Double.parseDouble(lat);
            //double dLng = Double.parseDouble(lng);
            //LatLng parkir = new LatLng(dLat, dLng);
            //cameraPosition = new
            CameraPosition.Builder().target(center).zoom(10).build();
        }
    }
}
```

```

//googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(park
ir, 15));
}*/

if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
    // TODO: Consider calling
    //     ActivityCompat#requestPermissions
    // here to request the missing permissions, and then
    // overriding
    //     public void onRequestPermissionsResult(int
    // requestCode, String[] permissions,
    //                                           int[]
    // grantResults)
    // to handle the case where the user grants the
    // permission. See the documentation
    // for ActivityCompat#requestPermissions for more
    // details.
    return;
}
if (android.os.Build.VERSION.SDK_INT >=
Build.VERSION_CODES.M) {
    if (ContextCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION)
        == PackageManager.PERMISSION_GRANTED) {
        buildGoogleApiClient();
        gMap.setMyLocationEnabled(true);
    }
}
else {
    buildGoogleApiClient();
    gMap.setMyLocationEnabled(true);
}
gMap.getUiSettings().setMyLocationButtonEnabled(true);
}

protected synchronized void buildGoogleApiClient() {
    mGoogleApiClient = new
    GoogleApiClient.Builder(this).addConnectionCallbacks(this).addOn
    ConnectionFailedListener(this).addApi(LocationServices.API).buil
    d();
    mGoogleApiClient.connect();
}

```

```

@SuppressLint("MissingPermission")
@Override
public void onConnected(@Nullable Bundle bundle) {
    LocationRequest mLocationRequest = new LocationRequest();
    mLocationRequest.setInterval(1000);
    mLocationRequest.setFastestInterval(1000);

    mLocationRequest.setPriority(LocationRequest.PRIORITY_BALANCED_P
OWER_ACCURACY);
    if (ContextCompat.checkSelfPermission(this,
        android.Manifest.permission.ACCESS_FINE_LOCATION)
        == PackageManager.PERMISSION_GRANTED) {

        LocationServices.FusedLocationApi.requestLocationUpdates(mGoogle
ApiClient, mLocationRequest, this);
    }
}
@Override
public void onConnectionSuspended(int i) {
}
@Override
public void onConnectionFailed(@NonNull ConnectionResult
connectionResult) {
}
@Override
public void onLocationChanged(Location location) {
    mLastLocation = location;
    if (mCurrLocationMarker != null) {
        mCurrLocationMarker.remove();
    }
    LatLng latLng = new LatLng(location.getLatitude(),
location.getLongitude());
    //addMarker(mLastLocation.getLatitude(),
mLastLocation.getLongitude());
    CameraPosition cameraPosition = new
CameraPosition.Builder().target(new LatLng(latLng.latitude,
latLng.longitude)).zoom(16).build();

    //gMap.animateCamera(CameraUpdateFactory.newCameraPosition(camer
aPosition));

    //menghentikan pembaruan lokasi
    if (mGoogleApiClient != null) {

        LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleA
piclient, this);
    }
}

```

Kode diatas dituliskan untuk mendapatkan setiap *latitude* dan *longitude* dari data tempat parkir yang di input pada web admin. Untuk penulisan

kode *API Google Maps* yang menghubungkan antara *web admin* dan *android* adalah sebagai berikut.

```
public static final int MY_PERMISSIONS_REQUEST_LOCATION = 99;
public boolean checkLocationPermission(){
    if (ContextCompat.checkSelfPermission(this,
        android.Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED) {
        if
        (ActivityCompat.shouldShowRequestPermissionRationale(this,
        android.Manifest.permission.ACCESS_FINE_LOCATION)) {
            ActivityCompat.requestPermissions(this,
                new
                String[]{android.Manifest.permission.ACCESS_FINE_LOCATION},
                MY_PERMISSIONS_REQUEST_LOCATION);
        } else {
            ActivityCompat.requestPermissions(this,
                new
                String[]{android.Manifest.permission.ACCESS_FINE_LOCATION},
                MY_PERMISSIONS_REQUEST_LOCATION);
        }
        return false;
    } else {
        return true;
    }
}
@SuppressWarnings("MissingPermission")
@Override
public void onRequestPermissionsResult(int requestCode, String
permissions[], int[] grantResults) {
    switch (requestCode) {
        case MY_PERMISSIONS_REQUEST_LOCATION: {
            if (grantResults.length > 0
                && grantResults[0] ==
                PackageManager.PERMISSION_GRANTED) {
                // Izin diberikan.
                if (ContextCompat.checkSelfPermission(this,
                android.Manifest.permission.ACCESS_FINE_LOCATION)
                == PackageManager.PERMISSION_GRANTED) {
                    if (mGoogleApiClient == null) {
                        buildGoogleApiClient();
                    }
                    gMap.setMyLocationEnabled(true);
                }
            } else {
                // Izin ditolak.
                Toast.makeText(this, "permission denied",
                Toast.LENGTH_LONG).show();
            }
            return;
        }
    }
}
```

```

private void addMarker(final double loclat, final double
loclong) {
    lat = getIntent().getStringExtra("lat");

    lng = getIntent().getStringExtra("lng");
    nama = getIntent().getStringExtra("alamat");
    double dLat = Double.parseDouble(lat);
    double dLng = Double.parseDouble(lng);
    LatLng latlng = new LatLng(dLat, dLng);
    float[] results = new float[1];
    Location.distanceBetween(loclat, loclong,
        Double.parseDouble(lat),
        Double.parseDouble(lng), results);
    int meter = (int) results[0];
    int distance = meter / 1000;
    markerOptions.position(latlng);
    markerOptions.title(nama);
    // markerOptions.snippet(keterangan);
    markerOptions.snippet("Jarak : " + distance + "Km");

    gMap.addMarker(markerOptions);

    gMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latlng,10))
    ;

    gMap.setOnInfoWindowClickListener(new
    GoogleMap.OnInfoWindowClickListener() {
        @Override
        public void onInfoWindowClick(Marker marker) {
            // Toast.makeText(getApplicationContext(),
            marker.getTitle(), Toast.LENGTH_SHORT).show();
            Toast.makeText(getApplicationContext(),
            marker.getSnippet(), Toast.LENGTH_SHORT).show();
        }
    });
}
}

```

4.2.4 Tahapan Build Database

Pada tahap ini merupakan proses pembuatan *database* yang digunakan untuk menampung seluruh data tempat parkir. Penulis membuat *database* dengan nama *skripsi_revo*, pada database ini dibuat tabel *parkir*, *tips*, *user* dan *wilayah*.

4.2.4.1 Tabel user

Pada tabel *user*, penulis membuat 7 kolom tabel yang terdiri *id*, *username*, *password*, *nama*, *email*, dan *telp*. Pada tabel ini, *id* dijadikan sebagai *primary key*, *email* sebagai *text*, sedangkan *username*, *password*, *nama*, *telp* dan *level* diberikan dengan tipe data *varchar*.

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	id_user	int(11)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Kunci Utama
<input type="checkbox"/>	2	username	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama
<input type="checkbox"/>	3	password	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama
<input type="checkbox"/>	4	nama	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama
<input type="checkbox"/>	5	email	text	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama
<input type="checkbox"/>	6	telp	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama
<input type="checkbox"/>	7	level	varchar(11)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama

Gambar 4. 16 Tabel User

4.2.4.2 Tabel Wilayah

Pada tabel *wilayah*, penulis membuat 2 kolom tabel yang terdiri *id_wilayah* dan *nama_wilayah*. Pada tabel ini, *id_wilayah* dijadikan sebagai *primary key* dan *nama_wilayah* diberikan tipe data *varchar*.

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	id_wilayah	int(11)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Kunci Utama
<input type="checkbox"/>	2	nama_wilayah	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama

Gambar 4. 17 Tabel Wilayah

4.2.4.3 Tabel Parkir

Pada tabel *parkir*, penulis membuat 8 kolom tabel yang terdiri *id_parkir* (*primary key*), *id_wilayah* (*int*), *alamat* (*text*), *lat* (*double*), *lng* (*double*), *perusahaan* (*varchar*), *mobil* (*int*) dan *motor* (*int*).

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	id_parkir	int(11)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Kunci Utama
<input type="checkbox"/>	2	id_wilayah	int(11)		Tidak	Tidak ada			Ubah Hapus Kunci Utama
<input type="checkbox"/>	3	alamat	text	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama
<input type="checkbox"/>	4	lat	double		Tidak	Tidak ada			Ubah Hapus Kunci Utama
<input type="checkbox"/>	5	lng	double		Tidak	Tidak ada			Ubah Hapus Kunci Utama
<input type="checkbox"/>	6	perusahaan	varchar(30)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama
<input type="checkbox"/>	7	mobil	int(11)		Tidak	Tidak ada			Ubah Hapus Kunci Utama
<input type="checkbox"/>	8	motor	int(11)		Tidak	Tidak ada			Ubah Hapus Kunci Utama

Gambar 4. 18 Tabel Parkir

4.2.4.4 Tabel Tips

Pada tabel *user*, penulis membuat 6 kolom tabel yang terdiri *id*, *username*, *password*, *nama*, *email*, dan *telp*. Pada tabel ini, *id* dijadikan sebagai *primary key*, *email* sebagai *text*, sedangkan *username*, *password*, *nama* dan *telp* diberikan dengan tipe data *varchar*.

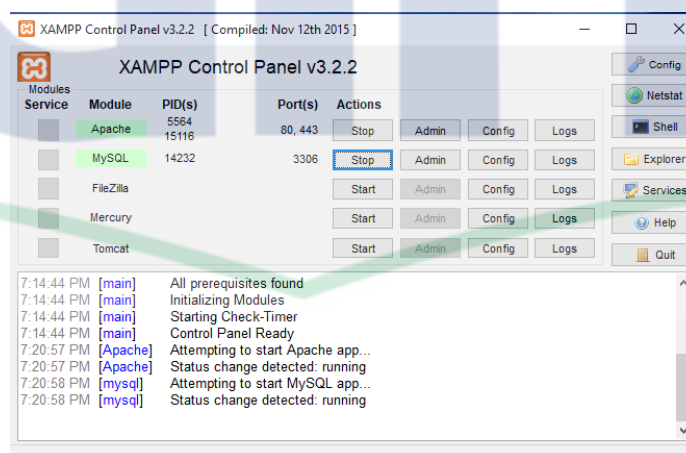
#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Komentar Ekstra	Tindakan
<input type="checkbox"/> 1	id_tip	int(11)		Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Kunci Utama
<input type="checkbox"/> 2	judul	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada		Ubah Hapus Kunci Utama
<input type="checkbox"/> 3	keterangan	text	utf8mb4_general_ci	Tidak	Tidak ada		Ubah Hapus Kunci Utama

Gambar 4. 19 Tabel Tips

4.2.5 Tahapan Build REST API Dengan PHP

Pada tahap ini, penulis juga menggunakan *API* yang berfungsi untuk mengembangkan sistem aplikasi informasi tempat parkir terdekat. *API* menjadi penghubung antara *latitude* dan *longitude* yang telah di input pada *web admin* dengan *android*. Penulis menggunakan *PHP* sebagai media penulisan *code* dalam membangun sistem. Berikut tahapan untuk membangun *API* seperti dibawah ini:

1. Dalam membangun *API*, tahap pertama yang diperlukan adalah untuk mengaktifkan *MySQL* dan *Apache*, *XAMPP Control Panel* yang digunakan penulis adalah versi 3.2.2



Gambar 4. 20 Mengaktifkan Apache dan MySQL

2. Penulis membuat database baru dengan menggunakan nama *skripsi_revo* sebagai databasenya dan mengatur pada *.env* pada project

php dengan mengganti DB_DATABASE menjadi nama database (*skripsi_revo*) dan DB_USERNAME menjadi root. Proses tersebut merupakan untuk mengkonfigurasi project yang dibuat.

```

8 DB_CONNECTION=mysql
9 DB_HOST=127.0.0.1
10 DB_PORT=3306
11 DB_DATABASE=coba_api
12 DB_USERNAME=root
13 DB_PASSWORD=

```

Gambar 4. 21 Konfigurasi Project

3. Membuat *Route* pada file *api.php*

Untuk membuat membuat API, *php* menyediakan routes yang dikhususkan untuk membuat API, yaitu *api.php*. Pada file ini penulis menambahkan route yang berfungsi untuk memasukkan perintah pada *php*, yaitu sebagai berikut:

```

Route::get('arsip', 'ArsipController@index');
Route::post('arsip', 'ArsipController@create');
Route::put('/arsip/edit/{id}', 'ArsipController@update');
Route::delete('/arsip/delete/{id}', 'ArsipController@delete');
Route::post('cari', 'ArsipController@cari');

```

Gambar 4. 22 Route Perintah pada PHP

Route tersebut memiliki fungsinya masing – masing yang nantinya akan mengarah ke file *ArsipController.php* yang berisikan perintah dari masing – masing fungsi tersebut.

Pada file ini juga dibuat route yang berfungsi untuk mendapatkan *get* dan *post* dari masing – masing fungsi pada API, yaitu sebagai berikut:

