APLIKASI PENCARIAN PARKIR JAKARTA BERBASIS ANDROID MENGGUNAKAN *RESTFUL API*

Muhammad Adam Dzulqarnain
UIN Sunan Gunung Djati Bandung
1157050100@student.uinsgd.ac.id

Muhammad Luthfi Aziz
UIN Sunan Gunung Djati Bandung
1157050107@student.uinsgd.ac.id

Muhammad Fauzi Rachman
UIN Sunan Gunung Djati Bandung
fauzirachman.fr@gmail.com

Aldy Rialdy Atmadja
Sekolah Tinggi Teknologi Garut
aldyrialdyatmadja@sttgarut.ac.id

ABSTRACT

Jakarta is one of the metropolitan cities in Indonesia. The increasing number of vehicles entering Jakarta has resulted in the difficult to find parking area. Moreover, the population density occurred in there also resulted in the difficulty of finding parking area. Each parking area have its own capacity, the driver must directly check the state of the parking area.According to a survey conducted by VIVA, it takes 21 minutes to find parking in Jakarta (Viva, 2017). Based on the background of the problem, it can be conclude, the focus of the research is how to implement RESTFul API on an android application , so it will be ease to find a parking area in Jakarta. The system development method uses Scrum, the implementation results in the form of an android-based application using the RESTFul API.

Keyword: *Application, Android, RESTFul, API*

ABSTRAK

Jakarta merupakan salah satu kota metropolitan di Indonesia. Semakin banyaknya kendaraan yang masuk ke Kota Jakarta mengakibatkan semakin meningkatnya kebutuhan lahan parkir. Karena akibat kepadatan penduduk yang terjadi di Jakarta mengakibatkan sulitnya mencari lahan parkir. Setiap area parkir pasti memiliki kapasitasnya tersendiri, pengendara harus meninjau secara langsung keadaan area parkir tersebut. Menurut survey yang dilakukan oleh viva butuh 21 menit untuk mencari lahan parkir di Jakarta (viva, 2017).Berdasarkan latar belakang permasalahan tersebut, maka dapat dirumuskan, fokus pada penelitian adalah, bagaimana implementsasi *RESTFul API* pada aplikasi android agar dapat memudahkan pencarian parkir di Jakarta.Metodologi pengembangan sistem menggunakan *Scrum*, hasil implementasi berupa aplikasi berbasis android menggunakan *RESTFul API*.

Kata Kunci: Aplikasi, Android, *RESTFul, API*

1. PENDAHULUAN

1.1 Latar Belakang

Jakarta merupakan salah satu kota metropolitan di Indonesia. Sebagai ibu kota Negara, Kota Jakarta menjadi kota yang terkenal dengan salah satu kota termacet di Indonesia. Hal tersebut membuat jakarta menjadi jantung perekonomian dan banyak menarik orang untuk datang ke Jakarta, Semakin banyaknya yang datang ke Kota Jakarta mengakibatkan semakin bertambah banyak juga kendaraan yang masuk ke Kota Jakarta.

Semakin banyaknya kendaraan yang masuk ke Kota Jakarta mengakibatkan semakin meningkatnya kebutuhan lahan parkir. Karena akibat kepadatan penduduk yang terjadi di Jakarta mengakibatkan sulitnya mencari lahan parkir. Setiap area parkir pasti memiliki kapasitasnya tersendiri, pengendara harus meninjau secara langsung keadaan area parkir tersebut. Menurut survey yang dilakukan oleh viva butuh 21 menit untuk mencari lahan parkir di Jakarta(viva, 2017).

Permasalahan tersebut dapat di pecahkan dengan kemajuan tekhnologi aplikasi berbasis android dengan menggunakan API, maka penulis tertarik untuk mengambil tema penelitian yang berjudul "APLIKASI PENCARIAN PARKIR JAKARTA BERBASIS ANDROID MENGGUNAKAN RESTFUL API".

1.2 Rumusan dan Batasan Masalah

Berdasarkan latar belakang permasalahan tersebut, maka dapat dirumuskan, fokus pada penelitian ini adalah: Bagaimana implementasi RESTFul API pada aplikasi android agar dapat memudahkan pencarian parkir di Jakarta.

