

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian ini melibatkan studi literatur terhadap literasi ilmiah penelitian terpilih yang relevan dengan tema penelitian, yakni perancangan dan pembangunan aplikasi *dashboard* berbasis *website* dengan menggunakan metode *Prototype*. Pemilihan jurnal penelitian didasarkan pada keterkaitannya dengan topik penelitian yang sedang dilakukan. Jurnal-jurnal penelitian sebelumnya membantu pengembangan teori pada penelitian ini. Penelitian terdahulu juga berperan dalam menentukan metodologi penelitian yang akan digunakan. Penulis dapat mengevaluasi hasil dari penelitian terdahulu untuk dibandingkan dengan hasil yang diperoleh dalam penelitian saat ini.

Penelitian sebelumnya yang pertama dilakukan oleh Ali Ikhwan dan Dina Amalia Putri Lubis pada tahun 2023 dengan topik perancangan sistem informasi berbasis *web* laporan pengaduan masyarakat. Penelitian ini bertujuan guna membantu masyarakat dalam menyampaikan aspirasi kepada Dinas ESDM atau Dinas Energi dan Sumber Daya Mineral Provinsi Sumatera Utara. Metode RAD (*Rapid Application Development*) digunakan dalam proses pengembangan serta pengujian dilakukan menggunakan metode *black box*. Hasil penelitian ini yaitu aplikasi berbasis *website* sistem informasi laporan pengaduan masyarakat di Dinas ESDM Provinsi Sumatera Utara dan meraih hasil yang cukup baik dimana sistem telah bekerja sesuai yang diharapkan [10].

Penelitian sebelumnya yang kedua dilakukan oleh Alfiani, Dedy Panji Agustino, dan I Gusti Agung Vony Purnama pada tahun 2023 dengan topik perancangan sistem informasi pengaduan berbasis *website*. Penelitian ini bertujuan guna membantu masyarakat dan atlet dalam memberikan informasi aduan kendala atau pelayanan dalam menyampaikan aspirasi kepada Komite Olahraga Nasional Indonesia Provinsi Bali. Metode pengembangan yang

digunakan yaitu metode *Waterfall* dengan metode pengujian *black box*. Hasil dari penelitian ini yaitu telah dibangun aplikasi sistem informasi pengaduan berbasis *website* pada Komite Olahraga Nasional Indonesia Provinsi Bali dan meraih hasil yang cukup baik dimana sistem telah bekerja sesuai yang diharapkan [11].

Penelitian sebelumnya yang ketiga dilakukan oleh Rima Tamara Aldis dan Afif Arofi pada tahun 2022 dengan topik merancang sistem berbasis *website* layanan pengaduan. Tujuan penelitian ini untuk membantu pengguna sistem dalam melakukan pengaduan yang telah terjadi. Metode pengembangan yang digunakan yaitu metode *Prototype* serta pengujian *black box*. Hasil dari penelitian tersebut yaitu telah dibangun aplikasi sistem layanan pengaduan berbasis *website* menggunakan metode *Prototype* dan meraih hasil yang baik dimana sistem telah bekerja sesuai yang diharapkan [12].

Penelitian sebelumnya yang keempat dilakukan oleh Gede Surya Mahendra, Ngurah Premadhira, I Gede Bagastia Widi Atmaja, Kadek Ngurah Adi Kusuma, Anak Agung Eka Wirayuda, dan I Komang Widiantera pada tahun 2023 dengan topik merancang sistem informasi pengaduan berbasis *website*. Penelitian ini bertujuan sebagai penyalur aspirasi masyarakat di Kabupaten Buleleng. Metode pengembangan yang digunakan yaitu metode *Prototype* serta pengujian *black box testing*. Hasil dari penelitian tersebut yaitu perancangan sistem informasi pengaduan telah berhasil menerapkan apa yang dibutuhkan oleh *user* dalam metode *Prototype* dan meraih hasil pengujian yang baik [13].

Penelitian sebelumnya yang kelima dilakukan oleh Okta Veza dan Sayuti pada tahun 2021 dengan topik perancangan *dashboard* informasi target pajak kendaraan bermotor. Penelitian ini bertujuan sebagai alat bantu monitoring target Pajak Kendaraan Bermotor pada Badan Pengelolaan Pajak dan Retribusi Daerah (BPPRD) atau Bapenda Provinsi Kepulauan Riau dengan metode *Waterfall*. Hasil dari penelitian ini bahwa sistem sudah dapat berjalan dengan baik dan sudah bisa digunakan oleh masyarakat untuk *monitoring* wajib pajak [14].

Penelitian sebelumnya yang keenam dilakukan oleh Iqbal Wahyudi dan Ahmad Syazili pada tahun 2021 dengan topik sistem *dashboard monitoring website* dosen. Penelitian ini bertujuan sebagai alat bantu *monitoring* aktifitas dosen di Universitas Bina Darma. Metode *Prototype* dipilih sebagai metode pengembangan yang digunakan. Hasil dari penelitian tersebut yaitu telah dibangun pengembangan *website monitoring website* untuk dosen, dengan menyertakan fitur-fitur seperti grafik untuk melihat semua postingan, kemampuan untuk menambah pengguna, pengaturan peran dan izin, serta integrasi dengan *database* melalui *form input* [15].

Penelitian sebelumnya yang ketujuh dilakukan oleh Supriyanto, Iskandar Fitri, dan Nurhayati pada tahun 2022 dengan topik pengembangan sistem informasi *inventory* peralatan mekanik. Penelitian ini diharapkan proses pengelolaan peralatan mekanik pada unit kerja di Universitas Nasional Jakarta pada bidang pelayanan sarana dan prasarana akan lebih baik dan efisien. Metode *Waterfall* digunakan sebagai metode pengembangan serta metode pengujian yang digunakan adalah *white box testing* dan *black box testing*. Hasil dari pengujian metode *white box testing* dan *black box testing* menunjukkan bahwa semua fungsionalitas aplikasi berjalan dengan baik [16].

Penelitian sebelumnya yang kedelapan dilakukan oleh Muhammad Luthfi Hamzah, Umi Fariha, Alfi Syahri, Qhoiril Aldi Giansyah, Fitriani Sandes dan Anisya Caty Praniffa pada tahun 2023 dengan topik sistem informasi parkir berbasis *website* Universitas Islam Negeri Sultan Syarif Kasim Riau. Penelitian ini mengeksekusi atau menemukan kegagalan dalam sistem yang ada dan memungkinkan dilakukannya perbaikan. Metode *black box testing* dan *white box testing* digunakan sebagai metode pengujian. Hasil dari penelitian tersebut yaitu pengujian menggunakan metode *black box testing* dan *white box testing* menunjukkan bahwa tidak ditemukan permasalahan dari sistem yang dibangun [17].

