

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Sebagai Penunjang untuk penyusunan laporan ini, maka diperlukan ide pokok dan kerangka pemikiran yang relevan dengan topik penelitian. Hal ini bertujuan untuk membangun pemahaman awal tentang kerangka teoritis yang melandasi penelitian ini. Selain itu, tinjauan pustaka ini juga akan mengidentifikasi celah – celah penelitian yang masih ada dalam penelitian sebelumnya. Dengan demikian, Penelitian ini diharapkan mampu memberikan kontribusi yang signifikan bagi kemajuan pengetahuan dibidang ini dan memberikan solusi yang bermanfaat bagi permasalahan yang ada. Berikut beberapa penelitian yang mengacu pada karya-karya sebelumnya sebagai referensi :

Penelitian lain yang dilakukan oleh Ridwan Romadhon dan rekan-rekannya dengan judul "Sistem Informasi *Booking* Wisata Pendakian Melalui *Online* di Kawasan Gunung Prau Dataran Tinggi Dieng Jawa Tengah" menyoroti dampak positif dari sistem terkomputerisasi otomatis dalam penyusunan laporan dan efisiensi pengolahan data, tanpa ketergantungan pada pengambilan data secara manual. Untuk memberikan solusi alternatif, penulis melakukan penelitian dengan membangun sistem informasi pemesanan *online*[8].

Penelitian yang dilakukan oleh Mahmud Cahyono dengan judul "*Implementation of Online Registration Information System for Mount Lawu Climbing on Mobile Web*" membahas kebutuhan akan sistem informasi pendaftaran online untuk pendakian Gunung Lawu. Sistem ini bertujuan menyediakan informasi mengenai data pendaki dan ketersediaan kuota pendakian. Diharapkan sistem ini dapat membantu pencatatan data pendaki yang akan mendaki Gunung Lawu di setiap jalur pendakian dan memberi informasi mengenai jumlah pendaki yang melakukan pendakian[9].

Penelitian yang dilakukan oleh Amir Hamzah. dengan judul "Sistem Informasi Pengelolaan Dan Pemesanan Wisata Pendakian Gunung Di Indonesia Berbasis *Web*" mengkaji *web* dimana petugas yang bertanggung jawab mengelola rute pendakian gunung dapat memperoleh informasi mengenai gunung dan melakukan pemesanan jalur pendakian. Fitur obrolan langsung memudahkan pendaki untuk bertanya atau berdiskusi mengenai pendakian. Situs *web* ini menampilkan beragam informasi seperti gambar, tulisan, dan laporan, serta menyediakan fasilitas pemesanan jalur pendakian gunung secara daring. Tahap pengujian hasil implementasi meliputi pengujian komponen, pengujian integrasi, dan pengujian sistem[10].

Pada tahun 2023, Maysce Christi, dkk melakukan penelitian dengan judul "Rancang Bangun Sistem Informasi dan Pelayanan *E-Ticket (Booking Online)* pada Wisata Pendakian Gunung Budheg Tulungagung menggunakan *Website* dengan *Framework Laravel*". Penelitian ini membahas pelayanan yang masih diaplikasikan secara konvensional, mulai dari membeli tiket, atau membayar tiket masuk pada wisata pendakian Gunung Budheg, sehingga pendataan pengunjung yang memakan waktu lama, dan juga pembayaran yang akan terbuang sia – sia saat pengunjung memiliki nominal uang yang sangat besar. Maka dari itu peneliti membangun Sistem Informasi dan Layanan *E-Ticketing (Online Booking)* dengan menggunakan metode *SDLC* atau *System Development Life Cycle* dengan *waterfall* model. Hasil dari penelitian ini adalah Sistem Informasi dan Layanan *E-Ticketing (Online Booking)* dengan memanfaatkan *framework laravel* Sehingga mempermudah dan mempercepat proses pendataan maupun pendaftaran yang akan dilakukan para pendaki. Dan mengurangi adanya antrean pada saat pendakian dilakukan[11].

