

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Parkir**

Kata parkir berasal dari kata “*park*” yang berarti taman. Menurut kamus bahasa Indonesia, parkir diartikan sebagai tempat menyimpan. (Menurut Hobbs (1995), dalam jurnal Hermawan (2020) parkir diartikan sebagai suatu kegiatan untuk meletakkan atau menyimpan kendaraan di suatu tempat tertentu yang lamanya tergantung kepada selesainya keperluan dari pengendara tersebut. Menurut Warpani (1990), dalam jurnal Hermawan (2020) definisi parkir adalah meletakkan kendaraan dari suatu tempat atau areal untuk jangka waktu (durasi) parkir tertentu.

#### **2.2 Mall Kartini**

Mall Kartini adalah salah satu pusat perbelanjaan Besar ke dua di Kota Bandar Lampung. Mall ini berdiri pada sekitar tahun 2003. Awalnya di mall ini terdapat Matahari *Department Store*. Kemudian *Giant Supermarket* dibuka di mall ini dan Matahari *Department Store* masih berdiri. Namun sekarang Matahari *Department Store* Mall Kartini telah digantikan oleh *Center Point Department Store*, sedangkan *Matahari Department Store* Lampung dipindahkan ke *Central Plaza* Lampung. Di mall ini terdapat *Food Court*, KFC, dan lain sebagainya. Di mall ini juga terdapat toko *furniture* yang terkemuka.

#### **2.3 Aplikasi Mobile**

Menurut Fitriani Sahara, Rena Adriana (2016), Aplikasi *Mobile* adalah program yang digunakan orang untuk melakukan sesuatu pada sistem komputer. *Mobile* dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi.

## 2.4 Android

Android adalah sebuah system operasi untuk smartphone dan Tablet. Sistem operasi dapat diilustrasikan sebagai „jembatan“ antara piranti (*device*) dan penggunanya, sehingga pengguna bisa berinteraksi dengan *device*-nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device*. *Mobile phone* adalah salah satu perangkat yang bergerak seperti telepon seluler atau komputer bergerak yang digunakan untuk mengakses jasa jaringannya. (Yuni Puspita Sari & Rionaldi Ali 2019).

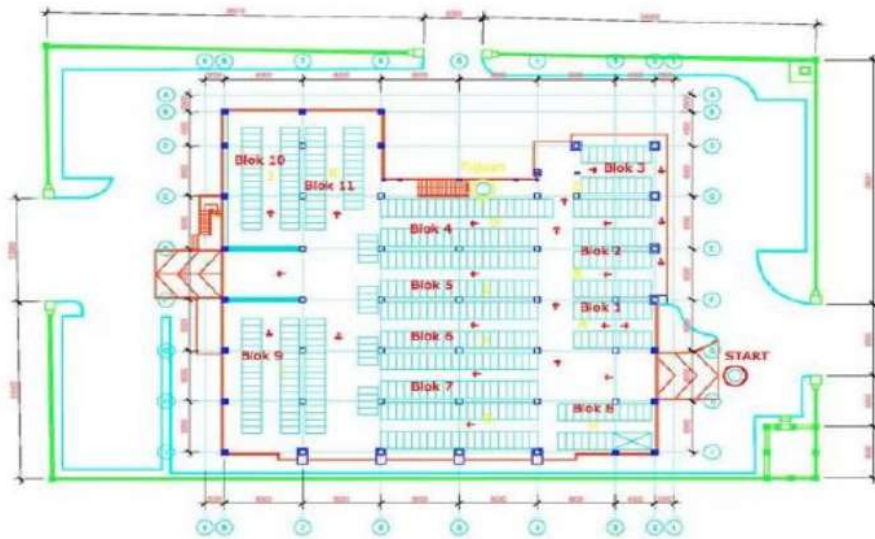
## 2.5 *Firestore Realtime Database*

*Firestore Realtime Database* merupakan *cloud database*. Data disimpan dalam format JSON dan disinkronkan secara *realtime* ke setiap klien yang terhubung. Ketika membangun aplikasi hybrid lintas platform, seperti Android dan iOS maka semua klien berbagi satu *instance Realtime Database* dan secara otomatis menerima pembaruan dengan data tertentu. *Firestore Realtime Database* adalah basis data NoSQL dan karena itu memiliki optimalisasi dan fungsionalitas yang berbeda dibandingkan dengan basis data relasional. Membuat *database Firestore* bisa melalui import file JSON ke konsol *Firestore*, atau dapat juga dibuat langsung melalui halaman konsol *Realtime Database* secara manual.