Berdasarkan judul penelitian yang diambil, terdapat beberapa penelitian yang berkaitan dan dapat mendukung penelitian saat ini yang dapat dijadikan acuan. Ringkasan penelitian ini dapat di lihat pada Tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

| No | Judul | Tahun | Metode | Masalah | Hasil |
|----|---|-------|---|---|--|
| 1 | Perancangan Sistem Informasi Laporan Pengaduan Masyarakat Berbasis Web pada Dinas ESDM Sumut [10] | 2023 | RAD (<i>Rapid Application Development</i>) dan <i>black box testing</i> | Di Dinas ESDM Provinsi Sumatera Utara, masih terjadi masalah di mana laporan pengaduan masyarakat tergabung dalam surat masuk lainnya. | Dalam rangka meningkatkan pelayanan kepada masyarakat dan efisiensi kerja di Dinas ESDM Sumatera Utara, telah diciptakan Sistem Informasi Laporan Pengaduan Masyarakat Berbasis Web. |
| 2 | Sistem Informasi Pengaduan Pada Komite Olahraga Nasional Indonesia Provinsi Bali Berbasis Web [11] | 2023 | <i>Waterfall</i> dan <i>black box testing</i> | Karena KONI atau Komite Olahraga Nasional Indonesia Provinsi Bali tidak memiliki sistem pengaduan, kendala yang dialami masyarakat saat berkunjung tidak dapat ditangani secara langsung. | Masyarakat dapat menggunakan <i>website</i> pengaduan KONI untuk mengadakan fasilitas dan pelayanan yang ada di KONI Provinsi Bali. |
| 3 | Penerapan Metode Metode <i>Prototyping</i> pada Perancangan Sistem Layanan Pengaduan berbasis <i>Website</i> [12] | 2022 | <i>Prototyping</i> dan <i>black box testing</i> | Sulit bagi pengguna untuk mengajukan keluhan atau melaporkan insiden yang mereka alami. | Penerapan metode <i>prototype</i> dalam perancangan sistem layanan pengaduan berbasis <i>web</i> ini telah diimplementasikan dan bertujuan untuk membantu dalam pengolahan pengaduan, serta dalam penyusunan laporan pengaduan penggunaan dan pengaduan. |
| 4 | Penerapan Metode Metode <i>Prototype</i> pada Perancangan Sistem | 2023 | <i>Prototype</i> dan <i>black box testing</i> | Pemerintah Buleleng masih menggunakan sistem pengaduan manual yang mengharuskan pencatatan semua | <i>Prototype</i> ini telah dirancang dengan cermat dan sesuai dengan rencana, sehingga menjadi patokan yang baik |

| No | Judul | Tahun | Metode | Masalah | Hasil |
|----|--|-------|------------------|---|--|
| | Informasi Pengaduan Masyarakat Buleleng berbasis <i>Website</i> [13] | | | proses pengaduan secara fisik di atas kertas. Akibatnya, terdapat potensi kesalahan dan ketidakefisienan dalam laporan yang dapat menghambat proses tindak lanjut oleh pemerintah. | untuk pengembangan sistem di masa mendatang. |
| 5 | Perancangan <i>Dashboard</i> Informasi <i>Target</i> Pajak Kendaraan Bermotor pada Badan Pengelolaan Pajak dan Retribusi Daerah Provinsi Kepulauan Riau [14] | 2021 | <i>Waterfall</i> | Manajemen dan pemangku kepentingan menghadapi kesulitan dalam memantau dan mengevaluasi <i>target</i> pajak kendaraan karena jumlah informasi yang diperlukan sangat besar. Selain itu, format pelaporan yang menggunakan tabel juga membuat sulit untuk mengestimasi <i>target</i> pajak kendaraan bermotor. | Program Sistem Informasi Perpajakan dan Administrasi Perpajakan Kepulauan Riau beroperasi dengan sukses, telah diadopsi oleh wajib pajak dan UPT PPD dan berfungsi dengan baik. |
| 6 | <i>Dashboard Monitoring Website</i> Dosen Studi Kasus Universitas Bina Darma [15] | 2021 | <i>Prototype</i> | Universitas Bina Darma memiliki beberapa aktivitas dosen. Namun, saat ini untuk melacak aktivitas para dosen, administrator harus membuka <i>database</i> secara bersamaan. Sistem ini kurang efisien dan tidak efektif dalam memantau karena | Dalam penelitian ini, dibangun sebuah situs <i>web</i> untuk melacak aktivitas yang dilengkapi dengan grafik untuk memvisualisasikan semua postingan, kemampuan untuk menambahkan pengguna, mengatur peran dan hak akses, serta memungkinkan |

| No | Judul | Tahun | Metode | Masalah | Hasil |
|----|--|-------|---|---|--|
| | | | | membutuhkan waktu yang banyak. | transfer data antar <i>database</i> melalui formulir. |
| 7 | Aplikasi <i>Inventory Peralatan Mekanik Unit BRT UNAS Berbasis Web</i> Menggunakan Metode <i>Black-Box</i> dan <i>White-Box Testing</i> [16] | 2021 | Metode <i>waterfall</i> , serta pengujian <i>black box</i> dan <i>white box</i> | Sistem peminjaman alat di unit BRT menggunakan proses manual dengan catatan alat yang disimpan dalam buku atau kertas. Namun, ini tidak menjamin penyimpanan data yang akurat dan pengembalian alat terkadang tidak sesuai, yang menyulitkan manajemen stok dan inventaris alat untuk pengumpulan data pekerjaan. | Untuk meningkatkan efisiensi dan mengurangi risiko kehilangan barang, Sistem Informasi Inventori Peralatan Mekanik Unit BRT UNAS berbasis web membantu mengelola proses peminjaman dan pemesanan alat mekanik. Semua fungsi aplikasi berjalan dengan baik, seperti yang ditunjukkan oleh pengujiannya. Ini memungkinkan pengguna menggunakan aplikasi dengan baik. |
| 8 | Pengujian <i>Black Box</i> dan <i>White Box</i> Sistem Informasi Parkir Berbasis <i>Web</i> [17] | 2023 | <i>Black box</i> serta <i>white box testing</i> | Implementasi sistem parkir di Universitas Islam Negeri Sultan Syarif Kasim Riau belum terstruktur dan terpantau dengan baik, serta kurangnya informasi verifikasi mengenai kendaraan siswa, guru, dan staf menghambat evaluasi parkir. Tidak adanya pengaturan yang efektif oleh satpam mengakibatkan masalah dalam mengatur lalu lintas kendaraan di kampus. | Hasil pengujian <i>white box</i> dan <i>black box</i> pada sistem informasi parkir berbasis <i>web</i> UIN SUSKA Riau menunjukkan bahwa sistem tersebut dibangun dengan baik dan berfungsi dengan baik sesuai kebutuhan, dan setiap pengguna memiliki akses yang tepat ke sistem. |

Berdasarkan penelitian terdahulu, dapat disimpulkan bahwa metode *Prototype* dapat dilakukan untuk pengembangan sistem secara optimal. Perbedaan dari penelitian sebelumnya yaitu terletak pada objek penelitian metode dan proses yang dianalisis yaitu dengan objek aplikasi *dashboard* pelaporan pengaduan kendala sistem internal metode perancangan dengan *Prototype* dan untuk metode pengujian menggunakan metode *black box testing* dan *white box testing*.

2.2 Landasan Teori

Beberapa teori yang digunakan sebagai pendukung dalam penelitian ini yang tersaji pada bagian berikut ini. Guna memahami dan menganalisis masalah yang diteliti, teori-teori ini memberikan dukungan konseptual dan metodologis yang penting.

2.2.1 Rancang Bangun

Rancang Bangun merupakan proses merancang, merencanakan, dan membuat pengaturan dari beberapa elemen yang terpisah untuk membentuk suatu kesatuan yang memiliki fungsi yang diinginkan. Ini melibatkan menginterpretasikan hasil analisis menjadi sebuah perangkat lunak yang lengkap dan menciptakan atau memperbaiki sistem yang sudah ada [18].

2.2.2 Aplikasi

Aplikasi adalah suatu program yang dirancang agar memberikan fungsi dan penggunaan kepada pengguna atau *target* yang dituju. Kata "aplikasi" berasal dari istilah "*application*" yang merujuk pada penggunaan atau implementasi suatu program [19]. Aplikasi juga dapat didefinisikan sebagai sekumpulan perintah atau kode yang disusun secara sistematis untuk menjalankan suatu perintah yang diberikan oleh manusia melalui komponen atau *hardware* komputer yang digunakan oleh manusia dalam menjalankan program aplikasi, yang memungkinkan manusia untuk menyelesaikan masalah yang diinginkan [20].