```

Route::get('upbji', 'upbjiController@index');
Route::get('grafik', 'grafikController@index'); //Untuk get grafik
Route::get('grafikadafoto', 'grafikController@adafoto'); //Untuk get grafik ada foto
Route::get('grafikutpusat', 'grafikController@utpusat'); //Untuk get grafik UT Pusat
Route::get('grafikutupbji', 'grafikController@utupbji'); //Untuk get grafik UT UPBJJ
Route::get('grafikijazahbelumadafoto', 'grafikController@belumadafoto'); //Untuk get grafik ijazah belum ada foto
Route::get('grafikijazahsudahadafoto', 'grafikController@sudahadafoto'); //Untuk get grafik ijazah sudah ada foto
Route::post('generatepass', 'ArsipController@pass'); //generate pw pake laravel
Route::post('login', 'ArsipController@login');

```

Gambar 4. 23 Route Perintah *get* dan *post*

Pada kode pengkodean route tersebut, akan mengarah kepada file wilayahController, parkirController, dan tips&trikController.

4. Wilayah Controller

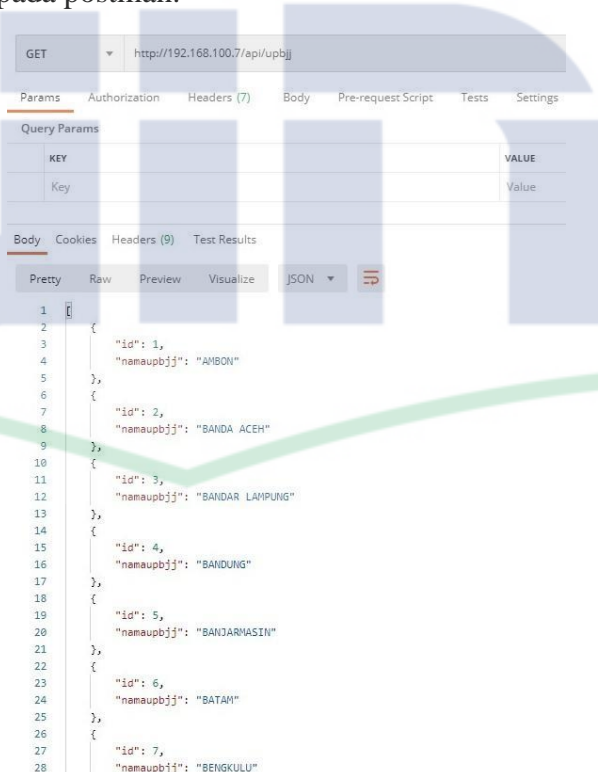
Pada file upbjjController.php, penulis memasukkan kode untuk menampilkan index mengenai Wilayah. Kode tersebut meliputi sebagai berikut:

```
//index (menampilkan
semua) public function
index(){

    return wilayah::all();

}
```

Kode diatas memiliki fungsi untuk mengambil data upbjj dan ditampilkan dihalaman index. Berikut merupakan contoh run API yang dijalankan pada postman.



Gambar 4. 24 Contoh Running pada API

Pada contoh menjalankan sebuah request diatas, penulis menjalankan perintah get untuk mendapatkan data mengenai nama wilayah. Penulis memasukkan alamat http yang sesuai dengan localhost pada perangkat yang terhubung. Setelah penulis memasukkan alamat http yang diinginkan, server akan melakukan response dengan menampilkan data berupa id dan namawilayahj yang telah tersimpan pada database.

5. Parkir Controller

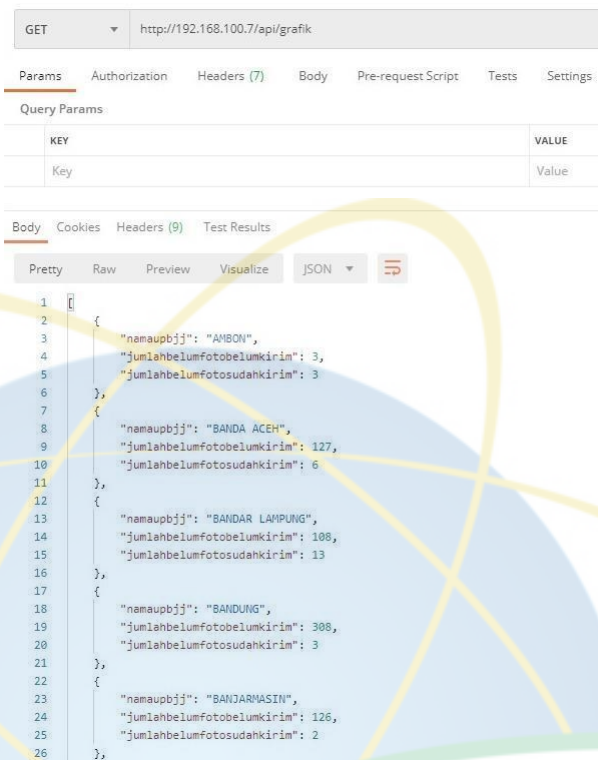
Pada file parkirController.php, penulis memasukkan beberapa perintah untuk menampilkan data detail tempat parkir, yaitu sebagai berikut.

```
//untuk memanggil query tabel wilayah (db)
$parkir = \DB::table('wilayah') -> get();
```

Kode diatas dituliskan untuk memanggil query tabel parkir pada database. Query yang diambil untuk meresponse ini didapat dari tabel database wilayah. Setelah query berhasil dipanggil, penulis melakukan looping untuk mencari data dari tabel parkir yang diberikan nama parkir yang telah dilakukan looping dari variabel wilayah. Berikut merupakan penulisan kode untuk melakukan proses looping.

```
foreach($parkir as $val)
```

Berikut adalah contoh request dan response dari server ketika dilakukan request untuk memanggil data detail tempat parkir dari seluruh wilayah.



Gambar 4. 25 Request dan Response Parkir API pada Postman

6. Tips&Trik Controller

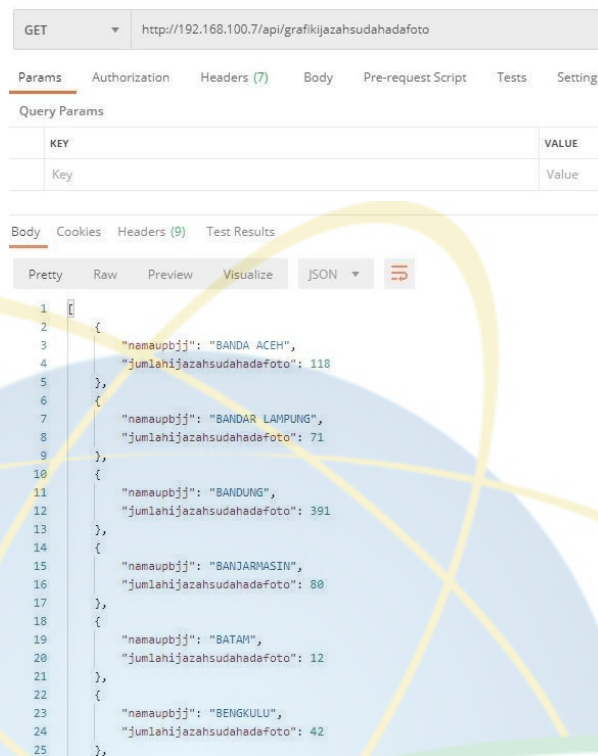
Pada file `tips&trikController.php`, penulis memasukkan beberapa perintah untuk menampilkan tips&trik dan informasi dalam kendaraan, yaitu sebagai berikut.

```
//untuk memanggil query tabel tips&trik (db)
$tips&trik = \DB::table('tipd&trik') -> get();
```

database. Query yang diambil untuk meresponse ini didapat dari tabel database api. Setelah query berhasil dipanggil, penulis melakukan looping untuk mencari data dari tabel tips&trik yang diberikan nama judul informasi yang telah dilakukan looping dari variabel tips&trik. Berikut merupakan penulisan kode untuk melakukan proses looping.

```
foreach($tips&trik as $val)
```

Berikut adalah contoh request dan response dari server ketika dilakukan request untuk memanggil informasi kendaraan dari seluruh list data.



Gambar 4. 26 Request dan Response Parkir API pada Postman

Setelah semua sudah selesai dibuat, selanjutnya penulis membuat kode untuk login melalui api untuk diaplikasikan pada android studio.

Untuk penulisan kode mengenai Login Controller, penulis melakukan request ke tabel useradmin pada database. Sistem akan melakukan request untuk mengecek username dan password dari akun admin. Berikut ini penulisan kode fungsinya.

```

//Login Controller

public function login(Request $request){

    $username = $request->username; (mengambil variabel
    $pass = $request->password;
    $status = "Username tidak ada";

    $user = DB::table('useradmin') -> where(["username" =>
    $username]) -> get();

    if($user != NULL){
        foreach($user as
        $val){
  
```

API memiliki fungsi sebagai penyedia sumber daya berupa data atau informasi yang bersumber dari database. Alamat API disediakan oleh server yang telah dibuat dan fungsinya akan ditunjukkan pada tabel berikut.

Tabel 4. 3 Deskripsi Api

URL Address	Metode	Deskripsi
api/wilayah	Get	Untuk menampilkan wilayah tempat parkir secara keseluruhan.
api/wilayah/edit/{id}	Put	Edit data wilayah parkir dengan pilihan id.
api/wilayah/delete/{id}	Delete	Menghapus data wilayah parkir dengan pilihan id.
api/cari	Post	Mencari data wilayah tempat parkir.
api/parkir	Get	Menampilkan data detail tempat parkir beserta list poin tempat parkir
api/parkir/edit/{id}	Put	Edit list data detail tempat parkir dengan pilihan id.
api/parkir/delete/{id}	Delete	Menghapus list data detail tempat parkir dengan pilihan id.
api/cari	Post	Mencari list data detail tempat parkir.
api/tips&trik	Get	Menampilkan list data tips&trik dan informasi dalam berkendara.

api/tips&trik/edit/{id}	Put	Edit list data tips&trik dan informasi dalam berkendara dengan pilihan id.
api/tips&trik/delete/{id}	Delete	Menghapus list data tips&trik dan informasi dalam berkendara dengan pilihan id.
api/cari	Post	Mencari list tips&trik dan informasi dalam berkendara

4.2.6 Tahapan Build Interface Menggunakan Android Studio

4.2.6.1 Perancangan Splash Screen

Berikut merupakan kode untuk membangun rancangan *splash screen* aplikasi informasi tempat parkir terdekat berbasis android. Penulis menggunakan beberapa komponen tambahan yang diperlukan seperti *image assets* pada android studio.

```
import android.content.Intent;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
```

Pada halaman ini, beberapa package digunakan untuk membangun sebuah halaman ini. Proses import diperlukan untuk memasukkan package kehalaman ini, Package ini merupakan sebuah library berisikan fungsi – fungsi widget dan material untuk desain. Untuk proses splash screen atau layar pembuka untuk sistem ini, penulis memberikan waktu selama 3 detik pada variabel duration.

```
public class SplashscreenActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splashscreen);
        final Handler handler = new Handler();

        Intent(getApplicationContext(), Login.class));
        finish();}
        }, 3000L); //3000L=3 detik
    }
}
```

Penggunaan image asset dibutuhkan untuk mengambil gambar pada assets yang sebelumnya telah diinput. Berikut gambar yang diambil pada fungsi asset diatas.



Gambar 4. 27 Logo Dishub.png

(Sumber: Penulis)

4.2.6.2 Perancangan Halaman Login

Berikut merupakan kode untuk membangun halaman login user pada aplikasi informasi tempat parkir terdekat berbasis android. Pada fitur ini hanya diperuntukkan user pada aplikasi yang bertujuan untuk mencari list tempat parkir.

Pada halaman ini, penulis menggunakan public class dalam membentuk fitur yang dibutuhkan seperti textfield untuk username, password dan button. Berikut ini merupakan kode tersebut.

```
public class Login extends AppCompatActivity {

    ProgressDialog pDialog;
    Button btn_login, btn_register;
    EditText txt_username, txt_password;
    Intent intent;

    int success;
    ConnectivityManager conMgr;

    private String url = Config.host + "android/login.php";

    private static final String TAG =
Login.class.getSimpleName();

    private static final String TAG_SUCCESS = "success";
    private static final String TAG_MESSAGE = "message";

    public final static String TAG_USERNAME = "username";
    public final static String TAG_ID = "id_user";
```

```
String tag_json_obj = "json_obj_req";

SharedPreferences sharedPreferences;
Boolean session = false;
String id, username, komoditas;
int level;
public static final String my_shared_preferences =
"my_shared_preferences";
public static final String session_status =
"session_status";
```

Kode diatas dituliskan untuk membuat sebuah textfield untuk memasukkan username user. Untuk penulisan kode membuat sebuah textfield password adalah sebagai berikut.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);

    conMgr = (ConnectivityManager)
    getSystemService(Context.CONNECTIVITY_SERVICE);
    {
        if (conMgr.getActiveNetworkInfo() != null
            && conMgr.getActiveNetworkInfo().isAvailable()
            && conMgr.getActiveNetworkInfo().isConnected())
        {
        } else {
            Toast.makeText(getApplicationContext(), "No Internet
            Connection",
                Toast.LENGTH_LONG).show();
        }
    }
}
```

```
btn_login = (Button) findViewById(R.id.btn_login);
btn_register = (Button) findViewById(R.id.btn_register);
txt_username = (EditText) findViewById(R.id.txt_username);
txt_password = (EditText) findViewById(R.id.txt_password);

// Cek session Login jika TRUE maka langsung buka
MainActivity
sharedpreferences =
getSharedPreferences(my_shared_preferences,
Context.MODE_PRIVATE);
session = sharedPreferences.getBoolean(session_status,
false);
id = sharedPreferences.getString(TAG_ID, null);
username = sharedPreferences.getString(TAG_USERNAME, null);
```

```

if (session) {
    Intent intent = new Intent(Login.this, Home.class);
    intent.putExtra(TAG_ID, id);
    intent.putExtra(TAG_USERNAME, username);
    finish();
    startActivity(intent);
}

```

Berikutnya adalah penulisan kode untuk button login, pada kode ini dimasukkan fungsi untuk masuk kedalam dashboard home. Berikut adalah penulisan kodenya.

```

btn_login.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        String username = txt_username.getText().toString();
        String password = txt_password.getText().toString();

        // mengecek kolom yang kosong
        if (username.trim().length() > 0 &&
            password.trim().length() > 0) {
            if (conMgr.getActiveNetworkInfo() != null
                &&
                conMgr.getActiveNetworkInfo().isAvailable()
                &&
                conMgr.getActiveNetworkInfo().isConnected()) {
                checkLogin(username, password);
            } else {
                Toast.makeText(getApplicationContext() , "No
                Internet Connection", Toast.LENGTH_LONG).show();
            }
        } else {
            // Prompt user to enter credentials
            Toast.makeText(getApplicationContext() , "Kolom tidak
            boleh kosong", Toast.LENGTH_LONG).show();
        }
    }
}

```

Pada kode diatas, ketika button login diklik maka sistem akan melakukan proses verifikasi terhadap username dan password. Jika email dan password yang dimasukkan sesuai, maka sistem akan menyatakan bahwa proses login tersebut telah sukses dan sistem akan langsung masuk ke halaman dashboard home (LoginPagee()).

Tetapi, jika salah satu variabel tersebut tidak memenuhi syarat, baik

variabel email atau password yang tidak sesuai, maka sistem akan mengeluarkan notifikasi yang menyatakan bahwa login salah atau error.

