

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 PERANCANGAN**

Perancangan adalah suatu proses yang bertujuan untuk menganalisis, menilai, memperbaiki dan menyusun suatu sistem (baik sistem fisik maupun non fisik) dari beberapa elemen terpisah kedalam satu kesatuan yang utuh dan berfungsi dengan memanfaatkan informasi yang ada.

Nofri Yudi Arifin, et al [2] mendefinisikan bahwa perancangan atau “design sistem merupakan tahap setelah analisa dari siklus pengembangan sistem: pendefinisian dari kebutuhan kebutuhan fungsional dan persiapan untuk rancang bangun implementasi; menggambarkan bagaimana suatu sistem dibentuk”.

Sedangkan menurut Rusdi Nur dan M. Arsyad Suyuti [3] mendefinisikan bahwa “Prancangan adalah tahapan perancangan (*design*) yang memiliki tujuan untuk mendesign sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik”.

Menurut Helmi Fauzi Siregar, Yustria Handika Siregar dan Melani [4] “Perancangan atau desain didefinisikan sebagai proses aplikasi berbagai teknik dan prinsip bagi tujuan pendefinisian suatu perangkat, suatu proses atau sistem dalam detail yang memadai untuk memungkinkan realisasi fisiknya”.

Berdasarkan definisi diatas penulis dapat menyimpulkan bahwa perancangan adalah suatu aktivitas dalam pembentukan sistem untuk membuat dan mendesign sistem yang baru dengan memanfaatkan informasi yang ada.

Serangkaian prinsip-prinsip dasar dalam perancangan / desain diusulkan oleh

A. Davis [4] yaitu sebagai “berikut:

- 1) Desain tidak boleh menderita karena tunnel vision (visi terowongan).
- 2) Desain tidak boleh berulang.
- 3) Desain harus terstruktur untuk mengakomodasi perubahan.
- 4) Desain harus terstruktur untuk berdegradasi dengan baik, bahkan pada saat data dan event-event (kejadian-kejadian) menyimpang atau menghadapi kondisi operasi.
- 5) Desain bukan pengkodean dan pengkodean bukanlah desain.
- 6) Desain harus dinilai kualitasnya pada saat desain dibuat, bahkan setelah jadi.
- 7) Desain harus dikaji untuk meminimalkan kesalahan-kesalahan konseptual (semantik)”.

## **2.2 SISTEM INFORMASI GEOGRAFIS**

Sistem informasi geografi terdiri dari tiga istilah yaitu sistem, informasi dan geografi. Sistem merupakan sekumpulan komponen atau elemen-elemen yang saling berhubungan antara komponen/elemen tersebut untuk mencapai tujuan. Informasi adalah hasil dari pengolahan data, dan geografi adalah ilmu yang mempelajari tentang permukaan bumi. Berikut ini definisi sistem informasi geografis menurut para ahli:

Jumadi, Danardono, dan Vidya N. Fikriyah [5] mendefinisikan bahwa “Sistem Informasi Geografis (SIG) atau juga dikenal sebagai *Geographic Information System* (GIS) merupakan sebuah sistem yang didesain untuk memperoleh, menyimpan, memanipulasi, menganalisis, mengatur, dan menampilkan data geografis”.

Menurut Burrough [6] “Sistem Informasi Geografis adalah sistem perangkat yang dapat melakukan pengumpulan, penyempurnaan, pengambilan kembali, transformasi dan visualisasi dari data spasial bumi untuk kebutuhan tertentu”.

Sedangkan Menurut Ni Nyoman Supuwiningsih dan Muhammad Rusli [7] mengungkapkan bahwa:

“SIG adalah sistem berbasis komputer yang memiliki kemampuan dalam menangani data bereferensi geografi, yaitu pemasukan data, manajemen data (penyimpanan dan pemanggilan kembali), manipulasi dan analisis data, serta keluaran sebagai hasil akhir (output). Hasil akhir (output) dapat dijadikan acuan dalam pengambilan keputusan pada masalah yang berhubungan dengan geografi”.

Berdasarkan definisi diatas maka dapat disimpulkan bahwa Sistem Informasi Geografis adalah sebuah sistem yang berbasis komputer yang mampu mengatasi data-data yang mempelajari tentang permukaan bumi (pemetaan bumi). SIG dapat mengintegrasikan macam-macam deskripsi lokasi dengan karakteristik-karakteristik fenomena yang ditemukan di lokasi tersebut.

