BAB II
TINJAUAN PUSTAKA

2.1 Rancang Bangun

Rancang bangun merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengatruran dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, dengan maksud lain rancang bangun sebuah kegiatan menerjemah hasil analisa ke dalam bentuk paket perangkat lunak kemudian akan menciptakan sistem atau memperbaiki sistem yang ada (Gunawan et al., 2021). Rancang bangun juga merupakan kegiatan menerjemahkan hasil analisis ke dalam bentuk perangkat lunak kemudian menciptakan sistem tersebut atau mengembangkan sistem yang sudah ada (Lutfiani et al., 2020). Menurut penjelasan diatas dapat disimpulkan bahwa rancang bangun adalah membuat suatu objek dari beberapa elemen yang terpisah dengan metode tertentu kedalam suatu kesatuan yang utuh dan berfungsi agar dapat diimplementasikan dan berguna dan memiliki nilai bagi penggunanya. Rancang bangun merupakan kegiatan menerjemahkan hasil analisa kedalam bentuk paket aplikasi lalu membangun sistem lama ataupun memperbaiki sistem yang telah ada.

2.2 HRIS (Human Resource Information System)

HRIS dapat didefinisikan sebagai suatu sistem terintegrasi yang digunakan untuk mengumpulkan, menyimpan dan menganalisa informasi tentang sumber daya manusia dalam sebuah organisasi yang terdiri dari *database* , komputer aplikasi, perangkat keras dan perangkat lunak yang diperlukan untuk mengumpulkan, merekam, menyimpan, mengelola, memberikan, menyajikan dan memanipulasi data untuk fungsi – fungsi sumber daya manusia. Menurut Jonni yang dikutip oleh (Ma'ruf et al., 2020), HRIS (*Human Resource Information System*) adalah sebuah sistem informasi yang menangani pengelola SDM pada sebuah perusahaan.

2.3 Aplikasi Mobile

Dalam penelitian (Afit Muhammad Lukman, 2019), menurut Turban *mobile* application atau biasa juga disebut dengan *mobile* apps, yaituistilah yang digunakan untuk mendeskripsikanaplikasi internet yang berjalan pada*smartphone* atau piranti *mobile* lainnya.Aplikasi *mobile* juga dapat membantupenggunanya untuk terkoneksi denganlayanan internet yang biasanya diakses pada PC (Personal Computer) menjadi dipermudah dengan piranti yang lebih nyaman dibawakemanapun berada (portable). Disimpulkan bahwa aplikasi *mobile* adalah program aplikasi yang digunakan dengan sumber daya berbasis web yang menyediakan akses ke beragam informasi yang relevan. Aplikasi ini juga dapat diakses akses melalui perangkat telepon seluler, *smartphone*, nirkabel, pager dan perangkat sejenisnya.

2.4 Website

Website dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan. dengan jaringanjaringan halaman (hyperlink). Bersifat statis apabila isi informasi *website* tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik *website*. Bersifat dinamis apabila isi informasi *website* selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna *website* (Hayaty & Kusmawan, 2020).

Kemudian menurut (Muhyidin, 2020) menyatakan "*Website* merupakan suatu layanan sajian informasi yang menggunakan konsep hyperlink, yang memudahkan surfer (sebutan bagi pemakai komputer yang melakukan penelusuran informasi di internet)". Selanjutnya menurut (Doni & Rahman, 2020) *Website* adalah sekumpulan dokumen yang berada pada *server* dan dapat dilihat oleh *user* dengan menggunakan browser. Dokumen itu bisa terdiri dari beberapa halaman. Tiap-tiap halamannya memberi informasi atau interaksi yang beraneka ragam. Informasi atau interaksi yang beraneka ragam. Informasi dan interaksi itu bisa berupa tulisan,

gambar atau bahkan dapat ditampilkan dalam bentuk video, animasi, suara, dan lain-lain (Agarina et al., 2020).