Berikut ini merupakan contoh proses login, proses ini menggunakan kode yang berfungsi untuk menjalankan coding didalam latar belakang.

```
public void onResponse(String response) {
    Log.e(TAG, "Login Response: " + response.toString());
    hideDialog();

    try {
        JSONObject jsonObj = new JSONObject(response);
        success = jsonObj.getInt(TAG_SUCCESS);
        System.out.println("Sukses = "+success);

        // Check for error node in json
        if (success == 1) {
            String username = jsonObj.getString(TAG_USERNAME);
            String id = jsonObj.getString(TAG_ID);

            System.out.println("Username = "+username);

            Log.e("Successfully Login!", jsonObj.toString());

            Toast.makeText(getApplicationContext(),
            jsonObj.getString(TAG_MESSAGE), Toast.LENGTH_LONG).show();

            // menyimpan Login ke session
            SharedPreferences.Editor editor =
            sharedPreferences.edit();
            editor.putBoolean(session_status, true);
            editor.putString(TAG_ID, id);
            editor.putString(TAG_USERNAME, username);
            editor.commit();

            // Memanggil main activity
            Intent intent = new Intent(Login.this,
            Home.class);

            intent.putExtra(TAG_ID, id);
            intent.putExtra(TAG_USERNAME, username);
            finish();
            startActivity(intent);
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
```

```

        } else {
            Toast.makeText(getApplicationContext(),
                jsonObj.getString(TAG_MESSAGE),
                Toast.LENGTH_LONG).show();
        }
    } catch (JSONException e) {
        // JSON error
        e.printStackTrace();
    }
}
}, new Response.ErrorListener() {

```

Kode diatas melakukan request ke server dalam mengambil data mengenai username dan password. Sistem akan melakukan verifikasi kepada server dan jika data yang dimasukkan sudah sesuai, berarti proses response ke HTTP telah berhasil dan akan mengarah ke pembuatan sumber daya baru secara efektif yang telah dibuat sebelum response ini dikirim. Namun jika salah satu tidak sesuai dengan data yang telah ada pada server, maka sistem akan menunjukkan bahwa request gagal atau error karena data tidak valid.

4.2.6.3 Perancangan Halaman Dashboard Menu

Berikut ini merupakan kode untuk merancang halaman dashboard yang terdiri dari fitur wilayah, parkir dan tips&trik serta fitur untuk logout user. Berikut ini merupakan package yang digunakan dalam membangun halaman dashboard menu.

```

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;

import android.content.Context;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

```

Pada halaman ini menggunakan beberapa fungsi untuk memanggil halaman lainnya, fungsi tersebut dibungkus dalam sebuah button untuk mengarahkan user ke dalam fitur yang diinginkannya.

```
//SESSION
sharedpreferences =
getSharedPreferences(Login.my_shared_preferences,
Context.MODE_PRIVATE);
id = getIntent().getStringExtra(TAG_ID);
username = getIntent().getStringExtra(TAG_USERNAME);
txt_id.setText(id);
txt_username.setText(username);

btn_wilayah.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivity(new Intent(Home.this, Wilayah.class));
    }
});
```

Pada kode diatas merupakan sebuah Material Button, jika user melakukan proses klik akan mengarah ke halaman untuk mencari wilayah tempat parkir dengan melakukan return ke Peta.

Berikutnya adalah kode untuk Material Button untuk mengarahkan ke halaman dashboard peta. Berikut ini merupakan penulisan dalam android studio.

```
//SESSION
sharedpreferences =
getSharedPreferences(Login.my_shared_preferences,
Context.MODE_PRIVATE);
id = getIntent().getStringExtra(TAG_ID);
username = getIntent().getStringExtra(TAG_USERNAME);
txt_id.setText(id);
txt_username.setText(username);

btn_peta.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivity(new Intent(Home.this,
MapsActivity.class));
    }
});
```

Selanjutnya adalah kode untuk Material Button yang mengarahkan ke halaman tips&trik, penulis membungkusnya berikut dengan kode

yang mengarahkan ke halaman logout untuk user. Berikut ini adalah penulisan kode untuk halaman menu.

```
//SESSION
sharedpreferences =
getSharedPreferences(Login.my_shared_preferences,
Context.MODE_PRIVATE);
id = getIntent().getStringExtra(TAG_ID);
username = getIntent().getStringExtra(TAG_USERNAME);
txt_id.setText(id);
txt_username.setText(username);

btn_tip.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //
        startActivity(new Intent(AdminHome.this,
        ALaporanUdara.class));
        startActivity(new Intent(Home.this, Tip.class));
    }
});
```

```
//SESSION
sharedpreferences =
getSharedPreferences(Login.my_shared_preferences,
Context.MODE_PRIVATE);
id = getIntent().getStringExtra(TAG_ID);
username = getIntent().getStringExtra(TAG_USERNAME);
txt_id.setText(id);
txt_username.setText(username);

btn_logout.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SharedPreferences.Editor editor =
        sharedpreferences.edit();
        editor.putBoolean(Login.session_status, false);
        editor.putString(TAG_ID, null);
        editor.putString(TAG_USERNAME, null);
        editor.commit();

        Intent intent = new Intent(Home.this, Login.class);
        finish();
        startActivity(intent);
    }
});
```

4.2.6.4 Perancangan Halaman Wilayah

Berikut merupakan kode untuk membangun halaman wilayah yang terdiri dari fitur pencarian. Data tempat parkir akan muncul jika user klik button wilayah tempat parkir itu masing-masing. Berikut package yang digunakan dalam halaman ini.

```
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v4.widget.SwipeRefreshLayout;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Toast;
```

Pada halaman ini, akan muncul sebuah text field berupa key input yang digunakan untuk mencari data tempat parkir berdasarkan dengan wilayah nya masing-masing. Berikut ini penulisan kode dalam android studio.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_wilayah);

    list_wilayah = (ListView)
    findViewById(R.id.list_wilayah);
    swipe_refresh = (SwipeRefreshLayout)
    findViewById(R.id.swipe_refresh);
    cari_wilayah = (EditText)
    findViewById(R.id.cari_wilayah);
    btn_cari = (Button) findViewById(R.id.btn_cari);

    adapter = new Adapterwilayah(Wilayah.this, itemList);
    list_wilayah.setAdapter(adapter);

    swipe_refresh.setOnRefreshListener(this);

    swipe_refresh.post(new Runnable() {
        @Override
        public void run() {
        }
    });
};
```

Pada halaman dibawah ini, akan muncul icon search berupa button untuk melakukan pencarian terhadap data tempat parkir berdasarkan wilayah tempat parkir tersebut. Berikut ini penulisan kode dalam

```
private void callVolley(){
    itemList.clear();
    adapter.notifyDataSetChanged();
    swipe_refresh.setRefreshing(true);

    // membuat request JSON
    JsonRequest jArr = new JsonRequest(url_select, new
    Response.Listener<JSONArray>() {
        @Override
        public void onResponse(JSONArray response) {
            Log.d(TAG, response.toString());

            // Parsing json
            for (int i = 0; i < response.length(); i++) {
                try {
                    JSONObject obj = response.getJSONObject(i);

                    Datawilayah item = new Datawilayah();

                    item.setId_wilayah(obj.getString(TAG_id_wilayah));
                    item.setNama_wilayah(obj.getString(TAG_nama_wilayah));
                    item.setJumlah(obj.getString(TAG_jumlah));

                    // menambah item ke array
                    itemList.add(item);
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }

            // notifikasi adanya perubahan data pada adapter
            adapter.notifyDataSetChanged();

            swipe_refresh.setRefreshing(false);
        }
    }, new Response.ErrorListener() {

        @Override
        public void onErrorResponse(VolleyError error) {
            VolleyLog.d(TAG, "Error: " + error.getMessage());
            swipe_refresh.setRefreshing(false);
        }
    });

    // menambah request ke request queue
    ApplicationController.getInstance().addToRequestQueue(jArr);
}
```



```

private void pencarian(){
    final ProgressDialog loading = ProgressDialog.show(this,
        "Loading...", "Please wait...", false, false);

    StringRequest strReq = new
    StringRequest(Request.Method.POST, url_cari, new
    Response.Listener<String>() {

        @Override
        public void onResponse(String response) {
            Log.e("Response: ", response.toString());

            try {
                JSONObject jsonObj = new JSONObject(response);

                int value = jsonObj.getInt(TAG_VALUE);

                if (value == 1) {
                    itemList.clear();
                    adapter.notifyDataSetChanged();

                    String getObject =
                    jsonObj.getString(TAG_RESULTS);
                    JSONArray jsonArray = new
                    JSONArray(getObject);

                    for (int i = 0; i < jsonArray.length();
                    i++) {
                        JSONObject obj =
                        jsonArray.getJSONObject(i);

                        Datawilayah item = new Datawilayah();

                        item.setId_wilayah(obj.getString(TAG_id_wilayah));
                        item.setNama_wilayah(obj.getString(TAG_nama_wilayah));
                        item.setJumlah(obj.getString(TAG_jumlah));

                        itemList.add(item);
                    }

                    } else {
                        Toast.makeText(getApplicationContext(),
                        jsonObj.getString(TAG_MESSAGE), Toast.LENGTH_SHORT).show();
                    }

                } catch (JSONException e) {
                    // JSON error
                    e.printStackTrace();
                }

                adapter.notifyDataSetChanged();
                loading.dismiss();
            }
        }, new Response.ErrorListener() {

```

4.2.6.5 Perancangan Halaman Peta

Berikut merupakan kode untuk membangun halaman peta yang terdiri dari fitur pencarian dan direktori ke tempat parkir yang dicari. Data detail tempat parkir akan muncul jika user klik button list poin tempat parkir itu masing-masing. Berikut package yang digunakan

```
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.location.LocationListener;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.CameraPosition;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.parkir.reno.app.AppController;
import com.parkir.reno.helper.Config;
```

Pada halaman ini, akan muncul sebuah maps beserta list poin tempat parkir apabila user klik button menu Peta. List poin tersebut memberikan data detail tempat parkir yang dicari oleh user. Berikut ini penulisan kode dalam android studio.

```
public void onMapReady(GoogleMap googleMap) {
    gMap = googleMap;
    checkLocationPermission();
    lat = getIntent().getStringExtra("lat");

    lng = getIntent().getStringExtra("lng");
    nama = getIntent().getStringExtra("alamat");
    // Mengarahkan ke monas
    double dLat = Double.parseDouble(lat);
    double dLng = Double.parseDouble(lng);
    Value = 10;
    center = new LatLng(dLat, dLng);
    cameraPosition = new
    CameraPosition.Builder().target(center).zoom(10).build();
    gMap.addMarker(new
    MarkerOptions().position(center).title(nama));

    gMap.animateCamera(CameraUpdateFactory.newLatLngZoom(center, Value
    ));
    /* if (lat.equals("123") && lng.equals("123")){
    }
    else{
        CameraPosition.Builder().target(center).zoom(10).build();
    }
}
```

```

//googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(par
kir, 15));
}*/
if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
    // TODO: Consider calling
    //     ActivityCompat#requestPermissions
    // here to request the missing permissions, and then
    // overriding
    //     public void onRequestPermissionsResult(int
    requestCode, String[] permissions,
    //                                     int[]
    grantResults)
    // to handle the case where the user grants the
    // permission. See the documentation
    // for ActivityCompat#requestPermissions for more
    // details.
    return;
}
if (android.os.Build.VERSION.SDK_INT >=
Build.VERSION_CODES.M) {
    if (ContextCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION)
        == PackageManager.PERMISSION_GRANTED) {
        buildGoogleApiClient();
        gMap.setMyLocationEnabled(true);
    }
} else {
    buildGoogleApiClient();
    gMap.setMyLocationEnabled(true);
}
gMap.getUiSettings().setMyLocationButtonEnabled(true);
}

```

Pada halaman dibawah ini, akan muncul sebuah direktori berupa result yang menampilkan arah untuk user jika ingin menuju ke tempat parkir yang telah dicari. Berikut ini masing-masing penulisan kode dalam android studio.

```

public void onRequestPermissionsResult(int requestCode, String
permissions[], int[] grantResults) {
    switch (requestCode) {
        case MY_PERMISSIONS_REQUEST_LOCATION: {
            if (grantResults.length > 0
                && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                // Izin diberikan.
                if (ContextCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_FINE_LOCATION)
                    == PackageManager.PERMISSION_GRANTED) {
                    if (mGoogleApiClient == null) {
                        buildGoogleApiClient();
                    }
                    gMap.setMyLocationEnabled(true);
                }
            } else {
                // Izin ditolak.
                Toast.makeText(this, "permission denied",
Toast.LENGTH_LONG).show();
            }
            return;
        }
    }
}

private void addMarker(final double loclat, final double
loclong) {
    lat = getIntent().getStringExtra("lat");

    lng = getIntent().getStringExtra("lng");
    nama = getIntent().getStringExtra("alamat");
    double dLat = Double.parseDouble(lat);
    double dLng = Double.parseDouble(lng);
    LatLng latLng = new LatLng(dLat, dLng);
    float[] results = new float[1];
    Location.distanceBetween(loclat, loclong,
        Double.parseDouble(lat),
        Double.parseDouble(lng), results);
    int meter = (int) results[0];
    int distance = meter / 1000;
    markerOptions.position(latLng);
    markerOptions.title(nama);
    //      markerOptions.snippet(keterangan);
    markerOptions.snippet("Jarak : " + distance + "Km");

    gMap.addMarker(markerOptions);

    gMap.setOnInfoWindowClickListener(new
GoogleMap.OnInfoWindowClickListener() {

```

```

        public void onInfoWindowClick(Marker marker) {
//            Toast.makeText(getApplicationContext(),
marker.getTitle(), Toast.LENGTH_SHORT).show();
            Toast.makeText(getApplicationContext(),
marker.getSnippet(), Toast.LENGTH_SHORT).show();
        }
    });
}

```

4.2.6.6 Perancangan Halaman Tips&Trik

Berikut merupakan kode untuk membangun halaman tips&trik yang terdiri dari fitur pencarian dan list informasi. Tips&trik dan informasi dalam berkendara akan muncul jika user klik button menu tips&trik tersebut. Berikut adalah package yang digunakan dalam halaman ini.

```

import android.app.ProgressDialog;
import android.os.Bundle;
import android.support.v4.widget.SwipeRefreshLayout;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Toast;

```

Pada halaman ini, akan muncul sebuah text field berupa key input yang digunakan untuk mencari tips&trik dan informasi dalam berkendara. Berikut ini penulisan kode dalam android studio.