## 2.6 Metode Algoritma FloydWarshall

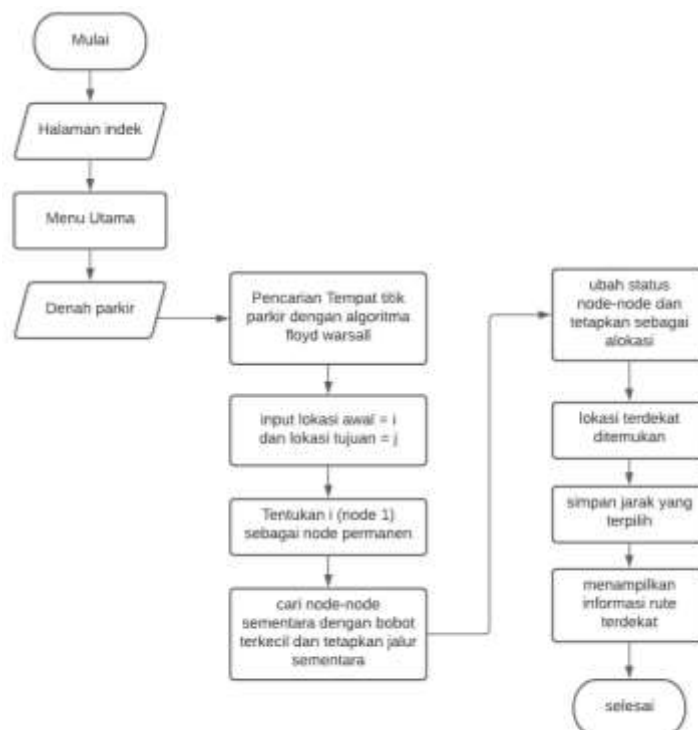
Menurut Jayanti Ari Dewi (2016), Penerapan algoritma *Floyd Warshall* dilakukan berdasarkan hasil yang diperoleh dari tahap analisa masalah. Algoritma *Floyd Warshall* sangat efisien dari sudut pandang penyimpanan data karena dapat diimplementasikan dengan hanya pengubahan sebuah matriks jarak. Untuk perolehan data, penelitian ini menggunakan jenis data sekunder yang diperoleh melalui *state of the art review* penelitian lainnya yang sejenis serta melalui akses internet (jurnal, *ebook*). letak parkir yang disarankan oleh sistem berdasar pada letak parkir dengan jalur terpendek yang dapat mengoptimalkan jarak tempuh pengunjung menuju pintu masuk gedung. Masalah jalur terpendek adalah masalah menemukan suatu jalur antara dua simpul sedemikian sehingga jumlah bobot dari busur penyusunnya dapat seminimal mungkin. Untuk

menentukan jalur terpendek pada penentuan tata letak parkir dalam penelitian ini menggunakan metode *Floyd Warshall* untuk melakukan perhitungan jalur terpendek dari pintu masuk menuju lot parkir.