Prahasta [7] mengungkapkan bahwa ada beberapa alasan yang menyebabkan konsep-konsep Sistem Informasi Geografi beserta aplikasi-aplikasinya menjadi menarik digunakan di berbagai disiplin ilmu adalah sebagai berikut:

- 1) Sistem Informasi Geografis sangat efektif dalam membantu proses pembentukan, pengembangan, atau perbaikan peta dan selalu berdampingan dengan lingkungan fisik dunia nyata yang penuh dengan kesan-kesan visual.
- 2) Sistem Informasi Geografis dapat digunakan sebagai alat bantu (tools atau tutorial) utama yang interaktif, menarik, dan menantang dalam usaha usaha untuk meningkatkan pemahaman, pengertian, pembelajaran, dan pendidikan mengenai ide - ide atau konsep - konsep lokasi, ruang, kependudukan, dan unsur-unsur geografis yang terdapat di permukaan bumi serta data atribut yang menyertainya.
- 3) Sistem Informasi Geografis menggunakan data spasial dan data non spasial (atribut) secara terintegrasi sehingga sistemnya dapat menjawab pertanyaan spasial maupun non-spasial.
- 4) Sistem Informasi Geografis memiliki kemampuan untuk menguraikan unsur - unsur yang terdapat di permukaan bumi dalam bentuk layer yang dapat memodelkan bentuk nyata di bumi.
- 5) Aplikasi Sistem Informasi Geografis dapat di-customize dengan menggunakan perintah - perintah bahasa script yang dimiliki oleh perangkat lunak Sistem Informasi Geografis yang bersangkutan sehingga memenuhi kebutuhan-kebutuhan pengguna secara otomatis, cepat, lebih menarik, informatif, dan user friendly.

## 2.3 APLIKASI MOBILE

*Mobile Application* atau dalam Bahasa Indonesia disebut sebagai perangkat bergerak, dan selanjutnya dapat disebut sebagai “*aplikasi mobile*” atau “*aplikasi*” saja, adalah perangkat lunak yang berjalan pada ponsel cerdas (*smartphone*), tablet atau perangkat yang serupa [8].

Aplikasi *mobile* merupakan aplikasi lokasi-ke-lokasi (*seluler*) yang memungkinkan penggunanya menggunakan aplikasi-aplikasi *smartphone* kapan saja, dimana saja sambil melakukan aktivitas sehari-hari. Contoh aplikasi mobile yang umum digunakan adalah game, web browser, musik, radio, dan lain sebagainya [9].

Dalam membangun aplikasi mobile, seorang pengembang (*developer*) bisa menggunakan beberapa pendekatan, yakni secara Native, Hybrid, atau Cross-Platform. Berikut ini beberapa informasi penting dari masing-masing pendekatan tersebut yaitu:

### 2.3.1 Pendekatan Secara Native

Aplikasi yang dibangun dengan pendekatan native adalah bentuk aplikasi yang ada pada perangkat bergerak (*mobile*) yang dikembangkan secara khusus dan ditulis dalam satu bahasa pemrograman dengan kumpulan kode-kode berbeda sesuai dengan sistem operasi tertentu, misalnya sistem operasi Android dan iOS. Untuk android, pengembangan aplikasinya menggunakan Java atau Kotlin, sedangkan iOS pengembangannya menggunakan Objective-C dan Swift. Karena dibangun menggunakan kode yang berbeda tersebut, aplikasi native menghasilkan kinerja yang juga baik saat *offline* dan memiliki performa terbaik dibandingkan dengan pendekatan lainnya [10].

Aplikasi native memiliki kelebihan yaitu sebagai berikut:

- 1) Performa yang sangat baik karena ditulis secara native untuk platform spesifik.
- 2) Mampu mengakses semua fitur perangkat keras smartphone, seperti info device, accelerometer, kamera, kompas, file, dan sebagainya.
- 3) Menghasilkan antarmuka look and feel yang alami dengan sangat baik.

Adapun kekurangan dari aplikasi native yaitu sebagai berikut:

- 1) Pengembangan yang tidak mudah karena menggunakan lingkungan, bahasa, dan API (*Application Programming Interface*) spesifik.
- 2) Aplikasi hanya berjalan pada platform yang sudah dispesifikasikan di awal pengembangan, apabila ingin tersedia di platform lain maka harus ditulis dari awal dengan menggunakan tool pengembangan yang sesuai.

Mengembangkan aplikasi secara native merupakan impian bagi banyak pemilik aplikasi seluler, tetapi sangat disayangkan karena tidak semua orang dapat memanfaatkan pendekatan ini. Faktor biaya adalah faktor pembatas terbesar, karena membangun dan memelihara aplikasi native membutuhkan lebih banyak waktu dan orang untuk mengerjakannya. Namun pendekatan ini sangat ideal untuk perusahaan yang sudah matang dan memiliki basis pengguna yang besar dan konsisten. Aplikasi dengan pendekatan native ini memiliki fungsionalitas yang cukup kompleks sehingga cocok untuk bisnis perusahaan yang menggunakan banyak teknologi perangkat keras yang didukung seperti sistem sidik jari, GPS, kamera, dan mikrofon [10].

### 2.3.2 Pendekatan Secara Hybrid

Pada umumnya aplikasi hybrid dibuat dengan menggunakan bahasa yang sama atau mirip dengan bahasa yang bisa digunakan untuk membuat aplikasi berbasis web, yaitu HTML, CSS, dan Javascript. Hybrid dapat mengatasi kekurangan yang ada pada aplikasi berbasis web dengan menambahkan fitur-fitur yang disediakan oleh aplikasi native. Aplikasi native dapat di-*deploy* pada berbagai platform sekaligus, sehingga dapat meminimalisasi usaha dalam proses pembuatan aplikasi. Pembangunan aplikasi dengan pendekatan hybrid memang lebih murah daripada aplikasi native, tetapi kinerjanya cenderung lebih lambat. Dalam hal ini diperlukan dukungan perangkat lunak spesifik, yaitu framework pengembangan aplikasi mobile [10].

