

## BAB II

### A. Studi Pustaka

Telaah pustaka merupakan kajian dari buku, skripsi, majalah, internet dan lain lain yang digunakan sebagai referensi yang terkait dengan variabel penelitian yang relevan dengan hasil penelitian lain. Adapun penelitian serupa yang dapat diasumsikan memiliki relevansi dengan penelitian ini adalah:

No.	Peneliti	Judul	Metode Penelitian	Hasil Penelitian
1.	Roby Adi Wibowo	Implementasi Autonomous Navigation Robot Menggunakan Global Positioning System (GPS) Untuk Pemetaan Kadar Gas Berbahaya	Penelitian ini mengimplementasikan Autonomous Navigation Robot yang dilengkapi dengan Global Positioning System (GPS) Ublox Neo M8 dan sensor gas TGS2600 untuk melakukan aktifitas pemetaan terhadap gas berbahaya pada suatu lokasi menggunakan aplikasi JAVA berbasis Google Map.	Hasil implementasi ini menunjukkan bahwa Autonomous Navigation Robot yang dilengkapi dengan GPS Ublox Neo M8 dan sensor TGS 2600 dapat memetakan kadar gas suatu lokasi dengan keakuratan posisi di bawah 5 meter.
2.	Siska Fadilah, Adian Fatchur Rochim, Rinta Krida Lukmana	Sistem Informasi Penjualan dan Pemesanan Layanan Berbasis Web dan Sms Gateway di Petshop PETZONE	Sistem informasi dibangun menggunakan bahasa pemrograman PHP framework Codeigniter, javascript untuk tampilan yang dinamis, dan database MySQL. Proses pembuatan dan pengembangan Sistem Informasi Penjualan ini menggunakan metode air terjun.	Hasil pengujian Sistem Informasi Penjualan dan Pemesanan Layanan Berbasis Web dan SMS Gateway menunjukkan bahwa semua fitur yang terdapat baik dalam sistem informasi maupun SMS gateway dapat bekerja dengan baik dengan ratusan sample data, dan server SMS gateway dapat memproses lebih dari satu SMS secara bersamaan.

3.	Yosnard Jackson Adu	Aplikasi Reservasi dan Member Lapangan Futsal di Kota Malang Menggunakan Metode Navigasi Waypoint Berbasis Android	Penelitian ini membahas tentang pemesanan lapangan futsal menggunakan algoritma first come first served berbasis android bermanfaat dalam memudahkan pelanggan untuk melakukan pemesanan lapangan futsal dengan mudah dan cepat.	Sistem ini menghasilkan kepuasan user dengan nilai Cukup dengan catatan masih banyak fitur yang belum ada dalam aplikasi.
4.	Yunanda Pratama dan Dadan Nur Ramadan	Perancangan GPS Tracking Untuk Penyewaan Kendaraan Bermotor	Penerapan teknologi Internet of Things dapat diterapkan untuk mengatasi permasalahan pemantauan terhadap sepeda motor. Data longitude & latitude didapatkan dari GPS Mobile Phone serta data dikirimkan ke firebase	melalui penelitian ini penulis ingin menggunakan fitur GPS Mobile Phone serta Firebase pada aplikasi Navigasi Waypoint untuk memberikan keamanan pada kendaraan sepeda motor
5.	Julianto Eko Prasetyo, IBK Widiartha, Moh. Ali Albar	Sistem Informasi Geografis Pencarian Lokasi Wisata Kuliner Terdekat di Kota Mataram Berbasis Website	menggunakan fitur informasi geografis. Namun bukan menggunakan Google Maps, melainkan menggunakan Mapbox.	Sistem yang akan dibangun dapat memberikan petunjuk arah untuk menuju lokasi wisata kuliner yang dituju.

**Tabel 2.1** Perbandingan Beberapa Penelitian

Relevansi penelitian ini dengan penelitian-penelitian diatas adalah penggunaan metoda UML, GPS, Firebase, Navigasi waypoint dalam penerapan aplikasinya. Perbedaan penelitian penulis dengan penelitian-penelitian diatas adalah pengaplikasiannya, dimana penulis mengaplikasi untuk pemesanan lapangan futsal yang menerapkan aplikasi android baik untuk admin ataupun untuk user.