```

private void pencarian(){
    final ProgressDialog loading = ProgressDialog.show(this,
    "Loading...", "Please wait...", false, false);

    StringRequest strReq = new
    StringRequest(Request.Method.POST, url_cari, new
    Response.Listener<String>() {

        @Override
        public void onResponse(String response) {
            Log.e("Response: ", response.toString());

            try {
                JSONObject jsonObj = new JSONObject(response);

                int value = jsonObj.getInt(TAG_VALUE);

```

```

if (value == 1) {
    itemList.clear();
    adapter.notifyDataSetChanged();

    String getObject = jsonObj.getString(TAG_RESULTS);
    JSONArray jsonArray = new JSONArray(getObject);

    for (int i = 0; i < jsonArray.length(); i++) {
        JSONObject obj = jsonArray.getJSONObject(i);

        Datatip item = new Datatip();

        item.setId_tip(obj.getString(TAG_id_tip));
        item.setJudul(obj.getString(TAG_judul));
        item.setKeterangan(obj.getString(TAG_keterangan));

        itemList.add(item);
    }

    } else {
        Toast.makeText(getApplicationContext(),
        jsonObj.getString(TAG_MESSAGE), Toast.LENGTH_SHORT).show();
    }

    } catch (JSONException e) {
        // JSON error
        e.printStackTrace();
    }

    adapter.notifyDataSetChanged();
    loading.dismiss();
}
}

```

4.2.6.7 Perancangan Halaman Login Admin

Berikut merupakan kode untuk membangun halaman login user pada aplikasi informasi tempat parkir terdekat berbasis android. Pada fitur ini hanya diperuntukkan admin pada aplikasi yang bertujuan untuk mengelola data tempat parkir.


```

        echo "<div class='alert alert-warning'
role='alert'>Login gagal! username dan password
salah!</p></div>";

        if($_GET['pesan'] == "simpan"){
            echo "<div class='alert alert-warning'
role='alert'>Berhasil menyimpan data</div></p>";
        }
    }
?>
<p>
        <span class="fontawesome-user"></span>
        <input type="text" placeholder="Username"
name="username" required>
    </p>
<p>
        <span class="fontawesome-lock"></span>
        <input type="password" placeholder="Password"
name="password" required>
    </p>
<p>
        <input type="submit" value="Sign In"
name="submit_login"/>
    </p>
</fieldset>
</form>

</div> <!-- end login -->

</div>

```

4.2.6.8 Perancangan Halaman Dashboard Admin

Berikut ini merupakan halaman dashboard menu untuk admin yang terdiri dari fitur wilayah, parkir dan tips&trik serta fitur. Berikut ini merupakan kode yang digunakan dalam membangun halaman dashboard menu.

```
<!-- Navigation -->
<nav class="navbar navbar-inverse navbar-fixed-top"
role="navigation">

  <!-- Brand and toggle get grouped for better mobile display
  -->

  <div class="navbar-header">

    <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-ex1-collapse">

      <span class="sr-only">Toggle navigation</span>

      <span class="icon-bar"></span>

      <span class="icon-bar"></span>

      <span class="icon-bar"></span>

    </button>

  </div>

  <!-- Sidebar Menu Items - These collapse to the responsive
navigation menu on small screens -->

  <div class="collapse navbar-collapse navbar-ex1-collapse">

    <ul class="nav navbar-nav side-nav">

      <li>

        <a href="index.php"><i class="fa fa-fw fa-
dashboard"></i> Dashboard</a>

      </li>

      <li >

        <a href="wilayah.php"><span class="glyphicon
glyphicon-inbox"></span> Wilayah</a>

      </li>
```

```

        <li >
            <a href="parkir.php"><span class="glyphicon
glyphicon-inbox"></span> Parkir</a>
        </li>
        <li >
            <a href="tip.php"><span class="glyphicon
glyphicon-inbox"></span> Tips & Trik</a>
        </li>
        <li >
            <a href="pengaturan.php"><span class="glyphicon
glyphicon-inbox"></span> Pengaturan</a>
        </li>
        <li>
            <a href="index.php?logout='1'"><i class="fa fa-
fw fa-power-off"></i> Keluar</a>
        </li>
    </ul>
</div>
<!-- /.navbar-collapse -->
</nav>

```

4.2.6.9 Perancangan Halaman Wilayah Admin

Halaman ini merupakan kode untuk membangun menu wilayah untuk admin yang terdiri dari menu search, input, edit dan delete. Data tempat parkir akan muncul jika admin klik button wilayah tempat parkir. Berikut kode yang digunakan dalam halaman ini.

```

<html lang="en">
<head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-
scale=1">

    <meta name="description" content="">

    <meta name="author" content="">

    <title>Produk</title>

    <!-- Bootstrap Core CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- Custom CSS -->
    <link href="css/sb-admin.css" rel="stylesheet">

    <link href="css/style.css" rel="stylesheet">

    <!-- Custom Fonts -->
    <link href="font-awesome/css/font-awesome.min.css"
rel="stylesheet" type="text/css">

</head>
<body>

    <div id="wrapper">

        <?php include "menu.php" ?>

        <div id="page-wrapper">

            <div class="container-fluid">

                <!-- Page Heading -->

                <div class="row">

                    <div class="col-lg-12">

                        <h1 class="page-header">

                            Wilayah

                        </h1>

```

```

<ol class="breadcrumb">
  <li>
    <i class="glyphicon glyphicon-inbox"></i>&nbsp;&nbsp;&nbsp;
    Wilayah
  </li>
</ol>
</div>
</div>

```

Pada halaman ini, akan muncul sebuah text field berupa key input yang digunakan untuk menambah data tempat parkir apabila admin ingin menambahkan sesuatu. Berikut ini penulisan kode dalam android studio.

```

<div class="col-lg-6">
  <form action="wilayah_aksi.php" enctype="multipart/form-
  data" method="post" accept-charset="utf-8">
    <div class="form-group">
      <input type="text" class="hidden" name="id_wilayah">
    </div>
    <div class="form-group">
      Nama Wilayah : <input type="text" class="form-control"
      name="nama_wilayah" required
      oninvalid="this.setCustomValidity('Kolom Nama Wilayah Tidak
      Boleh Kosong')" oninput="setCustomValidity('')">
    </div>
    <div class="form-group">
      <button type="submit" name="submit_tambah" class="btn
      btn-default">SUBMIT</button>
    </div>
  </form>
</div>
</div>
<!-- /.container-fluid -->
</div>
<!-- /#page-wrapper -->

```


Kemudian pada halaman dibawah ini, akan muncul icon edit berupa button untuk melakukan edit jika ada data yang berubah pada data tempat parkir. Berikut ini penulisan kode dalam android studio.

```
<div class="col-lg-6">

    <form action="wilayah_aksi.php" enctype="multipart/form-
data" method="post" accept-charset="utf-8">

        <div class="form-group">

            <input type="text" value='<?php echo $data['id_wilayah']
?>' class="hidden" name="id_wilayah">

        </div>

        <div class="form-group">

            Nama Wilayah : <input type="text" value='<?php echo
$data['nama_wilayah'] ?>' class="form-control"
name="nama_wilayah" required
oninvalid="this.setCustomValidity('Kolom Nama Wilayah Tidak
Boleh Kosong')" oninput="setCustomValidity('')">

        </div>

        <div class="form-group">

            <button type="submit" name="submit_edit" class="btn btn-
default">SUBMIT</button>

        </div>

    </form>

</div>

</div>

<!-- /.container-fluid -->

</div>

<!-- /#page-wrapper -->

</div>

<!-- /#wrapper -->

<!-- jQuery -->

<script src="js/jquery.js"></script>
```

Selanjutnya akan muncul icon delete berupa button yang berfungsi jika ada data yang salah atau ingin dihapus pada data tempat parkir. Berikut ini penulisan kode dalam android studio.

```
<?php
include "config/koneksi.php";
$id_wilayah = $_GET['id'];

$query = "DELETE FROM wilayah WHERE
id_wilayah='$id_wilayah'";

$sql = mysqli_query($con, $query);

if($sql){ // Cek jika proses simpan ke database sukses atau
tidak

    // Jika Sukses, Lakukan :

    header("location:wilayah.php");

}

?>
```

4.2.6.10 Perancangan Halaman Parkir Admin

Halaman ini merupakan kode untuk membangun menu parkir yang terdiri dari fitur search, input, edit dan delete. Menu ini hanya dapat diakses oleh admin yang dimana jika admin klik button Parkir, admin dapat mengelola fitur yang tersedia. Berikut ini adalah kode program yang digunakan dalam halaman ini.

```
<html lang="en">
<head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-
scale=1">

    <meta name="description" content="">

    <meta name="author" content="">

    <title>Parkir</title>
```

```

<!-- Bootstrap Core CSS -->

<link href="css/bootstrap.min.css" rel="stylesheet">

<!-- Custom CSS -->

<link href="css/sb-admin.css" rel="stylesheet">

<link href="css/style.css" rel="stylesheet">

<!-- Custom Fonts -->

<link href="font-awesome/css/font-awesome.min.css"
rel="stylesheet" type="text/css">
</head>
<body>
    <div id="wrapper">
        <?php include "menu.php" ?>
        <div id="page-wrapper">
            <div class="container-fluid">
                <!-- Page Heading -->
                <div class="row">
                    <div class="col-lg-12">
                        <h1 class="page-header">
                            Parkir
                        </h1>
                        <ol class="breadcrumb">
                            <li>
                                <i class="glyphicon glyphicon-inbox"></i>&nbsp;&nbsp;
                                Parkir
                            </li>
                        </ol>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

Pada halaman ini, akan muncul sebuah text field berupa key input yang digunakan untuk menambah list data tempat parkir apabila admin ingin menambahkan nya. Berikut ini penulisan kode dalam android studio.

```
<div class="col-lg-6">

    <form action="parkir_aksi.php" enctype="multipart/form-data"
method="post" accept-charset="utf-8">

        <div class="form-group">

            <input type="text" class="hidden" name="id_parkir">

        </div>

        <div class="form-group">

            Nama Wilayah : <select name="id_wilayah" class="form-
control" required oninvalid="this.setCustomValidity('Kolom Nama
Wilayah Tidak Boleh Kosong')" oninput="setCustomValidity('')">

                <option value="" selected>-Pilih-</option>

                <?php
                    $sqlwil      = "SELECT * FROM wilayah ";
                    $querywil    = mysqli_query($con, $sqlwil);
                    while($r=mysqli_fetch_array($querywil)){
                        echo "<option
value='". $r['id_wilayah']. "'>". $r['nama_wilayah']. "</option>";
                    }?>

            </select>

        </div>

        <div class="form-group">

            Alamat :<textarea name="alamat" class="form-control"
required oninvalid="this.setCustomValidity('Kolom Alamat Tidak
Boleh Kosong')" oninput="setCustomValidity('')"></textarea>

        </div>
```

```

<div class="form-group">

    Lat : <input type="text" class="form-control" name="lat"
required oninvalid="this.setCustomValidity('Kolom Lat Tidak
Boleh Kosong')" oninput="setCustomValidity('')">

</div>

<div class="form-group">

    Lang : <input type="text" class="form-control"
name="lng" required oninvalid="this.setCustomValidity('Kolom
Lang Tidak Boleh Kosong')" oninput="setCustomValidity('')">

</div>

<div class="form-group">

    Perusahaan : <input type="text" class="form-control"
name="perusahaan" required
oninvalid="this.setCustomValidity('Kolom Perusahaan Tidak Boleh
Kosong')" oninput="setCustomValidity('')">

</div>

<div class="form-group">

    Mobil :<input type="number" class="form-control"
name="mobil" required oninvalid="this.setCustomValidity('Kolom
Mobil Tidak Boleh Kosong')" oninput="setCustomValidity('')">

</div>

<div class="form-group">

    Motor :<input type="number" class="form-control"
name="motor" required oninvalid="this.setCustomValidity('Kolom
Motor Tidak Boleh Kosong')" oninput="setCustomValidity('')">

</div>

<div class="form-group">

    <button type="submit" name="submit_tambah" class="btn
btn-default">SUBMIT</button>

</div>

</form>

</div>

```

Kemudian pada halaman dibawah ini, akan muncul icon edit berupa button untuk melakukan edit jika ada data yang berubah pada data tempat parkir. Berikut ini penulisan kode dalam android studio.

```
<div class="col-lg-6">

    <form action="parkir_aksi.php" enctype="multipart/form-data"
method="post" accept-charset="utf-8">

        <div class="form-group">

            <input type="text" value='<?php echo $data['id_parkir']
?>' class="hidden" name="id_parkir">

        </div>

        <div class="form-group">

            Nama Wilayah : <select name="id_wilayah" class="form-
control" required oninvalid="this.setCustomValidity('Kolom Nama
Wilayah Tidak Boleh Kosong')" oninput="setCustomValidity('')">

                <option value='<?php echo $data['id_wilayah'] ?>'
selected><?php echo $data['nama_wilayah'] ?></option>";

                <?php
                $sqlwil      = "SELECT * FROM wilayah ";
                $querywil    = mysqli_query($con, $sqlwil);
                while($r=mysqli_fetch_array($querywil)){
                    echo "<option
value='". $r['id_wilayah'] ."'>". $r['nama_wilayah'] . "</option>";
                }?>

            </select>

        </div>

        <div class="form-group">

            Alamat :<textarea name="alamat" class="form-control"
required oninvalid="this.setCustomValidity('Kolom Alamat Tidak
Boleh Kosong')" oninput="setCustomValidity('')"><?php echo
$data['alamat'] ?></textarea>

        </div>

    </div>
```



```

<div class="form-group">

    Lat : <input type="text" value='<?php echo $data['lat']
?>' class="form-control" name="lat" required
oninvalid="this.setCustomValidity('Kolom Lat Tidak Boleh
Kosong')" oninput="setCustomValidity('')">

</div>

<div class="form-group">

    Lang : <input type="text" value='<?php echo $data['lng']
?>' class="form-control" name="lng" required
oninvalid="this.setCustomValidity('Kolom Lang Tidak Boleh
Kosong')" oninput="setCustomValidity('')">

</div>

<div class="form-group">

    Perusahaan : <input type="text" value='<?php echo
$data['perusahaan'] ?>' class="form-control" name="perusahaan"
required oninvalid="this.setCustomValidity('Kolom Perusahaan
Tidak Boleh Kosong')" oninput="setCustomValidity('')">

</div>

<div class="form-group">

    Mobil :<input type="number" value='<?php echo
$data['mobil'] ?>' class="form-control" name="mobil" required
oninvalid="this.setCustomValidity('Kolom Mobil Tidak Boleh
Kosong')" oninput="setCustomValidity('')">

</div>

<div class="form-group">

    Motor :<input type="number" value='<?php echo
$data['motor'] ?>' class="form-control" name="motor" required
oninvalid="this.setCustomValidity('Kolom Motor Tidak Boleh
Kosong')" oninput="setCustomValidity('')">

</div>

<div class="form-group">

    <button type="submit" name="submit_edit" class="btn btn-
default">SUBMIT</button>

</div>