2.2.3 Dashboard

Dashboard adalah representasi grafis yang digunakan oleh organisasi untuk melihat informasi yang relevan dan memantau kinerja organisasi dengan tujuan mencapai hasil yang diinginkan [21]. Informasi yang relevan dikompilasi dalam satu halaman pada *dashboard* untuk mempermudah pemahaman pengguna, sementara penggunaan *dashboard* dengan fitur interaktif seperti filter hari dan tahun menjadi pilihan terbaik untuk menyajikan visualisasi data yang interaktif [22].

2.2.4 Laporan Pengaduan

Laporan pengaduan merupakan mekanisme dalam pengendalian internal perusahaan yang bertujuan untuk mengungkapkan pelanggaran yang terjadi di dalam organisasi. Laporan pengaduan harus dilakukan dengan itikad baik, bukan berdasarkan niat jahat atau fitnah, dan tidak berhubungan dengan kebijakan tertentu. Melalui laporan pengaduan, pelapor dapat melaporkan tindakan curang atau pelanggaran yang dilakukan oleh bagian internal organisasi [23].

2.2.5 Kendala

Kendala adalah faktor yang menghalangi perusahaan dalam mencapai tujuannya, dapat disebabkan oleh berbagai aspek dan mempengaruhi semua elemen dalam organisasi, sistem, dan produk [24]. Selain itu, kendala dapat didefinisikan sebagai hambatan yang muncul saat seseorang melakukan suatu tugas yang menghalanginya dari melakukan tugas tersebut, baik secara disengaja maupun tidak disengaja [25].

2.2.6 Sistem

Sistem merupakan kumpulan dari beberapa bagian yang bekerja sama untuk mencapai suatu tujuan, dan yang peran utamanya adalah menerima masukan, memprosesnya, dan menghasilkan keluaran [26]. Sistem adalah kumpulan bagian yang saling terkait satu sama lain untuk

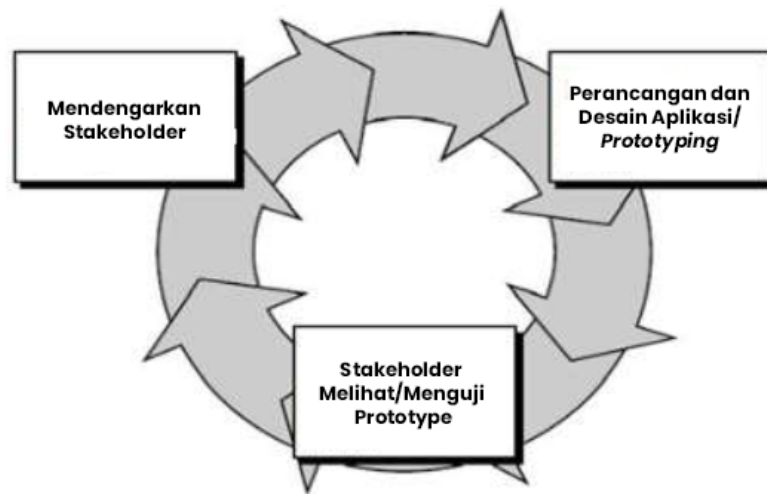
melakukan kegiatan penting untuk mencapai tujuan tertentu [27]. Sistem adalah kumpulan komponen sistem yang saling berhubungan untuk mencapai tujuan organisasi atau perusahaan yang paling penting [28]. Dapat disimpulkan bahwa sistem adalah kumpulan komponen yang saling terikat, yang bekerja sama mencapai tujuan tertentu, dan terikat satu sama lain untuk melakukan kegiatan penting guna mencapai tujuan tersebut.

2.2.7 Aplikasi *Website*

Aplikasi *website* merupakan sistem informasi yang menyediakan antarmuka pengguna berbasis *website* untuk mendukung interaksi pengguna. Interaksi pengguna pada internet terdiri dari tiga fase, yaitu permintaan, proses, dan tanggapan. Sebuah *website* terdiri dari kumpulan halaman yang dapat memberikan informasi dengan berbagai bentuk seperti teks, gambar, animasi, suara, atau kombinasi elemen statis dan dinamis. Halaman-halaman tersebut saling terhubung membentuk rangkaian bangunan yang membentuk jaringan situs yang terhubung [29]. Dapat disimpulkan bahwa *website* adalah kumpulan alamat *website* yang dibuat dengan menggunakan bahasa pemrograman HTML dengan isian informasi berupa teks, gambar, animasi, dan lainnya.

2.2.8 Metode *Prototype*

Prototyping adalah proses merancang model awal dari sebuah produk yang belum sepenuhnya memiliki semua fitur produk yang sebenarnya, tetapi sudah mencakup fitur-fitur utama. Model ini digunakan untuk keperluan pengujian dan sering kali menjadi dasar uji coba sebelum memasuki tahap produksi yang sebenarnya. Metode *Prototype* memungkinkan interaksi antara pengembang dan pengguna selama proses pembuatan produk. Dalam konteks perangkat lunak, *prototyping* merupakan salah satu metode dalam siklus hidup sistem yang bertujuan mengembangkan model menjadi sistem final dengan lebih cepat serta biaya yang lebih rendah. Ada berbagai cara dan pendekatan yang dapat digunakan dalam melakukan *prototyping* [30].



Gambar 2.1 Tahapan Metode *Prototype* [30][31]

Langkah-langkah metode *Prototype* adalah sebagai berikut.

1. Mendengarkan Pengguna

Pada tahap ini, keluhan pelanggan didengar untuk mengumpulkan kebutuhan sistem. Untuk membuat sistem yang sesuai dengan kebutuhan, Peneliti harus mengetahui bagaimana sistem yang sedang berjalan sebelum mengetahui masalah yang terjadi [30]. Contoh tahapan ini adalah sebagai berikut.

3.1. Analisa Kebutuhan

1. Kebutuhan Pengguna

Dalam sistem presensi disini pegawai memiliki hak akses terhadap pengaksesan aplikasi, akses tersebut antara lain sebagai berikut:

- 1) Pegawai dapat melakukan *login* sebagai Pegawai
- 2) Pegawai dapat melakukan absensi masuk
- 3) Pegawai dapat mengajukan izin
- 4) Pegawai dapat melihat riwayat absensi
- 5) Pegawai dapat edit *profile* dan *password*
- 6) Pegawai dapat melakukan *logout*

2. Kebutuhan Sistem

Dalam sistem digunakan harus memenuhi kebutuhan *user* agar dapat digunakan dengan mudah. Kebutuhan sistem presensi di desa Mekar Jaya meliputi antara lain seperti berikut:

- a. Setiap pegawai wajib melakukan *login* terlebih dahulu untuk dapat mengakses aplikasi dengan memasukkan NIP pegawai dan *password* agar keamanan privasi pegawai tetap terjaga.
- b. Pengguna dapat memiliki hak akses untuk menggunakan aplikasi presensi pegawai.
- c. Sistem dapat merekap hasil presensi pegawai untuk laporan setiap satu bulan sekali.
- d. Setelah menggunakan aplikasi pegawai dapat melakukan *logout*.

Gambar 2.2 Contoh Mendengarkan Pengguna[32]

2. Perancangan dan Desain Aplikasi (*Prototyping*)

Pada fase ini, sistem dirancang dan dibuat prototipenya. Prototipe dibangun sesuai dengan persyaratan sistem yang telah ditentukan berdasarkan keluhan pengguna [30].

3. Pengguna Melihat/Menguji *Prototype*

Pada fase ini, pengguna menguji prototipe sistem. Penilaian kekurangan sistem yang teridentifikasi kemudian dibuat berdasarkan kebutuhan pengguna. Pengembangan selanjutnya adalah mendengarkan keluhan pengguna kembali untuk meningkatkan prototipe yang telah di rancang [30]. Contoh tahapan ini adalah sebagai berikut.

