

# PERANCANGAN APLIKASI ANTARMUKA *SMART OPEN PARKING* BERBASIS *INTERNET OF THINGS (IoT)* PADA PERANGKAT ANDROID

Ardana Adwindea<sup>\*</sup>), Aghus Sofwan dan Muhammad Arfan

Departemen Teknik Elektro, Universitas Diponegoro  
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

<sup>1)</sup>E-mail: danadwnd@gmail.com

## Abstrak

Dengan pertumbuhan penduduk dan pembangunan ekonomi, jumlah kendaraan di jalan meningkat dari hari ke hari. Parkir menjadi salah satu masalah utama untuk kota, dan menjadi sangat mahal. Karena itu, tempat parkir menjadi terbatas di kota-kota besar. Dengan demikian, cara-cara parkir yang cerdas, inovatif dan efisien harus dibangun untuk mengakomodasi tuntutan parkir secara efisien. Untuk membantu pengemudi, dibutuhkan aplikasi perangkat bergerak yang memungkinkan pengemudi untuk melihat di mana tempat parkir kosong tanpa harus mendatangi tempat parkir. Penelitian ini bertujuan untuk membuat aplikasi parkir terbuka pintar berbasis internet of things pada perangkat android. Aplikasi ini dibuat menggunakan Android Studio. Perancangan aplikasi menggunakan Unified Modelling Language berupa diagram use case, diagram aktivitas, dan diagram kelas. Hasil pengujian menunjukkan setiap fitur dapat berjalan dengan baik.

*Kata kunci: Smart Open Parking, Internet of Things, Android*

## Abstract

With population growth and economic development, the number of vehicles on the road is increasing day by day. Parking became one of the main problems for the city, and became very expensive. Therefore, parking space is limited in big cities. Thus, smart, innovative and efficient parking methods must be built to accommodate parking demands efficiently. To help the driver, a mobile application is needed that allows the driver to see where the parking lot is empty without having to go to the parking lot. This study aims to create an internet of things-based smart open parking application on Android devices. This app was created using Android Studio. The design of this application using Unified Modeling Language, such as case diagrams, activity diagrams, and class diagrams. The test results show each feature works well.

*Keywords: Smart Open Parking, Internet of Things, Android*

## 1. Pendahuluan

Dengan pertumbuhan penduduk dan pembangunan ekonomi, jumlah kendaraan di jalan meningkat dari hari ke hari. Parkir menjadi salah satu masalah utama untuk kota, dan menjadi sangat mahal. Karena itu, parkir terbatas di kota-kota besar termasuk universitas dan perusahaan di seluruh dunia. Misalnya, menemukan tempat parkir selama acara besar (seperti selama hari pertandingan atau hari kelulusan) sangat menyulitkan. Pengemudi kehilangan jam kerja dan mengonsumsi bahan bakar saat mencari tempat parkir kosong. Ini bisa sangat membuat frustrasi bagi pengemudi untuk mencari tempat parkir di tempat parkir. Frustrasi ini dapat terjadi karena tempat parkir sulit ditemukan atau karena pengemudi lain telah mengambil tempat parkir sebelum mereka mencapai tempat itu.

Dengan demikian, cara-cara parkir yang cerdas, inovatif dan efisien harus dibangun untuk mengakomodasi tuntutan parkir secara efisien. Untuk membantu pengemudi, dibutuhkan aplikasi perangkat bergerak yang memungkinkan pengemudi untuk melihat di mana tempat parkir kosong tanpa harus mendatangi tempat parkir.

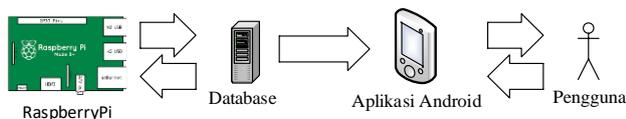
Pada penelitian sebelumnya, telah terdapat beberapa jurnal yang juga memanfaatkan aplikasi Android sebagai aplikasi Smart Parking. Di antaranya yaitu "IoT based Smart Parking System"[1], "Android Based Smart Parking System Using Slot Allocation & Reservations"[2], "A Smart Parking System Based on IoT Protocols and Emerging Enabling Technologies"[3], "Automatic Smart Parking System using Internet of Things (IOT)"[4], "Sistem Parkir Cerdas"[5].

Penelitian ini bertujuan untuk merancang dan membuat aplikasi *Smart Open Parking* berbasis *Internet of Things* pada perangkat Android yang dapat menampilkan informasi lokasi tempat parkir serta jumlah kapasitas dan jumlah kendaraan yang ada di tempat parkir. Aplikasi ini juga sebagai alat untuk *monitoring* status sumber daya yang digunakan oleh perangkat keras *Smart Open Parking* oleh pengelola *Smart Open Parking*.

