

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa penelitian tentang perancangan sistem informasi dan layanan *e-ticketing* telah dilakukan dan diterapkan dalam bidang pendakian. Hasil penelitian sebelumnya menunjukkan bahwa perancangan sistem informasi dan pelayanan *e-ticketing* dalam pendakian gunung sangat diperlukan sebagai dasar untuk penelitian selanjutnya dan memudahkan penelitian berikutnya.

Berdasarkan penelitian yang dilakukan oleh Frans Eduard Schaduw, Syaiful Anwar, Aburizal Fahmi (2018). Penelitian yang berjudul **“Penerapan Metode Waterfall Dalam Sistem Informasi Registrasi Pendakian Gunung Sindoro Berbasis Web”** menjelaskan bahwa penelitian ini dilaksanakan untuk mengatasi masalah yang terjadi dalam aktivitas pendakian gunung, khususnya terkait kecelakaan dan risiko hilangnya nyawa akibat jalur-jalur pendakian yang terlalu ekstrim dan berbahaya. Penyebab utama masalah ini adalah kurangnya pengetahuan tentang wisata pendakian gunung dan minimnya pemahaman mengenai kegiatan semacam ini. Penelitian ini menerapkan metode waterfall dalam pengembangan sistem informasi registrasi pendakian Gunung Sindoro dengan penekanan pada unsur pendidikan. Outputnya adalah aplikasi registrasi pendakian yang dilengkapi dengan informasi serta materi pendidikan tentang sejarah gunung, rute pendakian, peralatan yang diperlukan, estimasi biaya, dan prosedur pendakian yang tepat [15].

Berdasarkan penelitian yang dilakukan oleh Safry Pasau Tangdisosang, Indrastanti R Widiyanti (2022). Penelitian yang berjudul **“Sistem Informasi Pendakian Gunung Jawa Timur Berbasis Web Menggunakan Framework Codeigneter”** menjelaskan bahwa penelitian ini dilaksanakan untuk mengatasi masalah yang terjadi dalam aktivitas pendakian gunung di

Jawa Timur, khususnya terkait kecelakaan dan risiko hilangnya nyawa akibat kurangnya informasi yang memadai tentang pendakian gunung. Penyebab utama masalah ini adalah kurangnya pengetahuan tentang jalur pendakian, peralatan yang dibutuhkan, dan kondisi cuaca yang sering kali diabaikan oleh pendaki. Penelitian ini menerapkan metode waterfall dalam pengembangan sistem informasi pendakian gunung berbasis web menggunakan framework CodeIgniter. Hasilnya adalah aplikasi pendakian yang menyediakan informasi lengkap tentang estimasi waktu pendakian dari basecamp ke puncak, kondisi cuaca, rute pendakian, perlengkapan yang diperlukan, sejarah singkat gunung, tips pendakian yang aman, dan informasi transportasi ke basecamp. Tujuan utamanya adalah untuk mengurangi kecelakaan dalam pendakian gunung dan memudahkan pendaki dalam mengakses informasi yang relevan. Pengujian dilakukan menggunakan metode blackbox dan menunjukkan bahwa aplikasi berjalan sesuai dengan perancangan yang telah ditetapkan [16].

Berdasarkan penelitian yang dilakukan oleh Amir Hamzah, Dina Andayati, Guntara (2022). Penelitian yang berjudul **“Sistem Informasi Pengelolaan dan Pemesanan Wisata Pendakian Gunung di Indonesia berbasis Web”** menjelaskan bahwa penelitian ini dilakukan sebagai upaya untuk mengatasi beberapa permasalahan yang terjadi dalam wisata pendakian gunung. Permasalahan tersebut meliputi kurangnya pemahaman informasi terkait pendakian, pertumbuhan minat pendakian yang tinggi, dan kebutuhan akan sistem informasi pendakian yang lengkap dan mudah diakses. Perancangan sistem menggunakan UML (Unified Modeling Language) dan MySQL untuk manajemen data. SDLC (System Development Life Cycle) mencakup tahap analisis kebutuhan, perancangan, desain, implementasi, dan pengujian. Hasil penelitian menunjukkan bahwa sistem yang dibangun beroperasi dengan lancar dan memenuhi kebutuhan para pendaki. Diharapkan aplikasi sistem informasi ini dapat mengurangi risiko kecelakaan serta meningkatkan keselamatan para pendaki gunung dengan menyediakan informasi yang diperlukan secara menyeluruh dan mudah diakses [17].