Studi terkini yang dilakukan oleh Prasetyo Wibowo Yunanto. Dengan judul "Sistem Informasi Penjejak Pendakian Gunung Berbasis *Web*," penelitian ini mengeksplorasi perancangan sistem *monitoring*

pendaki gunung dengan jaringan nirkabel memakai protokol MQTT. Sistem yang dirancang menciptakan sebuah *platform monitoring* pendakian gunung yang memanfaatkan titik akses nirkabel terhubung ke perangkat pendaki, memakai metode *publish-subscribe* melalui protokol MQTT. Sistem ini beroperasi dengan membangun titik akses nirkabel bersamaan stasiun *Wi-Fi* pada node arsitektur (komputer kantor) yang berperan sebagai perantara dalam jaringan komunikasi antara klien dan broker, selanjutnya pertukaran informasi dijalankan dengan node bergerak (perangkat bergerak milik pendaki)[12].

Penelitian lain yang dilakukan oleh Martha Yogi dan timnya, berjudul "Implementasi Sistem Informasi Geografis Jalur Pendakian Gunung Penanggungan dengan Metode *Dijkstra* dan Penerapan *Fuzzy* dalam Rekomendasi Jalur" berfokus pada penggunaan metode *Fuzzy* untuk menyarankan awal titik di *website* sistem informasi geografis Gunung Penanggungan. Ini mendukung pendaki dalam memilih jalur yang cocok dengan preferensi mereka, apakah mereka menginginkan medan yang curam, normal, atau landai, serta apakah mereka ingin melintasi jalur yang ada candi atau tidak[13].

Tabel 1 Ringkasan Penelitian Sebelumnya

No	Judul	Penulis	Tahun	Studi Kasus	Metode/Algoritma	Hasil	Relevansi
1.	Sistem Informasi <i>Booking</i> Wisata Pendakian Melalui <i>Online</i> Di Kawasan Gunung Prau Dataran Tinggi Dieng Jawa Tengah	Ridwan Romadhon, Gregorius H.A.K	2021	Gunung Prau Dataran Tinggi Dieng Jawa Tengah	RAD (<i>Rapid Application Development</i>)	Setelah melakukan penelitian dan menyebarkan kuesioner, diciptakan <i>website</i> , yaitu Sistem Informasi <i>Booking Online</i> Gunung Prau. <i>Website</i> ini adalah bentuk peningkatan dalam manajemen dengan teknologi informasi. Program <i>website</i> ini dirancang sejalan sesuai kebutuhan pendaki dan wisatawan Gunung Prau.	Relevan dengan penelitian ini dalam perencanaan dan pengembangan sistem informasi <i>berbasis web</i> , metode yang diterapkan menggunakan RAD dan menekankan pentingnya manajemen data yang efektif dan efisien dalam mendukung operasional sistem operasi.
2.	Implementasi Sistem Informasi Pendaftaran <i>Online</i> Pendakian Gunung Lawu Berbasis <i>Web Mobile</i>	Roni, Setiawan dan Mustaqiem	2022	Gunung Lawu	<i>Waterfall</i>	Sistem ini memudahkan pendaki untuk mendaftar pendakian Gunung Lawu. Pendaki juga bisa mendapatkan informasi tentang jalur pendakian dan banyaknya pendaki yang tengah mendaki di tanggal tertentu. Namun, sistem ini memiliki keterbatasan karena tidak	Relevan dengan penelitian ini dalam pengembangan <i>front-end</i> nya menggunakan <i>HTML</i> dan berupaya meningkatkan efektivitas dan efisiensi dalam proses administrasi dan pelayanan.

No	Judul	Penulis	Tahun	Studi Kasus	Metode/Algoritma	Hasil	Relevansi
						mendukung pendaftaran pendakian dalam berkelompok, sehingga setiap pendaki harus mendaftar secara individual. Sistem pendaftaran <i>online</i> pendakian Gunung Lawu ini berbasis <i>web</i> .	
3.	Sistem Informasi Pengelolaan Dan Pemesanan Wisata Pendakian Gunung Di Indonesia Berbasis <i>Web</i>	Amir Hamzah, Dina Andayati, Guntara	2022	Gunung Sindoro	<i>Waterfall</i>	Penelitian ini menghasilkan sistem informasi untuk manajemen dan pemesanan pendakian gunung, yang memberikan <i>platform web</i> bagi petugas gunung untuk menata rute pendakian dan bagi pendaki untuk mencari informasi serta memesan jalur pendakian. Terdapat fitur obrolan langsung (<i>live chatting</i>) mempermudah pendaki supaya bertanya dan berdiskusi tentang pendakian.	Penelitian ini menguji sistem dengan metode <i>Black Box Testing</i> , yang dikerjakan dengan menguji langsung dari perspektif <i>admin</i> , petugas, dan pendaki.

