

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu merupakan salah satu acuan dalam proses penyelesaian tugas akhir/riset ini, maka akan diterapkan riset – riset yang relevan dalam topik pembahasan tugas akhir ini guna untuk memperdalam dan melanjutkan penelitian – penelitian yang ada dan juga sebagai bahan referensi otentik atau bukti nyata dalam proses penyelesaian tugas akhir ini. Beberapa penelitian yang di ambil oleh peneliti sebagai nilai acuan dan juga referensi tertulis adalah sebagai berikut :

1. Penelitian yang dilakukan oleh (Mutmainna & Andi Mashdarul Khair, 2022) mengenai “Rancang Bangun Aplikasi Pencarian Bengkel Aktif Berbasis Google Maps API Di Wilayah Kota Makassar” hasil dan kesimpulan penelitian menunjukkan bahwa aplikasi yang dirancang berguna dalam memudahkan *customer* untuk mencari lokasi bengkel yang aktif.
2. Penelitian yang dilakukan oleh (Liwang, Rasyid, 2013) mengenai “Rancang Bangun Aplikasi Pencarian Rute Terpendek Tempat Wisata Dengan Memanfaatkan Google Maps API (studi kasus: kabupaten kulon progo)” hasil dan kesimpulan penelitian menunjukkan bahwa aplikasi membantu para calon wisatawan untuk mencari perjalanan tujuan wisata mereka dalam menentukan jarak dan waktu tempuh ke tempat-tempat wisata yang dituju.
3. Penelitian yang dilakukan oleh (Daniel Oktodeli Sihombing, 2015) mengenai “Perancangan Aplikasi Web Untuk Pencarian Lokasi Dan Rute Rumah Sakit Berbasis Google Maps API” hasil dan kesimpulan penelitian

menunjukkan bahwa aplikasi web yang dirancang dapat menampilkan informasi mengenai rumah sakit yang ada di kota Pontianak dan mengkonversi koordinat latitude dan longitude rumah sakit ke peta digital menggunakan Google Maps API.

Berdasarkan penelitian – penelitian di atas dapat disimpulkan bahwa penggunaan dan pengimplementasian aplikasi berbasis web dapat memberikan banyak manfaat baik terhadap pribadi maupun Masyarakat tertentu dalam industri informasi teknologi dari berbagai segmentasi yang ada.

Berikut adalah beberapa hasil penelitian yang telah penulis riset dan pelajari sebelumnya sebagai bahan acuan dan referensi tertulis dalam proses penyelesaian tugas akhir ini.

Tabel 2. 2 Perbandingan Penelitian Terdahulu

No.	Identitas	Hasil	Persamaan	Perbedaan
1.	(Mutmainna & Andi Mashdarul Khair,2022) Rancang Bangun Aplikasi Pencarian Bengkel Aktif Berbasis Google Maps API Di Wilayah Kota Makassar	aplikasi yang dirancang berguna dalam memudahkan <i>customer</i> untuk mencari lokasi bengkel yang aktif.	Penggunaan <i>framework laravel</i>	Metode penelitian, desain sistem
2.	(Liwang, Rasyid, 2013) Implementasi Aplikasi Sistem Informasi Desa Dan Kependudukan	aplikasi membantu para calon wisatawan untuk mencari perjalanan tujuan wisata mereka	Menggunakan aplikasi berbasis website <i>PHP</i>	Teknik pengumpulan data dan adanya <i>domain</i> serta <i>hosting</i>

	Berbasis Web Di Desa Kukuh Kerambitan	dalam menentukan jarak dan waktu tempuh ke tempat-tempat wisata yang dituju.		
3.	(Daniel Oktodeli Sihombing, 2015) Perancangan Aplikasi Web Untuk Pencarian Lokasi Dan Rute Rumah Sakit Berbasis Google Maps API	dapat menampilkan informasi mengenai rumah sakit yang ada di kota Pontianak dan mengkonversi koordinat latitude dan longitude rumah sakit ke peta digital menggunakan Google Maps API	Teknik pengumpulan data	Pengujian aplikasi (<i>testing</i>)

2.2 Sistem Informasi

Sistem informasi merupakan kegiatan pengolahan data yang dapat diawali dengan mengumpulkan informasi, memprosesnya, menganalisis informasi yang didapat, menyimpan informasi, selanjutnya menyebarkan informasi yang telah disaring dari proses sebelumnya untuk kemanjuran serta kepentingan suatu individu maupun organisasi. (Nadialista Kurniawan, 2021). Rahman, Sustyo, & Primasari (2019) mendefinisikan sistem informasi bahwa, suatu sistem dalam suatu organisasi yang menyatukan kebutuhan pengolahan transaksi harian, mendukung operasi,

bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. UML merupakan singkatan dari *Unified Modeling Language*.