TABEL 2
TABEL *FEEDBACK* PENGGUNA

| Komponen | <i>Feedback</i> Pengguna | Kesimpulan |
|---------------------------------|---|------------------|
| Halaman Beranda. | Menyarankan untuk menambahkan grafik yang menampilkan diagram jumlah surat masuk, jumlah surat keluar, dan statistik jumlah surat masuk & surat keluar yang masuk setiap bulan dan tahun. | Perlu dievaluasi |
| Halaman Surat. | <ul style="list-style-type: none"> Menyarankan untuk menambahkan fitur log aktivitas dan riwayat surat, dan menambahkan kolom status surat pada tabel. Menyarankan untuk penambahan fitur <i>tracking</i> surat masuk, surat keluar, dan disposisi surat masuk. | Perlu dievaluasi |
| Halaman Tambah Surat. | Fungsi sudah berjalan sesuai dengan yang direncanakan. | Diterima |
| Halaman Disposisi Surat. | Fitur sudah berjalan dengan baik. | Diterima |
| Halaman Tambah Disposisi Surat. | Fungsi sudah berjalan dengan baik. | Diterima |
| Halaman Cetak Laporan Surat. | Fungsi cetak sudah berjalan dengan baik pada periode tanggal, bulan, maupun tahun. | Diterima |
| Halaman Cetak Lembar Disposisi. | Fitur cetak lembar disposisi sudah berjalan dengan baik. | Diterima |

Gambar 2.3 Contoh Evaluasi Pengguna[33]

Kelebihan dalam menggunakan metode *Prototype* adalah sebagai berikut [30].

1. Kemampuan komunikasi antara pengembang dan pengguna dapat terjalin dengan baik.
2. Pengembangan dapat memenuhi kebutuhan pengguna dengan lebih baik.
3. Menghemat lebih banyak waktu dalam proses pengembangan sistem.
4. Penerapan menjadi lebih mudah karena pengguna tahu apa yang diharapkannya.
5. Melibatkan pengguna dalam proses pengembangan sistem, pengembang dapat lebih mudah mengetahui produk yang diinginkan pengguna.

Pada dasarnya, prototipe berfungsi sebagai cara untuk menemukan kebutuhan perangkat lunak. Setelah pengembangan prototipe dimulai, pengembang berusaha untuk memberikan gambaran kepada pengguna atau pemilik, seperti *form input*, *form proses*, *form output*, laporan, *database*, sistem operasi dan sebagainya, yang memungkinkan program diselesaikan dengan cepat [34].

Metode *prototype* memungkinkan pengguna untuk memberikan informasi yang diperlukan untuk pengembangan sistem dan memungkinkan pengembang sendiri untuk membangun sistem dengan cepat [34]. Namun, metode ini memiliki kekurangan adalah sebagai berikut [34].

1. Pengguna kadang-kadang tidak menyadari bahwa perangkat lunak yang ada tidak memiliki kualitas perangkat lunak secara keseluruhan atau kemampuan pemeliharaan jangka panjang.
2. Pengembang biasanya ingin proyek mereka diselesaikan dengan cepat, jadi mereka dapat membuat prototyping lebih cepat dengan menggunakan algoritma dan bahasa pemrograman yang sederhana tanpa mempertimbangkan bahwa program tersebut hanyalah cetak biru sistem.
3. Hubungan antara pengguna dan komputer yang disediakan mungkin bukan mencerminkan contoh perancangan yang baik.

Model prototipe dapat menjadi paradigma yang efektif untuk rekayasa perangkat lunak, meskipun ada beberapa masalah yang mungkin muncul. Aturan main harus ditetapkan sejak awal. Artinya, pengembang dan pengguna harus setuju bahwa prototipe dibuat dengan tujuan yang jelas dan dengan batasan yang jelas juga. Selain itu, pengguna harus jelas dalam memberikan informasi yang dibutuhkan, memberikan data saat ini, dan memberikan aturan yang berlaku untuk sistem yang akan dibangun. Selain itu, pengembang harus bertindak dengan bijak dan hati-hati terhadap apa yang telah diberikan oleh pengguna [34].

2.2.9 PT. Telkom Indonesia Witel Purwokerto

PT. Telkom Indonesia (persero) Tbk. merupakan Badan Usaha Milik Negara (BUMN) yang berfokus pada layanan jaringan telekomunikasi serta teknologi informasi di Indonesia [35][36]. Pemerintah Indonesia mendirikan Perusahaan Negara Pos dan Telekomunikasi (PN Postel) pada tahun 1961, namun kemudian memisahkan bidang pos dan telekomunikasi melalui peraturan pada tahun 1965. Dengan pemisahan ini, terbentuklah PN Telekomunikasi yang merupakan awal dari PT. Telkom Indonesia yang saat ini menyediakan enam kategori produk yang melayani segmen bisnis, perumahan, personal, dan konsumen lainnya, termasuk *Enterprise Digital*, *Mobile*, *Fixed*, *Consumer Digital*, *Network*, dan *Wholesale & International* [36].

PT. Telkom Indonesia Witel Purwokerto merupakan bagian dari PT. Telkom Indonesia yang beroperasi sebagai unit bisnis di kota Purwokerto dengan fokus pada pendekatan berorientasi pada pelanggan dalam menjalankan kegiatan bisnisnya [37]. PT. Telkom Indonesia Witel Purwokerto mengelola di daerah Purwokerto sendiri, Purbalingga, Banjarnegara, dan Cilacap.

2.2.10 Indihome

Indihome merupakan salah satu produk milik PT. Telkom Indonesia yang menyediakan layanan jaringan internet yang telah menyebar ke seluruh wilayah Indonesia [38]. Keunggulan Indihome terletak pada penggunaan teknologi fiber optik yang memberikan kestabilan dan kecepatan tinggi dalam akses internet. Dengan kecepatan hingga 300Mbps dan ketahanan terhadap kondisi cuaca, Indihome menawarkan beragam pilihan kecepatan dan paket yang dapat disesuaikan dengan kebutuhan pelanggan [39]. Hal ini menjadikan Indihome sebagai pilihan yang menarik bagi masyarakat yang menginginkan layanan internet handal. Perkembangan Indihome terus berlanjut dengan peningkatan

minat masyarakat melalui penawaran teknologi fiber optik yang semakin cepat, stabil, andal, dan canggih [40].

2.2.11 Tipe Order

Activation Order (AO) adalah permintaan dari pelanggan untuk memasang layanan baru atau melakukan instalasi produk dari perusahaan.

Modification Order (MO) adalah permintaan dari pelanggan untuk menambah layanan yang telah dimiliki sebelumnya atau meningkatkan paket layanan internet dari perusahaan.

Disconnect Order (DO) adalah permintaan dari pelanggan untuk menghentikan layanan yang telah dimiliki sebelumnya atau berhenti berlangganan dari layanan perusahaan.

Suspend Order (SO) adalah tindakan sementara perusahaan untuk menghentikan layanan pelanggan karena pembayaran tagihan belum dilunasi, sehingga pelanggan tidak dapat menggunakan layanan dari perusahaan.

Resumption Order (RO) adalah permintaan dari pelanggan untuk mengaktifkan kembali layanan setelah terkena *suspend order* dari perusahaan, dengan syarat bahwa pembayaran tagihan yang tertunda sudah diselesaikan.