Gambar 2.1 Ilustrasi Denah Parkir

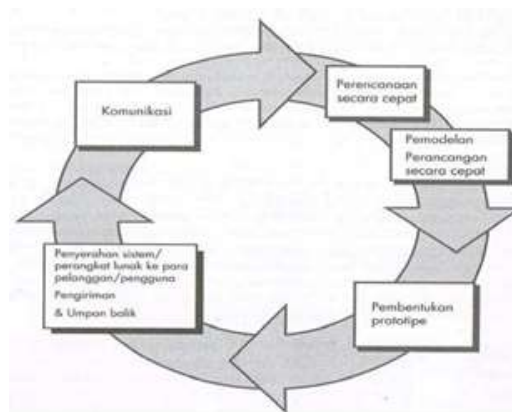
## 2.7 Flowchart Detektor Tempat Parkir Kosong



Gambar 2.2 Flowchart Detektor Tempat Parkir Kosong

## 2.8 Metode Pengembangan Perangkat Lunak *Prototype*

Metode pengembangan perangkat lunak yang digunakan adalah model *prototype*. Model *prototype* dapat digunakan untuk menyambung ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak. diuraikan seperti pada gambar berikut:



Gambar 2.3 Metode *Prototype* (sumber: Pressman, Rekayasa Perangkat Lunak, 2018)

## 2.9 Teori Pendukung

Teknik atau alat bantu digunakan dalam memvisualisasikan rancangan sistem. Teori-teori lain yang digunakan untuk mendukung penulisan ini terdiri dari *unified modeling language* (UML) yang terdiri dari *use case diagram* dan *activity diagram*, *class diagram* dan *black box testing*.

## 2.10 *Unified Modeling Language* (UML)











UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Hendini, 2016). Sedangkan Menurut Sukamto & Shalahuddin (2015:137) mendefinisikan bahwa “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”. Maka dari itu dapat ditarik kesimpulan bahwa *Unified Modeling Language* (UML) merupakan suatu bahasa standar yang digunakan untuk

pemodelan dan komunikasi rancangan perangkat lunak dengan menggunakan diagram atau simbol-simbol tertentu.

### 2.10.1 Use Case Diagram

*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat (Ade Hendini, 2016). *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case* Diagram yaitu:






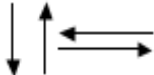
Tabel 2.1 Simbol – simbol *Use Case* Diagram

SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya ( <i>sinergi</i> ).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

### 2.10.2 Activity Diagram

*Activity* diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Ade Hendini, 2016). Simbol-simbol yang digunakan dalam *activity* diagram yaitu:








Tabel 2.2 Simbol – simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan diakhiri
5		<i>Decision</i>	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		<i>Line Connector</i>	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

### 2.10.3 Class Diagram

*Class* diagram merupakan salah satu diagram utama dari UML untuk menggambarkan *class* atau *blueprint object* pada sebuah sistem. (Aji Agustian, Samirah Rahayu, Lani Nurlani, 2016) Simbol-simbol yang digunakan dalam *Class* diagram yaitu:

Tabel 2.3 Simbol – simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	<u>Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor</u>
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	<u>Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri</u>
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

### 2.11 *Black box testing*

Rosa dan Shalahuddin (2015:275) *Black Box Testing* (Pengujian Kotak Pitam) yaitu” menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”. Maka dapat disimpulkan bahwa *Black Box Testing* (Penguji Kotak Hitam) merupakan pengujian perangkat lunak untuk menguji unit tanpa menguji desain dan pengkodean bahwa suatu program telah sesuai dengan proses yang diinginkan.

### 2.12 **Android Studio**

Android adalah sebuah sistem operasi untuk smartphone dan Tablet. Sistem operasi dapat diilustrasikan sebagai ‘jembatan’ antara piranti (*device*) dan pengguna, sehingga pengguna bisa berinteraksi dengan *device*-nya dan menjalankan aplikasiaplikasi yang tersedia pada *device* (Yuni Puspita Sari, 2016). Android Studio adalah sebuah IDE (*Integrated Development Environment*) untuk Android *development* yang diperkenalkan oleh Google. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android studio mempunyai fitur-fitur yang dapat mempermudah penulis dalam penelitiannya.