Berdasarkan penelitian yang dilakukan oleh Arief Ichwani, Nizirwan Anwar, Kundang Karsono, Mohamad Alrifqi (2021). Penelitian yang berjudul **“Sistem Informasi Penjualan Berbasis Website dengan Pendekatan Metode *Prototype*”** menjelaskan bahwa penelitian ini dilakukan untuk mengatasi beberapa permasalahan yang terjadi dalam operasional usaha Kinov Seprai. Permasalahan tersebut meliputi pencatatan data penjualan dan data produk yang masih manual, pemasaran produk yang terbatas pada wilayah sekitar dan dilakukan dari mulut ke mulut, serta pemesanan produk yang mengharuskan pelanggan datang ke lokasi sehingga jumlah penjualan terbatas dan rentan terjadi human error. Sistem informasi berbasis web merupakan teknologi yang memuat berbagai informasi, transaksi, dan pengolahan data untuk mengatasi masalah tersebut. Metode yang digunakan untuk membangun sistem informasi adalah *prototype*, dan analisis permasalahan menggunakan metode PIECES. Hasil penelitian menunjukkan bahwa sistem informasi berbasis web dapat menyediakan informasi yang informatif, berfungsi sebagai media promosi yang efektif, menjangkau target pasar yang lebih luas, serta memberikan pelayanan penjualan yang lebih mudah dan hemat waktu. Sistem ini diharapkan dapat meningkatkan efisiensi operasional dan mengurangi risiko human error dalam bisnis Kinov Seprai [18].

Tabel 2. 1 Ringkasan Penelitian Sebelumnya

No	Judul	Comparing	Constrasting	Criticize	Synthesize	Summarize
1	Penerapan Metode Waterfall Dalam Sistem Informasi Registrasi Pendakian Gunung Sindoro Berbasis Web [15]	Perbandingan antara kondisi pendakian gunung di Indonesia secara umum dan potensi risiko yang mungkin terjadi, termasuk kecelakaan dan korban meninggal dunia.	Meskipun gunung-gunung Indonesia menawarkan pemandangan yang indah, jalur-jalur pendakian yang ekstrim juga dapat menyebabkan risiko kecelakaan dan korban jiwa.	Sistem informasi untuk pendakian gunung Sindoro dibuat menggunakan metode waterfall. Namun, tidak dijelaskan secara rinci tentang alasan mengapa metode ini dipilih dan bagaimana metode tersebut berkontribusi pada kesuksesan sistem informasi.	Tujuan dari penelitian ini adalah untuk mengembangkan dan mengimplementasikan sistem informasi untuk proses pendaftaran pendakian Gunung Sindoro, yang juga mencakup unsur pedagogis.	Untuk mengembangkan dan menerapkan sistem informasi pada registrasi pendakian Gunung Sindoro, Indonesia.

No	Judul	Comparing	Constrasting	Criticize	Synthesize	Summarize
2	Sistem Informasi Pendakian Gunung Jawa Timur Berbasis Web Menggunakan Framework Codeigneter [16]	Perbandingan antara kondisi pendakian gunung di Jawa Timur secara umum dan potensi risiko yang mungkin terjadi, termasuk kecelakaan dan korban meninggal dunia.	Meskipun gunung-gunung di Jawa Timur menawarkan pemandangan yang indah, jalur-jalur pendakian yang ekstrem juga dapat menyebabkan risiko kecelakaan dan korban jiwa.	Sistem informasi untuk pendakian gunung di Jawa Timur dibuat menggunakan metode waterfall dan framework CodeIgniter. Namun, tidak dijelaskan secara rinci tentang alasan mengapa metode dan framework ini dipilih dan bagaimana keduanya berkontribusi pada kesuksesan sistem informasi.	Tujuan dari penelitian ini adalah untuk mengembangkan dan mengimplementasikan sistem informasi untuk proses pendaftaran pendakian gunung di Jawa Timur, yang juga mencakup informasi yang lengkap dan bermanfaat bagi para pendaki.	Untuk mengembangkan dan menerapkan sistem informasi pada registrasi pendakian gunung di Jawa Timur, Indonesia, menggunakan framework CodeIgniter dan metode waterfall.