```

Selanjutnya akan muncul icon delete berupa button yang berfungsi jika ada data yang salah atau ingin dihapus pada data tempat parkir. Berikut ini penulisan kode dalam android studio.

```
<?php
include "config/koneksi.php";
$id_parkir = $_GET['id'];
$query      = "DELETE FROM parkir WHERE id_parkir='$id_parkir'";
$sql = mysqli_query($con, $query);
if($sql){ // Cek jika proses simpan ke database sukses atau
tidak
    // ...
}
```

4.2.6.11 Perancangan Halaman Tips&Trik Admin

Halaman ini merupakan kode untuk membangun menu tips&trik yang terdiri dari fitur search, input, edit dan delete. Menu ini hanya dapat diakses oleh admin yang dimana jika admin klik button Tips&trik, admin dapat mengelola fitur yang tersedia. Berikut ini adalah kode program yang digunakan dalam halaman ini.

```
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>Tips & Trik</title>
    <!-- Bootstrap Core CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- Custom CSS -->
```

```

<link href="css/sb-admin.css" rel="stylesheet">

<link href="css/style.css" rel="stylesheet">

<!-- Custom Fonts -->

<link href="font-awesome/css/font-awesome.min.css"
rel="stylesheet" type="text/css">
</head>
<body>
    <div id="wrapper">
        <?php include "menu.php" ?>
        <div id="page-wrapper">
            <div class="container-fluid">
                <!-- Page Heading -->
                <div class="row">
                    <div class="col-lg-12">
                        <h1 class="page-header">
                            Tips & Trik
                        </h1>
                        <ol class="breadcrumb">
                            <li>
                                <i class="glyphicon glyphicon-inbox"></i>&nbsp;
                                Tips & Trik
                            </li>
                        </ol>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

Pada halaman ini, akan muncul sebuah text field berupa key input yang digunakan untuk menambah list tips&trik dan informasi dalam berkendara apabila admin ingin menambahkan nya. Berikut ini penulisan kode dalam android studio.

```
<div class="col-lg-6">
    <form action="tip_aksi.php" enctype="multipart/form-data"
method="post" accept-charset="utf-8">
        <div class="form-group">
            <input type="text" class="hidden" name="id_tip">
        </div>
        <div class="form-group">
            Judul : <input type="text" class="form-control"
name="judul" required oninvalid="this.setCustomValidity('Kolom
Judul Tidak Boleh Kosong')" oninput="setCustomValidity('')">
        </div>
        <div class="form-group">
            Keterangan : <textarea name="keterangan" class="form-
control" required oninvalid="this.setCustomValidity('Kolom
Keterangan Tidak Boleh Kosong')"
oninput="setCustomValidity('')"></textarea>
        </div>
        <div class="form-group">
            <button type="submit" name="submit_tambah" class="btn
btn-default">SUBMIT</button>
        </div>
    </form>
</div>
</div>
```

```

</div>

<!-- /.container-fluid -->

</div>

<!-- /#page-wrapper -->

</div>

<!-- /#wrapper -->

<!-- jQuery -->
<script src="js/jquery.js"></script>
<!-- Bootstrap Core JavaScript -->
<script src="js/bootstrap.min.js"></script>

```

Kemudian pada halaman dibawah ini, akan muncul icon edit berupa button untuk melakukan edit jika ada data yang berubah pada data list informasi dalam berkendara. Berikut ini penulisan kode dalam android

```

<div class="col-lg-6">
    <form action="tip_aksi.php" enctype="multipart/form-data"
    method="post" accept-charset="utf-8">
        <div class="form-group">
            <input type="text" value='<?php echo $data['id_tip'] ?>'
            class="hidden" name="id_tip">
        </div>
        <div class="form-group">
            Judul : <input type="text" value='<?php echo
            $data['judul'] ?>' class="form-control" name="judul" required
            oninvalid="this.setCustomValidity('Kolom Judul Tidak Boleh
            Kosong')" oninput="setCustomValidity('')">
        </div>
        <div class="form-group">

```

```

        Keterangan :<textarea name="keterangan" class="form-
control" required oninvalid="this.setCustomValidity('Kolom
Keterangan Tidak Boleh Kosong')">
oninput="setCustomValidity('')"><?php echo $data['keterangan']
?></textarea>

    </div>

    <div class="form-group">

        <button type="submit" name="submit_edit" class="btn btn-
default">SUBMIT</button>

    </div>
</form>
</div>
</div>

    <!-- /.container-fluid -->
</div>

    <!-- /#page-wrapper -->
</div>

    <!-- /#wrapper -->

    <!-- jQuery -->
    <script src="js/jquery.js"></script>

```

Selanjutnya akan muncul icon delete berupa button yang berfungsi jika ada data yang salah atau ingin dihapus pada list informasi dalam berkendara. Berikut ini penulisan kode dalam android studio.

```

<?php
include "config/koneksi.php";

$id_tip = $_GET['id'];

$query      = "DELETE FROM tip WHERE id_tip='$id_tip'";

$sql = mysqli_query($con, $query);

if($sql){ // Cek jika proses simpan ke database sukses atau
tidak

```


4.2.6.12 Perancangan Halaman Pengaturan Admin

Halaman ini merupakan kode untuk membangun menu pengaturan yang terdiri dari sebuah text field berupa key input yang digunakan untuk mengelola data username dan password admin, menu ini hanya dapat diakses oleh admin. Berikut ini adalah kode program yang digunakan dalam halaman ini.

```
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <meta name="description" content="">
  <meta name="author" content="">
  <title>Pengaturan</title>
  <!-- Bootstrap Core CSS -->
  <link href="css/bootstrap.min.css" rel="stylesheet">
  <!-- Custom CSS -->
  <link href="css/sb-admin.css" rel="stylesheet">
  <link href="css/style.css" rel="stylesheet">
  <!-- Custom Fonts -->
  <link href="font-awesome/css/font-awesome.min.css"
rel="stylesheet" type="text/css">
</head>
<body>
<div id="wrapper">
  <?php include "menu.php" ?>
  <div id="page-wrapper">
    <div class="container-fluid">
      <!-- Page Heading -->
```

```

<div class="row">

    <div class="col-lg-12">

        <h1 class="page-header">
            Pengaturan
        </h1>

        <ol class="breadcrumb">

            <li>

                <i class="glyphicon glyphicon-inbox"></i>&nbsp;  
                Pengaturan

            </li>

        </ol>

    </div>

</div>

<div class="col-lg-6">

    <form action="pengaturan_aksi.php" enctype="multipart/form-
data" method="post" accept-charset="utf-8">

        <div class="form-group">

            <input type="text" value='<?php echo $data['id_user']
?>' class="hidden" name="id_user">

        </div>

        <div class="form-group">

            Username : <input type="text" value='<?php echo
$data['username'] ?>' class="form-control" name="username"
required oninvalid="this.setCustomValidity('Kolom Username Tidak
Boleh Kosong')" oninput="setCustomValidity('')">

        </div>

        <div class="form-group">

            Password : <input type="text" value='<?php echo
$data['password'] ?>' class="form-control" name="password"
required oninvalid="this.setCustomValidity('Kolom Password Tidak
Boleh Kosong')" oninput="setCustomValidity('')">