No	Judul	Penulis	Tahun	Studi Kasus	Metode/Algoritma	Hasil	Relevansi
4.	Rancang Bangun Sistem Informasi dan Pelayanan <i>ETicket</i> (Booking <i>Online</i>) pada Wisata Pendakian Gunung Budheg Tulungagung menggunakan <i>Website</i> dengan <i>Framework Laravel</i>	Christi Maysce, Hayuhardhika Widhy, Putra Nugraha, Hanggara Buce Trias	2023	Gunung Budheg Tulungagung	<i>System Development Life Cycle</i> (SDLC)	Sistem Informasi dan Layanan <i>E-Ticketing</i> (<i>Online Booking</i>) dengan memanfaatkan <i>framework laravel</i> Sehingga mempermudah dan mempercepat proses pendataan maupun pendaftaran yang akan dilakukan para pendaki.	Terkait dengan penelitian ini, dalam pengembangan sistem informasi dan layanan <i>e-ticketing</i> untuk objek wisata pendakian, digunakan kerangka kerja <i>Laravel</i> yang berbasis <i>web</i> .
5.	Sistem Informasi Penjejak Pendakian Gunung Berbasis <i>Web</i>	Prasetyo Wibowo Yunanto, Murien Nugraheni, Nurman Nugraha	2019	Gunung Gede-Pangrango	<i>Prototype</i>	Hasilnya berupa Sistem Informasi Pemantauan Pendakian Gunung berbasis <i>web</i> yang akan diterapkan oleh tiga pengguna utama: super <i>admin</i> , admin pengelola gunung, dan pendaki gunung. Pengembangan sistem informasi pemantauan pendakian gunung menerapkan model <i>prototyping</i> dapat menciptakan sistem yang tepat	Dalam konteks penelitian ini, dalam kerangka sistem berbasis <i>web</i> , peramban <i>web</i> (<i>web browser</i>) digunakan untuk mendapatkan informasi dalam format <i>hypertext</i> . Peramban <i>web</i> akan menyampaikan permintaan (<i>request</i>) ke <i>server web</i> dan menampilkan hasilnya kepada pengguna. Selain itu, peramban <i>web</i> juga dapat mengirimkan formulir ke <i>server web</i> untuk diolah. Contoh dari peramban <i>web</i> termasuk <i>Opera Mini</i> ,

No	Judul	Penulis	Tahun	Studi Kasus	Metode/Algoritma	Hasil	Relevansi
						secara spesifik dengan kebutuhan pengguna.	<i>Mozilla Firefox</i> , <i>Internet Explorer</i> , dan lain-lain.
6.	Implementasi Sistem Informasi Geografis Jalur Pendakian Gunung Penanggungan Dengan Metode <i>Dijkstra</i> Dan Penerapan <i>Fuzzy</i> Dalam Rekomendasi Jalur	Martha Yogi Yuda Rifendy, Paramitha Nerisafitra	2023	Gunung Penanggungan	<i>Dijkstra</i>	Setelah perancangan sistem selesai dan semua data terkumpul, langkah selanjutnya adalah mengkalkulasi saran jalur pendakian Gunung Penanggungan yang paling pendek dan aman memakai metode <i>Fuzzy</i> dan <i>Dijkstra</i> . Tahap awal adalah menghitung jalur tercepat dari pendakian melalui Via Jolotundo, Via Kedungudi, dan Via Tamiajeng dengan memakai Algoritma <i>Dijkstra</i> .	Terkait dengan penelitian ini, dalam rancangan halaman pemesanan pendakian, halaman tersebut menawarkan bagian pemesanan pendakian. Tujuannya adalah untuk mempermudah pengguna dalam mencoba pemesanan sebelumnya, untuk menghindari kehabisan kuota atau antrean yang panjang saat berada di lokasi. Di sini, pengguna diminta untuk mengisi data pribadi dan jumlah rombongan yang akan ikut pendakian, yang kemudian akan disimpan dalam <i>database</i> .