No	Judul	Comparing	Constrasting	Criticize	Synthesize	Summarize
3	Sistem Informasi Pengelolaan dan Pemesanan Wisata Pendakian Gunung di Indonesia berbasis Web [17]	Sebelum aplikasi sistem informasi ini, masalahnya adalah minimnya pemahaman para pendaki tentang informasi terkait pendakian, seperti peta jalur, prosedur, dan aspek lainnya.	Tingkat minat untuk melakukan pendakian gunung semakin meningkat terutama di kalangan muda.	Sistem ini terbukti berfungsi dengan baik dan berhasil memenuhi kebutuhan para pendaki.	Untuk memenuhi kebutuhan para pendaki gunung di Indonesia, yang semakin lama semakin banyak, namun seringkali kurang memahami informasi terkait pendakian.	Untuk menciptakan aplikasi sistem informasi yang lengkap dan mudah diakses terkait proses pendakian gunung di Indonesia.

No	Judul	Comparing	Constrasting	Criticize	Synthesize	Summarize
4	Sistem Informasi Penjualan Berbasis Website dengan Pendekatan Metode Prototype [18]	Perbandingan antara metode manual yang digunakan oleh Kinov Seprai dalam pencatatan data penjualan dan produk dengan pendekatan sistem informasi berbasis web yang lebih modern dan efisien.	Meskipun metode manual memungkinkan Kinov Seprai untuk beroperasi selama empat tahun, metode ini memiliki keterbatasan seperti jangkauan pemasaran yang terbatas dan rentan terhadap human error, sementara sistem informasi berbasis web menawarkan solusi yang lebih efisien dan luas.	Sistem informasi untuk penjualan Kinov Seprai dibangun menggunakan metode prototype dan analisis permasalahan dengan metode PIECES. Namun, tidak dijelaskan secara rinci mengapa metode prototype dipilih dan bagaimana metode ini secara khusus mengatasi permasalahan yang dihadapi oleh Kinov Seprai.	Tujuan dari penelitian ini adalah untuk mengembangkan dan mengimplementasikan sistem informasi penjualan berbasis web untuk Kinov Seprai yang mencakup informasi produk, transaksi penjualan, dan pengolahan data, serta menyediakan media promosi yang efektif dan memperluas target pasar.	Untuk mengembangkan dan menerapkan sistem informasi penjualan berbasis website untuk Kinov Seprai menggunakan metode prototype dan analisis permasalahan dengan metode PIECES.

2.2 Landasan Teori

2.2.1 Aplikasi *Mobile*

Aplikasi *mobile* telah menjadi teknologi yang populer dan digunakan secara luas, terutama karena perkembangan pesat *smartphone* yang semakin canggih. Secara lebih rinci, aplikasi *mobile* adalah program yang dirancang khusus untuk dijalankan pada perangkat *mobile*, seperti *smartphone* dan *tablet* [19]. Selain itu, aplikasi *mobile* telah digunakan dalam berbagai konteks, seperti navigasi ke alamat pelanggan TV berbayar, menunjukkan sejauh mana aplikasi *mobile* dapat memberikan manfaat dalam berbagai situasi [20].

2.2.2 *E-Ticket*

E-Ticket atau tiket elektronik adalah sebuah inovasi dalam industri tiket yang telah menggantikan tiket fisik yang lebih umum digunakan beberapa tahun yang lalu. Dalam format elektronik, *E-Ticket* menjadi alternatif *modern* untuk menyediakan tiket kepada pengguna. Keunggulan utama *E-Ticket* adalah kemampuannya untuk memberikan dokumentasi yang lengkap dan terinci mengenai detail perjalanan atau proses penjualan tiket yang telah dilakukan oleh pengguna [21].