Didik Dwi Prasetyo [11] menyatakan bahwa:

“Intuisi dari aplikasi hybrid adalah menanamkan aplikasi mobile HTML5 kedalam kontainer native. Aplikasi ini berupa mengombinasikan kelebihan-kelebihan pendekatan aplikasi *web* mobile HTML5 dan aplikasi native smartphone target.”

Rohi Abdullah [12] mengungkapkan keunggulan dari aplikasi hybrid yaitu:

- 1) Cukup membuat aplikasi satu kali untuk digunakan oleh berbagai sistem operasi smartphone.
- 2) Menggunakan bahasa yang mudah dipahami yaitu CSS3, HTML5 dan javascript yang sudah sangat familiar bagi *web* programmer.
- 3) Biaya pengembangan yang lebih mudah.
- 4) Waktu yang dibutuhkan untuk pengembangan aplikasi jauh lebih sedikit.

Keunggulan aplikasi hybrid bila dibandingkan dengan aplikasi *web* yaitu:

- 1) Proses loading dan kinerja lebih cepat.
- 2) Dapat dijalankan secara *online* maupun offline.
- 3) Dapat mengakses fitur-fitur perangkat keras smartphone, seperti kamera, galeri, kontak, dan sebagainya, serta dapat di-upload ke app store.

Adapun kekurangan dari aplikasi hybrid yaitu sebagai berikut:

- 1) Performa tidak sebaik aplikasi native.
- 2) Tampilan antarmuka pengguna terkadang memiliki perbedaan dengan tampilan native pada platform, sehingga dapat menimbulkan inkonsistensi pada tampilan antarmuka pengguna.

### **2.3.3 Pendekatan Secara Cross-Platform**

Cross-Platform atau Multi-Platform merupakan sebuah software atau aplikasi yang bisa digunakan pada sistem operasi yang berbeda, karena ditulis dengan bahasa pemrograman dan teknologi khusus sehingga aplikasi bisa berjalan pada berbagai sistem operasi seperti iOS, Android, Windows Phone, dan lain-lain. Bahasa pemrograman yang sering digunakan untuk software atau aplikasi Cross-Platform biasanya menggunakan Java, C, atau C++ dan pada umumnya software atau aplikasi cross-platform merupakan perangkat lunak bebas (*open source*). Pada saat ini, aplikasi *mobile* dengan pendekatan Cross-Platforms juga bisa dikembangkan menggunakan platform seperti Flutter [9].

Flutter merupakan produk teknologi buatan Google yang beroperasi menggunakan bahasa pemrograman Dart. Flutter merupakan sebuah kumpulan UI kits yang berbasis open source dan dapat digunakan untuk mengembangkan aplikasi yang berbasis Android, iOS, Linux, Mac, Windows, Google Fuchsia, dan website dari satu basis kode [10].

Dari ketiga pendekatan diatas, jika dibandingkan maka pengembangan aplikasi menggunakan pendekatan *Cross-Platforms* adalah pilihan yang terbaik karena bisa menyeimbangkan faktor biaya dan kinerja. Hasil akhir aplikasi yang menggunakan pendekatan *Cross-Platform* dengan Flutter lebih murah dibandingkan dengan pendekatan native. Performa aplikasi untuk aplikasi *Cross-Platforms* juga lebih unggul dibandingkan dengan pendekatan hybrid.

## 2.4 INTERNET

Verra Scofica [13] mengungkapkan bahwa:

“Internet sebetulnya singkatan dari kata *Interconnected Networking*. *Networking* artinya jaringan, sedang *Interconnected* berarti saling berkaitan/terkoneksi. Sehingga internet adalah jaringan komputer yang saling terkoneksi”.

Sedangkan Janner Simarmata [14] mengungkapkan bahwa:

“Internet adalah kelompok atau kumpulan dari jutaan komputer. Penggunaan Internet memungkinkan kita untuk mendapatkan informasi dari komputer yang ada didalam kelompok tersebut dengan asumsi bahwa pemilik komputer memberikan izin akses. Untuk mendapatkan sebuah informasi, sekumpulan protokol harus digunakan, yaitu sekumpulan aturan yang menetapkan bagaimana suatu informasi dapat dikirim dan diterima.

Suminar Pujawati dan Bambang Bagus Harianto [15] menjelaskan bahwa “Internet dapat dikatakan sebagai suatu jaringan komputer yang sangat besar, dimana jaringan komputer tersebut terdiri dari beberapa jaringan kecil yang saling terhubung satu sama lain dan dapat saling bertukar informasi”.

Dani Ainur Rivai dan Sukadi [15] mengungkapkan bahwa “Internet merupakan sekumpulan jaringan yang terhubung satu dengan lainnya, dimana jaringan menjadikan sambungan menuju global informasi”.