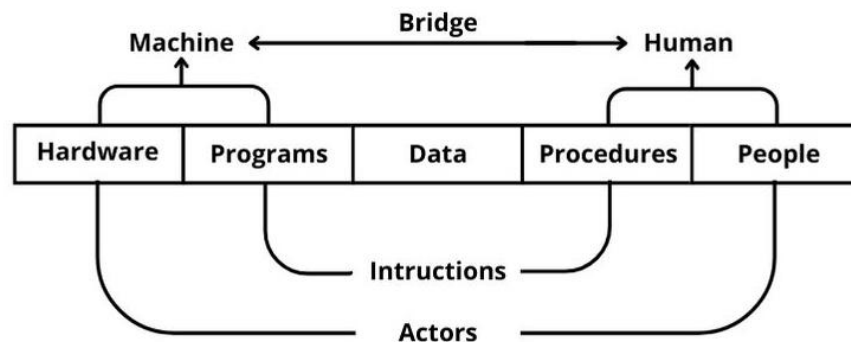
Penelitian terdahulu umumnya menggunakan metode *Waterfall*, *SLDC* dan *Prototype* dalam pengembangan perangkat lunak yang memiliki keterbatasan dalam fleksibilitas dan waktu penyelesaian, sedangkan penelitian sekarang ini menggunakan metode *Rapid Application Development (RAD)* yang lebih menekankan pada kecepatan pengembangan dan fleksibilitas adaptasi terhadap perubahan kebutuhan, sehingga memungkinkan iterasi lebih cepat dan kolaborasi yang lebih efektif antara pengembang dan pengguna akhir.

2.2 Dasar Teori

2.2.1 Sistem Informasi

Sistem informasi di organisasi dengan maksud untuk memenuhi kebutuhan pemrosesan transaksi harian, membantu operasi, manajerial, dan kegiatan strategis, serta menyajikan laporan yang dibutuhkan untuk pihak eksternal. Sistem ini terdiri dari kombinasi prosedur kerja, informasi, orang, dan teknologi informasi yang disusun secara terstruktur untuk memperoleh tujuan organisasi. Dengan demikian, sistem informasi adalah serangkaian sistem yang ada di organisasi yang mengumpulkan, mengerjakan, menyimpan, dan menyebarkan informasi untuk membantu pengambilan keputusan dan pengendalian di organisasi tersebut[14].

Ada lima komponen dalam sistem informasi. Gambar 2.1 yang akan disajikan selanjutnya akan mengilustrasikan setiap elemen tersebut.



Gambar 2.1 Lima Elemen Sistem Informasi

Lima elemen itu saling berhubungan satu sama lain, dan dapat dikelompokkan seperti yang tertera di bawah ini[15]:

1. Perangkat keras dan program, yang bertindak sebagai mesin.
2. Individu dan prosedur, mencakup orang dan bagaimana cara mengoperasikan mesin.
3. Data, berfungsi sebagai perantara manusia dan mesin untuk prosedur pengolahan data.

2.2.2 *E-Ticketing*

E-ticketing atau tiket elektronik merupakan inovasi teknologi dalam *e-commerce*, di mana penjualan tiket perjalanan dapat dilakukan tanpa menggunakan tiket kertas. Semua informasi terkait *e-ticketing* disimpan dengan cara digital pada sistem komputer. *E-ticketing* dapat meminimalisir biaya proses pembuatan tiket, meniadakan kebutuhan formulir kertas, dan menaikkan fleksibilitas bagi pembeli tiket[16].

E-ticketing memberikan beberapa keuntungan, antara lain:

- a. Meminimalisir biaya yang berhubungan dengan mencetak dan mengirimkan tiket kepada pembeli.
- b. Mengurangi kebutuhan tenaga kerja yang terlibat dalam pencetakan dan pengiriman tiket.
- c. Tiket elektronik aman dari risiko duplikasi.
- d. Pengiriman tiket biasanya dilakukan secara instan setelah proses pembelian selesai.
- e. Prosedur yang sederhana membantu pengguna dalam melakukan transaksi.
- f.

