

# Desain Antarmuka Aplikasi *Mobile* Pada Sistem Parkir Berbasis *Internet of Things*

David Ishak Kosasih, Resmana Lim, Petrus Santoso  
Program Studi Teknik Elektro, Universitas Kristen Petra  
Jl. Siwalankerto 121-131, Surabaya - 60236, Indonesia

Email: m23413005@john.petra.ac.id ; resmana@petra.ac.id ; petrus@petra.ac.id

**Abstrak**--Dalam paper ini dijelaskan desain aplikasi *mobile* pada sistem parkir berbasis *internet of things*. Pada sistem ini, data dari area parkir (status slot dan aktivitas penggunaan area parkir) akan dikirim menuju aplikasi *mobile*. Sistem ini memiliki 4 bagian utama yaitu server lokal, *ubidots cloud*, server publik, dan aplikasi *mobile*. Server lokal bertugas untuk mengelola data dari sebuah area parkir untuk kemudian dikirim menuju *ubidots cloud*. *Ubidots cloud* bertugas untuk menyimpan data dari area parkir. Selanjutnya, server publik akan mengambil data dari *ubidots cloud* untuk diproses dan dikirim menuju aplikasi *mobile*. Terdapat 3 macam pengguna aplikasi *mobile* yaitu pengguna biasa, pengguna pra bayar dan pengguna admin. Pengguna biasa dan pra bayar adalah orang yang menggunakan area parkir pada umumnya. Pengguna pra bayar sendiri merupakan pengguna area parkir yang melakukan proses pembayaran di depan dengan menggunakan kartu RFID. Pengguna admin adalah orang yang memiliki hak akses penuh atas sebuah area parkir seperti pengelola. Dari hasil pengujian pada sistem ini didapatkan bahwa penggunaan *memory* pada aplikasi *mobile* cukup stabil. Selain itu, penggunaan jaringan pada aplikasi *mobile* ini di bawah 1 detik untuk menerima dan mengirim data menuju server.

**Kata kunci:** *Smart Parking*, Aplikasi *Mobile*, *Ubidots cloud*

## I. PENDAHULUAN

Aplikasi *mobile* mulai banyak diterapkan pada berbagai macam aspek kehidupan. Contoh penerapan aplikasi *mobile* dapat kita lihat pada bidang komunikasi, informasi, pendidikan, dan hiburan. Di sisi lain, pengembangan aplikasi *mobile* untuk pendukung *smart city* masih sangat minim dan sulit di temukan terutama pada negara berkembang seperti Indonesia. Hal ini dapat dilihat dari sedikitnya jumlah aplikasi *mobile* untuk *smart city* di *play store* dan *app store*.

Konsep *smart city* sendiri saat ini mulai banyak dikembangkan. *Smart city* didefinisikan oleh IBM (*International Business Machines*) sebagai penggunaan teknologi informasi dan komunikasi yang bertujuan untuk merasakan, menganalisa, dan mengintegrasikan informasi dari sebuah sistem kota serta dapat merespon berbagai

macam kebutuhan masyarakat [1]. Mengacu pada definisi di atas, aplikasi *mobile* sangat mungkin dimanfaatkan karena dapat mempermudah dalam proses pengambilan atau penyebaran data pada konsep *smart city*.

Dari beberapa penerapan *smart city*, pada sistem ini dipilih penggunaan aplikasi *mobile* untuk *smart parking* atau sistem parkir berbasis *internet of things* (IoT). *Internet of things* adalah paradigma komunikasi terbaru dimana berbagai macam benda dari kehidupan sehari-hari akan dilengkapi dengan mikrokontroler, *transceiver*, dan susunan protokol agar benda-benda tersebut dan pengguna mampu berkomunikasi satu sama lain menjadi bagian terintegrasi dari internet [2].

Terdapat beberapa manfaat yang menjadi landasan dipilihnya *smart parking*. Dengan penggunaan aplikasi *mobile* untuk *smart parking* maka pengguna dapat mengetahui kondisi area parkir sebelum berada di lokasi bahkan ketika masih berada di rumah. Hal ini juga dapat menjadi pertimbangan untuk menentukan lokasi tujuan pengguna. Alasan utama pentingnya pembangunan *smart parking* untuk sebuah kota adalah menghemat waktu seorang pengendara menemukan slot parkir sehingga dapat memperkecil emisi CO dari mobil, mengurangi kemacetan lalu lintas dan membuat masyarakat lebih bahagia [2]. Pada akhirnya, aplikasi *mobile* untuk *smart city* akan didukung oleh konsep aplikasi *mobile* pada *smart parking* ini.