        </div>

```

4.3 Implementation (Penerapan Pengujian)

Setelah melakukan tahap analisis kebutuhan sistem, perancangan desain dan pengkodean, peneliti melakukan pengujian terhadap tahap implementasi sistem. Metode yang digunakan dalam pengujian adalah *black box testing* dan *user acceptance testing*.

4.3.1 Black Box Testing

Tahap pengujian ini dimaksudkan untuk mengetahui apakah setiap fungsi yang ada pada sistem aplikasi informasi tempat parkir terdekat berbasis android yang telah di bangun sudah berfungsi sesuai dengan rancangan yang telah dibuat. Pengujian yang dilakukan pada sistem aplikasi informasi tempat parkir terdekat ini menggunakan *black box testing*.

Tabel 4. 4 Pengujian Black Box Login Admin

Data Yang Dimasukan	Diharapkan	Pengamatan	Kesimpulan
Input Username dan Password diisi dengan data yang sesuai untuk admin.	Admin masuk ke halaman utama admin.	Menampilkan menu utama sesuai dengan yang dituju.	Diterima.
Klik button login	Admin masuk ke halaman utama admin.	Menampilkan menu utama sesuai dengan yang dituju.	Diterima.
Kasus dan Hasil Uji (Data Salah)			
Input Username atau Password diisi dengan data yang tidak sesuai untuk admin.	Tidak dapat masuk ke dalam halaman utama admin.	Admin tidak dapat masuk kedalam halaman utama admin dan muncul pesan	Diterima.

		bahwa admin memasukkan username atau password yang tidak sesuai.	
--	--	--	--

Tabel 4. 5 Pengujian Black Box Halaman Utama Admin

Data Yang Dimasukan	Diharapkan	Pengamatan	Kesimpulan
Klik button Wilayah	Admin masuk ke halaman wilayah.	Menampilkan halaman wilayah dengan menampilkan field pencarian dan tabel pembagian masing-masing wilayah	Diterima.
Klik button Parkir	Admin masuk ke halaman parkir.	Menampilkan halaman parkir dengan menampilkan list data tempat parkir	Diterima.
Klik button Tips&Trik	Admin masuk ke halaman tips&trik	Menampilkan halaman tips&trik dengan menampilkan keterangan	Diterima.

		berisikan informasi.	
Klik button Logout	Admin keluar dari halaman utama a menu admin dan menampilkan halaman utama a aplikasi.	Menampilkan halaman utama aplikasi informasi tempat parkir.	Diterima.

Tabel 4. 6 Pengujian Black Box Halaman Wilayah (Admin)

Data Yang Dimasukan	Diharapkan	Pengamatan	Kesimpulan
Klik field pencarian	Tampil key input wilayah berupa huruf.	Menampilkan key input berupa huruf dan mencari wilayah.	Diterima.
Klik button Tambah baru	Tampil key input Nama Wilayah berupa huruf	Menampilkan key input berupa huruf dan mengisi nama wilayah	Diterima.

Klik button search	Tampil data pencarian berupa wilayah yang dicari.	Menampilkan list tempat parkir yang dicari berdasarkan nama wilayah.	Diterima.
Klik button icon edit	Admin masuk ke halaman edit wilayah.	Menampilkan halaman edit wilayah.	Diterima.
Klik button icon delete	Tampil menu delete yang dapat menghapus data wilayah	Menampilkan pilihan untuk menghapus data wilayah tempat parkir. Jjika admin memilih delete, maka data wilayah tempat parkir tersebut akan terhapus dari database.	Diterima.

Tabel 4. 7 Pengujian Black Box Halaman Parkir (Admin)

Data Yang Dimasukan	Diharapkan	Pengamatan	Kesimpulan
Klik button Tambah baru	Admin masuk ke halaman parkir, dan tampil key input data tempat parkir berupa huruf.dan angka	Menampilkan halaman parkir dengan menampilkan halaman form input tambah baru untuk data tempat parkir	Diterima.
Klik field pencarian	Tampil key input tempat parkir berupa huruf.	Menampilkan key input berupa huruf dan mencari data tempat parkir.	Diterima.
Klik button icon edit	Admin masuk ke halaman edit parkir.	Menampilkan haliman edit parkir.dengan menampilkan halaman form edit untuk data tempat parkir.	Diterima.

Klik button icon delete	Tampil menu delete yang dapat menghapus data tempat parkir	Menampilkan pilihan untuk menghapus data tempat parkir. Jika admin memilih delete, maka data tempat parkir tersebut akan terhapus dari database.	Diterima.
-------------------------	--	--	-----------

Tabel 4. 8 Pengujian Black Box Halaman Tips&Trik (Admin)

Data Yang Dimasukan	Diharapkan	Pengamatan	Kesimpulan
Klik button Tambah baru	Admin masuk ke halaman Tips&trik, dan tampil key input tips&trik berupa huruf.dan angka.	Menampilkan halaman parkir dengan menampilkan halaman form input tambah baru untuk tips &trik.	Diterima.
Klik field pencarian	Tampil key input tips&trik berupa huruf.dan angka	Menampilkan key input berupa huruf dan angka kemudian mencari informasi	Diterima.

		berkendara.	
Klik button icon edit	Admin masuk ke halaman edit tips&trik.	Menampilkan halaman edit tips&trik dengan menampilkan halaman form edit untuk tips&trik.	Diterima.
Klik button icon delete	Tampil menu delete yang dapat menghapus informasi berkendara atau tips&trik.	Menampilkan pilihan untuk menghapus tips&trik. Jjika admin memilih delete, maka tips&trik tersebut akan terhapus dari database.	Diterima.

Tabel 4. 9 Pengujian Black Box Login (User)

Data Yang Dimasukan	Diharapkan	Pengamatan	Kesimpulan
Input Username dan Password diisi dengan data yang sesuai untuk user.	User masuk ke halaman utama user.	Menampilkan menu utama sesuai dengan yang dituju.	Diterima.
Klik button login	User masuk ke halaman utama user.	Menampilkan menu utama sesuai dengan yang dituju.	Diterima.
Kasus dan Hasil Uji (Data Salah)			
Input Username atau Password diisi dengan data yang tidak sesuai untuk user.	Tidak dapat masuk ke dalam halaman utama user.	User tidak dapat masuk kedalam halaman utama user dan muncul pesan bahwa user memasukkan username atau password yang tidak sesuai.	Diterima.

Tabel 4. 10 Pengujian Black Box Halaman Utama (User)

Data Yang Dimasukan	Diharapkan	Pengamatan	Kesimpulan
Klik button Wilayah	User masuk ke halaman pencarian wilayah tempat parkir berupa field pencarian dan button search.	Menampilkan halaman Wilayah dengan menampilkan field pencarian tempat parkir dan button search.	Diterima.
Klik button Peta	User masuk ke halaman dashboard Peta	Menampilkan halaman dashboard Peta	Diterima.
Klik button Tips&trik	User masuk ke halaman utama tips&trik	Menampilkan halaman utama tips&trik.	Diterima.

Tabel 4. 11 Pengujian Black Box Halaman Wilayah (User)

Data Yang Dimasukan	Diharapkan	Pengamatan	Kesimpulan
Klik field pencarian	Tampil key input wilayah berupa huruf.	Menampilkan key input berupa huruf, lalu mencari nama wilayah berupa huruf.	Diterima.

Klik button search	Tampil data pencarian berupa wilayah dan data tempat parkir yang dicari.	Menampilkan data detail tempat parkir yang dicari berdasarkan wilayah.	Diterima.
--------------------	--	--	-----------

Tabel 4. 12 Pengujian Black Box Halaman Peta (User)

Data Yang Dimasukan	Diharapkan	Pengamatan	Kesimpulan
Klik button Peta	Tampil halaman utama peta berupa maps	Menampilkan halaman Peta berupa maps beserta list poin tempat parkir.	Diterima.
Klik button list poin	Tampil data detail tempat parkir berupa kapasitas dan jarak.	Menampilkan data detail tempat parkir terdekat yang dicari.	Diterima.
Klik button direktori maps	Tampil rute jalan menuju tempat parkir terdekat yang dicari.	Menampilkan rute jalan menuju tempat parkir terdekat yang dicari.	Diterima.

Tabel 4. 13 Pengujian Black Box Halaman Tips&Trik (User)

Data Yang Dimasukan	Diharapkan	Pengamatan	Kesimpulan
Klik field pencarian	Tampil key input tips&trik berupa huruf.dan angka	Menampilkan key input tips&trik, lalu menuliskan informasi berkendara berupa huruf dan angka yang dicari.	Diterima.
Klik button search	Tampil data pencarian berupa tips&trik dalam berkendara.	Menampilkan informasi berkendara yang dicari berdasarkan input.	Diterima.

Berdasarkan hasil pengujian dengan menggunakan metode *black box testing* diatas, maka penulis dapat mengambil kesimpulan bahwa proses pencarian tempat parkir terdekat dan fungsi-fungsi pokok dalam aplikasi dapat berjalan dengan baik dan menghasilkan *output* seperti yang diharapkan.

4.3.2 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) merupakan metode pengujian yang dilakukan oleh *end-user* dimana user tersebut adalah staf/karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya. *User Acceptance Testing* adalah metode testing dimana pengguna akhir melakukan percobaan untuk memvalidasi produk berdasarkan kebutuhan yang telah ditentukan. Pengujian ini bertujuan untuk membantu menemukan *bug* yang

akan muncul pada saat penggunaan. Dengan *User Acceptance Testing*, sistem dapat diuji oleh *end-user* (pengguna), agar mendapat penilaian terhadap sistem ini. Berikut merupakan rumus perhitungan persentase dari pengujian UAT:

$$\text{tingkat kesesuaian} = \frac{\text{skor sesuai}}{\text{skor sesuai} + \text{skor tidak sesuai}} \times 100$$

Dengan menggunakan metode *user acceptance testing* (UAT), uji yang dilakukan berdasarkan fungsi-fungsi aplikasi yang dibutuhkan *stakeholder*. Pengujian sistem aplikasi ini dilakukan dengan wawancara oleh Developer Dinas Perhubungan dan 5 responden (Manajer UP Dinas Perhubungan dan 4 kepala *staff*) di bidangnya masing-masing yang berkaitan dengan perparkiran di DKI Jakarta, serta 5 responden (*user*) sebagai pengendara kendaraan roda 2 ataupun roda 4. Dari hasil pengujian, setiap responden memberikan penilaian yang sama terhadap sistem. Penulis meringkasnya dengan satu penilaian sesuai dari responden menjadi 3 poin.

Tabel 4. 14 UAT-01A

User Acceptance Test			
Nama Sistem		Pengembangan Aplikasi Informasi Tempat Parkir Terdekat Di Provinsi DKI Jakarta Menggunakan Location Based Service (LBS) dan Google Maps API (Studi Kasus: Unit Perparkiran Dinas Perhubungan Provinsi DKI Jakarta)	
Nomor Pengujian		UAT-01A	
Topik Pengujian		Fungsi Halaman Menu Wilayah	
Tanggal Pengujian		25 September 2021	
Penguji		Developer Dishub, 5 responden (Manajer UP Dishub, 4 kepala <i>staff</i>), 5 responden (<i>user</i>) pengendara roda 2 atau roda 4	
No	Fungsi pokok	Sesuai	
		Ya	Tidak
1.	Menampilkan key input berupa huruf, lalu mencari nama wilayah berdasarkan input.	36	0
2.	Menampilkan data detail tempat parkir yang dicari	36	0

	berdasarkan input nama wilayah.		
Jumlah		72	0

Tabel 4.14 diatas merupakan hasil dari penilaian terhadap fitur yang terdapat pada fungsi menu *wilayah*, pengujian dilakukan oleh para responden. Hasilnya **seluruh** kasus pengujian sesuai dengan harapan pengguna.

Tabel 4. 15 UAT-02A

User Acceptance Test			
Nama Sistem	Pengembangan Aplikasi Informasi Tempat Parkir Terdekat Di Provinsi DKI Jakarta Menggunakan Location Based Service (LBS) dan Google Maps API (Studi Kasus: Unit Perparkiran Dinas Perhubungan Provinsi DKI Jakarta)		
Nomor Pengujian	UAT-02A		
Topik Pengujian	Fungsi Halaman Menu Peta		
Tanggal Pengujian	25 September 2021		
Penguji	Developer Dishub, 5 responden (Manajer UP Dishub, 4 kepala <i>staff</i>), 5 responden (<i>user</i>) pengendara roda 2 atau roda 4		
No	Fungsi pokok	Sesuai	
		Ya	Tidak
1.	Menampilkan halaman Peta berupa tampilan maps beserta list poin tempat parkir.	36	0
2.	Menampilkan data detail tempat parkir terdekat yang dicari.	36	0
3.	Menampilkan rute jalan menuju tempat parkir terdekat yang dicari.	36	0
Jumlah		108	0

Tabel 4.15 diatas merupakan hasil dari penilaian terhadap fitur yang terdapat pada fungsi menu *peta*, pengujian dilakukan oleh para responden. Hasilnya **seluruh** kasus pengujian sesuai dengan harapan pengguna.

Tabel 4. 16 UAT-03A

User Acceptance Test			
