

BAB III

LANDASAN TEORI

Pada bab ini akan dipaparkan teori-teori yang melandasi didalam pembangunan aplikasi yang akan dibuat.

3.1. Definisi Parkir

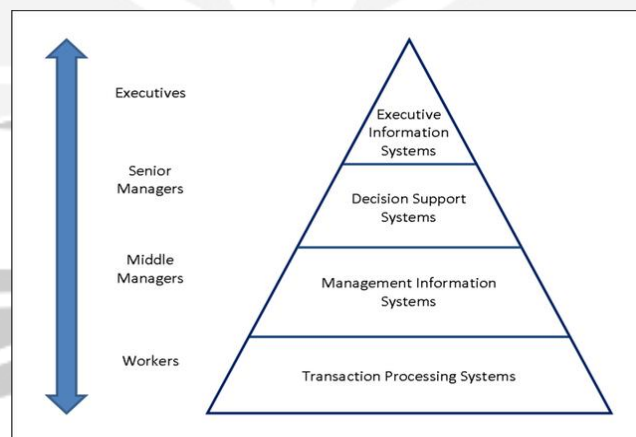
Menurut Andi Prasetya Utomo (2013), parkir merupakan keadaan tidak bergerak suatu kendaraan yang bersifat sementara karena ditinggalkan oleh pengemudinya. Secara hukum dilarang untuk parkir di tengah jalan raya, namun parkir di sisi jalan umumnya diperbolehkan. Fasilitas parkir dibangun bersama-sama 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 atau menurunkan orang atau barang.

3.2. Sistem Informasi

Menurut Eka Prasetya Adhy Sugara (2011), Sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Menurut Robert dan Roscoe (2005) sistem informasi dapat

didefinisikan sebagai sistem pada suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan laporan-laporan yang diperlukan.

Didalam sebuah buku karya Laudon, K.C. and Laudon, J.P. yang berjudul *Information Systems*, tahun 2006, menggambarkan sebuah piramida sistem yang mencerminkan hirarki organisasi, transaction processing systems biasanya diletakkan pada bagian bawah piramida, kemudian management information systems, decision support systems pada bagian atas diakhiri dengan executive information systems atau sistem informasi eksekutif.



Gambar 3.1 Piramida Sistem Informasi

Meski saat ini teknologi telah berkembang dengan pesat serta hadirnya beragam teknologi baru, namun gambaran piramida sistem informasi diatas masih menjadi panduan dasar dalam pengembangan sistem informasi. Setiap level manajemen membutuhkan informasi yang berbeda sesuai dengan

bidang dan tanggung jawabnya, baik dari informasi eksternal maupun internal organisasi sebaiknya dapat disajikan dengan mudah dan bermanfaat pada setiap level manajemen.

3.3. Sistem Operasi Android

Menurut Ableson (2009), Android merupakan aplikasi dengan *platform mobile*, dimana Android merupakan aplikasi *open source*. Android merupakan produk utama dari Google yang mempunyai bagian pada *Open Handset Alliance*. *Open Handset Alliance* adalah kumpulan 30 organisasi yang mempunyai komitmen menjadikan selular sebagai perangkat yang terbuka dan baik untuk pasar. Android merupakan *platform* yang pertama kali menjadi platform terbuka untuk perangkat seluler yang berasal dari perangkat lunak serta pembangunannya ditujukan untuk perangkat yang menggunakan basis *mobile* atau berbasis *mobile*. Android termasuk kernel berbasis Linux, aplikasi *end-user*, dan *framework* aplikasi. *User application* dibangun berbasiskan bahasa pemrograman Java. Bahkan aplikasi yang sedang dibangun juga berbasiskan Java. Menurut sebuah riset yang dilakukan oleh *strategy analytics* di tahun 2013, Android merupakan sistem operasi untuk perangkat *smartphone* terbesar dengan tingkat adopsi sebesar 81.3% disusul iOS dan Windows Phone masing-masing dengan 13.4% dan 4,1%. Menurut Nripin dan Bhat (2013), terdapat tiga kategori dalam *mobile application* yaitu *mobile native application*, *mobile web application*, dan *mobile hybrid application*. Perbedaan kategori tersebut berdasarkan bahasa pemrograman yang digunakan dan layanan

yang dapat didukung oleh aplikasi *mobile* dimana masing-masing memiliki kelebihan dan kekurangan. Aplikasi yang dikembangkan merupakan *mobile native application*.