## 2. Metode

### 2.1. Deskripsi Sistem

Dalam penelitian ini dirancang aplikasi *smart open parking* dengan sistem operasi yang berbasis Android. Aplikasi ini dirancang menggunakan bantuan IDE Android Studio dengan menggunakan bahasa XML untuk pembuatan antarmuka dan Java untuk proses logikanya[6].



Gambar 1. Desain Aplikasi

Pada Gambar 1 dapat dilihat alur ketika menggunakan aplikasi ini, yaitu ketika user mengakses aplikasi ini dan menggunakan fitur yang ada pada aplikasi. Aplikasi akan mengambil data dari basis data secara *real-time*. Hasil pengambilan data tersebut kemudian ditampilkan pada antarmuka aplikasi.

### 2.2 Analisa Kebutuhan

#### 2.2.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan gambaran mengenai fungsi-fungsi yang dapat dilakukan oleh sistem. Kebutuhan fungsional sistem meliputi:

- Pengguna dapat mengetahui lokasi tempat parkir.
- Pengguna dapat mengetahui kapasitas dan jumlah tersedia tempat parkir.
- Pengguna dapat mengetahui status sumber daya perangkat parkir pintar.
- Sistem dapat memberikan rute jarak terdekat dari posisi pengguna ke penanda parkir yang dipilih.

#### 2.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan sistem meliputi kinerja, kelengkapan operasi pada fungsi-fungsi yang ada, serta kesesuaian dengan lingkungan penggunaannya. Kebutuhan non-fungsional ini melingkupi beberapa kebutuhan yang mendukung kebutuhan fungsional, rumusan kebutuhan fungsional meliputi:

#### a. Kebutuhan Operasional

- Aplikasi dapat didistribusikan melalui internet dan harus dapat terinstall pada perangkat android.
- Sistem hanya dapat diakses melalui *file* format .apk yang telah terinstal di perangkat android.
- Antarmuka pengguna pada aplikasi dibuat dengan sederhana untuk memudahkan pengguna dari berbagai usia.
- Aplikasi dapat dijalankan pada minimal Android versi 4.3 (API 16).
- Sistem ini dibangun dengan menggunakan bahasa pemrograman Java.

#### b. Performasi Sistem

Sistem yang dibangun merupakan aplikasi yang berjalan pada lingkungan perangkat bergerak. Terdapat beberapa keterbatasan yang ditemui pada perangkat bergerak meskipun sistem operasi yang digunakan adalah Android. Oleh karena itu perlu diperhatikan guna menjadi acuan dalam pengembangan sistem, diantaranya:

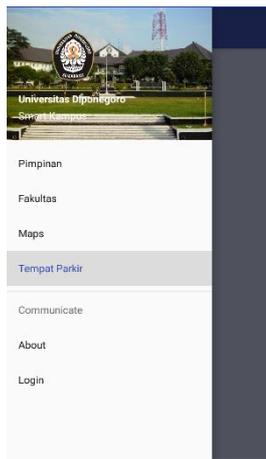
- Sumber daya baterai yang secara efektif hanya mampu bertahan selama kurang lebih 6 jam dengan penggunaan secara terus menerus dan kurang lebih 200 jam dalam keadaan *standby*.
- Tampilan aplikasi antarmuka yang berpengaruh terhadap waktu tunggu sehingga aplikasi benar-benar siap digunakan, semakin banyak komponen yang digunakan akan semakin lama pula waktu tunggu yang dibutuhkan.

Dari keterbatasan pada perangkat Android tersebut, maka diusulkan beberapa alternatif untuk menunjang performa aplikasi dengan keterbatasan yang ada, diantaranya:

- Merancang aplikasi dengan pemanfaatan sumber daya seefisien mungkin namun tidak mengurangi fungsi dan performa aplikasi.
- Merancang aplikasi dengan antarmuka sederhana namun tetap menarik dan mudah digunakan oleh pengguna.