2.3 Tactical Floor Game

Tactical Floor Game (TFG) merupakan kegiatan untuk mensimulasikan taktik dan menguji rencana operasi pertempuran. Kegiatan ini merupakan bagian penting dalam proses latihan di Sekolah Komando dan Staf (Sesko) Angkatan Darat (AD), Angkatan Laut (AL), Angkatan Udara (AU), dan Tentara Nasional Indonesia (Sesko TNI). Latihan *TFG* kadang-kadang dilakukan dengan menggunakan skema gabungan antara TNI dan Kepolisian Republik Indonesia untuk mereplikasi rencana operasional di ranah sipil sebagai bagian dari pengembangannya. *TFG* bertujuan untuk menguji rencana operasi, meliputi simulasi perencanaan, simulasi pergerakan, simulasi koordinasi antar unit, serta simulasi akomodasi dan perbekalan untuk memastikan operasi dapat berjalan sesuai teknik. Pelaksanaan kegiatan *TFG* dibagi menjadi beberapa tahapan antara lain koordinasi dengan komandan, melakukan perencanaan pergerakan dan logistik, melakukan perencanaan pertempuran, dan konsolidasi pasca operasi. Seluruh tahapan kegiatan ini dilakukan sebagai respons terhadap skenario berbasis cerita tertentu dan didokumentasikan dalam dokumen kerja yang nantinya akan dipresentasikan kepada komandan tertinggi dan disimulasikan oleh perwira senior.

Hal ini penting karena pemain terlibat dengan berbagai skenario atau studi kasus pertempuran yang harus ditangani. Para peserta harus mengembangkan dan menerapkan rencana taktis dan strategis yang dipikirkan dengan matang untuk

menghadapi tantangan-tantangan ini. Rencana-rencana ini tidak hanya didokumentasikan dalam dokumen kerja yang komprehensif tetapi juga diwujudkan melalui visualisasi medan perang menggunakan peta kertas dan model miniatur untuk mewakili unit sebenarnya.

2.4 *Laravel*

Laravel adalah kerangka web berbasis *PHP open source* gratis yang dibuat oleh Taylor Otwell untuk mengembangkan aplikasi web menggunakan paradigma MVC. Struktur pola MVC pada *Laravel* sedikit berbeda dengan struktur pola MVC pada umumnya. Pada *Laravel* terdapat *routing* yang menghubungkan permintaan user dan *controller*. Oleh karena itu, pengontrol data tidak segera menerima permintaan tersebut. (Informatika et al., 2023).

2.5 *Hypertext Pre-processor (PHP)*

PHP adalah bahasa pemrograman web yang mampu memproses data secara dinamis. PHP dianggap sebagai bahasa skrip sisi *server* yang terintegrasi, artinya sintaks dan perintah yang kami berikan akan dieksekusi sepenuhnya oleh *server* tetapi disertakan dalam halaman HTML standar. (Usada et al., 2012)

2.6 *Google Maps API*

Google Maps adalah layanan gratis yang diberikan oleh Google dan sangat populer. Google Maps adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, Google Maps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu browser. Kita dapat menambahkan fitur Google Maps dalam web yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan Google Maps API. Google Maps API

adalah suatu library yang berbentuk JavaScript.(Kindarto, 2008).

2.7 UML

Unified Modeling Language (UML) adalah metode pemodelan visual untuk merancang sistem berorientasi objek atau definisi UML adalah bahasa standar untuk memvisualisasikan, merancang, dan mendokumentasikan sistem. Dalam sistem berorientasi objek, pengembang menggunakan bahasa pemodelan untuk mendeskripsikan, membangun, dan mendokumentasikan sistem yang mereka desain. UML memungkinkan anggota tim untuk bekerja sama menggunakan bahasa pemodelan yang sama untuk menerapkan sistem yang berbeda. (Affandi & Syahputra, 2018).

Adapun beberapa bagian pada UML yang peneliti ambil sebagai bahan rancangan yang nantinya akan diimplemmentasikan dalam pembahasan – pembahasan selanjutnya, antara lain *use case diagram*, *activity diagram*, *class diagram*, dan beberapa bagan atau diagram lain, berikut adalah penjelasan dari setiap diagram yang telah disebutkan :

1. Use Case Diagram

Use Case diagram, merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* menggambarkan interaksi satu atau lebih aktor dengan sistem yang dibuat dan menggambarkan bagaimana sistem tersebut akan digunakan atau digunakan oleh administrator. Aktor merupakan orang yang berinteraksi

dengan sistem, sehingga *use case* merupakan abstraksi dan interaksi antara sistem dan aktor. (Affandi & Syahputra, 2018).