Agar masalah yang diteliti dapat sesuai dengan pokok permasalahan dan dapat dipahami maka perlu ada batasan masalah agar aplikasi yang dibangun sesuai dengan tujuan dan tidak menyimpang serta lebih terarah. Batasan masalah dalam penelitian ini sebagai berikut :

1) Aplikasi dibangun pada platform android
2) Aplikasi yang dikembangkan membutuhkan koneksi internet untuk mengakses API
3) Pembuat API didaptkan dari data.jakatra.go.id
4) Aplikasi hanya menampilkan sebagian data dari lokasi parkir yang ada di daerah jakarta.

1.3 Maksud dan Tujuan Penelitian

Maksud dari penelitian ini adalah untuk membangun aplikasi pencarian parkir jakarta berbasis android menggunakan *RESTFul API.* Sedangkan tujuan penelitian adalah untuk

1) Memberikan informasi tempat parkir di Jakarta
2) Meningkatkan efekrifitas pencarian parkir di Jakarta

1.4 Tijauan Pustaka

1.4.1 RESTFul API

REST (*REpresentational State Transfer*) merupakan standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web. Umumnya menggunakan HTTP (*Hypertext Transfer Protocol*) sebagai protocol untuk komunikasi data. REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000 (Feridi, 2016).

Pada arsitektur REST, REST server menyediakan *resources* (sumber daya/data) dan REST client mengakses dan menampilkan *resource* tersebut untuk penggunaan selanjutnya. Setiap *resource* diidentifikasi oleh URIs (*Universal Resource Identifier*s) atau global ID. *Resource* tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML (Feridi, 2016).

Metode HTTP yang umum digunakan dalam arsitektur berbasis REST adalah 1) GET, menyediakan hanya akses baca pada *resource*, 2) PUT, digunakan untuk menciptakan *resource* baru, 3) DELETE**,** digunakan untuk menghapus *resource*, 4) POST**,** digunakan untuk memperbarui *resource* yang ada atau membuat *resource* baru, 5) OPTIONS**,** digunakan untuk mendapatkan operasi yang disupport pada *resource*(Feridi, 2016).

1.4.2 Sistem Operasi Android

Android adalah sistem operasi yang dikeluarkan oleh Google. Android pada awalnya dikhususkan untuk sistem operasi smartphone dan tablet. Android juga mempunyai store dengan lebih dari 2 miliar pengguna aktif per Januari 2018 (Imaduddin & Permana, 2018).

Android Software Development Kit (SDK) merupakan *kit* yang bisa digunakan oleh para *developer* untuk mengembangkan aplikasi berbasis Android. Di dalamnya, terdapat beberapa tools seperti *debugger, software libraries, emulator,* dokumentasi, *sample code,* dan tutorial (Imaduddin & Permana, 2018).

Bahasa pemrograman yang sering digunakan untuk mengembangkan aplikasi Android adalah Java. Namun ada beberapa bahasa lainnya yang dapat digunakan, seperti C++, dan GO. Pada IO 2017, Google juga meresmikan Kotlin sebagai tambahan bahasa resmi (Imaduddin & Permana, 2018).

Android Studio adalah *Integrated Development Environment (IDE)* untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA . Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas (Android Developers , 2017).

1.4.3 Pemrograman Berorientasi Objek

Dalam pemrograman berorientasi objek , setiap objek akan memiliki data (sifat, berupa variabel maupun konstanta) dan method (perilaku atau kemampuan melakukan sesuatu, berupa fungsi). Jadi, objek dapat didefinisikan sebagai suatu entitas yang memiliki data dan method (Raharjo, Heryanto, & Haryono, 2012).

Semua bahasa pemrograman berorientasi objek menyediakan mekanisme untuk membantu mengimplementasikan model berorientasi objek. Prinsip tersebut adalah *encapsulation, inheritance,* dan *polymorphism* (Schildt, 2014).

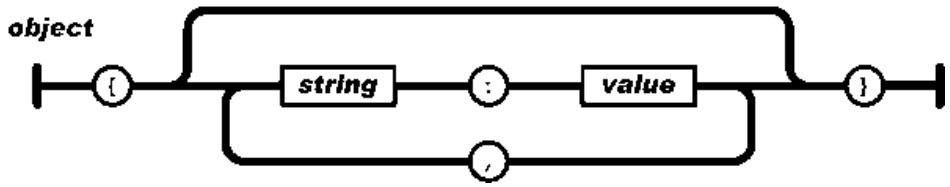
1.4.4 Bahasa Pemrograman Kotlin

Pada Google I/O 2017, Kotlin diumumkan sebagai bahasa pemrograman yang termasuk dalam bahasa kelas satu (First class) yang didukung untuk pembuatan aplikasi Android, selain Java dan C++. Kotlin adalah bahasa pemrograman yang dibuat oleh JetBrains. Google juga akan memastikan bahwa semua fitur baru di Android, framework, IDE dan keseluruhan library, akan dapat bekerja dan terintegrasi baik dengan bahasa pemrograman Kotlin serta interopable dengan fungsi-fungsi Java yang telah ada sehingga memungkinkan para(Rohman & Toro, 2018).