## **B. Landasan Teori**

### **1. Navigasi Mapbox**

Mapbox adalah salah satu penyedia peta daring kustom terbesar di situs-situs ternama seperti Foursquare, Pinterest, Evernote, Financial Times, dan Uber Technologies.[3] Sejak 2010, Mapbox memperbanyak pilihan peta kustomnya untuk mengisi keterbatasan yang dimiliki penyedia peta seperti Google Maps.[3] Mapbox merupakan pencipta atau kontributor sejumlah pustaka dan aplikasi peta bebas terkenal, misalnya spesifikasi MBTiles, kartografi TileMill IDE, pustaka JavaScript Leaflet, bahasa gaya dan parser peta CartoCSS, dan pustaka JavaScript mapbox.js.



*Gambar 2.1 Navigasi Mapbox*

### **2. Android Studio**

Android Studio adalah sebuah IDE yang bisa digunakan untuk pengembangan aplikasi android, dan dikembangkan oleh google. Android Studio merupakan pengembangan dari Eclipse IDE Java populer, yaitu IntelliJ IDEA. Android Studio direncanakan untuk menggantikan Eclipse kedepannya dengan IDE resmi untuk pengembangan aplikasi android. Sebagai pengembangan dari Eclipse Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan ADT, Android Studio menggunakan gradle sebagai *build environment* (Makiolor, Sinsuw, and B.N. Naj Joan 2017).



**Gambar 2.2** Android studio

### **3. Firebase**

Firebase adalah layanan Google yang berguna untuk membantu para developer untuk mengembangkan aplikasi. Firebase atau BaSS alias Backend as a Service merupakan cara developer yang fokus mengembangkan aplikasi tanpa perlu effort besar. Sejak ditemukan oleh Andrew Lee dan James Tamplin tahun 2011, Firebase memberikan produk Realtime Database yang berguna untuk menyimpan lebih banyak data dan sinkronisasi ke banyak pengguna. Pada tahun 2014 lalu, pihak Google mengakuisisi layanan pengembang aplikasi ini (Andy Nugroho 2021).



**Gambar 2.3** Firebase

### **4. Unified Modeling Language (UML)**







UML (*Unified Modeling Language*) adalah Bahasa untuk membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi, atau perangkat lunak) dari sistem perangkat lunak. Selain itu, *UML* adalah Bahasa pemodelan yang menggunakan konsep orientasi *object*. *UML* menyediakan notasi-notasi yang membantu

memodelkan sistem dari berbagai perspektif. *UML* tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan (Ipo Novianto 2017).

## 5. Flowchart

*Flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma (Ipo Novianto 2017). Simbol *Flowchart* dapat dilihat pada tabel berikut.

**Tabel 2.2** Simbol *Flowchart*



No	Simbol	Nama Simbol dan Keterangan
1.		<b>Simbol Terminal</b> yaitu untuk menyatakan permulaan atau akhir dari suatu program.
2.		<b>Simbol Arus / Flow</b> yaitu untuk menyatakan jalannya arus suatu proses (arah aliran program).
3.		<b>Simbol Proses</b> yaitu untuk menyatakan proses perhitungan / proses pengolahan data.
4.		<b>Simbol Input – Output</b> yaitu untuk memasukan data (proses input) maupun menunjukkan hasil (output data) dari suatu proses.
5.		<b>Simbol Decision / Logika</b> yaitu untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya / tidak.
6.		<b>Simbol Document</b> yaitu untuk mencetak laporan ke printer

## 6. Use Case Diagram

*Use case* adalah abstraksi dari interaksi antara sistem dan aktor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata *user*. Sedangkan *use case diagram*

memfasilitasi komunikasi antara analis dan pengguna serta antara analis dan *client* (Ipo Novianto 2017).



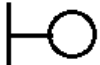



**Tabel 2.3** Simbol-simbol pada *Use Case Diagram*

No.	Simbol	Nama Simbol dan Keterangan
1.		<b>Actor</b> adalah menspesifikasikan himpunan perang yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<b>Use Case</b> menjelaskan urutan kegiatan yang dilakukan <i>actor</i> dan sistem untuk mencapai suatu tujuan tertentu.
3.	.	<b>Association</b> penghubung antara <i>Actor</i> dan <i>Use Case</i> .
4.	▼ Includes	<b>Include</b> yaitu perilaku <i>use case</i> merupakan bagian dari <i>use case</i> lain
5.	▲ Extend	<b>Extend</b> yaitu perilaku suatu <i>use case</i> memperluas perilaku <i>use case</i> lain. <i>Use case</i> boleh dilakukan boleh tidak (optional).
6.		<b>Sistem Roundary</b> yaitu batasan dari sebuah sistem.