Berdasarkan beberapa definisi diatas, maka dapat disimpulkan bahwa internet adalah singkatan dari *interconnected-networking* yang dapat menghubungkan masing-masing jaringan komputer agar dapat saling berkomunikasi dengan jaringan secara global. Internet mampu menghubungkan komputer diseluruh dunia sehingga memudahkan untuk mendapatkan segala informasi.

## **2.5 ALAT BANTU PEMODELAN SISTEM**

Dalam memodelkan sebuah sistem, dibutuhkan suatu metode yang dapat membantu dalam proses analisis sistem dan perancangan sistem. Penulis menggunakan model UML (*Unified Modelling Language*) yang merupakan sekumpulan simbol dan diagram yang dapat memodelkan perangkat lunak untuk diterjemahkan kedalam kode program menggunakan bahasa pemrograman yang mendukung pemrograman berorientasi objek. Tujuan pemodelan ini adalah untuk memahami perilaku sistem yang akan dirancang dengan lebih mendetail dan menggambarkan sistem secara lengkap dan sederhana sehingga dapat digunakan untuk menjelaskan jalanya sistem.

Rosa A. S. and M. Shalahuddin [17] mengungkapkan bahwa “*Unified Modelling Language (UML)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Dalam jurnal Tri A. Kurniawan, Object Management Group (OMG) mengabarkan bahwa penggunaan UML tingkat lanjut didukung oleh konsep pemodelan yang lebih matang yang dirumuskan dalam setiap versi spesifikasi UML yang dikembangkan oleh OMG. UML dikembangkan untuk menyediakan perangkat yang dibutuhkan oleh para pengembang perangkat lunak dalam menganalisis, merancang, dan mengimplementasikan sistem berbasis perangkat lunak [18].

Pada pembahasan ini, UML yang akan penulis gunakan yaitu *use case diagram*, *activity diagram*, dan *class diagram*. Masing-masing diagram tersebut akan diuraikan dalam penjelasan berikut ini:

### **2.5.1 Use Case Diagram**

*Use case diagram* sangat membantu dalam penyusunan kebutuhan (*requirement*) sebuah sistem dan perancangan semua fitur-fitur yang terdapat dalam sistem dan dapat mendeskripsikan apa yang seharusnya dilakukan oleh sebuah sistem.

Tri A. Kurniawan [18] menjelaskan bahwa:

“Salah satu diagram penting yang digunakan untuk mengilustrasikan kebutuhan (*requirements*) dari sistem adalah *use case* (UC) diagram, yang menjelaskan secara visual konteks dari interaksi antara aktor dengan sistem. Setiap *use case* menyatakan spesifikasi perilaku (fungsionalitas) dari sistem yang sedang dijelaskan yang memang dibutuhkan oleh aktor untuk memenuhi tujuannya”.

Rosa A. S and M. Salahuddin [19] mengungkapkan bahwa:

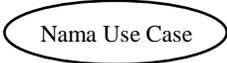
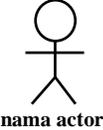
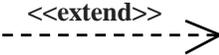
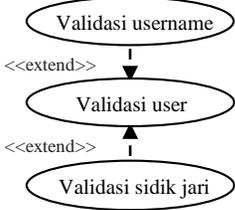
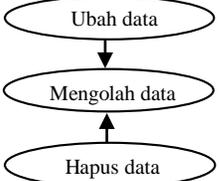
“*Use case* atau diagram *use case* merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu”.

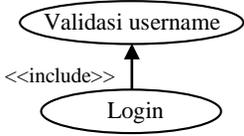
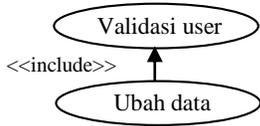
Rosa A. S dan M. Salahuddin [19] mengungkapkan bahwa “syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case* yaitu sebagai berikut:

- 1) Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- 2) *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar esan antarunit atau aktor.

Berdasarkan beberapa definisi diatas, maka dapat disimpulkan bahwa *use case diagram* dapat menggambarkan dan memodelkan interaksi antara satu atau lebih aktor dalam sistem informasi yang akan dibuat. *Use case diagram* juga dapat mendeskripsikan apa yang akan dilakukan oleh sistem. *Use case diagram* dibuat menggunakan simbol-simbol yang dapat dilihat pada tabel 2.1 berikut:

**Tabel 2.1 Simbol Use Case Diagram**  
Menurut Rosa A. S dan M. Shalauddin [19]

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p>
<p><b>Aktor / Actor</b></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.</p>
<p><b>Asosiasi / Association</b></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> yang memiliki interaksi dengan aktor.</p>
<p><b>Ektensi / Extend</b></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritence</i> pada pemograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan misal</p>  <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan.</p>
<p><b>Generalisasi / generalization</b></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p> 

	Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)
<p><b>Menggunakan / <i>include</i> / <i>uses</i></b></p> <p><code>&lt;&lt;include&gt;&gt;</code> →</p> <p>→ <code>&lt;&lt;uses&gt;&gt;</code></p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan semua fungsinya atau sebagai syarat dijalankan <i>use case</i> ini,</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ul style="list-style-type: none"> <li>• <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</li> </ul>  <pre> graph TD     Login([Login]) -- "&lt;&lt;include&gt;&gt;" --&gt; ValidasiUsername([Validasi username])   </pre> <ul style="list-style-type: none"> <li>• <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, missal pada kasus berikut:</li> </ul>  <pre> graph TD     UbahData([Ubah data]) -- "&lt;&lt;include&gt;&gt;" --&gt; ValidasiUser([Validasi user])   </pre> <p>Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

### 2.5.2 Activity Diagram

*Activity diagram* dapat merepresentasikan grafis dari seluruh tahapan alur kerja. Diagram ini digunakan untuk menjelaskan macam-macam langkah alir dari aktifitas yang akan dirancang dalam sebuah sistem. Dimana masing-masing alir memiliki awal, kemudian *decision* yang mungkin terjadi pada sistem, dan akhir dalam sistem tersebut.

A. I. S. Azis Zohrahayaty dan Y. A. Mustofa [20] mengungkapkan bahwa: “Activity Diagram menggambarkan setiap aktivitas yang ada pada sistem. Setiap aktivitas sistem mewakili suatu activity diagram”.

Rosa A. S. dan M. Salahuddin [19] mengungkapkan bahwa:

“Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktifitas apa yang dapat dilakukan oleh sistem”.

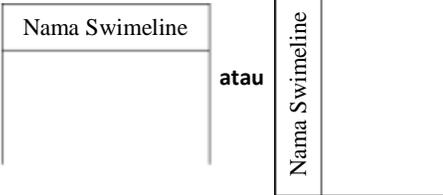
E. Triandini dan I. G. Suardika [21] mengungkapkan bahwa: “*Activity Diagram* adalah Sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau sistem), orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut.

*Activity diagram* juga banyak digunakan untuk mendefinisikan hal-hal seperti berikut [19]:

- 1) Rancangan proses bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- 2) Urutan atau pengelompokan tampilan dari sistem / user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- 3) Rancangan pengujian di mana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
- 4) Rancangan menu yang ditampilkan pada perangkat lunak.

Sehingga dapat disimpulkan bahwa *activity diagram* adalah suatu diagram yang menggambarkan aliran aktifitas yang sedang dirancang untuk memodelkan event-event yang terjadi dalam suatu *use case*. *Activity diagram* dibuat menggunakan simbol-simbol yang dapat dilihat pada tabel 2.2 berikut:

**Tabel 2.2 Simbol *Activity Diagram*  
Menurut Rosa A. S dan M. Shalahuddin [19]**

Simbol	Deskripsi
<b>Status awal</b> 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
<b>Aktivitas</b> 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
<b>Percabangan / <i>decision</i></b> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
<b>Penggabungan / <i>join</i></b> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
<b>Status akhir</b> 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
<b>Swimline</b> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

### 2.5.3 *Class Diagram*

*Class diagram* adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang digunakan. Class diagram memberikan gambaran (*diagram statis*) tentang sistem/perangkat lunak dan relas-relasi yang ada didalamnya.

Rosa A. S dan M. Shalahuddin [19] mengungkapkan bahwa: “Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi”.

Sedangkan A. I. S. Azis Zohrahayaty dan Y. A. Mustofa [20] mendefinisikan bahwa: “*Class Diagram* menggambarkan setiap objek atau data yang terdapat pada sistem. Setiap objek/data tersebut memiliki anggota, baik itu berupa *attributes (fields dan properties)*, *operations (methods)*, dan *events*”.

Surmayati [22] juga mendefinisikan bahwa:

“Class diagram adalah sebuah class yang menggambarkan struktur dan penjelasan class, paket, dan objek serta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Class diagram juga menjelaskan hubungan antar class dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan”.

Dadang Amiruddin, et al, [23] mengungkapkan bahwa: “*Class Diagram* yaitu salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa *class diagram* adalah gambaran yang menampilkan suatu hubungan dari struktur, kelas, paket, dan

objek dari sebuah sistem atau alat perancangan untuk pengembangan system. *Class diagram* dibuat menggunakan simbol-simbol yang dapat dilihat pada tabel 2.3 berikut:

**Tabel 2.3 Simbol *Class Diagram***  
Menurut Rosa A. S dan M. Shalahuddin [19]

Simbol	Deskripsi
<p><b>Kelas</b></p> 	Kelas pada struktur sistem
<p><b>Antarmuka / <i>interface</i></b></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
<p><b>Asosiasi / <i>association</i></b></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p><b>Asosiasi bararah / <i>directed association</i></b></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p><b>Generalisasi</b></p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
<p><b>Kebergantungan / <i>dependency</i></b></p> 	Relasi antar kelas dengan makna kebergantungan antar kelas

<p><b>Agregasi / <i>aggregation</i></b></p> 	<p>Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)</p>
---	---

## 2.6 ALAT BANTU PERANCANGAN SISTEM

### 2.6.1 Android Studio

Romanus Damanik, Parasian D. P Silitonga, dan Wasit Ginting [24] menjelaskan bahwa Android Studio adalah *IDE / Integrated Development Environment* (Lingkungan Pengembangan Terpadu) yang resmi untuk membuat aplikasi Android, yang didasarkan pada IntelliJ IDEA. Android Studio dibentuk dari IDE Java terkenal yaitu IntelliJ IDE. Android Studio adalah IDE resmi untuk pengembangan Android, tetapi itu tidak berarti pengembang tidak dapat menggunakan metode lain. Dan tentu saja, untuk menginstal Android Studio IDE, Anda membutuhkan aplikasi Android Studio terlebih dahulu. Pengembang juga dapat menggunakan Android Studio untuk membuat aplikasi Flutter [24].