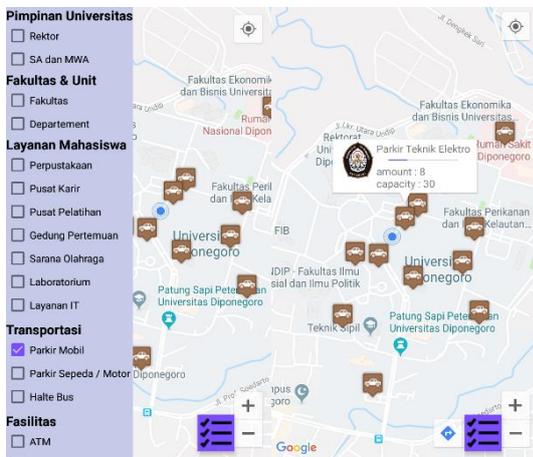
### 2.3 Desain Aplikasi

Desain aplikasi *smart open parking* berfungsi sebagai antar muka pengguna menyajikan beberapa menu utama yang akan mengantarkan pengguna kepada berbagai fungsionalitas sistem. Aplikasi *Smart Open Parking* memiliki beberapa *layout* utama diantaranya pada gambar berikut.



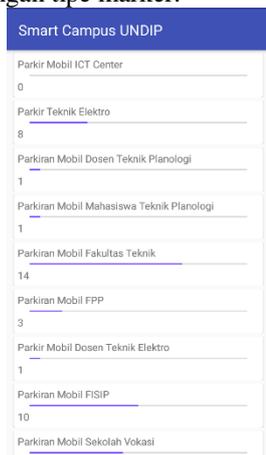
Gambar 2. Desain Antarmuka menu utama

Gambar 2 menunjukkan halaman menu utama dengan beberapa pilihan yang berfungsi untuk menavigasi ke layout lain.



Gambar 3. Desain Antarmuka halaman Map Marker

Gambar 3 menampilkan antarmuka map marker dengan menggunakan widget checkbox untuk memfilter marker lokasi sesuai dengan tipe marker.



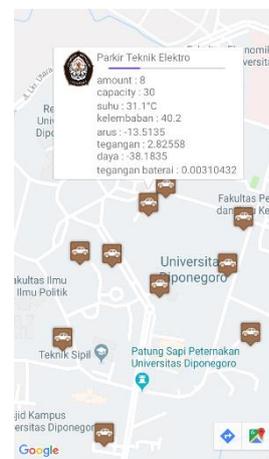
Gambar 4. Desain Antarmuka halaman Tempat Parkir

Gambar 4 menampilkan antarmuka tempat parkir dengan menggunakan recycler view.



Gambar 5. Desain Antarmuka halaman Login

Gambar 5 menampilkan antarmuka login dengan menggunakan widget edittext sebagai masukan otentifikasi dan *button* untuk mengeksekusi perintah login.

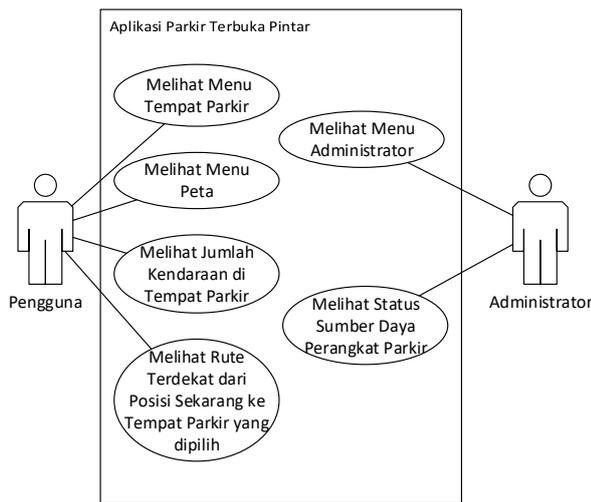


Gambar 6. Desain Antarmuka halaman Admin Monitor

Gambar 6 menampilkan antarmuka halaman admin monitor menggunakan fragment dan marker pada peta menampilkan informasi dari perangkat parkir pintar.

## 2.4 Diagram Use Case Aplikasi

Diagram *use case* menggambarkan fungsi-fungsi yang ada pada sistem. Diagram ini lebih berfokus pada fitur-fitur sistem dari sudut pandang pihak luar, yang dalam hal ini adalah pengguna aplikasi. Gambar 7 berikut merupakan diagram *Use Case* aplikasi *Smart Open Parking*[7].



Gambar 7. Diagram use case pengguna

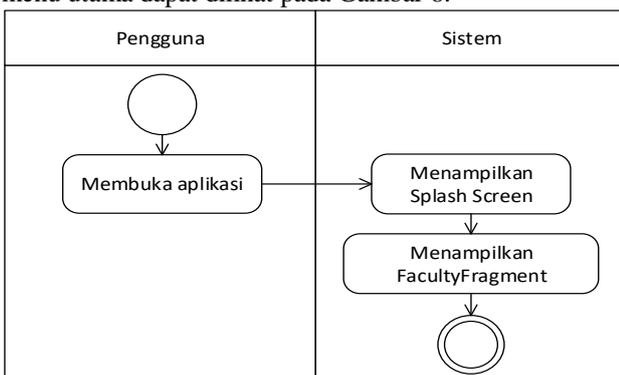
Gambar 7 menggambarkan fitur-fitur yang dapat diakses oleh pengguna. Pengguna dapat mengakses semua menu yang terdapat dalam aplikasi ini. Yaitu menu Tempat Parkir, menu Peta, dan menu Navigasi. Pengguna yang berperan sebagai administrator juga dapat melihat menu administrator dan melihat status sumber daya perangkat *Smart Open Parking*.