1.4.5 JSON

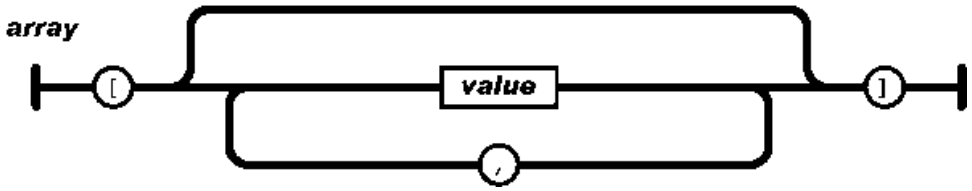
JSON (Java Script Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data (json.org, 2017). JSON menggunakan bentuk sebagai berikut:

1) Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).



Gambar 1 Struktur JSON Objek (json.org, 2017)

2) Larik (*array*) adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh ,

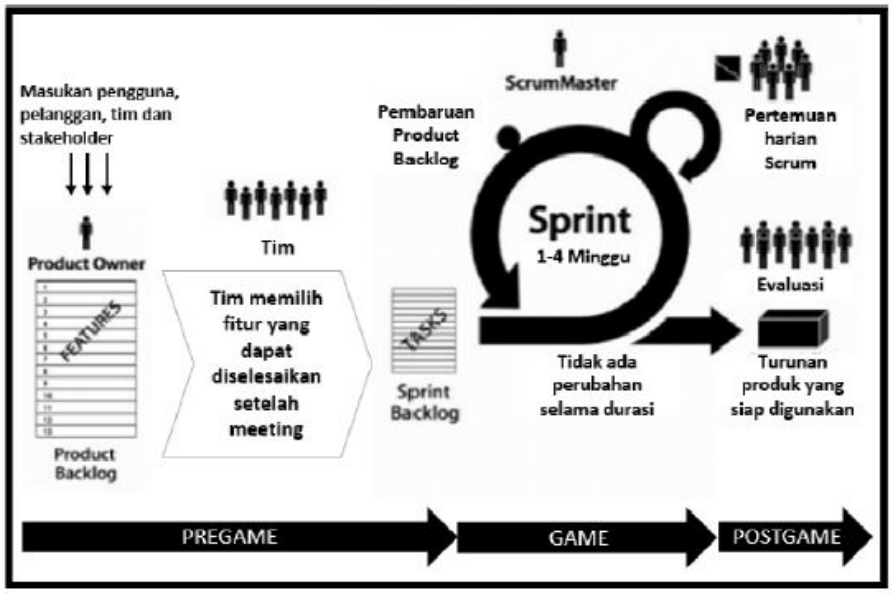


(koma).

Gambar 2 Struktur JSON Array (json.org, 2017)

2. METODOLOGI

Untuk pembuatan aplikasi ini menggunakan metode *scrum.Scrum* menggunakan pendekatan berkala (iterative) dan bertahap (incremental) untuk meningkatkan prediktibilitas dan mengendalikan risiko (Schwaber & Sutherland, The Scrum GuideTM The Definitive Guide to Scrum: The Rules of the Game, 2016).



Gambar 3 Tahapan dan pihak yang terlibat dalam metode *Scrum* (Schwaber, Agile Project Management with Scrum, 2004)

Berdasarkan gambar 3 metode *scrum* dapat dijelaskan sebagai berikut.

1) *Sprint*

Jantung dari *Scrum* adalah Sprint, yaitu sebuah batasan waktu dengan durasi satu bulan atau kurang, dimana terdapat proses pembuatan Increment yang "Selesai", dapat digunakan dan berpotensi untuk dirilis. Sprint memiliki durasi yang konsisten sepanjang daur

hidup pengembangan produk. Sprint yang baru langsung dimulai setelah Sprint sebelumnya selesai (Schwaber & Sutherland, The Scrum GuideTM The Definitive Guide to Scrum: The Rules of the Game, 2016).