Selain sebagai editor kode dan fitur developer IntelliJ yang handal, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas dalam membuat aplikasi Android, seperti:

- 1) Sistem build berbasis Gradle yang fleksibel.
- 2) Emulator yang cepat dan kaya fitur.
- 3) Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat Android.

- 4) Terapkan Perubahan untuk melakukan push pada perubahan kode dan resource ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi.
- 5) Template kode dan integrasi GitHub untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel.
- 6) Framework dan fitur pengujian yang lengkap.
- 7) Fitur lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya.
- 8) Dukungan C++ dan NDK.
- 9) Dukungan bawaan untuk Google Cloud Platform, yang memudahkan integrasi Google.
- 10) Cloud Messaging dan App Engine.

### **2.6.2 Framework Flutter**

Sigit Suryono and Hardiansah [25] menjelaskan tentang flutter bahwa “Flutter merupakan sebuah toolkit/framework yang dikeluarkan dan dikembangkan oleh Google untuk membuat sebuah aplikasi multi-platform baik mobile, web ataupun desktop dari sebuah basis code”.

Flutter memiliki kelebihan, diantaranya yaitu Fast Development, Expressive dan Flexible UI serta Native Performance. Fast Development yang berarti proses pengembangan berlangsung cepat. Flutter tidak membangun kembali setiap status yang ada dalam aplikasi, karena Flutter menyediakan fitur hot reload untuk membantu dalam membangun UI, menambahkan fitur, dan memperbaiki bugs. Expressive and Flexible

UI menyediakan berbagai API pilihan, scrolling alami, dan *platform awareness*, serta antarmuka pengguna yang indah untuk Android (*Material Design*) dan iOS (Cupertino) yang indah [25].

Aplikasi yang dibangun dengan Flutter ditulis menggunakan bahasa pemrograman Dart. Dart sendiri merupakan bahasa pemrograman yang dikeluarkan oleh Google dan digunakan untuk membuat aplikasi berbasis mobile, desktop, backend dan web. Dart merupakan bahasa pemrograman yang berorientasi pada objek serta menggunakan gaya penulisan bahasa C yang terkompilasi secara opsional kedalam bahasa pemrograman JavaScript. Kini framework Flutter (dengan bahasa pemrograman Dart) bisa digunakan untuk berbagai platform, berikut alasan kenapa flutter penting bagi developer [25]:

- 1) Flutter itu Indah, Flutter dulu diciptakan dari janji ide ngawang bahwa seharusnya ada cara lebih baik untuk membuat user interface (UI) aplikasi. Di Google I/O 2019, Google telah menyatakan sukses memenuhi janji itu. aplikasi-aplikasi yang dihasilkan flutter bisa jalan dengan 60 *frame-per-secon* dan responsif kepada pengguna aplikasi.
- 2) Flutter itu Adaptif untuk Mobile, Pada aplikasi yang sama tapi dalam platform yang berbeda seperti iOS dan Android, tidak ada perubahan pada codingan sama sekali akan tetapi sekilas sangat mirip. Flutter otomatis akan merubah aplikasi yang dirancang agar cocok dengan operating system yang digunakan. Aplikasi jadi terlihat seakan akan dibuat khusus untuk operating system tersebut.

- 3) Flutter ada Stateful Hot Reload, Stateful hot reload adalah fitur di mana codingan bisa langsung muncul di virtual machine. Pengguna bisa lihat langsung hasil dari perubahan yang dilakukan. Hal ini tentu dapat membantu untuk melihat perubahan estetika seperti merubah warna background.
- 4) Flutter bisa untuk ChromeOS, Ketika mencoba agar sebuah produk aplikasi bisa ada versi desktop, hal yang paling susah adalah memastikan UI yang tadinya di layar kecil bisa terlihat bagus di layar besar (layar desktop). Selain itu harus ada perubahan pada bagaimana aplikasi menerima dan mengenali input dari pengguna. Dengan Flutter, hanya perlu menambahkan beberapa kode baris seperti MediaQuery.
- 5) Flutter bisa untuk Web, Tentu keahlian untuk membuat aplikasi web akan sangat berguna mengingat 4.4 miliar orang adalah pengguna aktif internet dan memiliki akses ke web. Untungnya semua widget yang disediakan Flutter untuk mobile dapat digunakan dalam membuat aplikasi web. Dart akan diterjemahkan ke HTML, CSS dan Canvas. Selanjutnya akan melalui Dart2js *compiler* sebelum menjadi JavaScript yang dapat dibaca oleh browser.