## II. PERANCANGAN SISTEM

Berikut adalah skema sistem secara keseluruhan.



Gambar 1. Skema Sistem

Gambar 1 menunjukkan desain secara keseluruhan dari sistem ini. Pada desain sistem ini,

1 area parkir akan dilayani oleh 1 raspberry pi sebagai server lokal. Data dari setiap server lokal akan langsung dikirim menuju *cloud* dimana pada sistem ini digunakan *ubidots cloud*.

Selanjutnya, data yang terkumpul pada *ubidots cloud* akan diterima oleh aplikasi *mobile*. Pada saat pengguna melakukan permintaan data area parkir, server publik akan mengambil data tersebut dari *ubidots cloud* dan mengirimkannya menuju aplikasi *mobile*. Pada sistem ini digunakan *ubidots cloud* gratis sehingga dalam 1 menit server hanya dapat mengambil maksimal 60 data dari *ubidots cloud* [3]. Data ini dikirim menuju aplikasi *mobile* menggunakan koneksi TCP.

Sistem ini memiliki 2 jenis aplikasi *mobile* yaitu aplikasi *mobile* pengguna dan aplikasi *mobile* admin. Pengguna sendiri dapat dibagi menjadi 2 macam yaitu pengguna biasa dan pra bayar. Pengguna biasa adalah orang yang akan menggunakan jasa tempat parkir tanpa melalui sistem pra bayar. Pengguna pra bayar adalah pengguna jasa area parkir yang melakukan pembayaran melalui sistem kartu RFID. Admin adalah orang yang bertanggung jawab pada area parkir tersebut seperti karyawan hingga pemilik.

Berikut ini adalah layanan dari aplikasi *mobile* pengguna.

Tabel 1. Layanan Dari Aplikasi *Mobile* Pengguna

	Pengguna Biasa	Pengguna Pra Bayar
Masuk Aplikasi	√	√
Daftar Aplikasi	√	√
Mengubah Kata Sandi	√	√
Reset Kata Sandi	√	√
Hapus Akun	√	√
Melihat Daftar Area Parkir	√	√
Melihat Lokasi Area Parkir	√	√
Melihat Daftar Lantai Parkir	√	√
Melihat Data Slot Parkir	√	√
Melihat Aktivitas Parkir	x	√

Tabel 1 menunjukkan layanan dari aplikasi *mobile* pengguna. Pengguna perlu melakukan prosedur keamanan yaitu proses daftar dan masuk (*login*) aplikasi. Selain itu pengguna juga dapat mengubah kata sandi atau mengembalikan kata sandi menjadi kata sandi umum yang disediakan oleh produsen aplikasi (*reset* kata sandi). Apabila pengguna ingin menonaktifkan akun, maka dapat memanfaatkan layanan hapus akun. Daftar area parkir, daftar lantai parkir dan data slot (skema slot parkir) dapat dilihat oleh kedua jenis pengguna. Data aktivitas parkir yang berisi nama lokasi dan waktu area parkir digunakan hanya diperuntukan bagi pengguna pra bayar.

Aplikasi *mobile* admin memiliki layanan sebagai berikut :

1. Masuk aplikasi
2. Daftar aplikasi
3. Mengubah kata sandi
4. Reset kata sandi
5. Hapus akun

6. Melihat daftar lantai parkir
7. Melihat data slot parkir
8. Melihat pengguna area parkir

Admin juga memiliki prosedur keamanan yaitu proses daftar, masuk, mengubah kata sandi, *reset* kata sandi, dan hapus akun. Admin hanya dapat melihat informasi lantai parkir dan data slot pada area parkir dari admin yang bersangkutan. Selain itu, admin dapat melihat data pengguna area parkir yang berisi nama dan waktu penggunaan area parkir.