2.2.3 Definisi *Web*

Web adalah sebuah *domain* digital di internet yang berisi ribuan informasi tentang berbagai topik, disusun dalam format dokumen *hypertext*. *Hypertext* adalah prinsip dasar dari *World Wide Web*, di mana dokumen-

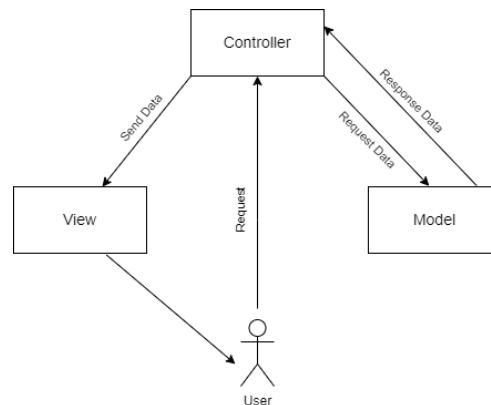
dokumen ini memungkinkan pengguna untuk menelusuri dari satu halaman *web* ke halaman lainnya menggunakan tautan[17].

2.2.4 Framework Laravel

Framework merupakan sekumpulan kode yang terdiri dari pustaka (*library*) dan alat (*tool*) yang disatukan secara terstruktur untuk membentuk suatu kerangka kerja. Tujuannya adalah untuk membantu dan mempercepat proses pengembangan aplikasi *web*. Dengan kata lain, *framework* adalah gabungan dari berbagai potongan program yang digabungkan menjadi suatu kesatuan kerja, membantu dalam pembuatan aplikasi[18].

Laravel adalah sebuah kerangka kerja aplikasi *web* yang dirancang dengan memakai bahasa pemrograman PHP, memiliki sifat terbuka (*open source*), dan mengadopsi pola pengembangan *Model-View-Controller (MVC)*. Laravel didistribusikan di bawah lisensi *MIT License* dan memakai *GitHub* sebagai platform untuk berbagi kode. Seperti kebanyakan kerangka kerja PHP lainnya, Laravel juga mengikuti pola MVC. Selain itu, Laravel disertai dengan perangkat baris perintah yang disebut "*Artisan*", yang dapat digunakan untuk menyusun dan menginstal paket[19].

Laravel menggunakan konsep MVC (*Model-View-Controller*), MVC adalah konsep arsitektur dalam pembangunan aplikasi berbasis *web* yang membagi aplikasi *web* menjadi 3 bagian besar. Yang mana setiap bagian memiliki tugas-tugas serta tanggung jawab masing-masing. Tiga bagian tersebut adalah: *model*, *view* dan *controller*. Pada gambar 2.2 merupakan konsep dari MVC.



Gambar 2.2 Konsep MVC (*Model-View-Controller*)[20]

2.2.5 Rapid Application Development (RAD)

Rapid Application Development (RAD) adalah metode peningkatan perangkat lunak yang bersifat inkremental, khususnya ditujukan pada durasi pengerjaan yang singkat[21]. Gambar 2.3 menggambarkan siklus RAD yang terdiri dari tiga fase, yaitu perencanaan, perancangan, dan penerapan sistem yang melibatkan kolaborasi antara analis dan pengguna.



Gambar 2. 3 Siklus RAD[22]

Pada tahap perencanaan, pengguna dan analis berkolaborasi untuk mengenali tujuan sistem dan syarat informasi yang ada dari tujuan tersebut. Tahap ini, fokusnya memecahkan masalah pengguna. Di tahap perancangan, analis dan pengguna bekerja sama untuk merancang dan membangun sistem. Durasi tahap ini dapat bervariasi tergantung

kompleksitas aplikasi yang direncanakan akan dikembangkan. Selama tahap ini, pengguna memberikan tanggapan terhadap prototipe yang ada dan analis memperbaiki desain berdasarkan masukan dan respons pengguna. Pendekatan ini akan mendorong pengembangan hingga tahap akhir. Pada tahap implementasi, setelah analis yang ikut serta pada tahap perencanaan dan perancangan memperoleh persetujuan dari pengguna, sistem baru akan diuji dan diperkenalkan kepada organisasi[23].