### 2.6.3 Google Maps API

Andy Juansyah [26] mendefinisikan Google Maps API sebagai “layanan dari google yang mempermudah pgunanya untuk melakukan kemampuan pemetaan untuk aplikasi yang dibuat”.

Sedangkan Danang Adi Sumarto dan Tedy Setiadi [27] menjelaskan bahwa “Google Maps API adalah Library Java Script. Dengan menggunakan google Maps API jadi kita hanya beronsentrasi tentang data dan biarkan urusan peta ditangani oleh google, sehingga dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal”.

Google Maps API merupakan library khusus yang disediakan Google Maps sendiri untuk para developer atau programmer dalam mengembangkan sistem/aplikasinya. API disini adalah kepanjangan dari *Application Programming Interface* dimana semua aplikasi yang menggunakan data secara realtime selalu menggunakan API. Dalam perkembangannya Google Maps API diberikan kemampuan untuk mengambil gambar peta statis. Melakukan geocoding, dan memberikan penuntun arah. Google Maps API bersifat gratis untuk publik [28].

#### **2.6.4 MariaDB**

MariaDB adalah software sistem manajemen basis data yang nyaris sama dengan MySQL, karena pengembang dari MariaDB ini adalah founder dari MySQL itu sendiri yaitu Michael Monty Widenius. Alasan dia untuk membuat MariaDB adalah karena MySQL telah di ambil alih oleh ORACLE sehingga menyebabkan MySQL menjadi produk yang berlisensi dan nantinya akan menjadi product komersil. Untuk penamaan MariaDB ini diambil dari salah satu nama anak Monty yaitu Maria, dan MySQL juga dinamai dengan salah satu nama anaknya yaitu My. Database MySQL

dan MariaDB juga dapat dikelola lebih mudah melalui antarmuka (*interface*) grafis dengan menggunakan aplikasi berbasis web yaitu phpMyAdmin [29].

MariaDB merupakan aplikasi yang dapat digunakan untuk memproses data SQL dalam waktu yang bersamaan. MariaDB juga bisa digunakan sebagai aplikasi database untuk ranah yang lebih besar dimana jumlah rekamannya bisa mencapai kurang lebih 50 juta. Berikut ini terdapat fungsi dari database MariaDB yaitu [29]:

- 1) Untuk menghubungkan client dengan menggunakan protokol TCP / IP, Named Pipes atau NT, dan juga UNIX Socket.
- 2) Untuk mendeteksi pesan error yang diterima oleh klien dengan setidaknya menggunakan 20 bahasa yang berbeda - beda. Meskipun memang untuk bahasa Indonesia masih belum termasuk ke dalam 20 bahasa tersebut.
- 3) Untuk menyediakan data yang bermacam-macam yang tentunya akan mempermudah pengguna ketika ingin menggunakan aplikasi database yang bukan hanya mampu menangani skala besar, melainkan juga menyediakan banyak pilihan data.
- 4) MariaDB memiliki fungsi multi-user. Artinya, aplikasi ini bisa digunakan oleh banyak user dalam waktu yang bersamaan. Inilah mengapa software yang satu ini lebih banyak dipilih dibandingkan dengan yang lainnya.

### **2.6.5 XAMPP**

XAMPP didefinisikan oleh Bay Haqi dan Heri Satria Setiawan [30] bahwa:

“XAMPP adalah perangkat lunak bebas (Free Software), yang mendukung banyak system operasi, merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri sebagai server yang berdiri sendiri (*localhost*), yang terdiri dari beberapa program antara lain: Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl”.

Wicaksono [31] juga mendefinisikan bahwa "XAMPP adalah sebuah software yang berfungsi untuk menjalankan website berbasis PHP dan menggunakan pengolahan data MYSQL di komputer lokal".

Sedangkan menurut Siregar & Sundari [32] “Xampp merupakan salah satu paket instalasi Apache, PHP, dan MySQL dengan secara instan yang dapat digunakan untuk membantu proses instalasi yang instan”.

Nama XAMP merupakan singkatan dari X (empat sistem operasi dari jenis apapun), Apache, MySQL, PHP, dan Perl. Program ini berbasis GNU General Public License yang merupakan web server gratis dan mudah digunakan, serta dapat memberikan tampilan halaman web yang dinamis [30]. XAMPP berperilaku seperti server web di komputer. XAMPP juga bisa disebut sebagai Cpanel server virtual yang dapat membantu pengguna untuk melihat pratinjau sehingga pengguna dapat memodifikasi situs web tanpa harus terhubung atau mengakses melalui internet [31].

Dapat disimpulkan bahwa XAMPP adalah suatu software open source yang didaamnya terdapat software-software pembantu seperti Apache, MYSQL, PHP, dan php myadmin, yang dapat digunakan sebagai alat bantu perancangan ataupun pengembangan sistem berbasis PHP.