Menurut Leonardus Ari Wardana (2016), *mobile native application* memiliki karakteristik yang terkait dengan *single platform*. Dalam pengembangan aplikasi diturunkan dari suatu platform perangkat *mobile* seperti android atau IOS yang memiliki kelebihan aplikasi berjalan lebih fleksibel dan lebih cepat. Aplikasi *native* biasanya dibuat menggunakan bahasa Java untuk Android dan Objective C untuk iOS. Aplikasi *native* bisa memanfaatkan penuh sistem operasi mereka terutama menyangkut akses ke perangkat keras *smartphone*. Aplikasi *native* dapat mengakses ke semua *native API (Application Programming Interface)*. Setelah pembuatan aplikasi selesai dilakukan dan akan didistribusikan, pendistribusiannya dapat dilakukan melalui *appstore (marketplace)* yang mendukung platform tersebut.

3.4. Location Based Service

Menurut Badrul Anwar (2014), LBS (*Location Based Service*) 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. LBS mengirimkan data lokasi berupa koordinat posisi dan gambar yang nantinya akan diterima *server* dan disajikan dalam bentuk tampilan lokasi. Layanan berbasis lokasi dapat digambarkan sebagai suatu layanan yang berada pada pertemuan tiga

teknologi yaitu: *Internet Service*, *Mobile Devices*, dan *Geographic Information System*. Terdapat beberapa unsur utama dari penggunaan LBS, yaitu dengan *Location Manager* atau API Maps untuk menyediakan perangkat bagi sumber untuk LBS kemudian API Maps untuk menyediakan fasilitas untuk menampilkan peta paket ini berada pada `com.google.android.maps`. Pada penyedia teknologi pencarian lokasi yang digunakan perangkat menggunakan *API Location* dengan data GPS dan data *real-time*. *API Location* berada pada paket `android.location`. dimana lokasi tertentu dapat ditentukan melalui *location manager*.

Menurut Steineger (2006), penggunaan LBS dapat digunakan menampilkan informasi posisi secara geografis keberadaan tempat parkir. Terdapat komponen pendukung layanan LBS, diantaranya:

1. Perangkat *mobile*. Perangkat ini berfungsi sebagai alat bantu (*tool*) bagi pengguna untuk meminta informasi. Hasil dari informasi yang diminta dapat berupa teks, suara, gambar dan lain sebagainya. Perangkat *mobile* yang dapat digunakan bisa berupa PDA, *smartphone*, laptop. Selain itu, perangkat *mobile* dapat juga berfungsi sebagai alat navigasi di kendaraan seperti halnya alat navigasi berbasis GPS.
2. Jaringan komunikasi. Komponen ini berfungsi sebagai jalur penghubung yang dapat mengirimkan data-data yang dikirim oleh pengguna dari perangkat *mobile* untuk kemudian dikirimkan ke penyedia layanan dan kemudian hasil permintaan tersebut dikirimkan kembali oleh penyedia layanan kepada pengguna.

3. Penyedia layanan. Merupakan komponen LBS yang memberikan berbagai macam layanan yang bisa digunakan oleh pengguna. Sebagai contoh ketika pengguna meminta layanan agar bisa tahu posisinya saat itu, maka aplikasi dan penyedia layanan langsung memproses permintaan tersebut, mulai dari menghitung dan menentukan posisi pengguna, menemukan rute jalan, mencari data di *Yellow Pages* sesuai dengan permintaan, dan masih banyak lagi yang lainnya.
4. Komponen penunjuk lokasi. Setiap layanan yang diberikan oleh penyedia layanan biasanya akan berdasarkan pada posisi pengguna yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang berfungsi sebagai pengolah yang akan menentukan posisi pengguna layanan saat itu. Posisi pengguna tersebut bisa didapatkan melalui jaringan komunikasi *mobile* atau juga menggunakan *Global Positioning System* (GPS).
5. Penyedia data. 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.

Penggambaran cara kerja LBS pada aplikasi SIMPARKIR, LBS akan mencari informasi parkir yang berada di sekitar posisi pengguna.