### 2.13 **Penelitian Terkait**

Dalam penyusunan skripsi ini, peneliti terinspirasi dan merefrensi dari penelitian-penelitian sebelumnya yang berkaitan dengan skripsi ini. Daftar penelitian terkait sebagai berikut :

1. Pemodelan Sistem Pelacakan LOT Parkir Kosong Berbasis Sensor *Ultrasonic* dan *Internet of Things* (IOT) pada Lahan Parkir di Luar Jalan Arthur Daniel Limantara (2017)

Menguraikan parkir diluar ruang milik jalan baik pada mall-mall yang biasanya lahan parkir bertingkat atau pada perkantoran (biasanya kantor Pemerintahan) yang mempunyai lahan parkir horizontal yang luas, seringkali menimbulkan persoalan dalam masalah pencarian atau pelacakan tempat (lot) parkir yang masing kosong dimana kendaraan

(mobil) akan berputar-putar atau naik-turun untuk mencari lot parkir yang masih kosong tersebut. Dalam mengatasi persoalan diatas pengelola parkir atau manajemen parkir biasanya membantu pengguna parkir dengan memberikan info jumlah lot parkir yang kosong pada jalur yang dilalui pengguna parkir. Walaupun membantu tetapi masih sering pengguna parkir mencari posisi lot yang kosong tersebut. Sensor *ultrasonic* dan *internet of things* akan digunakan dalam menyampaikan informasi posisi lot parkir yang kosong kepada pengguna parkir dipakai untuk mendeteksi ada-tidaknya kendaraan pada lot tersebut yang akan dikombinasikan dengan *internet of things* yaitu suatu alat (objek) yang dapat memancarkan sinyal melalui suatu jaringan baik kabel maupun nirkabel. Model yang dipakai adalah sebuah sensor *ultrasonic* (sensor jarak) dan *internet of things* (IoT) dalam hal ini adalah sebuah chip yang diprogram dan ditempatkan pada tiap-tiap lot parkir sehingga dapat memancarkan sinyal informasi digital baik ke server maupun ke gadget pengguna parkir (dengan perangkat lunak khusus). Harapannya solusi ini akan bermanfaat bagi manajemen pengelolaan parkir maupun pengguna parkir.

## 2. Rancang Bangun *Prototype* Sistem *Smart Parking* Berbasis Arduino dan Pemantauan Melalui *Smartphone*

R. Rudi, I. Dinata, dan R. Kurniawan (2017), Menguraikan Perancangan *Smart Parking* bertujuan untuk merancang suatu alat mikrokontroler yaitu *Smart Parking* yang dapat menginformasikan dan mengarahkan pengemudi mobil ke area parkir yang kosong. Lahan parkir yang dijadikan sebagai objek penelitian terdiri dari beberapa lokasi parkir dengan kapasitas beberapa kendaraan, namun penelitian ini hanya memilih beberapa slot parkir sebagai sampel. Pada perancangan ini memiliki beberapa bagian umum yang digunakan, yaitu sensor ultrasonik HC-SR04, Arduino Mega, PC/laptop, LCD (*Liquid Cristal Display*) dan IOT (*Internet Of Things*) penelitian menunjukkan bahwa LCD akan menampilkan beberapa slot parkir yang sudah terisi dan beberapa lagi slot parkir yang kosong dan pemantauan melalui *Smartphone* dengan



menggunakan aplikasi Blynk. Sedangkan untuk bagian palang pintu masuk area parkir menggunakan alat yaitu servo, dan Arduino UNO sebagai pengoperasian palang pintu parkir tersebut. Pada perancangan ini menghasilkan jarak pada sensor parkir di bawah 6 cm menandakan bahwa keadaan slot parkir terisi dan di atas 6 cm menandakan keadaan slot parkir tersedia pada LCD dan aplikasi Blynk tersebut. Sedangkan untuk servo palang masuk mendeteksi mobil pada jarak di bawah 5cm servo akan terbuka dengan *delay* 3 detik dan jika di atas 5 cm menandakan bahwa tidak ada mobil di depan palang sensor.