## 7. Sequence Diagram

*Diagram sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima oleh objek. Oleh karena itu untuk menggambar diagram sequence maka harus diketahui objek objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat *diagram sequence* juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Berikut adalah simbol-simbol yang ada pada *diagram sequence* (Ipo Novianto 2017).

**Tabel 2.4** Simbol-simbol *Diagram Sequence*

No.	Simbol	Nama	Nama Simbol dan Keterangan
1.		Actor	Menggambarkan orang yang sedang berinteraksi dengan sistem
2.		Entity Class	Menggambarkan hubungan yang akan dilakukan
3.		Boundary Class	Menggambarkan sebuah gambaran dari form
4.		Control Class	Menggambarkan penghubung antara boundary dengan tabel
5.		A focus of Control & a Life Line	Menggambarkan tempat mulai dan berakhirnya message
6.		A message	Menggambarkan pengiriman pesan

## 8. Class Diagram

*Class diagram* adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang kita gunakan. *Class diagram* memberi kita gambaran tentang perangkat lunak dan relas-relasi yang ada didalamnya. Menurut (Whitten L. Jeffery et al 2004:432) *Class Diagram* adalah diagram yang menunjukkan class-class yang ada dari sebuah sistem dan hubungannya secara logika. *Class diagram* menggambarkan struktur statis dari sebuah sistem. Karena itu *class diagram* merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk *UML* (Ipo Novianto 2017).

Atribut dan operation dapat memiliki salah satu sifat berikut :

- Private* hanya bisa dipanggil dari dalam kelas itu sendiri. Atribut diawali “-”.
- Protected* hanya dapat dipanggil oleh class yang bersangkutan dan class turunannya. Atribut diawali dengan tanda “#”.
- Public* dapat dipanggil dari semua objek atribut diawali tanda “+”

Ada beberapa simbol *relationship* antar class yg digunakan pada diagram class. Berikut adalah beberapa simbol-simbol yang ada pada class diagram dapat dilihat pada Tabel 2.5.

**Tabel 2.5** Simbol-simbol *Class Diagram*

No.	Simbol	Nama Simbol dan Keterangan
1.	.	<b>Assosiasi</b> adalah hubungan antar kelas.
2.	▼	<b>Generalization</b> adalah relasi antar kelas dengan makna generalisasi – spesialisasi (umum-khusus)
3.	◆	<b>Composition</b> adalah bentuk khusus dari agregasi di mana kelas yang menjadi bagian diciptakan setelah kelas menjadi <i>whole</i> dibuat. Misal kelas <i>whole</i> dihapus, maka kelas yang menjadi <i>part</i> ikut masalah.

*Multiplicity* adalah jumlah banyaknya objek sebuah kelas yang berelasi dengan sebuah objek *dari* kelas lain yang berasosiasi dengan kelas tersebut (Ipo Novianto 2017). Macam-macam *multiplicity* disajikan dalam Tabel 2.6.

**Tabel 2.6** Macam-macam *Multiplicity*

Nilai Kardinalitas	Arti
0 .. 1	Nol atau satu
1	Hanya satu
0 .. *	Not atau lebih
1 .. *	Satu atau lebih
N	Hanya n (dengan $n > 1$ )
0 .. n	Nol sampai n (dengan $n > 1$ )
1	Satu n (dengan $n > 1$ )



## 9. Prototype

*Prototype* merupakan salah satu dari implementasi sebuah desain produk yang akan dibangun. *Prototype* adalah sebuah bukti fisik atau konsep dari sebuah konsep perancangan. *Prototype* dapat membantu untuk menentukan apakah pengembang aplikasi telah berada pada jalur yang benar selama proses pengembangan. Pelatih dan pengembang profesional menggunakan teknik *prototyping* untuk menghasilkan umpan balik dari pengguna akhir selama proses perancangan sebelum akhirnya mulai membuat code aplikasi. Mendapatkan umpan balik yang kritis akan membantu dalam menghasilkan desain akhir yang pada akhirnya berdampak besar pada pengguna (Martono 2018).