Nama Sistem	Pengembangan Aplikasi Informasi Tempat Parkir Terdekat Di Provinsi DKI Jakarta Menggunakan Location Based Service (LBS) dan Google Maps API (Studi Kasus: Unit Perparkiran Dinas Perhubungan Provinsi DKI Jakarta)		
Nomor Pengujian	UAT-03A		
Topik Pengujian	Fungsi Halaman Menu Tips&Trik		
Tanggal Pengujian	25 September 2021		
Penguji	Developer Dishub, 5 responden (Manajer UP Dishub, 4 kepala <i>staff</i>), 5 responden (<i>user</i>) pengendara roda 2 atau roda 4		
No	Fungsi pokok	Sesuai	
		Ya	Tidak
1.	Menampilkan key input tips&trik, lalu mencari informasi berkendara dengan teks huruf atau angka.	36	0
2.	Menampilkan informasi berkendara yang dicari berdasarkan input.	36	0
Jumlah		72	0

Tabel 4.16 diatas merupakan hasil dari penilaian terhadap fitur yang terdapat pada fungsi menu *tips&trik*, pengujian dilakukan oleh para responden. Hasilnya seluruh kasus pengujian sesuai dengan harapan pengguna.

Dari keseluruhan hasil pengujian yang dilakukan, setiap responden memberikan penilaian yang sama terhadap sistem. Penulis meringkasnya dengan satu penilaian sesuai dari responden menjadi 3 poin. Berdasarkan pengujian dengan menggunakan UAT, jumlah skor kesesuaian yang diperoleh dalam penelitian ini adalah total 252 poin dengan persentase kesesuaian 100%.

BAB V

HASIL DAN PEMBAHASAN

5.1 Hasil Akhir Sistem

Pada penelitian ini menghasilkan sebuah rekomendasi sistem informasi tempat parkir terdekat yang berfungsi untuk memberikan kemudahan kepada *user* (pengendara) dan bagian perparkiran Dinas Perhubungan DKI Jakarta dalam mencari informasi titik lokasi parkir terdekat. *User* dapat mencari titik lokasi parkir terdekat berdasarkan lokasi *user* ataupun wilayah nya masing-masing (Jakarta), mendapatkan data detail tempat parkir terdekat berupa kapasitas roda 2 ataupun roda 4 dan jarak menuju tempat parkir, lalu mendapatkan informasi berkendara pada menu *tips&trik*. Setelah melalui beberapa tahap proses pengembangan sistem dan pengujian, akhirnya dapat disimpulkan bahwa sistem ini berhasil memberikan informasi titik lokasi parkir terdekat kepada pengguna. Terdapat juga bagian admin yang diperuntukkan pihak developer dari Dinas Perhubungan untuk mengelola setiap data tempat parkir.

Penelitian ini memiliki beberapa manfaat dan keunggulan dari penelitian sejenisnya ataupun sistem informasi tempat parkir (*e-parkir*) yang saat ini berada dalam Dishub DKI Jakarta. Sistem yang dikembangkan ini cukup memberikan kemudahan untuk *user* dalam mencari titik lokasi parkir di sekitar tempat tujuan, dengan melakukan pencarian tempat parkir terdekat berdasarkan wilayah yang diinginkan di DKI Jakarta melalui fitur *Wilayah*. *User* melakukan pencarian titik tempat parkir terdekat dengan tampilan maps yang menampilkan seluruh titik lokasi parkir di DKI Jakarta dan detail data tempat parkirnya (kapasitas roda dua/roda 4, jarak menuju lokasi parkir dari posisi *user*) melalui fitur *Peta*. *User* juga dapat mendapatkan informasi ataupun *safety* dalam berkendara melalui fitur *Tips&trik*. Sistem ini juga bermanfaat untuk *user* yang mencari tempat parkir dengan cara konvensional dan tidak menemukan titik parkir terdekat di sekitar destinasi.

Selain memiliki beberapa keunggulan, penelitian ini juga masih memiliki kekurangan. *User* tidak dapat memantau ketersediaan tempat parkir secara *real time*

melalui sistem dan *user* tidak dapat melakukan reservasi *online (booking)* tempat parkir melalui sistem. Sistem ini berfokus untuk menampilkan titik lokasi parkir terdekat dan detail data parkir di wilayah DKI Jakarta. Dimana dalam sistem ini, *user* akan mengetahui ketersediaan tempat parkir dengan menuju langsung ke tempat parkir tersebut.

5.2 Hasil Pengujian Black Box Testing

Pada tahap ini, penulis melakukan pengujian dengan metode *black box testing* untuk mengetahui setiap fungsi yang ada pada sistem aplikasi informasi tempat parkir terdekat berbasis android yang telah di bangun sudah berfungsi sesuai dengan rancangan yang telah dibuat. Berdasarkan hasil pengujian dengan menggunakan metode pengujian *black box testing* diatas, maka penulis dapat mengambil kesimpulan bahwa proses pencarian tempat parkir terdekat dan fungsi-fungsi pokok dalam aplikasi dapat berjalan dengan baik dan menghasilkan *output* seperti yang diharapkan. Terdapat beberapa kekurangan pada sistem ini, yaitu belum maksimalnya menu *register* atau *login* untuk kebutuhan data *user* dan pencarian titik lokasi parkir terdekat pada fitur *peta* masih belum maksimal (masih menggunakan *zoom in/out* pada layar aplikasi).

5.3 Hasil Pengujian User Acceptance Testing (UAT)

Pada tahap pengujian lainnya, penulis melakukan pengujian sistem aplikasi dengan menggunakan metode *user acceptance testing* untuk mengetahui penilaian dari responden terhadap fungsi-fungsi yang ada pada sistem aplikasi informasi tempat parkir terdekat berbasis android yang telah di bangun. Data hasil pengujian dari fungsi-fungsi yang di dapat kemudian disederhanakan sebagai berikut:

Tabel 5. 1 Pengujian User Acceptance Testing

No. Uji	Fitur	Jumlah Sesuai	
		Ya	Tidak
UAT-01A	Fungsi Halaman Menu Wilayah	72	0
UAT-02A	Fungsi Halaman Menu Parkir	108	0
UAT-03A	Fungsi Halaman Menu Tips&trik	72	0

Berdasarkan tabel 5.1 dari pengujian diatas, jumlah skor kesesuaian yang diperoleh dalam penelitian ini adalah total 252 poin. Sedangkan penilaian akhir untuk tingkat kesesuaian yang diharapkan dalam *user acceptance testing* ini adalah jumlah skor sesuai ditambah dengan jumlah skor tidak sesuai dari hasil pengujian yaitu $252+0 = 252$. Maka presentase tingkat kesesuaian aplikasi ini dengan responden adalah 100%, dengan perhitungan sebagai berikut:

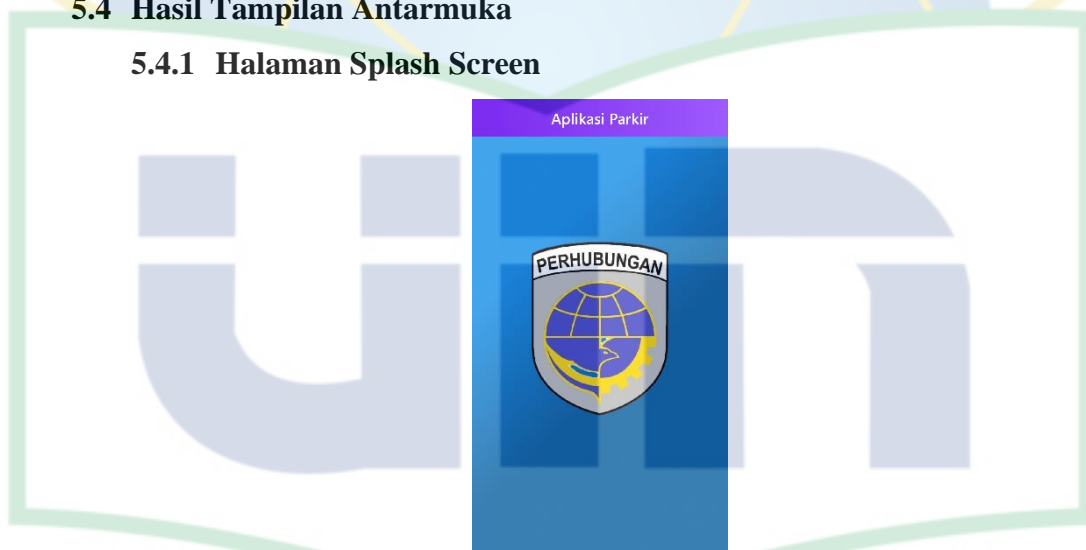
$$\text{tingkat kesesuaian} = \frac{\text{skor sesuai}}{\text{skor sesuai} + \text{skor tidak sesuai}} \times 100$$

$$\text{tingkat kesesuaian} = \frac{252}{252 + 0} \times 100$$

$$\text{tingkat kesesuaian} = 100\%$$

5.4 Hasil Tampilan Antarmuka

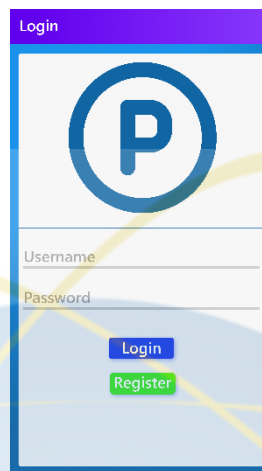
5.4.1 Halaman Splash Screen



Gambar 5. 1 Splash Screen Aplikasi

Tampilan *Splash Screen* merupakan tampilan awal ketika Aplikasi Informasi Tempat Parkir Terdekat Berbasis Android ini dibuka. *Splash Screen* akan muncul dalam waktu beberapa detik sebelum sistem menampilkan halaman utama aplikasi.

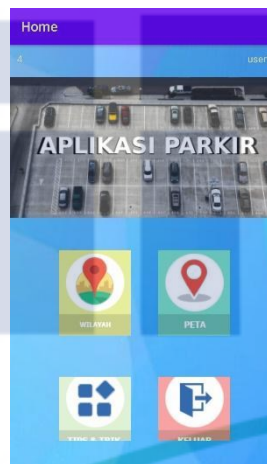
5.4.2 Halaman Login User



Gambar 5. 2 Halaman Login User

Menu Login ini merupakan fitur yang dapat digunakan oleh user. User diharuskan mengisi username dan password yang sesuai agar bisa masuk kedalam sistem.

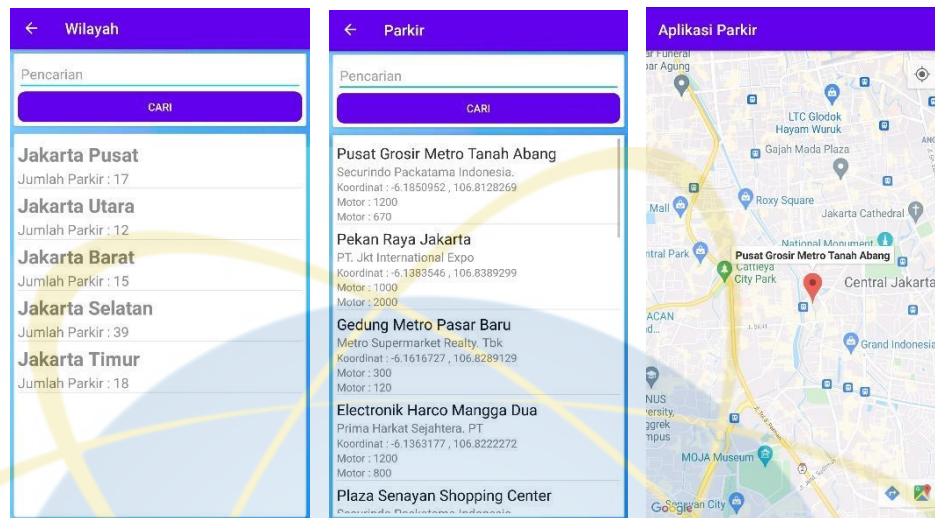
5.4.3 Halaman Home



Gambar 5. 3 Halaman Home User

Menu ini adalah halaman utama (Home Page) dari Aplikasi Informasi Tempat Parkir Terdekat Berbasis Android. Menu ini menampilkan beberapa fitur utama, yaitu fitur Wilayah, Peta, Tips&Trik dan Login.

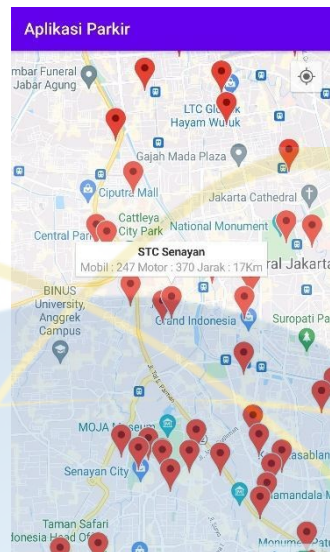
5.4.4 Halaman Menu Wilayah



Gambar 5. 4 Halaman Menu Wilayah

Menu ini merupakan sebuah fitur yang dapat diakses oleh user maupun admin, menu ini tidak memerlukan aktivasi login. Target pada fitur ini adalah user (pengendara) yang dengan mengakses menu ini, user tidak perlu repot lagi untuk mencari letak tempat parkir terdekat sesuai dengan posisi pengguna atau destinasi yang dituju. Cara penggunaan menu ini, user hanya perlu mencari tempat parkir sesuai wilayah nya masing-masing (Jakarta). Dengan memasukkan nama wilayah, sistem akan otomatis melakukan request ke webservice dan akan menampilkan informasi mengenai list tempat parkir.

5.4.5 Halaman Menu Peta



Gambar 5. 5 Halaman Menu Peta

Menu ini merupakan fitur yang menampilkan peta berupa maps beserta list poin tempat parkir. Fitur ini memberikan informasi data detail tempat parkir yang dicari apabila user klik button icon list poin tempat parkir, dapat menampilkan rute menuju tempat parkir terdekat yang dicari apabila user klik button direktori maps.

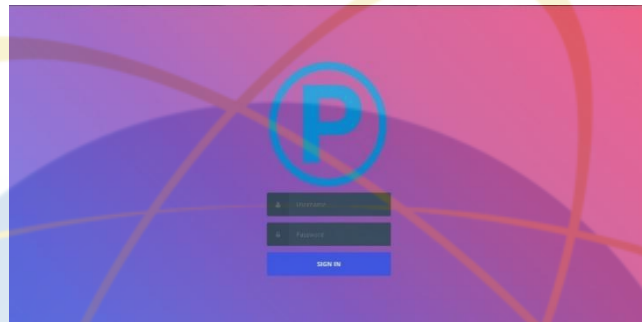
5.4.6 Halaman Menu Tips&Trik



Gambar 5. 6 Halaman Menu Tips & Trik

Menu ini merupakan fitur yang menampilkan tips&trik dan informasi dalam berkendara. Fitur ini memberikan informasi berkendara untuk user apabila user klik list beberapa informasi yang tersedia dan sudah diinput oleh admin.

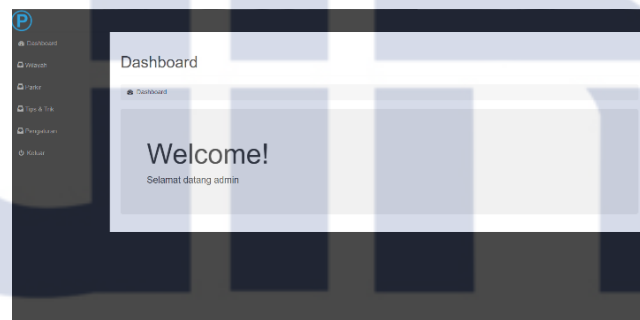
5.4.7 Halaman Login Admin



Gambar 5. 7 Halaman Login Admin

Menu Login ini merupakan fitur yang dapat digunakan oleh user. User diharuskan mengisi username dan password yang sesuai agar bisa masuk kedalam sistem.

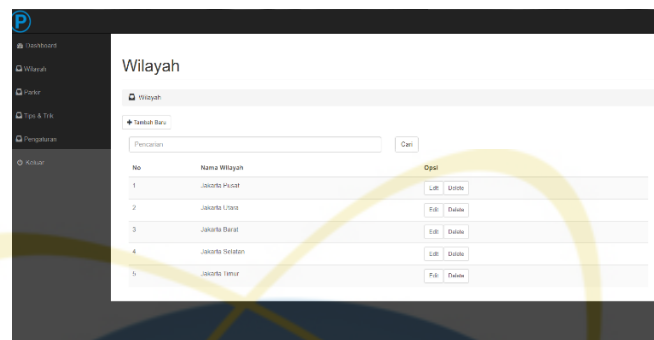
5.4.8 Halaman Dashboard Admin



Gambar 5. 8 Halaman Dashboard Admin

Menu ini adalah halaman dashboard untuk admin dari Aplikasi Informasi Tempat Parkir Terdekat Berbasis Android. Menu ini menampilkan beberapa fitur utama, yaitu fitur Wilayah, Parkir, Tips&Trik dan Logout.

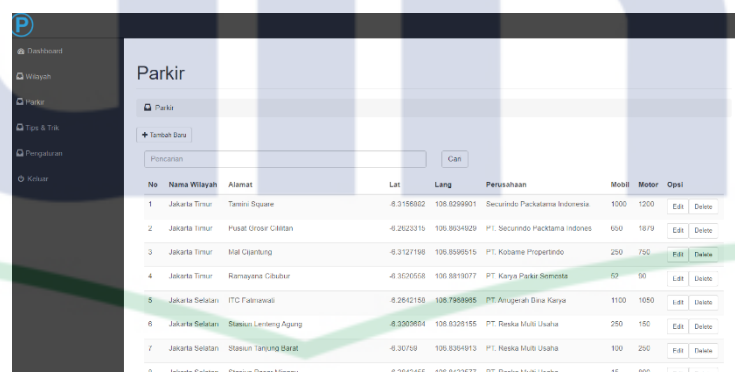
5.4.9 Halaman Menu Wilayah Admin



Gambar 5. 9 Halaman Menu Wilayah Admin

Merupakan halaman menu yang hanya dapat diakses oleh admin, menu ini berisikan mengenai data tempat parkir berdasarkan wilayah nya masing-masing (Jakarta). Admin dapat mencari data tempat parkir dengan memasukkan wilayah yang dicari dengan sesuai, admin juga dapat mengelola wilayah berupa input jika ada data yang ingin ditambah, edit jika ada data yang berubah dan delete jika ada data yang salah atau ingin dihapus. Dengan adanya menu ini, admin diberikan kemudahan dalam input data tempat parkir karena data yang di input sesuai dengan wilayah masing-masing.

5.4.10 Halaman Menu Parkir Admin

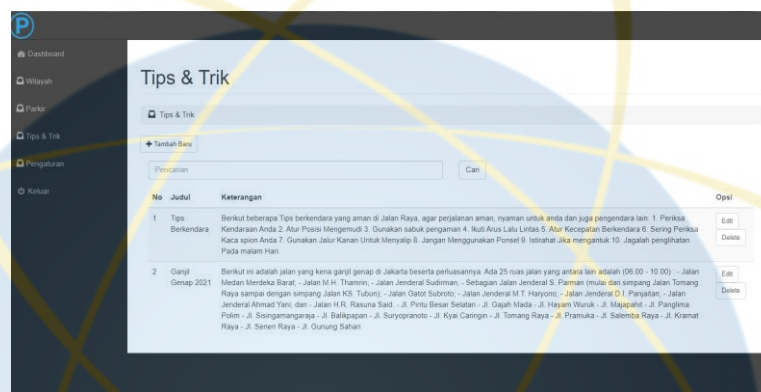


Gambar 5. 10 Halaman Menu Parkir Admin

Menu ini merupakan halaman menu yang berisikan mengenai data detail keseluruhan tempat parkir. Admin dapat mencari data tempat parkir dengan memasukkan wilayah atau nama tempat parkir yang dicari dengan sesuai, admin juga dapat mengelola data tempat parkir berupa input jika ada data yang

ingin ditambah, edit jika ada data yang berubah dan delete jika ada data yang salah atau ingin dihapus. Dengan adanya menu ini, admin diberikan kemudahan dalam input data tempat parkir dan tidak terkendala dengan crash dikarenakan data yang terlalu banyak.

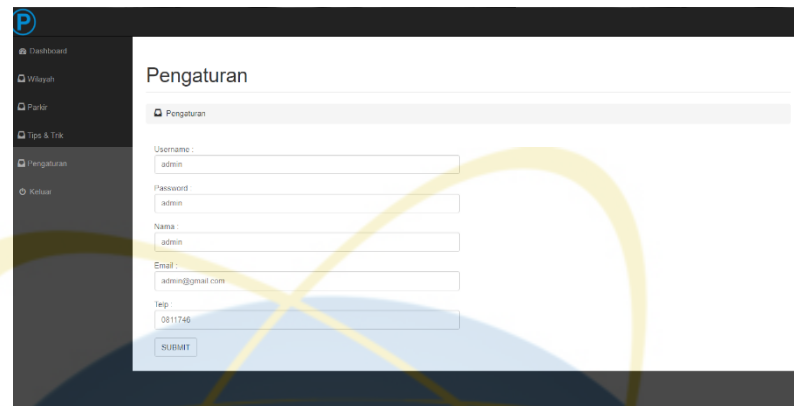
5.4.11 Halaman Menu Tips&Trik Admin



Gambar 5. 11 Halaman Menu Tips&Trik Admin

Menu ini merupakan halaman menu yang berisikan mengenai tips&trik dan informasi dalam berkendara yang hanya dapat diakses oleh admin. Admin juga dapat mengelola list informasi dalam berkendara berupa input jika ada data yang ingin ditambah, edit jika ada data yang berubah dan delete jika ada data yang salah atau ingin dihapus. Dengan adanya menu ini, admin diharapkan dapat mengelola dengan baik untuk memberikan kemudahan terhadap user yang ingin mencari tips&trik dan informasi dalam berkendara.

5.4.12 Halaman Menu Pengaturan Admin



The screenshot displays a web application interface. On the left is a dark sidebar with a menu containing: Dashboard, Wilayah, Pabrik, Tips & Trik, Pengaturan (highlighted), and Kotak. The main content area is titled 'Pengaturan' and contains a sub-section 'Pengaturan' with a list icon. Below this is a form with the following fields: Username (containing 'admin'), Password (containing 'admin'), Nama (containing 'admin'), Email (containing 'admin@gmail.com'), and Telp (containing '0811749'). A 'SUBMIT' button is located at the bottom of the form.