## 2.7 PENELITIAN SEJENIS

Penelitian ini dilakukan tentunya tak lepas dari penelitian sejenis yang telah dahulu. Penelitian sejenis ini penulis gunakan sebagai bahan perbandingan atau acuan dalam penulisan karya ilmiah. Output-output penelitian yg dijadikan acuan juga tak lepas dari topik penelitian yang penulis lakukan yaitu tentang Sistem Informasi Geografis. Berikut ini merupakan beberapa penelitian terdahulu yang penulis jadikan sebagai bahan acuan dalam penulisan karya ilmiah yaitu sebagai berikut:

**Tabel 2.4 Penelitian Sejenis**

No	Nama dan Judul Peneliti	Metode Penelitian	Hasil
1	Nama: Ferly Ardhy  Judul: “ <i>Sistem Informasi Geografis Penyedia Jasa Rumah Kos Berbasis Website (Studi Kasus: Wilayah Kotabumi Lampung Utara)</i> ” [33].	Metode: <i>Prototyping</i>  Teknik Pengumpulan Data: 1. Observasi 2. Wawancara 3. Studi Pustaka	Website Sistem Informasi Geografis tentang jasa rumah kost di Kotabumi yang telah berhasil dibangun sangat membantu para pencari jasa rumah kost untuk mengetahui lokasi rumah kost dan informasi jasa rumah kost. Dengan adanya Website ini juga dapat membantu bagi pemilik penyedia jasa rumah kost dalam memasarkan jasa rumah kost yang dimilikinya.
2	Nama: Rika Oviani  Judul: “ <i>Perancangan Aplikasi Sistem Informasi Geografis</i> ”	Metode: <i>Waterfall (Air Terjun)</i>  Teknik Pengumpulan Data:	Sistem yang dihasilkan dapat memberi kemudahan dan menghemat waktu dalam mencari informasi tentang pemetaan lokasi

No	Nama dan Judul Peneliti	Metode Penelitian	Hasil
	<i>Pencarian Rumah Kos dan Kontrakan Sekitaran Kampus di Kota Jambi Berbasis Android</i> ” [34].	1. Observasi 2. Kuesioner	dari kos dan kontrakan di wilayah sekitaran kampus STIKOM Dinamika Bangsa dan UNBARI dengan tampilan sistem yang menarik dan mudah dipahami. Dan akan sangat membantu untuk melakukan navigasi di dalam smartphone yang berbasis Android.
3	Nama: Dani Bagus Saputra  Judul: “ <i>Perancangan Aplikasi GIS Panduan Informasi Bengkel Resmi Kendaraan Bermotor Berbasis Android Di Kota Jambi</i> ” [35].	Metode: <i>Waterfall (Air Terjun)</i>  Teknik Pengumpulan Data: 1. Wawancara 2. Observasi	Penelitian ini menghasilkan sebuah aplikasi Mobile GIS Panduan Lokasi dan Informasi bengkel resmi kendaraan bermotor di Kota Jambi sehingga membantu pengguna mencari lokasi bengkel resmi kendaraan bermotor terdekat yang ada di kota jambi.
4	Nama: Ramadhan Febri Andani  Judul: “ <i>Perancangan Aplikasi Sistem Informasi Geografis Lokasi Praktek Dokter Berbasis Android di Kota Jambi</i> ” [36].	Metode: <i>Waterfall (Air Terjun)</i>  Teknik Pengumpulan Data: 1. Wawancara	Aplikasi yang dirancang menyajikan gambar, suara, dan lokasi dalam aplikasi, dan dapat digunakan oleh siapapun dengan menggunakan smartphone, aplikasi tersebut dapat mengakses lokasi praktek dokter, dan memberikan informasi sekilas tentang kesehatan.

No	Nama dan Judul Peneliti	Metode Penelitian	Hasil
5	Nama: Deddy Ackbar Rianto  Judul: “Perancangan Aplikasi Sistem Informasi Geografis (SIG) Lokasi Minimarket di Kota Jambi Berbasis Android” [37].	Metode: <i>Waterfall (Air Terjun)</i>  Teknik Pengumpulan Data: 1. Penelitian Pustaka 2. Penelitian Labor 3. Dokumentasi	Penelitian ini menghasilkan sebuah aplikasi mobile GIS berbasis Android yang interaktif, dapat diakses oleh siapa saja, dimana saja, dan kapan saja. Dapat memudahkan dalam mencari informasi tentang pemetaan lokasi minimarket yang ada di Kota Jambi dengan tampilan yang menarik dan mudah digunakan.

Pada penelitian ini penulis mengambil judul tentang “Perancangan Sistem Informasi Geografis Rumah Kost dan Kontrakan Berbasis Mobile di Kota Jambi. Dari penelitian sejenis yang dijelaskan dalam tabel 2.4 diatas, maka dapat disimpulkan bahwa penelitian sejenis tersebut memiliki perbedaan dengan penelitian yang dilakukan oleh penulis, perbedaan tersebut terletak pada:

1. Rancangan sistem informasi geografis yang dibangun yaitu aplikasi berbasis mobile yang dapat diakses oleh pengguna Android dan iOS.
2. Konten dari sistem informasi geografis yang dibangun pada penelitian ini yaitu tentang rumah kost dan kontrakan.
3. Objek penelitian yang penulis ambil yaitu di wilayah Kota Jambi.
4. Penulis menggunakan metode penelitian Waterfall (Air Terjun).