Agar lebih mudah dipahami, berikut adalah penjelasan dari alur tahapan pengembangan perangkat lunak dengan Metode *Rapid Application Development (RAD)*:

1. Perencanaan Syarat-syarat

Tahapan ini merupakan tahap awal dalam suatu pengembangan sistem, dimana pada tahap ini dilakukan identifikasi masalah dan pengumpulan data yang diperoleh dari pengguna atau stakeholder pengguna yang bertujuan untuk mengidentifikasi maksud akhir atau tujuan dari sistem dan kebutuhan informasi yang diinginkan berupa studi literatur dan wawancara. Pada tahap ini keterlibatan kedua belah sangatlah penting dalam mengidentifikasi kebutuhan untuk pengembangan suatu sistem.

2. Merancang Sistem

Di dalam tahap desain sistem, keaktifan pengguna yang terlibat sangatlah penting untuk mencapai tujuan karena pada tahapan ini dilakukan proses desain dan proses perbaikan desain secara berulang-ulang apabila masih terdapat ketidaksesuaian desain terhadap kebutuhan pengguna yang telah diidentifikasi pada tahapan sebelumnya. Luaran dari tahapan ini adalah spesifikasi *software* yang yang dibutuhkan di dalam sistem secara umum, *Database*, *Unified Modelling Language* dan Desain sistem.

3. Pembangunan Sistem

Pada tahap ini desain sistem yang telah dibuat dan disepakati, diubah ke dalam bentuk program versi awal sampai dengan versi final. Pada tahapan ini juga peneliti harus terus-menerus melakukan kegiatan pengembangan dan integrasi dengan bagian-bagian lainnya sambil terus mempertimbangkan *feedback* dari pengguna atau klien. Jika proses berjalan lancar maka dapat berlanjut ke tahapan berikutnya, sedangkan jika aplikasi yang dikembangkan belum menjawab kebutuhan, peneliti akan kembali ke tahapan rancang sistem.

4. Implementasi atau penyelesaian

Tahapan ini merupakan tahapan dimana peneliti menerapkan desain dari suatu sistem yang telah disetujui pada tahapan sebelumnya. Sebelum sistem diterapkan, terlebih dahulu dilakukan proses pengujian terhadap program menggunakan *Blackbox Testing* untuk mendeteksi kesalahan yang ada pada sistem yang dikembangkan. Pada tahap ini biasa memberikan tanggapan akan sistem yang sudah dibuat dan mendapat persetujuan mengenai sistem tersebut.

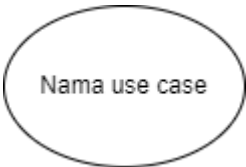
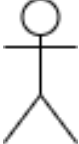


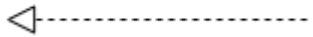
2.2.6 *Unified Modeling Language (UML)*

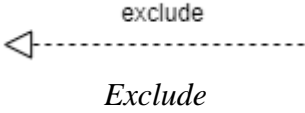
UML merupakan kependekan dari *Unified Modelling Language*, adalah sebuah pendekatan pemodelan yang menggunakan representasi visual untuk merancang sistem berbasis objek. UML sering dianggap sebagai standar bahasa yang digunakan untuk gambaran, perancangan, dan dokumentasi sistem perangkat lunak[24]. Beberapa jenis pemodelan yang termasuk dalam *UML* adalah *Use Case Diagram*, *Class Diagram*, *Activity Diagram*, dan *Sequence Diagram*[25].

1. Use Case Diagram

Diagram *use case* menguraikan kegunaan aplikasi dari perspektif pengguna luar sistem (aktor). Diagram ini menggambarkan fungsi sistem atau kelas dan interaksi sistem dengan lingkungan eksternal. Diagram *use case* dapat diterapkan dalam analisis untuk menangkap kebutuhan atau permintaan terhadap sistem serta memahami operasional yang diperlukan oleh sistem[26]. Pada Tabel 2 adalah simbol – simbol *Use Case Diagram*:

Tabel 2 Simbol *Use Case Diagram*

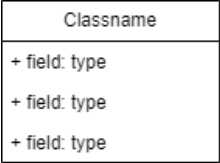
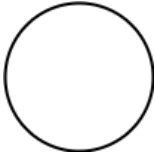


Simbol	Despkripsi
 <i>Use Case</i>	Abstraksi dan interaksi antara sistem dan aktor
 <i>Actor</i>	Mewakili peran orang, proses atau sistem yang berinteraksi dengan <i>use case</i>
 <i>Association</i>	Abstraksi penghubung antara <i>actor</i> dan <i>use case</i>
 Generalisasi	Menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dengan <i>use case</i> .
 <i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.