Gambar 5. 12 Halaman Menu Pengaturan Admin

Menu ini merupakan halaman menu yang berisikan mengenai pengaturan username dan password admin yang hanya dapat diakses oleh admin. Admin dapat mengelola username dan password dengan icon edit jika ada data yang berubah dan icon submit jika data tersebut sudah benar atau sesuai.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan rumusan masalah dan hasil penelitian dengan judul Pengembangan Aplikasi Informasi Tempat Parkir Terdekat Di Provinsi DKI Jakarta Menggunakan *Location Based Service* (LBS) dan *Google Maps API* (Studi Kasus: Unit Perparkiran Dinas Perhubungan Provinsi DKI Jakarta), maka dapat diambil kesimpulan bahwa sistem aplikasi informasi tempat parkir terdekat berbasis android ini dapat dikembangkan dengan menggunakan teknik *Location Based Service* (LBS) dan *Google Maps API*. Dapat memberikan kemudahan kepada *user* (pengendara roda dua ataupun roda empat) dalam mencari titik lokasi parkir terdekat sesuai dengan posisi *user* ataupun destinasi yang dituju.

Berdasarkan hasil pengujian menggunakan metode *user acceptance testing* (UAT) jumlah skor kesesuaian yang diperoleh dalam penelitian ini adalah total 252 poin. Sedangkan penilaian akhir untuk tingkat kesesuaian yang diharapkan dalam *user acceptance testing* ini adalah jumlah skor sesuai dibagi dengan jumlah skor sesuai ditambah jumlah skor tidak sesuai. Dari hasil pengujian, presentase tingkat kesesuaian sistem ini dengan responden adalah 100%. Penggunaan android studio dengan beberapa kelebihan juga memudahkan dalam penerapan LBS pada sistem ini, karena memiliki emulator yang cepat serta mendukung untuk digunakan semua perangkat *android*. Selain itu, penggunaan API juga dapat memudahkan dalam melakukan *request* dan *response* terhadap data yang telah tersimpan pada server.

6.2 Saran

Setelah dilakukan nya penelitian, penulis memiliki beberapa saran yang dapat diberikan, yaitu dalam pengembangan sistem berikutnya dapat di maksimalkannya menu *Register/Login* untuk kebutuhan data *user* dan di maksimalkannya fitur *Peta* (pencarian titik lokasi parkir terdekat dapat dicari berdasarkan jarak yang

diinginkan oleh *user*, sebagai contoh pencarian titik lokasi parkir terdekat dalam jarak 200m dari posisi *user*). Dapat diterapkannya pemantauan ketersediaan tempat parkir secara *real time* pada lokasi parkir dan fitur reservasi *online* (*booking*), dapat dikembangkan dengan *double platform* untuk pengguna sistem operasi *iOS* dan pengguna *website*.



DAFTAR PUSTAKA

- A. Mudzakir and R. Arifudin. (2017). Aplikasi Location Based Service Fasilitas Umum Berbasis Android. *Journal of Mathematics*. 4(6), 107-112.
- Abady, R. (2016). PENELITIAN DAN PENGEMBANGAN. 3(2), 60-63.
- Anwar, B., Jaya, H., Kusuma, P. I., Studi, P., & Komputer, S. (2018). Issn : 1978-6603 implementasi location based service berbasis android untuk mengetahui posisi user. 121–133.
- Ariati, N., & Kom, S. (2017). MODUL PERKULIAHAN – ANALISA SISTEM INFORMASI (A&PSI1). 1–49.
- Arifudin, A. M. and R. (2017). Aplikasi Location Based Service Fasilitas Umum Berbasis Android. *Math*. 12, 107-112.
- B. Anwar, H. Jaya, and P. I. K. (2017). Implementasi Locations Based Service Berbasis Android Untuk Mengetahui Posisi User. 13, 121–133.
- Budiman, E. (2018). Pemanfaatan Teknologi Location Based Service Dalam Mulawarman Berbasis Mobile. *Jurnal Ilmiah ILKOM*, 8(Desember), 137–144.
- Dian, I. U. (2019). Pelanggar Parkir Liar di DKI Jakarta Kian Meningkat, Kendaraan Tak Diambil Denda Bisa Bertambah. Retrieved from Minggu, 15 September website: <https://wartakota.tribunnews.com/2019/09/15/pelanggar-parkir-liar-di-dki-jakarta-kian-meningkat-kendaraan-tak-diambil-denda-bisa-bertambah>
- Gatra, S. (2020). Sebulan, Ada 197 Kendaraan Diderek Gara-gara Parkir Sembarangan di Jaksel. Retrieved from 04 Februari website: <https://megapolitan.kompas.com/read/2020/02/04/23032581/sebulan-ada-197-kendaraan-diderek-gara-gara-parkir-sembarangan-di-jaksel>
- Hakim, A. R., Harefa, K., & Widodo, B. (2019). Pengembangan Sistem Informasi Akademik Berbasis Android Menggunakan Flutter Di Politeknik. *SCAN - Jurnal Teknologi Informasi Dan Komunikasi*, 14(3), 27–32. <https://doi.org/10.33005/scan.v14i3.1684>
- Hakim, R. (2017). Teori & Konsep Aplikasi. 6–16.
- it-jurnal.com. (2018). biomas Chem Eng. *Biomass Chem Eng*, 49(23–6), 3–16.
- Izzaty, R. E., Astuti, B., & Cholimah, N. (2017). Pengembangan Sistem Informasi. *Angewandte Chemie International Edition*, 6(11), 951–952., 5–24.
- Jakarta, P. N. (2017). Rachmat Fajrin. *Jurnal Komputer Terapan*, 3(1), 33–40.

Retrieved from <http://jurnal.pcr.ac.id>

Jogiyanto. (2017). *Aplikasi & Pembelajaran*. 4(2), 8–19.

Kamus Besar Bahasa Indonesia. (2017). Jakarta: Pusat Bahasa, Departemen Pendidikan Nasional Indonesia. 201.

Kendall, J.E. & Kendall, K. . (2017). *ANALISIS DAN PERANCANGAN SISTEM: INDEKS*. JAKARTA. 1.

Literate, S., & Indonesia, J. I. (2020). View metadata, citation and similar papers at core.ac.uk. 274–282.

Majid, A. (2017). *Perencanaan Pembelajaran*, (Bandung: Remaja Rosdakarya, 2005). 24.

Nurhayati, S., & Ilmi, E. N. (2019). Sistem Aplikasi Pencarian Lokasi Parkir Terdekat Menggunakan Location Based Service Berbasis Android. *Komputika : Jurnal Sistem Komputer*, 6(2), 1–7. <https://doi.org/10.34010/komputika.v6i2.1677>

Priyadi, I., & Wellem Sagay, R. J. (2019). Perancangan Dan Implementasi Aplikasi Pasar Komoditas Andalan Desa (Pakades) Berbasis Mobile Android Untuk Petani Desa. *Jurnal SISTEM INFORMASI*, 1(2), 9–14. Retrieved from www.journal.ibmasmi.ac.id

Putri, M. P., & Effendi, H. (2018). Implementasi Metode Rapid Application Development Pada Website Service Guide “Waterfall Tour South Sumatera.” *Jurnal SISFOKOM*, 07(September), 130–136.

Rahmat, N. (2021). Efektivitas Kegiatan Bermanfaat Pemuda Mahasiswa Dalam Mengisi Masa Muda Dikehidupan Sehari-Hari. 1–9. Retrieved from <https://osf.io/preprints/socarxiv/x3ewz/>

Ramzi, M. (2017). ISSN 2338-137X Rancang Bangun Aplikasi Pengolahan Nilai Rapor dengan ISSN 2338-137X. Rancang Bangun Aplikasi Penjadwalan Mata Pelajaran Berbasis Web Pada SMK Negeri I Cerme, 5(7), 1–8.

Rohman, H. A., Radiyah, U., Maulana, A., Studi, P., Informatika, T., & Mandiri, S. N. (2019). 2191-5100-1-Pb. (November), 1–6.

Rusdina, R., Syafarina, G. A., & Amin, M. (2020). Prototype Aplikasi Banjar Berbasis Android Studio Sebagai Salah Satu Petunjuk Wisata Di Banjarmasin. *Technologia: Jurnal Ilmiah*, 11(1), 59. <https://doi.org/10.31602/tji.v11i1.2706>

Sani, A. A., Medi, A., Malik, I., & Ginting, M. (2018). Desain Tempat Parkir Sepeda Motor Bertingkat. *Austenit*, 10(April). Retrieved from <https://jurnal.polsri.ac.id/index.php/austenit/article/view/871/667>

Satyaputra, A. & E. M. A. (2018). Let's Build Your Android Apps with Android Studio. PT. Elex Media Komputindo, 2.

Späth, P. (2018). Pro Android with Kotlin. In Pro Android with Kotlin. <https://doi.org/10.1007/978-1-4842-3820-2>

Sudjiman, P. E., & Sudjiman, L. S. (2020). Analisis Sistem Informasi Manajemen Berbasis Komputer Dalam Proses Pengambilan Keputusan. *TelKa*, 8(2), 55–66. <https://doi.org/10.36342/teika.v8i2.2327>

Vandry Eko Haris Setiyanto, Cahya Rahmad, U. D. R. (2017). Pendeteksi Lokasi Parkir Mobil Menggunakan Metode Frame Differences Dan Static Template Matching. 8, 79–86.


Venti, C., Sholva, Y., & Nyoto, R. D. (2020). Sistem Manajemen Parkir Mobil On Street dan Off Street Berbasis Location Based Service (LBS) dan Google Maps API. 8(1), 48–57.

Yudhanto, Y., & Wijayanto, A. (2017). Mudah Membuat dan Berbisnis Aplikasi Android dengan Android Studio. Kompas Gramedia, 176.

Zaningsih, R. D. (2018). Pengembangan Aplikasi E-Modul Mobile Pembelajaran Matematika Berbasis Android Studio Pokok Bahasan Matriks. *Skripsi*, 1–129.

LAMPIRAN

Lampiran Surat Permohonan Wawancara



**KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI (UIN)
SYARIF HIDAYATULLAH JAKARTA
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. Ir. H. Juanda No. 95 Ciputat 15412 Indonesia
Telp. : (02-21) 7493606, 7493547, Fax. : (02-21) 7493315

Email : fat@uinjkt.ac.id
Website : fat.uinjkt.ac.id

Nomor : B-429E/F.9/TL.00/2/2020 Jakarta, 13 Februari 2020

Lampiran : -

Hal : **Permohonan Data/Wawancara**

Kepada
Yth. Unit Pengelola (UP) Perparkiran
Dinas Perhubungan Kota Depok

di
Tempat

Assalamu'alaikum Wr. Wb.
Dekan Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta
menerangkan bahwa :

Nama : Revo Yulianggara
Tempat/Tanggal : Jakarta / 21 Juli 1998
NIM : 11160910000017
Semester : 8
Program Studi : Teknik Informatika
Alamat : Jl. Pendidikan II No.44 Rt.05 Rw.09 Kel. Pekayon Kec.
Pasar Rebo, Kota Administrasi Jakarta Timur, DKI
Telp/Hp : 08986784469

adalah benar yang bersangkutan mahasiswa Fakultas Sains dan Teknologi
UIN Syarif Hidayatullah Jakarta yang sedang menyusun skripsi dengan judul:

***Pengembangan Prototipe Aplikasi Informasi Tempat Parkir di Kota Depok Menggunakan Metode
Location Based Service (LBS) dan Restfull API***

Untuk melengkapi bahan penulisan skripsi, dimohon kiranya Bapak/Ibu dapat menerima
yang bersangkutan untuk wawancara serta memperoleh data guna penulisan skripsi
dimaksud.

Atas kerjasama dan bantuannya, kami ucapkan terima kasih.
Wassalamu'alaikum Wr. Wb.

a.n. Dekan
Wakil Dekan Bidang Akademik


Nashrudin Hakiem, S.Si, MT, Ph.D
NIP. 19710608 200501 1 005

Lampiran 3. List Pertanyaan Wawancara 1

Tanggal Wawancara :

Narasumber : 5 Responden Expert di bidangnya (Dinas Perhubungan)

Jabatan : Manajer Unit Perparkiran

Tempat : Unit Perparkiran Dinas Perhubungan Provinsi DKI Jakarta

1. Penulis : Apakah Dishub Provinsi DKI Jakarta memiliki Data Lahan Parkir?

2. Penulis : Jika Iya, bagaimana kelengkapan data lahan parkirnya?

1. Detail luas tempat parkir?

2. Volume kendaraan Roda 2 dan Roda 4?

3. Tarif kendaraan Roda 2 dan Roda 4?

4. Apakah terdapat lokasi titik koordinat bujur dan lintang berdasarkan data setiap tempat parkir?

3. Penulis : Apakah setiap lahan parkir yang berada di wilayah Provinsi DKI Jakarta harus tercatat oleh Dishub?

4. Penulis : Kapankah data lahan parkir terakhir kali diperbaharui atau di update?

5. Penulis : Bagaimana sistem pencarian tempat parkir yang berlaku di Dishub? sudah menerapkan sistem aplikasi milik sendiri atau masih menggunakan software bantuan lainnya?

Lampiran 4. List Pertanyaan Wawancara 2

Tanggal Wawancara :

Narasumber : Bimo, Eka, Dzaky

Jabatan : Tim Developer Dishub

Tempat : Unit Perparkiran Dinas Perhubungan Provinsi DKI Jakarta

1. Penulis : Apakah sistem aplikasi yang berjalan sudah cukup baik? baik dari segi interface maupun fitur-fitur yang tersedia.
2. Penulis : Siapa saja target user aplikasi dari Dishub untuk menggunakan aplikasi pencarian tempat parkir?
3. Penulis : Dari sistem aplikasi pencarian tempat parkir milik Dishub yang sudah berjalan saat ini, apakah user aplikasi sudah merasa cukup dengan fitur yang ada?
4. Penulis : Kapan terakhir kali admin yang mengelola aplikasi tersebut melakukan perbaharuan atau update untuk data lahan parkir yang tersedia?
5. Penulis : Bagaimanakah tahap maintenance dari sistem aplikasi ini, berjalan secara terjadwal atau hanya berdasarkan kebutuhan sistem saja?