2) *Daily Scrum*

Daily Scrum adalah acara untuk *Development Team* yang memiliki batasan waktu 15 menit. Acara ini dilakukan setiap hari selama *Sprint* berlangsung. Di acara ini, *Development Team* membuat rencana kerja untuk 24 jam ke depan. Acara ini mengoptimalkan kolaborasi dan performa dari tim dengan melakukan inspeksi pada pekerjaan yang dilakukan semenjak *Daily Scrum* sebelumnya dan melakukan prakiraan terhadap pekerjaan selanjutnya di dalam Sprint. *Daily Scrum* dilakukan di waktu dan tempat yang sama setiap harinya untuk mengurangi kompleksitas (Schwaber & Sutherland, The Scrum GuideTM The Definitive Guide to Scrum: The Rules of the Game, 2016).

3) *Sprint Review*

Sprint Review diselenggarakan di akhir *Sprint* untuk menginspeksi *Increment* dan mengadaptasi *Product Backlog* bila diperlukan. Pada saat *Sprint Review*, *Scrum Team* dan pemegang kepentingan berkolaborasi untuk meninjau apa yang sudah diselesaikan di Sprint. Berdasarkan hasil tinjauan tersebut dan perubahan terhadap *Product Backlog* di dalam *Sprint*, hadirin berkolaborasi untuk menentukan pekerjaan selanjutnya yang dapat dilakukan untuk mengoptimalkan nilai bisnis. Ini adalah pertemuan informal, bukan pertemuan laporan status, dan presentasi Increment dilakukan guna mendapatkan umpan balik dan mengembangkan kemampuan kolaborasi (Schwaber & Sutherland, The Scrum GuideTM The Definitive Guide to Scrum: The Rules of the Game, 2016).

4) *Sprint Retrospective*

Sprint Retrospective adalah sebuah kesempatan bagi *Scrum Team* untuk menginspeksi dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di *Sprint* berikutnya (Schwaber & Sutherland, The Scrum GuideTM The Definitive Guide to Scrum: The Rules of the Game, 2016).

3. ANALISA DAN PERANCANGAN SISTEM

3.1 Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan penggambaran kebutuhan-kebutuhan utama yang dibutuhkan pada aplikasi yang akan dibangun, berikut kebutuhan fungsional aplikasi yang akan dibangun.

1) Aplikasi dapat menampilkan *list* parkir di Jakarta
2) Aplikasi dapat menampilkan *list* parkir di Jakarta berdasarkan kategori parkiran
3) Aplikasi dapat memfasilitasi penambahan favorit parkiran pengguna
4) Aplikasi dapat memfasilitasi pencarian parkiran di Jakarta
5) Aplikasi dapat menunjukan arah ke parkiran

3.2 Kebutuhan Non Fungsional

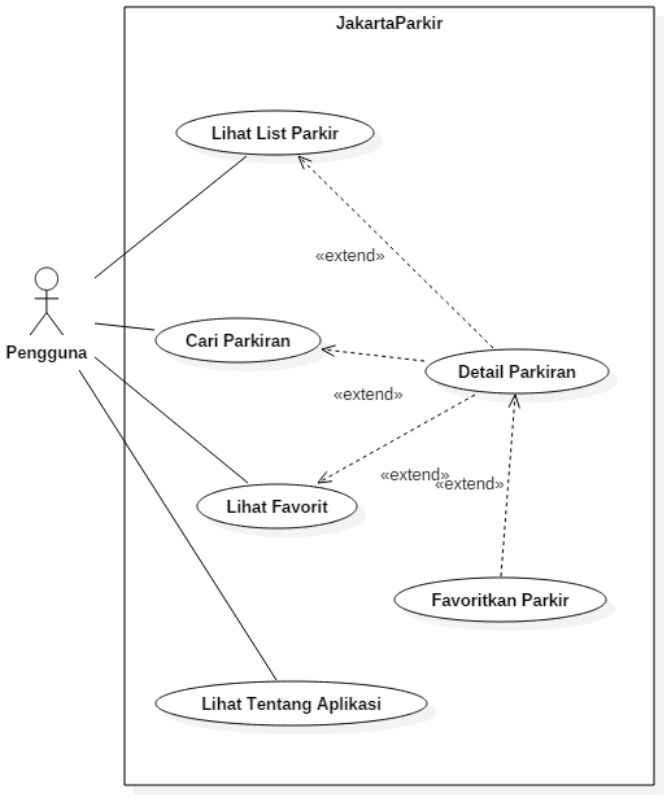
Selain kebutuhan fungsional kebutuhan non fungsional juga dibutuhkan untuk berjalannya sistem, dan kebutuhan non fungsional tersebut adalah kebutuhan sistem meliputi performa, kelengkapan operasi pada fungsi-fungsi yang ada, serta kesesuaian dengan lingkungan penggunanya. Rumusan kebutuhan non fungsional meliputi.