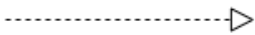
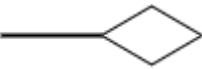
Simbol	Deskripsi
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.

2. Class Diagram

Diagram kelas memberikan ilustrasi tentang keterkaitan antara tabel yang terletak dalam basis data. Setiap kelas memiliki atribut dan metode, sesuai dengan proses yang berlangsung[26]. Pada Tabel 3 adalah simbol – simbol *Class Diagram*.

Tabel 3 Simbol Class Diagram

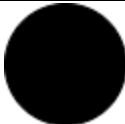

Simbol	Deskripsi
	Kelas pada struktur sistem.
 <p>Antarmuka/<i>Interface</i></p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
 <p>Asosiasi/<i>Association</i></p>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
 <p>Asosiasi Berarah/<i>Directed Association</i></p>	Hubungan antar kelas, dimana satu kelas digunakan oleh kelas lain, biasanya disertai dengan <i>multiplicity</i> .

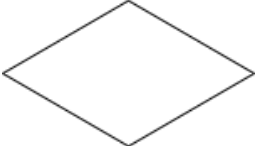



Simbol	Despkripsi
 Kebergantungan	Relasi antar kelas dengan makna kebergantungan antar kelas.
 Agregasi/Agregation	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

3. Activity Diagram

Activity diagram, yang juga dikenal sebagai diagram aktivitas, mengilustrasikan aktivitas-aktivitas yang dilakukan oleh sistem, tidak memperhitungkan aktivitas yang dilakukan oleh aktor. Dalam diagram ini, aktivitas sistem digambarkan sebagai proses yang sedang berlangsung. Sebuah aktivitas bisa dihubungkan dengan satu atau lebih *use case*. *Use case* memvisualisasikan cara aktor memakai sistem untuk melaksanakan suatu aktivitas. Dengan demikian, diagram aktivitas fokus pada aktivitas internal sistem, sementara *use case* memvisualisasikan interaksi antara aktor dan sistem[26]. Pada tabel 4 adalah simbol – simbol Activity Diagram.

Tabel 4 Simbol *Activity Diagram*

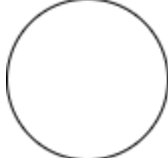
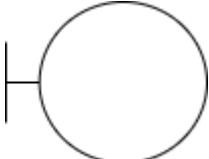

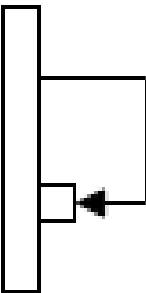
Simbol	Deskripsi
 Status Awal	Menandakan suatu titik awal aktivitas.
 Aktivitas	Menyatakan suatu aktivitas yang terjadi.



Simbol	Despkripsi
 <p data-bbox="571 528 863 562">Percabangan/<i>Decision</i></p>	<p data-bbox="962 365 1353 477">Percabangan dimana jika ada pilihan aktivitas lebih dari satu.</p>
 <p data-bbox="587 645 847 678">Penggabungan/<i>Join</i></p>	<p data-bbox="962 589 1353 701">Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.</p>
 <p data-bbox="635 864 802 898">Status Akhir</p>	<p data-bbox="962 730 1353 797">Menandakan suatu titik akhir aktivitas.</p>
 <p data-bbox="655 1144 786 1178"><i>Swimlane</i></p>	<p data-bbox="962 925 1353 1081">Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>

4. *Sequence Diagram*

Sequence diagram adalah representasi visual yang memodelkan keterkaitan antara objek di dalam dan luar sistem melalui pesan-pesan yang diurutkan berdasarkan waktu. Diagram ini mengilustrasikan secara detail bagaimana operasi-operasi dijalankan, pesan yang terkirim, dan kapan pesan tersebut terjadi. Dengan dimensi vertikal yang menggambarkan waktu dan dimensi horizontal yang menunjukkan objek-objek yang terlibat, *sequence diagram* memberikan gambaran yang jelas tentang urutan interaksi dalam sistem[26]. Pada Tabel 5 adalah simbol – simbol *sequence diagram*.