## 2.5 Diagram Aktivitas

Diagram aktivitas menggambarkan logika prosedural, dan aliran kerja dalam sistem yang sedang dirancang. Diagram aktivitas juga dapat digunakan untuk menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Diagram aktivitas dapat dibagi menjadi beberapa *swimlane object* untuk menggambarkan objek mana yang bertanggung jawab terhadap aktivitas tertentu[8].

### 2.5.1 Diagram Aktivitas Menu Utama

Diagram aktivitas ini menggambarkan aktivitas pengguna saat pertama kali memulai aplikasi. Pengguna akan melihat tampilan splash screen dan menu utama. Diagram aktivitas menu utama dapat dilihat pada Gambar 8.

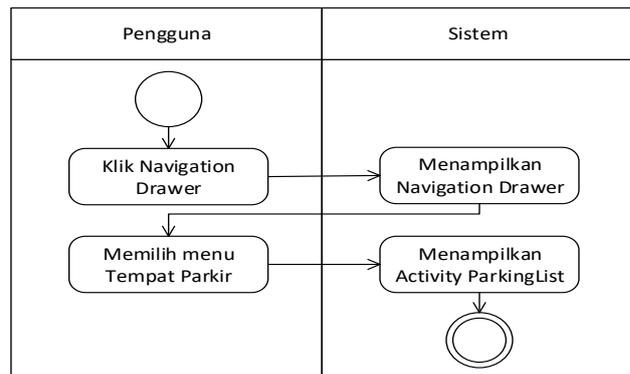


Gambar 8. Diagram aktivitas Menu Utama

Gambar 8 merupakan gambar diagram aktivitas menu utama untuk pengguna yang ingin memulai menggunakan aplikasi. Setelah pengguna membuka aplikasi, sistem akan menampilkan layout splash screen. Setelah itu sistem akan langsung membawa pengguna kepada layout menu utama yang berisi fragment fakultas.

### 2.5.2 Diagram Aktivitas Melihat Daftar Tempat Parkir

Diagram aktivitas ini menggambarkan aktivitas pengguna supaya dapat melihat daftar tempat parkir.



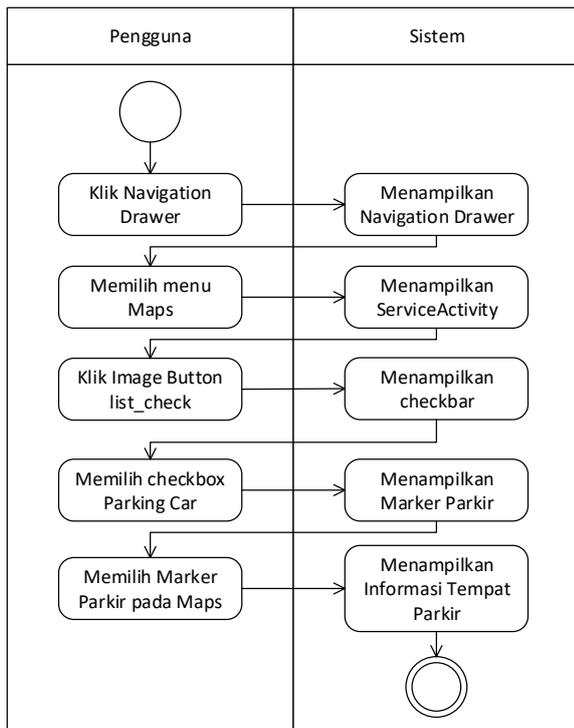
Gambar 9. Diagram aktivitas melihat daftar tempat parker

Gambar 9 merupakan gambar diagram aktivitas untuk melihat daftar tempat parkir. Setelah klik menu, akan muncul navigation drawer. Kemudian pilih menu Tempat Parkir dan akan muncul daftar tempat parkir.

### 2.5.3 Diagram Aktivitas Melihat Tempat Parkir pada Peta

Diagram aktivitas ini menggambarkan aktivitas pengguna supaya dapat melihat lokasi tempat parkir pada peta.