#### 1. Sistem Pendeteksi Lahan Parkir Menggunakan Raspberry Pi, Sensor Ultrasonik dan Mikrokontroller

Hafif Bustani Wahyudi (2016), menguraikan Dewasa ini lahan parkir menjadi kebutuhan yang penting bagi manusia. Semakin luasnya lahan parkir, maka akan berdampak semakin sulitnya mencari lahan parkir yang masih kosong maupun yang sudah digunakan. Untuk membantu menemukan lahan parkir baik itu yang masih kosong maupun yang sudah digunakan, diperlukan suatu sistem pemetaan parkir yang dapat memberikan informasi kepada pengguna parkir lahan yang masih kosong maupun yang sudah digunakan sebelumnya. Informasi lokasi parkir yang masih kosong berupa led yang menyala merah ketika terisi dan hijau ketika lahan kosong yang terhubung dengan Arduino yang bertindak sebagai mikrokontroler menghubungkan sensor sebagai pendeteksi ada tidaknya kendaraan di lahan parkir dan LED sebagai penyedia informasi kepada pengguna parkir. Sedangkan untuk mencari dimana lokasi parkir yang digunakan sebelumnya user dapat menggunakan *QR Code* untuk dipindai dengan *webcam* yang terhubung dengan Raspberry Pi. dan Raspberry Pi akan menampilkan informasi melalui monitor yang terhubung dengannya.

2. Sistem Parkir Menggunakan Ocr (*Optical Character Recognition*) Plat Nomor dan Iot (*Internet Of Things*)

Julian Sahertian, Muzan Ihda Khotmuniza, Risa Helilintar (2020), menguraikan Banyaknya kendaraan pribadi yang dimiliki penduduk perkotaan besar mengakibatkan berbagai permasalahan salah satunya adalah sulitnya menemukan lahan parkir yang kosong di tempat umum. Untuk mengatasi permasalahan tersebut pada penelitian ini dibuat sebuah sistem parkir dengan pendekatan IoT (*Internet of Things*) serta memanfaatkan teknologi OCR (*Optical Character Recognition*) pengenalan plat nomor. Sistem ini memanfaatkan sensor ultrasonik sebagai pendeteksi kendaraan masuk, dan modul kamera untuk mengambil gambar plat nomor kendaraan yang terparkir. Sistem ini dirancang untuk mempermudah pendataan lahan kosong parkir dan pengguna dapat mengakses informasi lahan parkir dimanapun dan kapanpun. Pengujian sistem dilakukan dibagi menjadi dua bagian yaitu pengujian deteksi plat nomor kendaraan menggunakan kamera dan dan pengujian pengiriman data ke server. Hasil yang didapat dari pengujian yang dilakukan cukup memuaskan yaitu sistem mampu mengenali plat nomor kendaraan dengan akurasi 85, 21%, dengan kecepatan pengiriman data maksimal saat semua slot parkir penuh adalah 17 detik.

3. Analisa *Highest And Best Use* (Hbu) Pada Lahan Kosong Nomor 52-58 Di Jalan Dinoyo Surabaya

Muhammad Rizal Azinuddin, Retno Indryani (2016), menguraikan Pada Jalan Dinoyo terdapat lahan kosong yang berfungsi sebagai lahan parkir. Lahan ini terletak pada lokasi yang terdapat pemukiman penduduk, perkantoran, pertokoan, dan Universitas. Berdasarkan potensi yang ada, lahan seluas  $\pm 3145$  m<sup>2</sup> ini dapat dikembangkan menjadi properti lain tanpa menghilangkan fungsi awal sebagai lahan parkir. Analisa lahan kosong ini menggunakan metode *Highest and Best Use* (HBU). Analisis ini meliputi empat hal pokok yaitu, analisis kelayakan secara fisik (*physically feasible*), analisis kelayakan secara peraturan (*legally permissible*), analisis kelayakan secara keuangan (*financially feasible*), dan

analisis produktivitas yang maksimal (*maximally productive*). Berdasarkan hasil analisa, alternatif properti yang memungkinkan dibangun pada lahan tersebut adalah hotel, pertokoan, dan gedung perkantoran. Dari alternatif properti tersebut, alternatif yang bisa meningkatkan nilai lahan dan merupakan produktivitas maksimum adalah alternatif hotel dengan nilai lahan sebesar Rp 13.088.424/m<sup>2</sup>.