2.2.12 Wireframe

Wireframe merupakan metode yang digunakan untuk merancang tampilan suatu aplikasi, baik itu *website* atau aplikasi *mobile*, dengan fokus pada struktur dan *layout*. *Wireframe* digunakan sebagai dasar untuk membuat prototipe dengan cepat dan menjadi awal dari ide desain yang konkret [41]. Sebelum proses desain dimulai, *wireframe* adalah kerangka atau sketsa kasar yang digunakan untuk menata komponen pada sebuah *website* atau aplikasi *mobile*. Seorang desainer UI biasanya ditugaskan untuk merancang *wireframe*. Mereka dapat menggunakan coretan kasar pada kertas atau alat desain khusus untuk *wireframing* seperti aplikasi

Balsamiq *Wireframe*. *Wireframe* hanya terdiri dari garis dan kotak yang mengatur tata letak elemen elemen pada *website* atau aplikasi *mobile* [42].

2.2.13 Desain UI/UX

UI dan UX merupakan akronim dari *User Interface* (antarmuka pengguna) dan *User Experience* (pengalaman pengguna), yang mengacu pada elemen visual dalam sebuah aplikasi seperti *website*, yang bertujuan untuk meningkatkan citra merek perusahaan atau bisnis [43].

2.2.13.1 *User Interface* (UI)

UI atau *User Interface* merupakan disiplin yang berkaitan dengan penyusunan grafis dalam suatu situs *website* atau aplikasi. Fokus *User Interface* mencakup berbagai elemen interaktif seperti tombol, teks, gambar, dan bidang *input* teks, serta mengatur tata letak, animasi, transisi, dan elemen interaksi kecil lainnya. Desainer *User Interface* bertanggung jawab untuk merancang aspek visual, termasuk pemilihan skema jenis font, bentuk tombol, dan warna yang digunakan, dengan tujuan menciptakan tampilan yang menarik dan meningkatkan keterikatan pengguna [43].

2.2.13.2 *User Experience* (UX)

Desainer UX atau *User Experience* bertanggung jawab dalam menciptakan produk yang memiliki nilai dan mengubah alur pengguna menjadi desain produk yang teruji dan menarik. Mereka bekerja sama dengan elemen lain untuk mencapai keselarasan antara kemajuan teknologi, tujuan bisnis, kebutuhan pengguna serta menghasilkan produk yang berarti, bermanfaat, dan menyenangkan. Desain *User Experience* yang dibuat akan mempengaruhi tingkat keterjangkauan dan pengalaman pengguna dalam berinteraksi dengan situs *website*. Kemampuan dasar seperti pembuatan *wireframe* dan desain *mockup* menjadi penting bagi seorang desainer *User Experience* [43].

2.2.14 Bahasa Pemrograman

Bahasa pemrograman adalah seperangkat aturan sintaks yang digunakan untuk menciptakan program komputer dan menghasilkan keluaran yang digunakan untuk membuat program. Bahasa pemrograman memungkinkan pemrogram untuk mengidentifikasi data apa yang sedang diproses komputer, cara menyimpan atau mengirimkannya, dan prosedur apa yang harus dilakukan dalam berbagai situasi [44].

2.2.14.1 HTML

Bahasa pemrograman HTML (*HyperText Markup Language*) digunakan untuk membuat halaman *website* yang menggunakan tag $\langle \rangle$ untuk mengatur tampilan halaman sesuai dengan posisi yang ditentukan. HTML berfungsi sebagai struktur dasar dalam merancang halaman *website* sebelum dilakukan desain dan pengembangan fungsionalitas. Selain HTML, akan digunakan juga bahasa pemrograman *Cascading Style Sheets* (CSS) untuk mengatur tampilan dan gaya halaman *website* [45].

2.2.14.2 CSS

Cascading Style Sheets (CSS) adalah serangkaian bahasa yang digunakan dalam mengontrol penampilan halaman *website*, seperti warna, tata letak, dan jenis huruf. Dengan menggunakan CSS, pengembang *website* dapat menciptakan halaman *website* yang responsif terhadap berbagai ukuran layar. Biasanya, pemrosesan CSS terpisah dari halaman HTML, namun juga memungkinkan untuk menambahkan kode CSS langsung ke dalam halaman HTML guna mempermudah pengaturan pada halaman yang memiliki desain serupa. [46].

2.2.14.3 PHP

PHP merupakan bahasa pemrograman *script server-side* yang digunakan dalam mengembangkan *website* serta dapat digunakan

secara umum. PHP awalnya dibuat oleh Rasmus Lerdorf pada tahun 1994 dan saat ini PHP merupakan kepanjangan dari *Hypertext Preprocessor*. PHP adalah sebuah bahasa pemrograman yang bebas digunakan dan bersifat *open source*, dengan lisensi PHP yang sedikit berbeda dari lisensi GNU *General Public License* yang umum digunakan oleh proyek *open source* [47].

2.2.15 Database MySQL

Database adalah koleksi informasi yang terstruktur dan disimpan di dalam komputer, yang dapat diakses dan diambil menggunakan program komputer. Kabinet basis data atau *database* dapat dianggap sebagai tempat penyimpanan untuk basis data tersebut [48]. *Database* memiliki peran sentral dalam pengembangan sistem informasi, karena memiliki fungsi sebagai tempat penyimpanan dan pengaturan data yang ada dalam sistem, sehingga memungkinkan informasi dapat diakses dan disusun dalam berbagai format. *Database* merupakan kumpulan data terkait yang diorganisasi serta disimpan secara terpadu dalam komputer, sehingga dapat menyediakan informasi yang optimal sesuai dengan kebutuhan pengguna [49].

MySQL merupakan *database multi-user* menggunakan SQL (*Structured Query Language*) menjadi sumber terbuka sistem manajemen basis data SQL yang paling populer dan yang tersedia saat ini. Fitur-fitur seperti kemampuan *multithreading*, dukungan untuk banyak pengguna, dan sistem manajemen basis data SQL membuat MySQL menjadi pilihan yang ideal untuk kebutuhan sistem basis data yang cepat, andal, dan mudah digunakan [49].

2.2.16 UML (*Unified Modeling Language*)

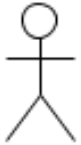


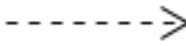
Bahasa standar *Unified Modeling Language* (UML) digunakan untuk mendokumentasikan desain dan membuat model perangkat lunak. Dengan menggunakan UML, pengembang perangkat lunak dapat memvisualisasikan dan mendefinisikan berbagai bagian sistem secara

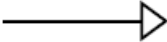
visual. Diagram UML berfungsi sebagai dokumen arsitektur yang digunakan oleh kontraktor saat membangun gedung, dan membantu pemrogram membuat perangkat lunak. Dengan menjadi lebih akrab dengan kosakata yang digunakan dalam UML, detailnya menjadi lebih mudah dipahami [50]. Jenis diagram UML yang biasanya digunakan dalam pengembangan sistem adalah sebagai berikut.

2.2.16.1 Use Case Diagram

Use case menggambarkan tahapan tindakan yang memberikan nilai terukur bagi aktor, dengan menggunakan *actor*, *use case*, dan relasi di dalamnya. Dalam diagram, *use case* digambarkan sebagai elips horizontal [51].

Tabel 2.2 Lambang *Use Case Diagram*

| Lambang | Nama | Keterangan |
|---|-----------------|--|
|  | Aktor | Aktor adalah entitas luar sistem yang berinteraksi dengannya. Ini dapat berupa orang, proses, atau sistem lain, dan biasanya disebut dengan kata benda pada awal nama aktor. |
|  | <i>Use Case</i> | Antar unit dapat berkomunikasi satu sama lain melalui pertukaran pesan antara aktor atau unit, yang biasanya ditunjukkan dengan kata kerja di awal nama <i>use case</i> . Ini menunjukkan fungsionalitas sistem. |
|  | Asosiasi | Aktor dan <i>use case</i> berinteraksi atau partisipasi dalam <i>use case</i> . |
|  | Ekstensi | <i>Use case</i> yang dapat berdiri sendiri tanpa bergantung pada <i>use case</i> tambahan |





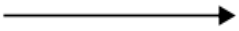
| Lambang | Nama | Keterangan |
|---|--------------|--|
| | | lainnya dihubungkan melalui relasi <i>use case</i> tambahan. |
|  | Generalisasi | Sebuah hubungan antara generalisasi dan spesialisasi antara dua <i>use case</i> menunjukkan bahwa satu memiliki fungsi yang lebih umum daripada yang lain. |

Sumber: Septilia A., Hariyanto W. dan Ade Frima S. (2020) [52]

2.2.16.2 Activity Diagram

Diagram *activity* berguna untuk memvisualisasikan urutan aktivitas, objek, status, transisi status, dan peristiwa dalam suatu sistem. Secara sederhana, diagram aktivitas menjelaskan bagaimana sistem berperilaku saat menjalankan operasinya [51].