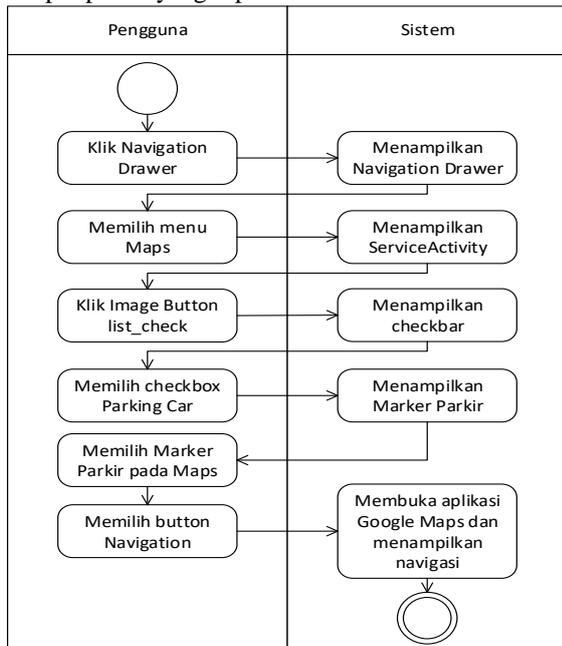
Gambar 10 merupakan gambar diagram aktivitas untuk melihat lokasi dan informasi tempat parkir pada peta. Setelah klik menu, akan muncul navigation drawer. Kemudian pilih menu Maps dan akan muncul peta. Klik image button dan menampilkan checklist, pilih Parking Car. Akan muncul marker pada peta. Kemudian pilih marker pada peta untuk memunculkan informasi tempat parkir.



Gambar 10. Diagram aktivitas melihat tempat parkir pada peta

### 2.5.4 Diagram Aktivitas Melihat Navigasi

Diagram aktivitas ini menggambarkan aktivitas pengguna supaya dapat menampilkan navigasi dari lokasi pengguna ke tempat parkir yang dipilih.

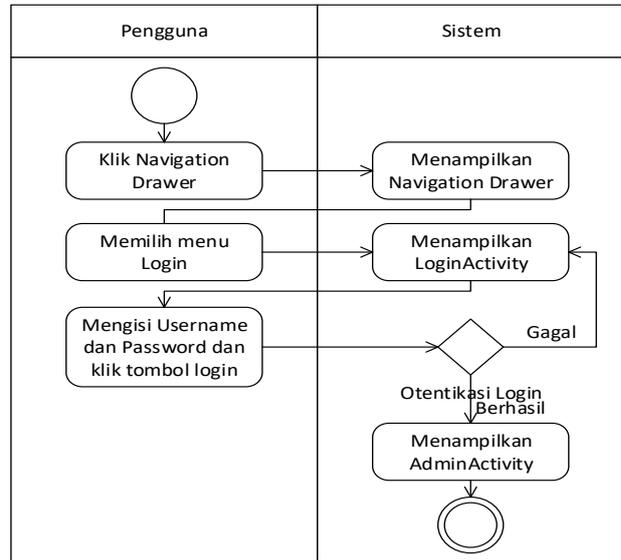


Gambar 11. Diagram aktivitas melihat navigasi

Gambar 11 merupakan gambar diagram aktivitas untuk melihat navigasi dari lokasi pengguna ke tempat parkir yang dipilih. Setelah memilih marker parkir pada peta, klik icon navigasi. Kemudian akan terbuka aplikasi Google Maps dan menampilkan navigasi.

### 2.5.5 Diagram Aktivitas Login

Diagram aktivitas ini menggambarkan aktivitas pengguna untuk login.