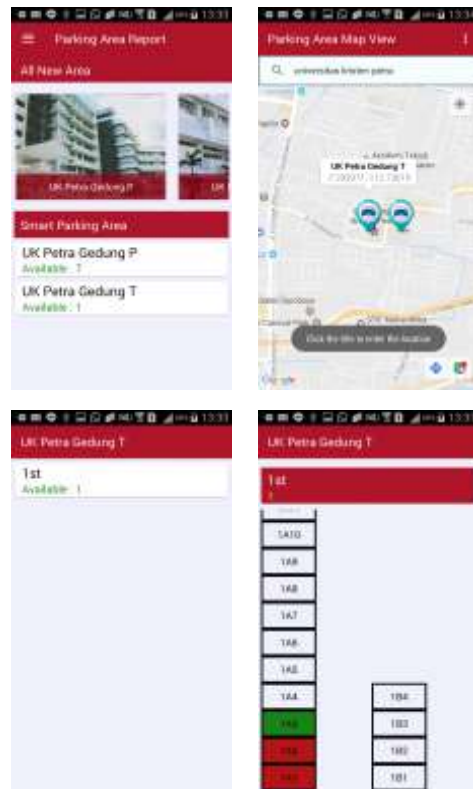
A. *Desain Antarmuka Aplikasi*

Terdapat 7 kelompok tampilan antarmuka aplikasi yaitu :

1. Antarmuka proses memuat aplikasi
2. Antarmuka halaman menu
3. Antarmuka halaman keamanan
4. Antarmuka halaman data kondisi slot parkir
5. Antarmuka halaman data aktivitas parkir
6. Antarmuka halaman data pengguna area parkir
7. Antarmuka halaman informasi aplikasi

Pada paper ini akan dijelaskan 3 kelompok tampilan antarmuka utama yaitu tampilan antarmuka 4, 5, dan 6.

Gambar 2 adalah antarmuka keempat yaitu data kondisi slot parkir.

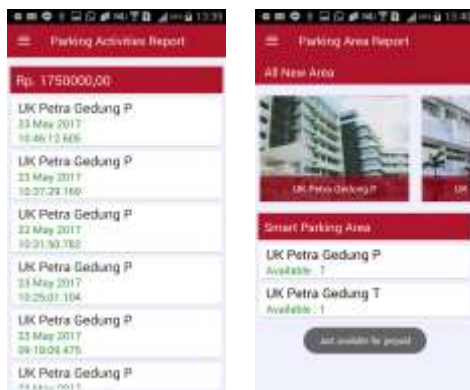


Gambar 2. Antarmuka Halaman Kondisi Slot Parkir

Pada gambar 2 terdapat 4 jenis halaman kondisi slot parkir yaitu :

1. Halaman *parking area report*. Pada halaman ini terdapat daftar nama dari seluruh area parkir dan jumlah slot yang masih tersedia dalam bentuk *list*. Halaman ini hanya tersedia pada aplikasi *mobile* pengguna.
2. Halaman *parking area map view*. Pada halaman ini pengguna dapat melihat daftar lokasi area parkir dalam bentuk *marker* pada *google map*. Selain itu pengguna juga dapat melihat posisi lintang dan bujur dari setiap lokasi parkir. Halaman ini hanya tersedia pada aplikasi *mobile* pengguna.
3. Halaman lantai parkir. Pada halaman ini pengguna dapat melihat daftar nama lantai dari area parkir tertentu dan jumlah slot yang masih tersedia. Halaman ini tersedia pada aplikasi *mobile* pengguna dan admin.
4. Halaman skema slot parkir. Pada halaman ini pengguna dapat melihat posisi dan status dari setiap slot pada area dan lantai parkir tertentu. Warna hijau menandakan kondisi slot kosong sedangkan warna merah menandakan kondisi slot terisi. Selain itu, pengguna juga dapat melihat jumlah slot yang masih tersedia. Halaman ini tersedia pada aplikasi *mobile* pengguna dan admin.

Gambar 3 adalah antarmuka kelima yaitu data aktivitas parkir.



Gambar 3. Antarmuka Halaman Data Aktivitas Parkir

Aktivitas parkir adalah halaman yang tersedia pada aplikasi *mobile* penggunaan hanya dapat diakses oleh pengguna pra bayar. Apabila halaman ini diakses oleh pengguna biasa maka akan muncul notifikasi "*just available for prepaid*" seperti pada gambar 3 kanan. Pada halaman ini terdapat nama area parkir yang telah digunakan oleh pengguna yang bersangkutan. Selain itu pengguna juga dapat melihat waktu penggunaan setiap area parkir dalam bentuk tanggal dan jam seperti pada gambar 3 kiri. Pada bagian atas *list* pengguna dapat melihat jumlah uang yang telah dikeluarkan untuk menggunakan area parkir.