1) Kebutuhan Operasional
    a. Dapat dijalankan di sistem operasi android
    b. Beroperasi selama 24 jam perhari dan 7 hari perminggu

2) Kebutuhan Perangkat Keras
    a. Komputer dengan spesifikasi minimal *Intel Core i3, RAM4Gb, hardisk 500Gb.*
    b. *Smarthpone* dengan spesifikasi minimal sistem operasi android 4.0 (*Ice Cream Sandwitch), RAM* 500 Mb, Memori 2 Gb.
3) Kebutuhan Perangkat Lunak
    a. Aplikasi *API Checker* (Postman)
    b. *IDE Software Android Studio 3.0*
    c. Aplikasi pemodelan *Star UML*

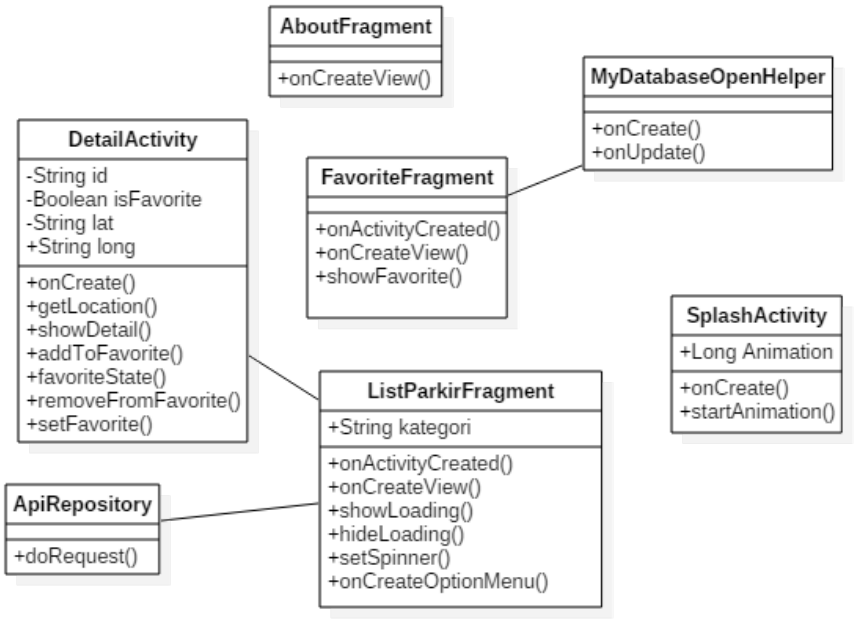
3.3 Use Case

Use Case Model berfokus pada faktor-faktor penentu keberhasilan sistem, dalam hal fungsi atau fitur yang perlu berinteraksi dengan pengguna. Dengan berfokus pada fitur sistem, Anda membuat serangkaian slot konseptual di mana semua persyaratan yang sangat bervariasi dapat ditempatkan (Pender, 2002)