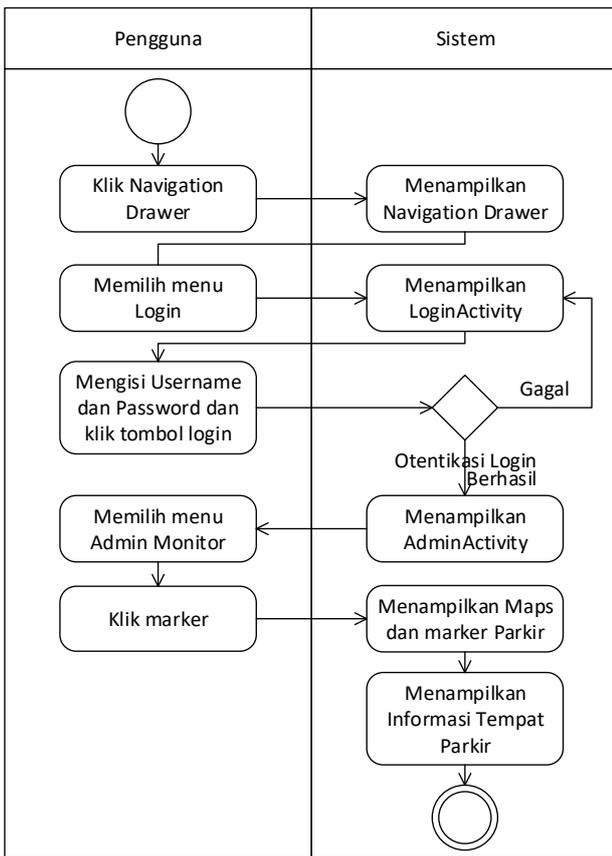


Gambar 12. Diagram aktivitas login

Gambar 12 merupakan gambar diagram aktivitas untuk login. Setelah klik navigation drawer, pilih login. Kemudian masukkan username dan password, setelah sistem mengotentifikasi, jika salah akan kembali ke halaman login, jika benar akan muncul halaman utama admin.

### 2.5.6 Diagram Aktivitas Melihat Status Sumber Daya Perangkat

Diagram aktivitas ini menggambarkan aktivitas pengguna supaya dapat melihat informasi sumber daya perangkat keras smart open parking.

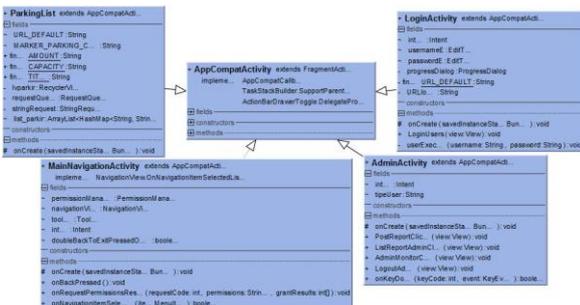


Gambar 13. Diagram aktivitas Melihat Sumber Daya

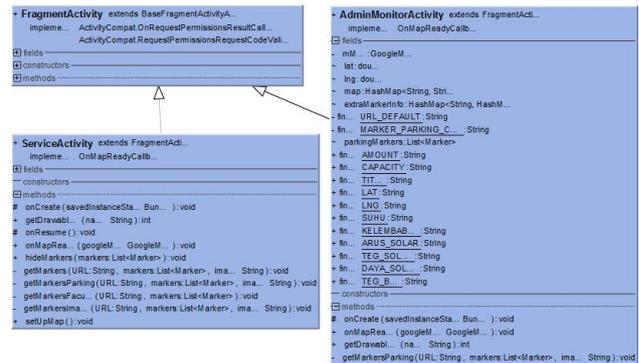
Gambar 13 merupakan gambar diagram aktivitas untuk melihat sumber daya perangkat. Setelah berhasil login, pilih menu Admin Monitor. Sistem menampilkan marker parkir pada peta. Pilih marker dan informasi sumber daya perangkat ditampilkan dalam bentuk pop-up.

### 2.6 Diagram Kelas

Diagram kelas menggambarkan komponen-komponen pembentuk sistem yang berupa kelas atau interface, dan hubungan antar komponen tersebut. Pada aplikasi Smart Open Parking, diagram kelas dibagi mencakup semua fungsi pada aplikasi.



Gambar 14. Hubungan diagram kelas appcompat



Gambar 15. Hubungan diagram kelas fragment



Gambar 16. Kelas-kelas independen

Dapat dilihat pada gambar 14 bahwa kelas-kelas utama yang terdapat pada aplikasi ini extends ke AppCompatActivity, sedangkan pada gambar 15 menunjukkan kelas-kelas yang sifatnya pengganti layout extends ke class Fragment. Gambar 16 menunjukkan ParkingListAdapter yang extends ke RecyclerView.Adapter, sedangkan RequestHandler dan URLString merupakan kelas yang berdiri sendiri.

### 3. Hasil dan Analisis

Pada penelitian ini, pengujian dilakukan dengan melakukan pengujian alfa yaitu pengujian fungsionalitas dari aplikasi. Pengujian alfa bertujuan untuk identifikasi dan menghilangkan masalah sebelum akhirnya sampai ke pengguna yang sebenarnya[9]. Pengujian alfa pada aplikasi ini dilakukan dengan menggunakan 3 devices. Hal ini untuk mengetahui performa aplikasi pada spesifikasi device yang berbeda. Media tersebut diantaranya :

1. Xiaomi Redmi Note 5 Pro, OS Android version 8.1 (Oreo), Layar 5.99 inch IPS LCD, Resolusi 1080x2160, CPU Octa-core 1.8 GHz
2. Zenfone 5, OS Android versi 5.1 (Lollipop), Layar 5,0 inch IPS LCD, Resolusi 720x1280, CPU Quad Core 1.2 GHz
3. Xperia S, OS Android version 4.1.2 (Jelly Bean), Layar 4.3 inch LCD, Resolusi 720x 1280.