2.2.3 *Framework Flutter*



Gambar 2.1 *Framework Flutter* [22]

Flutter merupakan suatu *Software Development Kit* (SDK) atau kerangka kerja *open source* yang dibangun oleh *Google* untuk memfasilitasi pembuatan dan pengembangan aplikasi yang kompatibel dengan sistem operasi *Android*, *iOS*, *Windows*, *Linux* dan masih banyak

lagi [23]. Kelebihan utama dari *Flutter* meliputi tingkat produktivitas yang tinggi, performa yang sangat baik, proses pengembangan yang ringkas dan cepat, serta statusnya sebagai proyek *open source* [24].

2.2.4 Wisata Alam

Wisata alam Gunung Prau tidak hanya menawarkan pemandangan yang menakjubkan, tetapi juga berfungsi sebagai batas wilayah yang memisahkan tiga kabupaten, yaitu Kabupaten Batang, Kabupaten Kendal, dan Kabupaten Wonosobo. Salah satu kota di Provinsi Jawa Tengah yang berdekatan dengan Gunung Prau adalah Kota Wonosobo, yang memiliki jumlah penduduk sekitar 769.396 jiwa [25]. Gunung Prau menyediakan beberapa jalur pendakian populer, termasuk Jalur Patak Banteng, Jalur Dwarawati, Jalur Dieng Wetan, Jalur Kalilembu, Jalur Wates, dan Jalur Igrimrana [26].

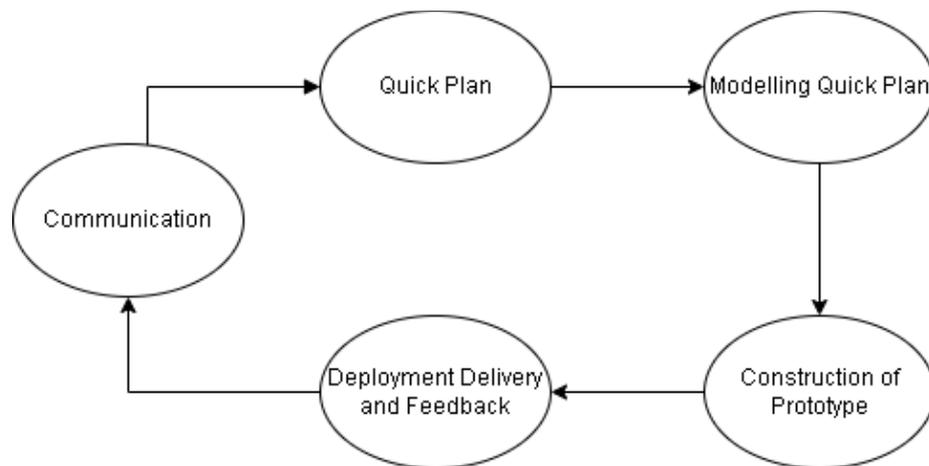
2.2.5 Sistem Informasi Pendaftaran

Sistem informasi pendaftaran *online* adalah sebuah solusi teknologi yang memungkinkan pendaki untuk mendaftar dan melakukan pendakian di gunung tanpa perlu hadir langsung di lokasi. Ini memungkinkan efisiensi dalam proses pendaftaran dan memanfaatkan berbagai fitur teknologi [27]. Implementasi *e-ticketing* juga dapat membantu mengurangi antrian dan waktu yang diperlukan dalam pendaftaran manual, memungkinkan pendaki untuk merencanakan kunjungan mereka dengan lebih baik. Selain itu, ini adalah upaya untuk menyediakan pengalaman pendakian yang lebih modern dan nyaman dengan memanfaatkan teknologi informasi [5].

2.2.6 Metode *Prototype*

Model *prototype* merupakan salah satu metode pengembangan perangkat lunak yang bertujuan untuk membuat mockup atau model awal dari aplikasi sebagai representasi dari produk akhir yang diinginkan. Metode ini sangat berguna ketika pengguna tidak dapat dengan jelas menyajikan informasi mengenai kebutuhan yang mereka inginkan. Model *prototype* memungkinkan pengembang untuk menerima masukan

dan umpan balik pengguna selama proses pengembangan, sehingga produk yang dibuat sesuai dengan kebutuhan pengguna. Selain itu, metode ini memudahkan komunikasi pengembang-pengguna dan mempercepat proses pengembangan aplikasi [28].