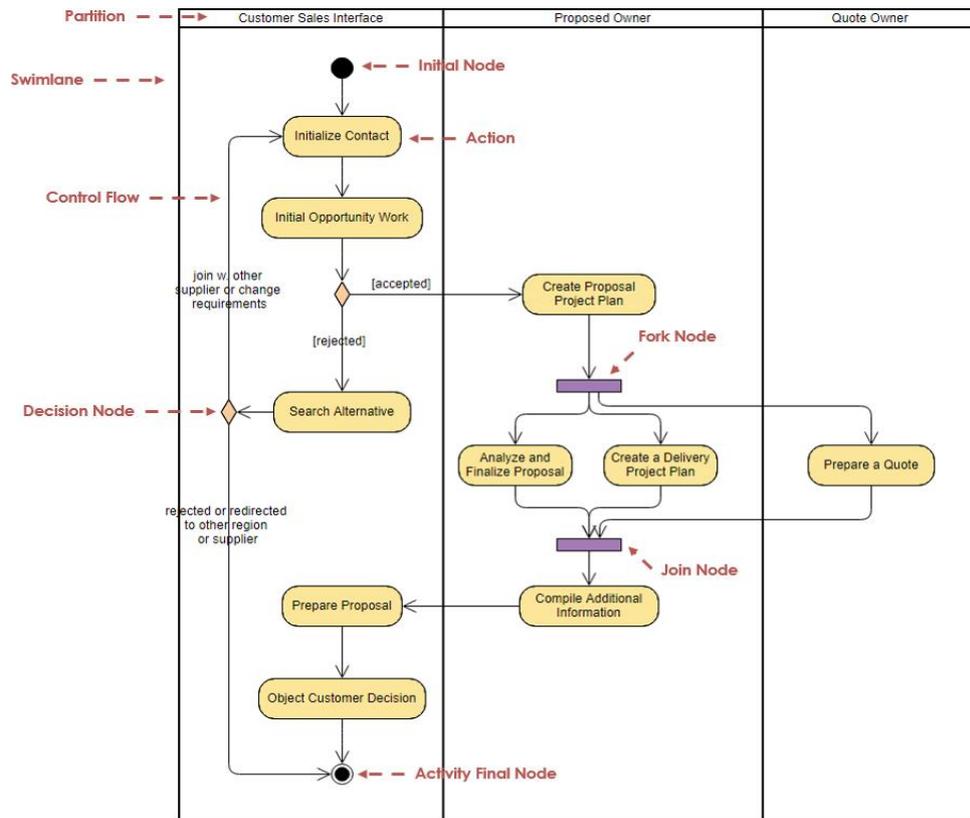
SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang 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.

Gambar 2. 4 Tabel Simbol - simbol *use case diagram*

(Sumber : www.dumetschool.com)

2. Activity Digram

Acitivity Digram menggambarkan serangkaian aliran aktivitas, digunakan untuk menggambarkan aktivitas yang dibentuk dalam suatu aktivitas sehingga dapat digunakan untuk aktivitas lain seperti *use case* atau interaksi. *Acitivity Digram* berupa diagram digunakan untuk menunjukkan proses kerja sistem. *Acitivity Digram* juga dapat menggambarkan proses paralel yang mungkin terjadi dalam beberapa eksekusi. (Affandi & Syahputra, 2018).