Pengujian alfa yang diterapkan pada aplikasi ini menggunakan model pengujian blackbox. Secara garis besar, ada 2 macam pengujian, yaitu pengujian komponen aplikasi Android yang dibuat dengan Android Studio dan pengujian waktu respon server pertukaran data.

### 3.1 Pengujian Android

Bentuk-bentuk pengujian yang digunakan untuk melihat performa tiap-tiap fitur pada aplikasi Android. Tabel 1 berikut adalah hasil uji aplikasi android.

**Tabel 1 Hasil Pengujian Aplikasi Android**

No	Bentuk Pengujian	Hasil Pengujian		
		Redmi Note 5	Zenfone 5	Xperia S
1	Memasang aplikasi pada android.	Berhasil	Berhasil	Gagal
2	Menampilkan marker parkir mobil	Berhasil	Berhasil	Gagal
3	Menampilkan informasi tempat parkir	Berhasil	Berhasil	Gagal
4	Menampilkan navigasi dari posisi pengguna ke marker yang dipilih	Berhasil	Berhasil	Gagal
5	Menampilkan daftar parkir mobil	Berhasil	Berhasil	Gagal
6	Login admin	Berhasil	Berhasil	Gagal
7	Menampilkan status sumber daya perangkat parkir pintar untuk admin	Berhasil	Berhasil	Gagal
8	Logout	Berhasil	Berhasil	Gagal

Dapat dilihat pada Tabel 1 bahwa pengujian tiap fitur memiliki status berhasil.

### 3.2 Pengujian Waktu Respon

Pengujian ini dilakukan dengan pada dua komponen aplikasi yaitu waktu respon aktivitas dan waktu respon server komunikasi data. Berikut ini adalah pengujian yang dilakukan.

#### 3.1.2 Pengujian Waktu Respon Aktivitas

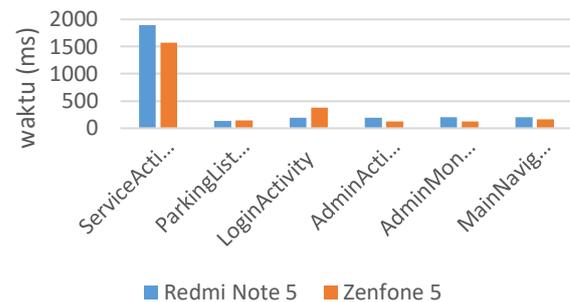
Pengujian waktu respons aktivitas dilakukan dengan cara mengukur waktu tunda antara aktivitas satu dengan aktivitas selanjutnya. Dalam pengujian ini dilakukan 5 kali percobaan dan didapatkan data pada Tabel 2 berikut.

**Tabel 2. Hasil Pengujian Waktu Respon Aktivitas**

Aktivitas	Waktu Respon		
	Redmi Note 5	Zenfone 5	Rata-rata
MainNavigation Activity	201	162	166
Service Activity	1895	1575	1587
ParkingList Activity	137	146	149
Login Activity	192	377	293
AdminMonitor Activity	198	124	172

Data pada Tabel 2 dapat direpresentasikan dalam grafik seperti Gambar 17 berikut.

**Waktu Respon Aktivitas**



**Gambar 17. Grafik Waktu Respon Aktivitas**

Pada Gambar 17 terlihat bahwa waktu respon paling lama yaitu ServiceActivity dan paling singkat yaitu ParkingListActivity. Hal ini dikarenakan pada ServiceActivity terdapat banyak fungsi seperti memilah tipe marker, menampilkan informasi marker, dll sehingga membuat aktivitas tersebut berat. Sedangkan pada aktivitas lainnya hanya terdapat sedikit fungsi sehingga waktu responnya lebih cepat. Perbandingan waktu respon antara 3 perangkat tidak terlalu jauh berbeda pada MainNavigationActivity, ParkingListActivity, LoginActivity, dan AdminMonitoringActivity. Sedangkan pada AdminActivity terlihat perbedaan waktu respon pada setiap perangkat.