Gambar 2.2 Model *Prototype* [29]

1. *Communication*

Developer dan klien bertemu untuk menetapkan kebutuhan dan tujuan umum. Mereka juga menetapkan komponen yang diperlukan selama proses pengembangan.

2. *Quick Plan*

Pada tahap perancangan, developer dan klien bertemu untuk menentukan tujuan dan kebutuhan software, serta membuat gambaran tentang bagian-bagian yang akan dibutuhkan untuk software yang akan dikembangkan. Perancangan ini mencakup semua komponen software yang sudah diketahui dan berjalan dengan cepat. Rancangan ini dapat digunakan sebagai dasar untuk membuat prototype.

3. *Modelling Quick Design*

Modeling Quick Design adalah metode perancangan yang menekankan representasi aspek perangkat lunak yang dapat dilihat oleh pengguna. Model ini lebih berorientasi pada pembuatan prototype yang mencakup semua aspek perangkat lunak yang

diketahui. Perancangan menggunakan model ini biasanya dilakukan dengan cepat dan menjadi dasar untuk pembuatan prototipe.

4. *Construction of Prototype*

Membangun versi awal perangkat lunak yang menyoroti elemen-elemen yang dapat dilihat dan digunakan oleh pengguna dikenal sebagai *Construction of Prototype*. Ini adalah model atau sketsa awal dari perangkat lunak yang akan dikembangkan.

5. *Deployment Delivery & Feedback*

Prototipe perangkat lunak akan diserahkan kepada pengguna untuk dievaluasi dan memberikan masukan. Masukan ini akan digunakan oleh pengembang untuk merevisi kebutuhan dan mengembangkan perangkat lunak yang akan dibuat [30].

2.2.7 UML (*Unified Modelling Language*)



Gambar 2.3 UML (*Unified Modelling Language*)

Unified Modeling Language (UML) adalah sebuah bahasa yang digunakan untuk pemodelan sistem berorientasi objek [31]. Tujuannya adalah untuk menyediakan cara *visual* yang efektif dalam merancang, mendokumentasikan, dan memahami sistem perangkat lunak yang kompleks [32]. UML membantu dalam menggambarkan berbagai aspek dari sistem, termasuk struktur, interaksi, dan perilaku. Terdapat beberapa jenis diagram UML yang digunakan:

1. *Use Case Diagram*: Menggambarkan bagaimana sistem dan orang-orang yang terlibat dalamnya berinteraksi, membantu memahami kasus pengguna.

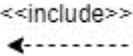
2. *Sequence Diagram*: Diagram ini memodelkan urutan interaksi antara objek-objek dalam sistem, menggambarkan bagaimana pesan atau panggilan metode dikirimkan.
3. *Activity Diagram*: Diagram ini membantu dalam pemodelan proses bisnis dengan menunjukkan aktivitas atau alur kerja sistem [33].

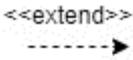
Jenis-jenis diagram UML tersebut membantu tim pengembang perangkat lunak untuk berkomunikasi, merencanakan, dan memahami sistem dengan lebih baik. Dalam pengembangan perangkat lunak, UML menjadi alat yang sangat penting dalam mencapai pemahaman yang jelas dan pemodelan yang akurat.

2.2.8 Use Case

Use Case Diagram adalah sebuah diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk memvisualisasikan interaksi antara sistem perangkat lunak dan aktor-aktor eksternal yang berinteraksi dengannya. Diagram ini memberikan gambaran umum tentang fungsionalitas sistem dengan menampilkan berbagai skenario atau kasus penggunaan yang mungkin terjadi [34]. Tabel 2.2 menunjukkan simbol yang ada di Use Case Diagram.