Tabel 5 Simbol *Sequence Diagram*

Simbol	Deskripsi
 <p data-bbox="639 658 807 696">Entity Class</p>	<p data-bbox="962 427 1345 539">Simbolisasi sistem sebagai landasan dalam menyusun basis data.</p>
 <p data-bbox="611 972 820 1010">Boundary Class</p>	<p data-bbox="962 741 1345 819">Menangani komunikasi antar lingkungan sistem.</p>
 <p data-bbox="624 1330 820 1368">Control Class</p>	<p data-bbox="962 1061 1345 1173">Bertanggung jawab untuk kelas dengan objek yang berisi logika.</p>
 <p data-bbox="647 1778 783 1816">Recursive</p>	<p data-bbox="962 1420 1174 1458">Pesan pembalik.</p>

Simbol	Deskripsi
 <p data-bbox="651 680 788 712">Activation</p>	<p data-bbox="963 367 1351 439">Proses durasi aktivasi sebuah operasi</p>
 <p data-bbox="655 1021 778 1052">Life Line</p>	<p data-bbox="963 763 1351 875">Komponen yang disimbolkan garis putus-putus terhubung ke objek</p>

2.2.7 Black Box Testing

Black Box Testing, juga dikenal sebagai pengujian fungsional, menitikberatkan pada perilaku luar perangkat lunak atau komponennya, dengan memperlakukannya sebagai "kotak hitam" tanpa memperhatikan detail internal. Oleh karena itu, penguji tidak melihat implementasi atau kode di dalamnya. Tujuan dari *Black Box Testing* yaitu memastikan bahwa fungsi-fungsi eksternal yang disediakan oleh perangkat lunak dikelola dengan benar. Pendekatan *Black Box Testing* menekankan pada pengujian berdasarkan pada spesifikasi kebutuhan fungsional yang sudah ditetapkan. *Black Box Testing* bukan teknik pengujian alternatif untuk *whitebox testing*, melainkan merupakan pendekatan yang melengkapi dengan memperhatikan jenis kesalahan yang berbeda dari metode *whitebox testing*[27].

Terdapat beberapa langkah yang menjadi bagian dari cara kerja *black box testing* ini. Adapun beberapa langkah tersebut, meliputi :

1. Memahami kebutuhan dan Spesifikasi

Sebelum melakukan pengujian, penguji harus memahami dengan baik kebutuhan dan spesifikasi dari *software* atau aplikasi yang akan ia uji. Selain itu, penguji juga perlu memahami sejumlah aspek pendukung yang diinginkan dari *software* atau aplikasi, seperti kinerja hingga fungsi.

2. Penentuan *Input*

Langkah lainnya yang juga termasuk dalam cara kerja pengujian ini adalah penentuan *input*. Di dalam hal ini, penguji harus menentukan input apa saja yang ingin diberikan ke *software*. Hal ini penting, untuk memastikan semua jenis *input* telah dicoba ke sistem *software*. Jadi, penilaian fungsionalitas *software* menjadi akurat.

3. Penentuan *Output*

Selain penentuan *input*, penentuan *output* juga termasuk *task* yang harus penguji lakukan sebelum pengujian. Demi memastikan apakah *output* yang sistem berikan sesuai dengan *input* yang ia terima nantinya, *output* ini akan menjadi parameter untuk menentukan apakah *software* berfungsi dengan baik atau tidak.

4. Tahap Penyeleksian *Input*

Di dalam tahap penyeleksian ini, penguji harus menerapkan skenario buatan (*test case*) untuk menguji *input* yang telah penguji tentukan sebelumnya. Misalnya, penguji melakukan *input* yang salah untuk mengetahui *output* apa yang akan sistem berikan atas *input* salah tersebut.

5. Tahapan Pengujian

Di dalam tahapan pengujian, sejumlah *input* yang telah penguji tentukan sebelumnya akan diuji untuk melihat *output* yang sistem berikan. Tahapan ini sendiri adalah tahapan inti untuk memastikan apakah *software* atau aplikasi terkait berjalan dengan baik atau tidak.

6. *Review* dan Evaluasi

Tahap terakhir dari metode *blackbox testing* adalah *review* dan juga evaluasi. Secara umum, dalam tahapan ini, penguji umumnya membuat laporan dan catatan hasil pengujian. Melalui data laporan pengujian ini, penguji dapat melihat apakah terdapat ketidaksesuaian di dalam sistem *software* atau aplikasi.