Gambar 4 adalah antarmuka keenam yaitu data pengguna area parkir.



Gambar 4. Antarmuka Halaman Data Pengguna Area Parkir

Data pengguna area parkir adalah halaman yang tersedia pada aplikasi *mobile* admin. Pada halaman ini terdapat nama pemakai dari area parkir tertentu dan waktu pemakaian berupa tanggal dan jam seperti pada gambar 4. Pada bagian atas *list* admin juga dapat melihat pemasukan yang didapatkan dari pemakaian area parkir.

### III. PENGUJIAN APLIKASI MOBILE

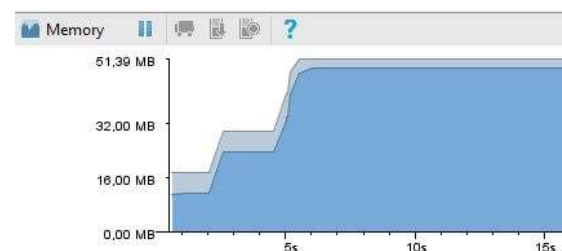
Pada bagian ini terdapat 2 macam pengujian dari aplikasi *mobile* pengguna yaitu :

1. Pengujian *memory*
2. Pengujian *network*

Berikut ini adalah hasil dari masing-masing pengujian.

#### A. Pengujian Memory

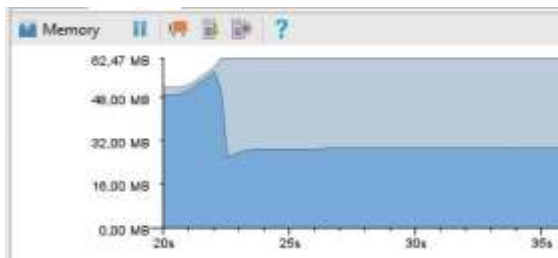
Pengujian *memory* adalah pengujian yang bertujuan untuk mengukur jumlah *memory* yang digunakan [4]. Dari hasil pengujian, aplikasi *mobile* ini menggunakan *memory* kurang lebih 50 MB. Gambar 5 adalah grafik penggunaan *memory* pada proses permintaan daftar area parkir.



Gambar 5. Grafik Alokasi Memory Pada Proses Permintaan Daftar Area Parkir

Pada gambar 5 terlihat bahwa saat aplikasi dibuka, kebutuhan *memory* akan mulai meningkat hingga stabil pada angka tertentu. Pada proses permintaan daftar area parkir, penggunaan *memory* stabil pada angka 48.52 MB. Penggunaan *memory* yang stabil menandakan bahwa aplikasi ini berjalan dengan ideal.

Namun, terjadi perbedaan yang signifikan dari alokasi *memory* pada permintaan lokasi area parkir. Gambar 6 adalah grafik alokasi *memory* pada proses permintaan lokasi area parkir.



Gambar 6. Grafik Alokasi *Memory* Pada Proses Permintaan Lokasi Area Parkir

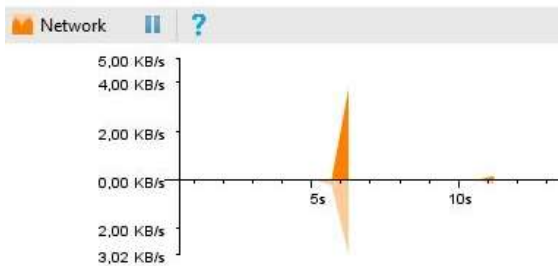
Pada gambar 6 terlihat bahwa saat halaman lokasi area parkir dibuka, penggunaan *memory* akan turun dengan cepat hingga stabil pada angka 29,18 MB. Artinya, pada halaman lokasi area parkir dibutuhkan alokasi *memory* yang lebih rendah daripada halaman lain.

**B. Pengujian Jaringan**

Pengujian jaringan bertujuan untuk menganalisa frekuensi penggunaan jaringan pada aplikasi *mobile* [5]. Terdapat 3 perbedaan yang cukup signifikan pada hasil pengujian ini yaitu pada proses :

1. Permintaan daftar area parkir
2. Permintaan data slot parkir
3. Permintaan lokasi area parkir

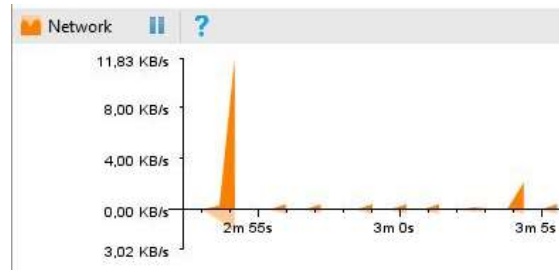
Gambar 7 adalah pengujian jaringan pada proses permintaan daftar area parkir.