Tabel 2.2 Simbol pada Use Case Diagram

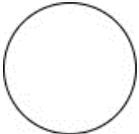
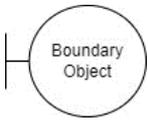
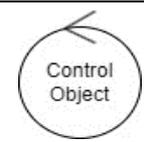
Notasi	Nama Elemen	Fungsi
 Actor	<i>Actor</i>	<i>Actor</i> dapat berinteraksi melalui hubungan spesialisasi atau asosiasi superclass. Aktor di taruh di luar batas subjek.
 Use Case	<i>Use Case</i>	Mewakili bagian dari fungsionalitas sistem di dalam batas sistem.
 Subject	<i>Subject Boundary</i>	Menunjukkan cakupan dari subjek.
	<i>Association Relationship</i>	Menghubungkan actor untuk berinteraksi dengan use case.
 <<include>>	<i>Include Relationship</i>	Menunjukkan bagaimana fungsionalitas satu use case dapat digabungkan dengan use case lain. Arah panah bergerak dari kasus dasar ke kasus yang termasuk.

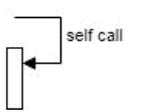
Notasi	Nama Elemen	Fungsi
	<i>Extend Relationship</i>	Menampilkan bagian tambahan dari sebuah use case untuk menambah perilaku opsional. Arah panah bergerak dari extension use case ke base use case.
	<i>Generalization relationship</i>	Menunjukkan generalisasi dari use case khusus ke use case umum.
	<i>Collaboration</i>	Interaksi antara aturan dan elemen lain menghasilkan perilaku yang lebih besar daripada jumlah bagian-bagiannya sendiri.

2.2.9 Sequence Diagram

Salah satu jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan interaksi antara objek-objek dalam sistem perangkat lunak adalah sequence diagram. Sequence diagram memodelkan urutan atau rangkaian pesan yang dikirim antara objek-objek dalam sistem dan juga menunjukkan bagaimana objek-objek tersebut berinteraksi satu sama lain dalam skenario tertentu [35]. Tabel 2.3 menunjukkan simbol yang ada di Sequence Diagram.

Tabel 2.3 Simbol pada Sequence Diagram

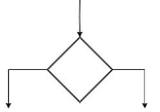
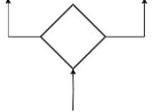
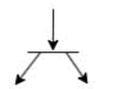
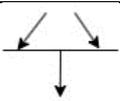
Notasi	Nama Elemen	Fungsi
	<i>Entity Class</i>	Sekumpulan kelas yang terdiri dari entitas-entitas memberikan gambaran awal sistem dan berfungsi sebagai dasar untuk desain basis data.
	<i>Boundary Class</i>	Sekumpulan kelas membentuk interaksi antara aktor dan sistem.
	<i>Control Class</i>	Sebuah item yang memiliki logika aplikasi dan tidak bertanggung jawab atas entitas tertentu..
	<i>Message</i>	Simbol untuk mengirim pesan antar kelas.

Notasi	Nama Elemen	Fungsi
	<i>Recursive</i>	Menggambarkan pengiriman pesan yang dikirimkan ke dirinya sendiri.
	<i>Activation</i>	Activation mewakili eksekusi operasi suatu objek, di mana panjang kotak sebanding dengan durasi aktivitas operasi tersebut.
	<i>Lifeline</i>	Adanya activation ditunjukkan oleh garis putus-putus pada objek sepanjang life line.

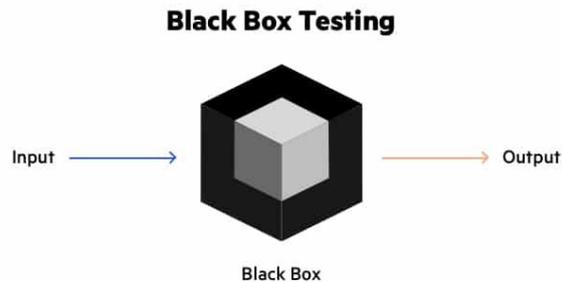
2.2.10 Activity Diagram

Dalam UML, *Activity Diagram* menunjukkan aktivitas atau langkah-langkah yang terjadi dalam proses atau alur kerja [36]. Tabel 2.4 menunjukkan simbol yang digunakan dalam diagram aktivitas.