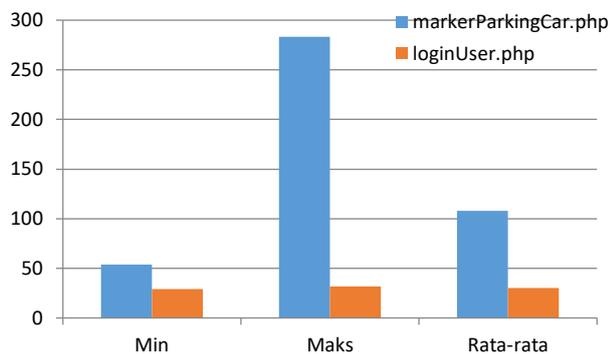
#### 3.1.3 Pengujian Waktu Respon Server Komunikasi Data

Pengujian waktu respon server ini dilakukan dengan menggunakan software JMeter. Pengujian dilakukan dengan 10 kali percobaan. Hasil yang didapat terdapat pada Tabel 3 berikut.

**Tabel 3. Hasil Pengujian Waktu Respon Server**

Pengujian ke-	Waktu Respon	
	markerParkingCar.php	loginUser.php
1	41	15
2	285	12
3	32	13
4	44	15
5	33	17
6	53	14
7	28	14
8	27	26
9	25	21
10	27	14
Rata-rata	59,5	16,1

Dari data yang ada pada Tabel 3 dapat dibuat grafik seperti pada Gambar 18 berikut.



Gambar 18. Grafik Waktu Respon Server

Gambar 4.7 menunjukkan bahwa markerParkingCar memiliki waktu respon sebesar 59,5ms dan loginUser memiliki waktu respon sebesar 16,1ms. markerParkingCar memiliki waktu respons yang lebih lama.

#### 4. Kesimpulan

Aplikasi antarmuka smart open parking berbasis internet of things pada perangkat Android telah dapat berjalan dengan baik pada perangkat Android dengan versi OS Android Lollipop (5.1) dan Oreo (8.1), dan tidak dapat berjalan pada perangkat Android dengan versi OS Jelly Bean (4.1.2).

Pada pengujian blackbox, semua bentuk pengujian masing-masing fitur aplikasi berjalan dengan baik pada Redmi Note 5 dan Zenfone 5, sedangkan pada Xperia S fitur aplikasi tidak dapat berjalan.

Adanya aplikasi Android ini dapat memberikan informasi tempat parkir terbuka yang dibangun dengan menggunakan IDE Android Studio. Sistem ini dapat memberikan informasi berupa jumlah mobil, total kapasitas, tegangan perangkat, arus perangkat, suhu dan kelembaban perangkat. Sistem ini juga dapat memberikan navigasi dari lokasi pengguna ke tempat parkir yang dipilih.

#### Referensi

- [1]. Khanna, A. and Anand, R. (2016) 'IoT based smart parking system', 2016 International Conference on Internet of Things and Applications, IOTA 2016, (January 2016), pp. 266–270. doi: 10.1109/IOTA.2016.7562735.
- [2]. Mainetti, L. et al. (2015) 'A Smart Parking System based on IoT protocols and emerging enabling technologies', IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings, pp. 764–769. doi: 10.1109/WF-IoT.2015.7389150.
- [3]. Renuka, R. and Dhanalakshmi, S. (2015) 'Android based smart parking system using slot allocation & reservations', ARPN Journal of Engineering and Applied Sciences, 10(7), pp. 3116–3120.
- [4]. S. R. Basavaraju, "Automatic Smart Parking System using Internet of Things (IOT)," Int. J. Sci. Res. Publ., vol. 5, no. 12, pp. 629–632, 2015.
- [5]. M. Sabang, Mahruz; S. Sahjad, Rihiza; Baharuddin, "Sistem Parkir Cerdas," 2012.
- [6]. Faizin, Ahmad Arif. 2017. *Aplikasi Media Pemasaran Properti dengan Menggunakan Teknologi Augmented Reality pada Perangkat Android*. TRANSIENT, [S.l.], p. 341-348, nov. 2017. ISSN 2302-9927
- [7]. Syarif, M. 2016. *Perancangan Aplikasi Bernama My Landmark Berbasis SIG untuk Informasi Penjualan Tanah Pada Perangkat Bergerak Android*. TRANSIENT, vol. 5, no. 2, Juni 2016 page. 117-124
- [8]. Iswahyudi, Catur & Rudi Prasetyo, Argo & Feby Prakoso, Andung & Nega, Muntaha. (2017). *Purwarupa Sistem Parkir Cerdas Berbasis Arduino Sebagai Upaya Mewujudkan Smart City*.
- [9]. Djamaluddin, W. 2011. *Implementasi Handphone Locator Dalam Sistem Operasi Android dan Google Maps*. penelitian. Teknik Informatika Institut Teknologi Sepuluh November Surabaya