Tabel 2.3 Lambang *Activity Diagram*


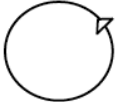

| Lambang | Nama | Keterangan |
|---|-------------------------------------|---|
|  | <i>Initial State/Starting Point</i> | Proses pertama kali di dalam <i>activity diagram</i> |
|  | <i>Final State/End Point</i> | Proses terakhir kali di dalam <i>activity diagram</i> |
|  | <i>State/Activity</i> | Aktivitas yang terjadi di dalam <i>activity diagram</i> |
|  | <i>Decision</i> | Menggambarkan pilihan untuk pengambilan keputusan |
|  | <i>Control Flow</i> | Urutan perpindahan suatu aktivitas |



Sumber: Marwan M. Londol dan Mirawati Suruanggi (2020) [53]

2.2.16.3 Sequence Diagram

Sequence diagram merupakan suatu bentuk diagram yang memiliki interaksi berurutan antara objek-objek dalam sistem. Diagram *sequence* memberikan representasi visual tentang bagaimana pesan-pesan dikirim antara objek-objek yang terlibat dalam interaksi, seperti pengguna, iklan, dan lainnya. *Sequence diagram* juga digunakan untuk menggambarkan interaksi antara kelas-kelas yang terhubung melalui pesan-pesan yang dikirimkan pada waktu tertentu [54].

Tabel 2.4 Lambang *Sequence Diagram*

| Lambang | Nama | Keterangan |
|---|-----------------------|---|
|  | <i>Entity Class</i> | Bagian dari sistem yang bertanggung jawab menyimpan data atau informasi ke dalam basis data |
|  | <i>Boundary Class</i> | Kumpulan kelas yang berfungsi sebagai <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem. Ini termasuk tampilan <i>form entry</i> dan <i>form cetak</i> . |
|  | <i>Control Class</i> | Suatu objek yang berisi logika aplikasi yang tidak bertanggung jawab kepada entitas. |
|  | <i>Message</i> | Komponen untuk menandakan hubungan komunikasi berbentuk pesan antara dua buah objek. |
|  | <i>Recursive</i> | Pengiriman pesan terhadap dirinya sendiri |





| Lambang | Nama | Keterangan |
|---|-------------------|--|
|  | <i>Activation</i> | Panjang kotak aktivasi berbanding lurus dengan durasi aktivitas operasi. |
|  | <i>Lifeline</i> | Garis putus-putus sepanjang <i>lifeline</i> terdapat <i>activation</i> |

Sumber: Sinta Maria dan Jevrian Efendi (2021) [55]

2.2.16.4 Class Diagram

Class diagram menunjukkan struktur statis kelas-kelas dalam sistem, yang mencakup atribut, fungsi, dan hubungan antar kelas. Tujuan dari *class diagram* adalah memvisualisasikan secara jelas struktur kelas dalam sistem dan digunakan secara luas dalam proses desain perangkat lunak. Dengan menggunakan diagram kelas, dapat dilihat dengan mudah bagaimana kelas-kelas saling terhubung dan berinteraksi dalam sistem yang sedang dibangun. Setiap kelas dalam *class diagram* memiliki tiga komponen penting yang harus ditunjukkan, yaitu nama kelas, atribut dan metoda [51].

Tabel 2.5 Lambang *Class Diagram*

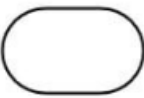



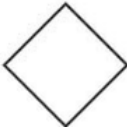

| Lambang | Nama | Keterangan |
|---|-----------------------------|--|
|  | <i>Class</i> | Elemen yang termasuk dalam struktur sistem. |
|  | <i>Association</i> | Asosiasi biasanya disertai dengan <i>multiplicity</i> dalam hubungan kelas. |
|  | <i>Directed Association</i> | Kelas memiliki hubungan dengan maknanya. |
|  | <i>Aggregation</i> | Hubungan antara kelas yang menunjukkan bahwa satu kelas merupakan bagian dari kelas lainnya. |


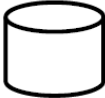
Sumber: Minda Mora Purba (2020) [56]

2.2.17 Flowchart

Flowchart merupakan representasi visual yang menggambarkan tahapan dan susunan prosedur dari sebuah program [56].

Tabel 2.6 Lambang *Flowchart*

| Lambang | Nama | Keterangan |
|---|--------------------------|---|
|  | Terminator | Simbol awal atau akhir digunakan dalam <i>flowchart</i> untuk menandai titik awal atau akhir dari proses. |
|  | Proses | Simbol pemrosesan digunakan dalam alur kerja untuk menggambarkan tahapan atau tindakan yang terjadi dalam proses. |
|  | <i>Input/Output</i> | Simbol <i>input/output</i> digunakan untuk menunjukkan <i>input</i> yang diperlukan dan <i>output</i> yang dihasilkan oleh proses. |
|  | <i>On-Page Reference</i> | Dalam <i>flowchart</i> , proses-proses yang berada pada lembar kerja yang sama dapat dihubungkan melalui simbol konektor. |
|  | <i>Decision</i> | Simbol pengambil keputusan digunakan untuk menentukan proses lanjutan berdasarkan kondisi-kondisi tertentu dalam suatu <i>flowchart</i> . |
|  | <i>Arrow</i> | Simbol penghubung digunakan untuk menghubungkan atau menyambungkan antara proses atau simbol-simbol lain dalam sebuah <i>flowchart</i> . |

| Lambang | Nama | Keterangan |
|---|---------------------------|---|
|  | <i>Off-Page Reference</i> | Simbol penghubung digunakan untuk menghubungkan proses antara lembar kerja yang berbeda dalam sebuah <i>flowchart</i> . |
|  | <i>Database</i> | Simbol yang merepresentasikan <i>database</i> atau basis data dalam sebuah <i>flowchart</i> . |

Sumber: Minda Mora Purba (2020) [56]

2.2.18 Framework

Frameworks adalah bagian dari perangkat lunak yang dapat digunakan kembali, mengeliminasi kebutuhan pengembang untuk membuat *script* yang serupa dalam tugas yang serupa. Sebagai contoh, jika seorang pengembang ingin menampilkan data dengan fungsi halaman pada halaman *website*, sebuah *framework* menyediakan fungsi paginasi yang dapat digunakan, sehingga pengembang hanya perlu mengimplementasikannya selama proses pengkodean, mengikuti aturan dan struktur yang telah ditentukan oleh *framework* tersebut [57].

2.2.18.1 Bootstrap

Bootstrap adalah sebuah alat yang membantu dalam pembuatan tampilan *website*, yang mempercepat proses pengembangan bagi pengembang dan desainer *website* [58]. Bootstrap adalah *framework* CSS yang memungkinkan pengembang membuat tampilan *website* dengan *class* dan komponen yang siap digunakan [59]. Pada tahun 2011 Mark Otto dan Jacob Thornton dua programmer Twitter membuat Bootstrap, sebuah *framework* yang dapat digunakan secara bersama di lingkungan Twitter, Oleh karena itu, meskipun nama resminya Bootstrap, *framework* ini juga dikenal sebagai Twitter Bootstrap di kalangan para pengembang [58].