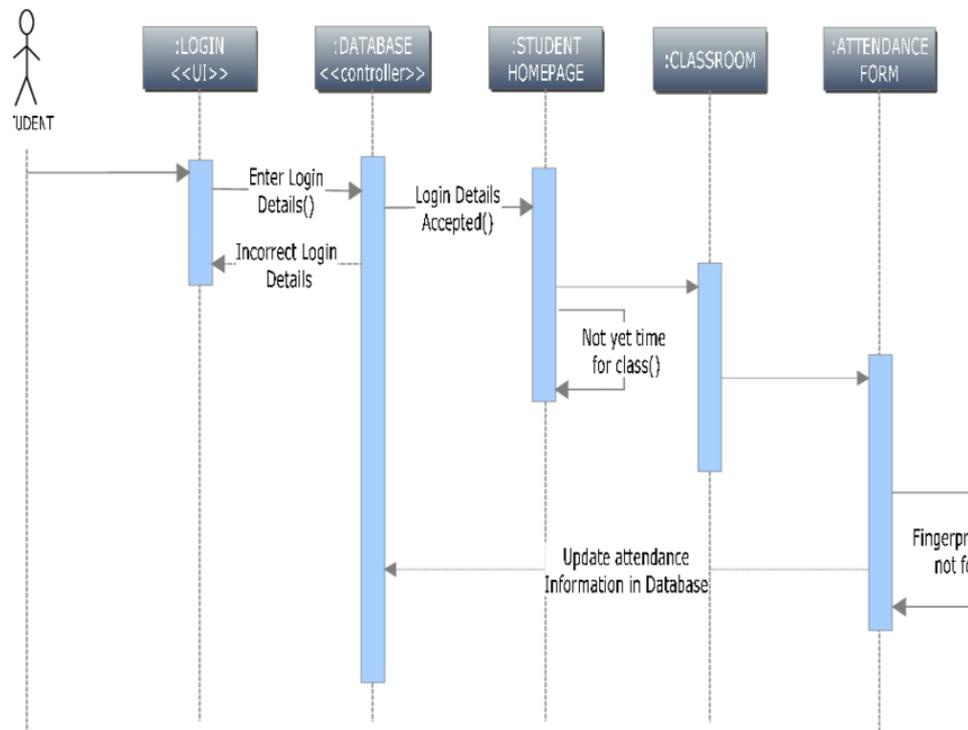


Gambar 2. 5 Contoh gambar *activity diagram*

(Sumber : <https://2.bp.blogspot.com/>)

3. *Sequence Diagram*

Sequence diagram, menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang menjadi objek tersebut. (Affandi & Syahputra, 2018).



Gambar 2. 6 Contoh *Sequence Digram*

(Sumber : Dokumentasi pribadi)

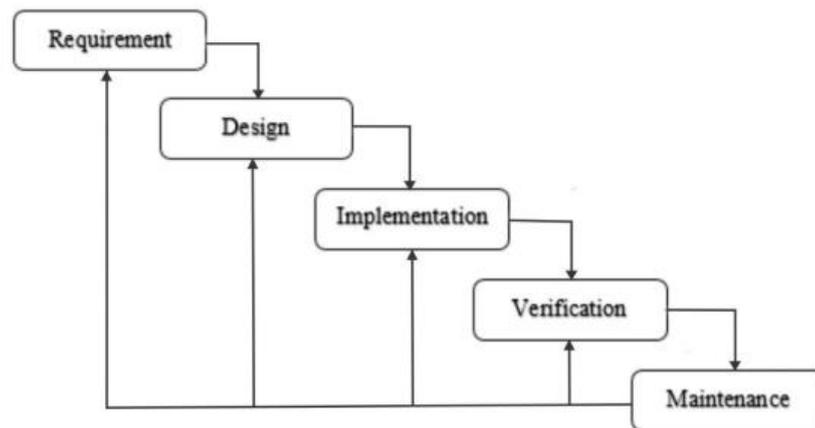
2.8 *Black Box Testing*

Black Box Testing merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional perangkat lunak yang dikembangkan. (Roohullah Jan et al., 2016). *Black box testing* cenderung dapat menemukan beberapa hal seperti fungsional yang tidak benar atau tidak ada, kesalahan struktur data, kesalahan akses basis data, kesalahan antar muka, kesalahan *performance* serta kesalahan inisialisasi dan terminasi. (Pratama et al., 2015).

2.9 *Metode Waterfall*

Metode penelitian yang digunakan pada penelitian ini yaitu suatu metode pengembangan yang bisa disebut waterfall, yang merupakan metode pengembangan dari pendekatan yang digunakan dalam penelitian deskriptif

kualitatif. Metode yang digunakan ini merupakan proses pengembangan pada perangkat lunak yang dilakukan secara berurutan, dimana kemajuan dipandang seperti air terjun yaitu air yang terus mengalir ke bawah. Waterfall melewati fase-fase seperti, perencanaan, desain, implementasi dan pengujian (Roger, 2015)



Gambar 2. 9 Metode Waterfall

(Sumber : Pressman, 2012)

1. *Requirement*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2. *Design*

Pada tahap ini, pengembang membuat desain sistem yang dapat membantu menentukan perangkat keras (hardware) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation*

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing.

4. *Verification*

Pada tahap ini, sistem dilakukan verifikasi dan pengujian apakah sistem sepenuhnya atau sebagian memenuhi persyaratan sistem, pengujian dapat dikategorikan ke dalam unit testing (dilakukan pada modul tertentu kode), sistem pengujian (untuk melihat bagaimana sistem bereaksi ketika semua modul yang terintegrasi) dan penerimaan pengujian (dilakukan dengan atau nama pelanggan untuk melihat apakah semua kebutuhan pelanggan puas).

5. *Maintenance*

Ini adalah tahap akhir dari metode waterfall. Perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.