Gambar 4 ***Use Case Diagram***

3.4 Class Diagram

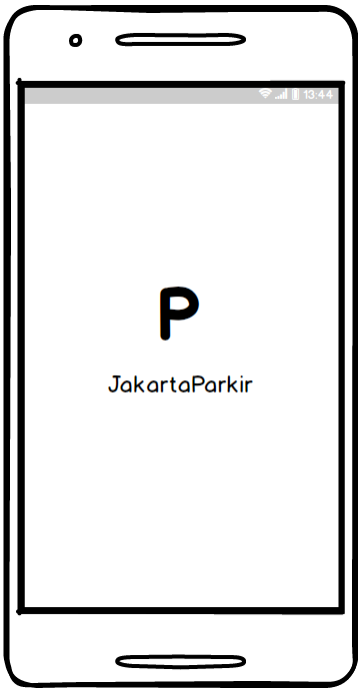


Gambar 5 *Class Diagram*

3.5 Desain

3.5.1 Tampilan Halaman Splash Screen

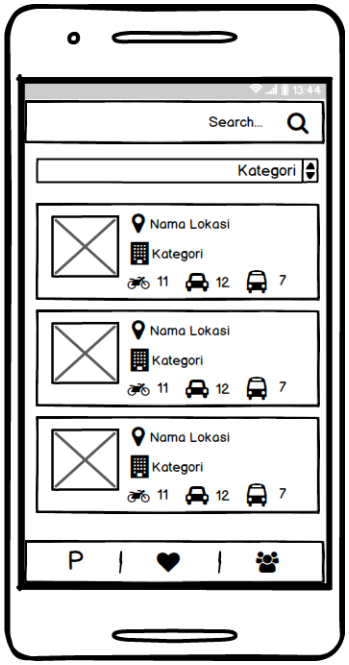
Antar muka ini logo JakartaParkir ketika aplikasi pertama kali dibuka.



Gambar 6 Halaman *Splash Screen*

3.5.2 Tampilan Halaman List Parkir

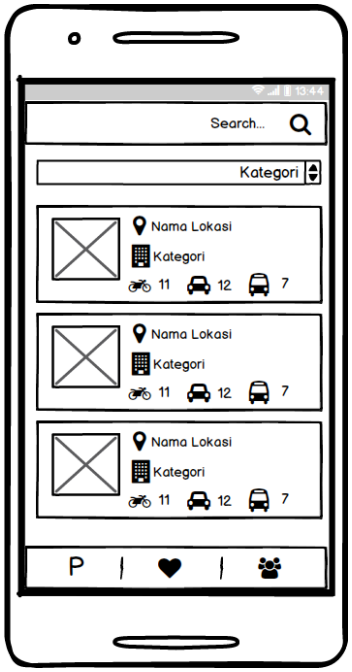
Antar muka ini menampilkan list daftar parkiran di Jakarta.



Gambar 7 Halaman *List* Parkir

3.5.3 Tampilan Halaman List Favorit

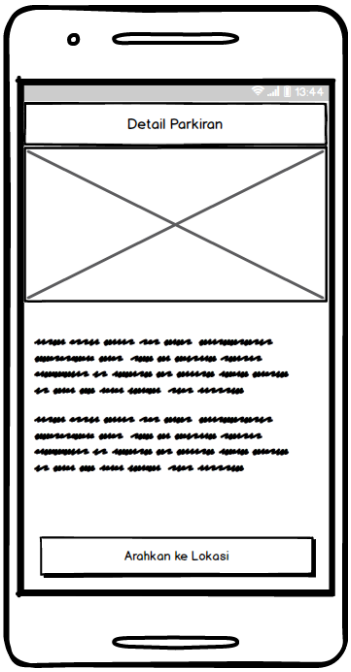
Antar muka ini menampilkan list daftar parkiran di Jakarta yang sudah difavoritkan pengguna.



Gambar 8 Halaman *List* Parkir

3.5.4 Tampilan Halaman Detail Parkir

Antar muka ini menampilkan detail parkiran yang berisi informasi dan petunjuk arah menuju parkiran.



Gambar 9 Halaman DetailParkir

3.6 Implementasi

RESTFul API yang dibangun untuk aplikasi Pencarian Parkir di Jakarta dijelaskan pada tabel 1.

Tabel 1 *Endpoint API*

No	URL	Method	Keterangan
1	/parkir/	Get	Menampilkan semua informasi tempat parkir di Jakarta
2	/parkir/detail/{id}	Get	Menampilkan detail informasi suatu tempat parkir berdasarika parameter id
3	/parkir/kategori/{kategori}	Get	Menampilkan semua informasi tempat parkir berdasarkan kategori tempat parkir
4	/parkir/search/	Get	Menampilkan semua informasi tempat parkir berdasarkan hasil pencarian alamat atau kategori parkir

Dari RESTFul API tersebut diimplementasikan pada aplikasi berbasis android, dengan beberapa fungsi berikut.