1. Fungsi pencarian telah diaktifkan, posisi pengguna didapatkan dari *Positioning Service* pada perangkat mobile pengguna. Hal ini dilakukan oleh perangkat menggunakan GPS sendiri ataupun layanan posisi jaringan yang berasal dari *provider* (*Cell Tower*). Setelah itu, perangkat *mobile* pengguna akan 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, *gateway* 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. Kemudian, *service* akan melakukan analisis kembali pesan dan memutuskan mana informasi dan posisi pengguna yang diperlukan untuk menjawab permintaan pengguna.
4. *Service* akan menemukan bahwa pengguna membutuhkan informasi parkir pada wilayah tertentu dan kemudian *service* tersebut akan meminta penyedia data untuk memberikan data tersebut.

5. Kemudian, lokasi parkir akan disampaikan kepada pengguna dalam bentuk peta digital (Badrul Anwar, 2014).

3.5. Google Maps Service

Menurut Faya Mahdia (2013), *Google Map Service* adalah sebuah jasa peta global virtual gratis dan *online* yang disediakan oleh perusahaan Google. Google Maps yang dapat ditemukan di alamat <http://maps.google.com>. Google Maps menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia. *Google Maps* juga menawarkan pencarian suatu tempat dan rute perjalanan. *Google Maps API* adalah sebuah layanan (*service*) yang diberikan oleh Google kepada para pengguna untuk memanfaatkan *Google Maps* dalam mengembangkan aplikasi. *Google Maps API* menyediakan beberapa fitur untuk memanipulasi peta, dan menambah konten melalui berbagai jenis layanan yang dimiliki, serta mengizinkan kepada pengguna untuk membangun aplikasi enterprise di dalam *website*-nya.

3.6. Web Service

Menurut Balachandar (2005), *web service* dibangun untuk memenuhi kebutuhan untuk berinteraksi bisa beroperasi *machine-to-machine* di atas jaringan dan *web service* merupakan sistem perangkat lunak. *Web service* mempunyai alat penghubung yang diuraikan pada format *machine-processable* atau WSDL. Sistem yang lain saling

berhubungan pada *web service* di menggunakan cara yang telah ditentukan pada deskripsinya yang dengan pesan SOAP, secara khas kemudian akan dihantarkan menggunakan HTTP dengan menggunakan XML *serialization*, bersama standar lain yang berhubungan dengan *web*.

3.7. Database MySQL

Menurut Welling (2005), MySQL merupakan sistem manajemen hubungan antar basis data yang sangat cepat dan sempurna. MySQL merupakan alat bantu untuk memanipulasi basis data, sehingga basis data dapat diisi dengan mudah, diambil, disusun dan diubah datanya. Server MySQL pun dapat mengatur kontrol akses dari data, sehingga beberapa pengguna dapat sekaligus bekerja pada waktu bersamaan.

3.8. PHP

Menurut Teguh Wahyono (2005), PHP atau yang dikenal dulu dengan sebutan *Personal Home Page* dan sekarang lebih familiar dengan sebutan *Hypertext Preprocessor* merupakan sebuah program yang dibangun oleh para programmer yang menekuni dunia *open source* bersama-bersama dari seluruh dunia. Dikembangkannya PHP untuk keperluan manipulasi data serta akses pada *database server open source*, seperti MySQL.

3.9. CodeIgniter

Menurut Ibnu Daqiqil (2011), CodeIgniter adalah sebuah *framework* aplikasi web yang bersifat *open source* digunakan untuk membangun aplikasi php dinamis. Tujuan utama dari pengembangan adalah membantu para *developer* mengerjakan aplikasi lebih cepat. CodeIgniter menyediakan berbagai macam *library* yang dapat mempermudah pengembangan. CodeIgniter dibangun dengan konsep *Model-View-Controller development pattern*, yang merupakan salah satu *framework* tercepat dibandingkan *framework* lainnya.

Demikian akhir dari pembahasan bab landasan teori, pada bab selanjutnya akan dibahas mengenai analisis dan perancangan system, yang meliputi lingkup masalah, perspektif produk, kebutuhan antarmuka eksternal, kebutuhan fungsionalitas perangkat lunak, ERD, *sequence diagram*, *class diagram*, *class diagram specific descriptions*, dan deskripsi perancangan antarmuka.