Tabel 2.4 Simbol pada Activity Diagram

Notasi	Nama Elemen	Fungsi
	<i>Activity</i>	Merepresentasikan sebuah kumpulan aktivitas.
	<i>Control flow</i>	Menunjukkan urutan eksekusi.
	<i>Initial node</i>	Tanda dari awal aktivitas.
	<i>Final activity node</i>	Untuk menunjukkan akhir dari sebuah aktivitas.
	<i>Decision node</i>	Memastikan bahwa alur objek bergerak hanya dalam satu arah.
	<i>Merge node</i>	Mengembalikan berbagai decision path menjadi satu.
	<i>Fork node</i>	Membagi perilaku menjadi aktivitas yang berjalan secara bersamaan.
	<i>Join node</i>	Menggabungkan aktivitas yang berjalan secara simultan.

2.2.11 Pengujian *Black Box*



Gambar 2.4 Pengujian *Black Box*

Pengujian black box adalah proses pengujian yang hanya menguji sistem atau perangkat lunak dari luar, tanpa memerlukan pengetahuan tentang bagaimana sistem atau perangkat lunak tersebut bekerja secara internal. Tujuannya adalah untuk memastikan bahwa sistem atau perangkat lunak dapat menyelesaikan tugas-tugas yang diinginkan dengan tepat sesuai dengan spesifikasi yang ditetapkan, tanpa memperhatikan bagaimana cara kerja sistem atau perangkat lunak tersebut [37].

2.2.12 Metode Kelayakan SUS (*System Usability Scale*)

System Usability Scale (SUS) adalah alat evaluasi *usability* yang dilakukan oleh pengguna dan memberikan nilai terhadap desain sistem. SUS lebih mudah dan efisien dibanding metode evaluasi lain karena hanya menggunakan template pertanyaan yang siap pakai dan dapat langsung digunakan untuk mengevaluasi sistem [38]. Contoh pertanyaan yang digunakan untuk pengujian SUS terdapat pada table 2.6.

Tabel 2.5 Pertanyaan SUS

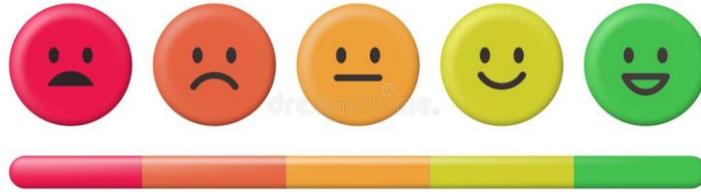
Kode	Pertanyaan	Skor				
P1	Saya berpikir akan sering menggunakan aplikasi ini	1	2	3	4	5
P2	Aplikasi ini tidak terlalu rumit	1	2	3	4	5

Kode	Pertanyaan	Skor				
		1	2	3	4	5
P3	Saya berpikir aplikasi ini mudah digunakan	1	2	3	4	5
P4	Sepertinya saya membutuhkan bantuan untuk dapat menggunakan aplikasi ini	1	2	3	4	5
P5	Saya menebukan berbagai fitur dalam aplikasi ini yang terintegrasikan dengan baik	1	2	3	4	5
P6	Saya menemukan ketidak konsistenan dalam aplikasi ini	1	2	3	4	5
P7	Saya membayangkan bahwa aplikasi ini dapat dipelajari oleh semua orang dengan cepat	1	2	3	4	5
P8	Saya menemukan kerumitan dalam aplikasi ini	1	2	3	4	5
P9	Saya sangat yakin dapat menggunakan aplikasi ini	1	2	3	4	5
P10	Sepertinya saya harus belajar banyak untuk dapat menggunakan aplikasi ini	1	2	3	4	5

Perhitungan Skala *Usability* (*System Usability Scale/SUS*) memiliki beberapa peraturan yang harus diikuti, antara lain:

1. Kuesioner SUS terdiri dari 10 pernyataan yang akan diajukan kepada setiap pengguna aplikasi.
2. Setiap pernyataan pada kuesioner SUS diberi skor kepuasan dalam skala 1 hingga 5, di mana 1 menunjukkan ketidakpuasan dan 5 menunjukkan kepuasan penuh.
3. Nilai akhir kegunaan (*usability level*) dihasilkan dari total skor proses perhitungan kemudian dikalikan 2.

4. Untuk mendapatkan nilai rata-rata survei SUS, hasil dari seluruh responden dijumlahkan dan hasilnya dibagi dengan jumlah responden.



Gambar 2.5 Skala Likert