1) Menampilkan semua informasi parkir di Jakarta
2) Menampilkan informasi parkir berdasarkan kategori tempat parkir
3) Menampilkan kuota motor, mobil, dan bus atau truk pada tempat parkir
4) Memfavoritkan parkiran
5) Pencarian parkiran
6) Petunjuk arah menuju parkiran

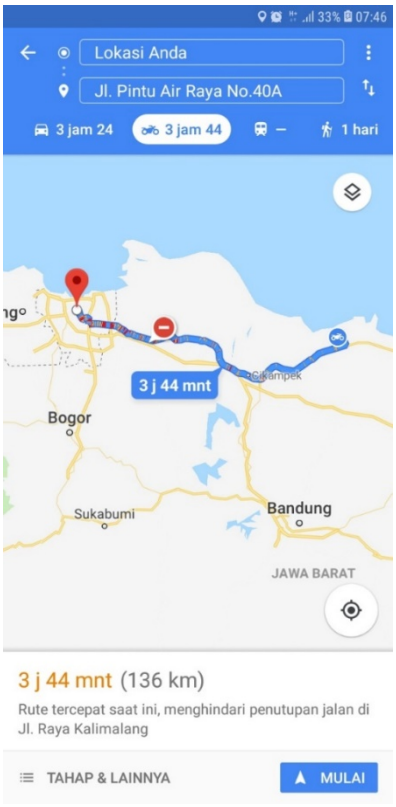
Implementasi antarmuka merupakan proses tahapan yang dilakukan pada desain yang diimplementasikan menjadi bentuk aplikasi yang akan dipakai pengguna.

Gambar 10 Implementasi *Splash Screen*

Gambar 11 Implementasi *List* Parkir

Gambar 12 Implementasi List FavoritParkir

Gambar 13Implementasi DetailParkir



Gambar 14 Implementasi Petunjuk Arah Parkir

4. PENUTUP

4.1 Kesimpulan

Berikut kesimpulan yang dapat disimpulkan berdasarkan analisis, perancangan serta implementasi terhadap program yang telah dibangun, yaitu :

1) Sistem yang dibangun dapat menampilkan list lokasi parkir dijakarta
2) Sistem mampu menunjukan detail Tempat parkir
3) Sistem mampu mengarahkan user ke tempat parkir yang akan dituju
4) Sistem mampu menyimpan tempat parkir favorite yang dipilih user
5) Sistem mampu melakukan pencarian tempat parkir
6) Sistem mampu menampilkan tempat parkir berdasarkan kategori yang tersedia.

4.2 Saran

Adapun saran penulis dalam pengembangan aplikasi ini yakni :

1) Menambahkan data list parkir yang tersedia
2) Memperluas daerah wilayah tempat parkir
3) Menambahkan fitur untuk melakukan booking parkir
4) Menampilkan kuota parkir yang tersedia
5) Menambahkan fitur untuk menghitung biaya parkir di tempat parkir yang akan dipilih

DAFTAR PUSTAKA

Android Developers . (2017, Agustus 7). (Google) Dipetik Oktober 7, 2018, dari https://developer.android.com/studio/intro/

Feridi. (2016, Februari 25). *codepolitan*. (codepolitan) Dipetik Oktober 7, 2018, dari https://www.codepolitan.com/mengenal-restful-web-services

Imaduddin, A., & Permana, S. (2018). *Menjadi Android Developer Expert.* Bandung: PT Presentologics.

json.org. (2017, Desember). Dipetik Oktober 7, 2018, dari https://www.json.org/json-id.html

Pender, T. (2002). *UML Weekend Crash Course.* Indianapolis: Wiley Publishing, Inc.

Raharjo, B., Heryanto, I., & Haryono, A. (2012). *Mudah Belajar Java.* Bandung: Informatika Bandung.

Rohman, N., & Toro, R. (2018). *Kotlin Android Developer Expert.* Bandung: PT. Presentologic.

Schildt, H. (2014). *Java The Complete Reference Ninth Edition.* New York: McGrawHil.

Schwaber, K. (2004). *Agile Project Management with Scrum.* Redmond: Microsoft Press.

Schwaber, K., & Sutherland, J. (2016, Juli). *The Scrum GuideTM The Definitive Guide to Scrum: The Rules of the Game*. (scrumguides) Dipetik Oktober 7, 2018, dari https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf

viva. (2017, 10 27). *Butuh 21 Menit Mencari Tempat Parkir di Jakarta*. (viva) Dipetik 01 09, 2019, dari https://www.viva.co.id/otomotif/mobil/971400-survei-butuh-21-menit-mencari-tempat-parkir-di-jakarta