Gambar 7. Grafik Frekuensi Penggunaan Jaringan Pada Proses Permintaan Daftar Area Parkir

Garis dengan warna oranye tua menggambarkan pola penerimaan data (rx) sedangkan garis dengan warna oranye muda menggambarkan pola pengiriman data (tx). Dari gambar 7, terlihat bahwa proses permintaan daftar area parkir membutuhkan waktu kurang dari 1 detik untuk menerima (rx) dan mengirim (tx) data. Kecepatan maksimal penerimaan data (rx) adalah 3.73 KB/s sedangkan kecepatan maksimal pengiriman data (tx) adalah 3.02 KB/s. Hasil pengujian jaringan dari setiap proses lainnya mendekati hasil pengujian ini.

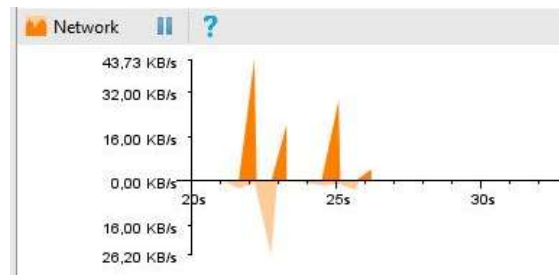
Gambar 8 adalah pengujian jaringan pada proses permintaan daftar slot parkir.



Gambar 8. . Grafik Frekuensi Penggunaan Jaringan Pada Proses Permintaan Daftar Slot Parkir

Dari gambar 8, terlihat bahwa proses permintaan daftar slot parkir membutuhkan waktu kurang dari 1 detik untuk menerima (rx) dan mengirim (tx) data. Kecepatan maksimal penerimaan data (rx) adalah 11.83 KB/s sedangkan kecepatan maksimal pengiriman data (tx) adalah 1.79 KB/s.

Gambar 9 adalah pengujian jaringan pada proses permintaan lokasi area parkir.



Gambar 9. Grafik Frekuensi Penggunaan Jaringan Pada Proses Permintaan Lokasi Area Parkir

Dari gambar 9, terlihat bahwa proses permintaan lokasi area parkir membutuhkan waktu kurang dari 1 detik untuk menerima (rx) dan mengirim (tx) data. Kecepatan maksimal penerimaan data (rx) adalah 43.73 KB/s sedangkan kecepatan maksimal pengiriman data (tx) adalah 26.20 KB/s.

Dari pengujian jaringan didapatkan 2 kesimpulan yaitu :

1. Rata-rata waktu penerimaan dan pengiriman data menuju server di bawah 1 detik.
2. Pertambahan jumlah data yang dikirim maupun diterima berbanding lurus dengan pertambahan kecepatan.

**IV. KESIMPULAN**

Kesimpulan yang dapat diambil dari perancangan sistem dan pengujian pada aplikasi *mobile* ini adalah :

1. Penggunaan *memory* dari aplikasi *mobile* pada sistem ini cukup stabil.
2. Hasil penggunaan jaringan pada aplikasi *mobile* ini cukup baik, karena waktu yang dibutuhkan kurang dari 1 detik untuk menerima maupun mengirim data menuju server.

DAFTAR PUSTAKA

- [1] K. Su, J. Li, and H. Fu, "Smart city and the applications," *IEEE Xplore*, pp. 1028–1031, 2011.
- [2] a Zanella, N. Bui, a Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014.
- [3] Ubidots, "Flexible pricing to take your project from prototype to production," 2017. [Online]. Available: <https://ubidots.com/pricing>. [Accessed: 27-Jun-2017].
- [4] Android, "Android Studio - Memory Monitor," 2016. [Online]. Available: <https://developer.android.com/tools/performance/memory-monitor/index.html>. [Accessed: 16-Jul-2017].
- [5] Android Developers, "Network Monitor | Android Studio," 2017. [Online]. Available: <https://developer.android.com/studio/profile/am-network.html>. [Accessed: 16-Jul-2017].