2.2.18.2 Laravel

Laravel merupakan sebuah *framework web* PHP *open-source* yang dikembangkan oleh Taylor Otwell. *Framework* ini dirancang untuk memudahkan pengembangan aplikasi *web* dengan pola MVC (*Model View Controller*). Salah satu perbedaannya adalah adanya fitur *routing* yang menghubungkan antara permintaan dari pengguna dan *controller*, sehingga *controller* tidak langsung menerima permintaan tersebut [57].

2.2.19 Chatbot

Chatbot merupakan sebuah aplikasi komputer yang dapat berkomunikasi dengan pengguna menggunakan bahasa alami. Teknologi ini pertama kali dikembangkan pada tahun 1960-an dengan tujuan untuk menguji kemampuan suatu sistem dalam meniru interaksi manusia [60][61]. *Chatbot* atau *bot* atau pesan instan adalah aplikasi yang memberikan respons otomatis terhadap pertanyaan yang diajukan oleh pengguna secara tepat waktu. Fungsinya meliputi layanan pelanggan dan pengumpulan informasi, dengan beberapa *chatbot* menggunakan pendekatan untuk mengenali pola kata yang digunakan oleh pengguna. Meskipun beberapa *chatbot* hanya mencocokkan kata kunci dengan *database* jawaban, ada juga yang lebih kompleks dalam memberikan respons yang lebih terarah [62].

2.2.20 Telegram API

API Telegram adalah *platform* yang disediakan oleh Telegram untuk pengembang membuat aplikasi secara gratis. Pengembang menggunakan API ini untuk membuat aplikasi yang berfungsi sebagai penghubung antara Telegram dan sistem pesan instan lainnya, memungkinkan pengguna aplikasi pihak ketiga berkomunikasi dengan pengguna Telegram. Dengan potensi pertumbuhan yang besar, API Telegram menjadi alat penting bagi pengembang untuk membangun aplikasi yang meningkatkan kualitas hidup pengguna [63].

2.2.21 *White Box Testing*

Pengujian *white box* atau yang biasa dikenal sebagai *structural testing*, *glass box testing*, *clear box testing*, dan/atau *transparent box testing* merupakan metode pengujian perangkat lunak yang fokus pada cara kerja aplikasi dan struktur internal. Dalam metode ini, pengetahuan tentang keterampilan pemrograman dan sistem internal digunakan untuk merancang kasus pengujian. Dengan menggunakan *white box testing*, peneliti dapat melakukan pengujian yang lebih mendalam, memeriksa kompleksitas pemrograman dan proses yang terlibat. Tujuan utama pengujian ini untuk mencapai cakupan struktural yang tinggi dari kode yang sedang diuji, termasuk pemanggilan metode sekuensial yang diinginkan [64]. Dalam konteks ini, *cyclomatic complexity* atau $V(G)$ digunakan sebagai ukuran tingkat kompleksitas sebuah program. Angka $V(G)$ menunjukkan jumlah jalur dasar yang minimal harus diuji dalam program tersebut. Terdapat beberapa pendekatan matematis yang dapat digunakan untuk menghitung $V(G)$ ini, salah satunya adalah dengan menggunakan cara berikut:

$$V(G) = R \quad (2.1)$$

$$V(G) = E - N + 2 \quad (2.2)$$

$$V(G) = P + 1 \quad (2.3)$$

Keterangan:

R = *Region* atau Jumlah Area

E = *Edge* adalah jalur atau sambungan antara dua *node* dalam sebuah graf

N = *Node* adalah titik atau blok dalam program

P = *Predicate Node* adalah jenis node khusus dalam *cyclomatic complexity* yang menggambarkan titik percabangan

Berdasarkan penelitian yang dilakukan oleh peneliti lain, ditemukan bahwa semakin tinggi nilai $V(G)$, semakin tinggi pula probabilitas terjadinya kesalahan dalam program [65].

2.2.22 *Black Box Testing*

Pengujian *black box* merupakan metode pengujian perangkat lunak yang berfokus pada kebutuhan fungsional dari *software* yang sedang dikembangkan. Melalui pengujian *black box*, pengembang dapat membuat serangkaian kondisi *input* yang akan menguji semua persyaratan fungsional program dengan sempurna. Metode ini tidak menggantikan uji coba *white box*, melainkan pendekatan yang melengkapi untuk menemukan kesalahan yang mungkin terlewatkan dengan menggunakan metode *white box* [66][67]. Pengujian *black box* bertujuan untuk menemukan kesalahan dalam kategori [66], [67] sebagai berikut:

1. Fungsi yang kurang tepat atau tidak lengkap.
2. Kekurangan atau kesalahan dalam antarmuka sistem.
3. Masalah dengan akses ke *database* eksternal ataupun struktur data.
4. Kinerja atau efisiensi yang mengakibatkan kinerja buruk.
5. Kesalahan dalam proses awal dan akhir program atau komponen.

2.2.23 *Sampling Jenuh*

Sampling Jenuh yaitu teknik penentuan sampel bila semua anggota populasi digunakan sebagai sampel. Hal ini sering digunakan untuk penelitian dengan jumlah sampel dibawah 30 orang, atau untuk penelitian yang ingin membuat generalisasi dengan tingkat kesalahan yang sedikit atau kecil. Misalnya jika jumlah populasi 20 orang, maka 20 orang tersebutlah yang dijadikan sampel [68]. Sampling jenuh masuk ke dalam kategori *nonprobability sampling*. *Nonprobability sampling* adalah metode pengambilan sampel di mana setiap anggota populasi memiliki peluang atau kesempatan yang berbeda untuk diambil sebagai sampel [68].

2.2.24 *Usability*

Bagian dari keilmuan *Human Computer Interaction* (HCI) yang dikenal sebagai *usability* berfokus pada studi desain antarmuka dan interaksi antara manusia dan komputer. Kajian *usability* ini akan

membahas bagaimana pengguna menggunakan teknologi, aplikasi, atau situs *website* tertentu. Indikator *usability* saat ini juga digunakan untuk mengukur seberapa puas pengguna dalam menggunakan aplikasi, teknologi, atau produk tersebut untuk mencapai tujuan atau tujuan. Ukuran keberhasilan sebuah aplikasi atau teknologi dapat di lihat dari seberapa baik aplikasi atau teknologi memberikan layanan berkualitas tinggi kepada pengguna dan seberapa jauh aplikasi atau teknologi dapat mengurangi kemungkinan kesalahan yang disebabkan [69]. *Usability* memiliki komponen-komponen [70], [71] yang digunakan untuk mengukur kualitas *usability* adalah sebagai berikut.

1. *Learnbility*

Saat pertama kali menggunakan *website*, pengguna dapat dengan cepat memahami tampilan dan perintah sistem dasar dan fitur navigasi dari setiap aspeknya.

2. *Efficiency*

Tingkat kemampuan kecepatan pengguna dalam menggunakan *website* berkorelasi dengan kinerja *website* saat berjalan dan saat mengerjakan tugas.

3. *Memorability*

Pengguna dapat kembali menggunakan sistem setelah lama tidak menggunakannya tanpa harus mempelajari tampilannya lagi. Tampilan sistem sangat mudah dipahami dan tidak rumit.

4. *Errors*

Sistem tidak membuat banyak kesalahan saat digunakan, dan jika pengguna mengalami kesalahan saat menggunakannya, mereka dapat dengan mudah menyelesaikannya.

5. *Satisfaction*

Pengguna merasa nyaman saat menggunakan sistem tersebut, tidak ada kendala yang membuatnya menjadi malas, dan tampilan sistem terlihat baik sehingga pengguna tidak merasa bosan saat menggunakannya. Pengalaman pengguna yang baik.

2.2.25 *System Usability Scale (SUS)*

Metode *System Usability Scale (SUS)* adalah salah satu alat pengujian kegunaan yang paling terkenal. John Brooke mengembangkan *System Usability Scale (SUS)* pada tahun 1986. *System Usability Scale (SUS)* adalah alat pengukuran kegunaan yang handal, populer, efektif, dan ekonomis yang terdiri dari 10 pertanyaan dengan 5 jawaban dalam skala Likert yang diterapkan dalam kuesioner, dengan penilaian kuesioner berkisar antara 1 hingga 5. Berikut adalah pertanyaan-pertanyaan dalam kuesioner tersebut.

1. Saya berpikir akan menggunakan sistem ini lagi.
2. Saya merasa sistem ini rumit untuk digunakan.
3. Saya merasa sistem ini mudah digunakan.
4. Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
5. Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.
6. Saya merasa ada banyak hal yang tidak konsisten pada sistem ini.
7. Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8. Saya merasa sistem ini membingungkan.
9. Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10. Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

Setiap pernyataan akan dijawab menggunakan skala 1 hingga 5, di mana 1 berarti "Sangat Tidak Setuju" dan 5 berarti "Sangat Setuju". Jika responden merasa ragu, mereka dapat memilih skala 3. Setelah semua responden telah mengisi kuesioner, skor akan dihitung untuk setiap responden [72].

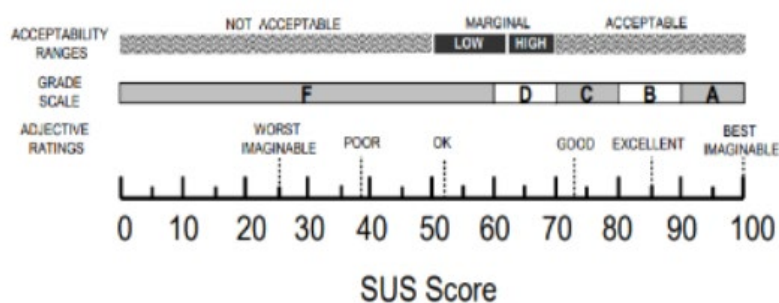
Setiap pertanyaan memiliki skor kontribusi yang berkisar antara 0 hingga 4. Untuk pertanyaan dengan nomor ganjil, skor kontribusi dihitung dengan mengurangi 1 dari nilai pertanyaan tersebut. Sedangkan untuk pertanyaan dengan nomor genap, skor kontribusi dihitung dengan

mengurangi nilai pertanyaan tersebut dari 5. Adapun rumus untuk menghitung skor pengujian SUS adalah sebagai berikut.

$$x = \left(\sum_{n=1}^5 (U_{2n-1} - 1) + (5 - U_{2n}) \right) \times 2,5 \quad (2.4)$$

$$\bar{x} = \frac{\sum x}{\text{jumlah responden}} \quad (2.5)$$

Untuk menyimpulkan bahwa suatu sistem dianggap baik, skor keseluruhan SUS harus melebihi 68. Hasil perhitungan skor rata-rata (2.5) dapat digunakan untuk menentukan tingkat kepuasan pengguna berdasarkan tiga kategori, yaitu *Adjective Ratings*, *Grade Scale*, dan *Acceptable Ranges*. Kategori-kategori dalam SUS Skor dapat digunakan sebagai acuan dalam mengevaluasi tingkat kepuasan pengguna berdasar gambar 2.4 [73].



Gambar 2.4 Kategori Skor SUS [73]

Jeff Sauro menginterpretasikan nilai SUS dengan menggunakan peringkat prosentase (*percentile ranks*) dan kelas huruf (*letter grades*), yang dimulai dengan A dan berakhir pada F, yang masing-masing menunjukkan kelas terbaik dan terburuk [69]. Untuk *letter grade*, ketentuan berikut harus dipenuhi tertampil pada tabel 2.7.

Tabel 2.7 Letter Grade SUS

| Nilai | Grade |
|-------------------------------|-------|
| nilai $\geq 80,3$ | A |
| $74 \leq \text{nilai} < 80,3$ | B |
| $68 \leq \text{nilai} < 74$ | C |

| Nilai | Grade |
|-----------------------------|-------|
| $51 \leq \text{nilai} < 68$ | D |
| nilai < 51 | F |

Sumber: Alvian Kosim M, Restu Aji S, Darwis M (2022) [69]

Penulis menggunakan Teori Nielsen dan *System Usability Scale* (SUS) untuk penelitian sebelumnya. Peneliti sebelumnya mengelompokkan pertanyaan-pertanyaan dalam SUS ke dalam kategori *Learnbility*, *Efficiency*, *Memorability*, *Errors*, dan *Satisfaction* berdasarkan teori Nielsen yang digunakan dalam penelitian sebelumnya. Pembagian kategorinya tersaji dalam tabel 2.8 berikut.

Tabel 2.8 Pengelompokan SUS ke dalam Kategori Nielsen

| No | Pertanyaan | Kategori |
|----|---|---------------------|
| 1 | Saya berpikir akan menggunakan sistem ini lagi. | <i>Satisfaction</i> |
| 2 | Saya merasa sistem ini rumit untuk digunakan. | <i>Learnbility</i> |
| 3 | Saya merasa sistem ini mudah digunakan. | <i>Learnbility</i> |
| 4 | Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini. | <i>Error</i> |
| 5 | Saya merasa fitur-fitur sistem ini berjalan dengan semestinya. | <i>Efficiency</i> |
| 6 | Saya merasa ada banyak hal yang tidak konsisten pada sistem ini. | <i>Error</i> |
| 7 | Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat. | <i>Memorability</i> |
| 8 | Saya merasa sistem ini membingungkan. | <i>Efficiency</i> |
| 9 | Saya merasa tidak ada hambatan dalam menggunakan sistem ini. | <i>Satisfaction</i> |
| 10 | Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini. | <i>Memorability</i> |

Sumber: Yogananti A, Pratama B, Akrom A (2022) [70]

2.2.26 Skala Likert

Likert scale adalah skala yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang

fenomena sosial. Seseorang dapat menunjukkan sikapnya dengan menyatakan setuju atau tidak setuju sesuai topik, objek, atau peristiwa tertentu. Setelah itu, deskriptor digunakan sebagai dasar untuk membuat instrumen yang terdiri pernyataan atau pertanyaan yang harus dijawab oleh responden [74]. Tabel berikut menunjukkan jenjang skala likert.

Tabel 2.9 Tabel Skala Likert

| Pernyataan | Nilai Pernyataan | |
|---------------------------|------------------|---------|
| | Positif | Negatif |
| Sangat Setuju (SS) | 5 | 1 |
| Setuju (S) | 4 | 2 |
| Ragu-Ragu/Netral (N) | 3 | 3 |
| Tidak Setuju (TS) | 2 | 4 |
| Sangat Tidak Setuju (STS) | 1 | 5 |

Sumber: Wahyuddin R, Sucipto A, Susanto T (2022) [74]

Adapun rumus yang digunakan untuk menghitung skor skala likert serta indeks adalah sebagai berikut [75].

Rumus skor Skala Likert:

$$skor = \sum_{i=1}^n (T \times P_i) \quad (2.6)$$

Rumus *Index*:

$$index = \frac{skor}{Y} \times 100 \quad (2.7)$$

Keterangan:

$$Y = \text{skor tertinggi} \times \text{jumlah responden} \quad (2.8)$$

$$X = \text{skor terendah} \times \text{jumlah responden} \quad (2.9)$$

T = Total jumlah responden

P_i = Pilihan angka skor likert

n = Banyaknya data