2.5 Front-end

Dalam penelitian (Arhandi n.d, 2018), menurut Goldbot *front end* adalah segala sesuatu yang mengubungkan antara *user* dengan sistem *back end*. Biasanya merupakan sebuah *user interface* dimana *user* akan berinteraksi dengan sistem. Pekerjaan yang sering muncul sebagai seorang pengembang *front end* adalah desainer *user interface* dan desainer *user experience*. Seorang pengembang *front end* tidak akan membuat program atau aplikasi yang berjalan di *logic* bisnis tapi fokusnya akan lebih banyak ke antarmuka, desain grafis (*User Interface Designer*) dan bagaimana membuat desain yang nyaman digunakan oleh *user* (*User Experience Designer*). Bahasa pemrograman yang biasanya digunakan dalam pengembangan *front end* adalah html dan html.

2.6 Back-end

Back end adalah bagian belakang layar dari sebuah *website*. Setiap *back end* dari situs web meliputi dari tiga bagian: *server*, *database* , dan aplikasi. Pengembang *back end* menulis kode yang memungkinkan ketiga komponen ini berinteraksi dan bekerja sama untuk melakukan fungsi dan menyampaikan informasi kepada pengguna akhir (Salim & Ishaq, 2021). Disebut dengan *back end* atau *server* side pada dasarnya adalah dimana aplikasi atau proses sistem yang berjalan di *back end* adalah data yang ditambahkan, diubah atau dihapus. *Back end* menangani segala sesuatu yang biasanya tidak terlihat atau berinteraksi langsung dengan pengguna, seperti *database* dan *server*. Pengembang *back end* biasanya adalah programmer atau pengembang yang pekerjaannya berfokus pada keamanan, desain sistem, dan manajemen data di dalam sistem. Mengembangkan sistem dan aplikasi dinamis dengan data yang terus berubah membutuhkan pengembang *back end* (Arhandi, n.d., 2019). Berikut beberapa alat bantu dalam mengembangkan *Back end* yang digunakan dalam perancangan *REST API* :

2.6.1 *Application Programming Interface* (*API*)

API adalah antarmuka yang digunakan oleh untuk mengakses aplikasi atau layanan dari program. *API* memungkinkan pengembang untuk menggunakan fungsionalitas yang ada dari aplikasi lain, jadi tidak perlu membuatnya ulang dari awal. . Dalam konteks situs web, *API* memanggil fungsi melalui *Hypertext Transfer Protocol* (HTTP) dan menerima respons dalam bentuk *Extensible Markup Language* (XML) atau *JavaScript Object Notation* (JSON).

Tujuan penggunaan *API* adalah untuk berbagi data antar aplikasi yang berbeda, Tujuan penggunaan *API* lain adalah untuk mempercepat pengembangan aplikasi dengan menyediakan fungsionalitas yang terpisah sehingga pengembang tidak perlu mendesain Fitur yang sama lagi. OS-level *API* membantu aplikasi berkomunikasi dengan dan di antara lapisan dasar dengan mengikuti serangkaian protokol dan spesifikasi khusus (Hasanuddin & Budi Hartono, 2022).

2.6.2 *Representional State Transfer* (REST)

REST adalah seperangkat prinsip arsitektur yang mengirimkan data melalui antarmuka yang distandarisasi seperti HTTP. REST berperilaku seperti aplikasi web biasa. Klien dapat mengirim permintaan ke *server* melalui protokol HTTP, kemudian *server* merespons kembali ke klien. REST dikembangkan oleh Roy Fielding, salah satu pendiri proyek Apache HTTP *Server*. Dalam REST sendiri, *server* REST menyediakan sumber daya (sumber daya/data) dan klien REST mengakses dan mengekspos sumber daya untuk nanti. Setiap sumber daya diidentifikasi dengan URI (Pengidentifikasi Sumber Daya Universal atau pengidentifikasi global. daya disajikan sebagai teks, JSON, atau format XML (Hasanuddin & Budi Hartono, 2022).

2.6.3 *JavaScript Object Notation* (JSON)

JSON adalah format berbagi data, seperti namanya JSON berasal dari bahasa pemrograman Javascript, tetapi formatnya tersedia dalam banyak bahasa lain, termasuk Python, Ruby, PHP, dan Java. JSON sering diucapkan seperti nama

"Jason". JSON menggunakan ekstensi .json karena berdiri sendiri. Ketika didefinisikan dalam format file lain (seperti dalam .html), mungkin muncul dalam tanda kutip sebagai string JSON atau diteruskan dalam variabel. Format ini mudah untuk ditransfer antara *server* web dan klien atau browser (Hasanuddin & Budi Hartono, 2022).

2.6.3 XAMPP

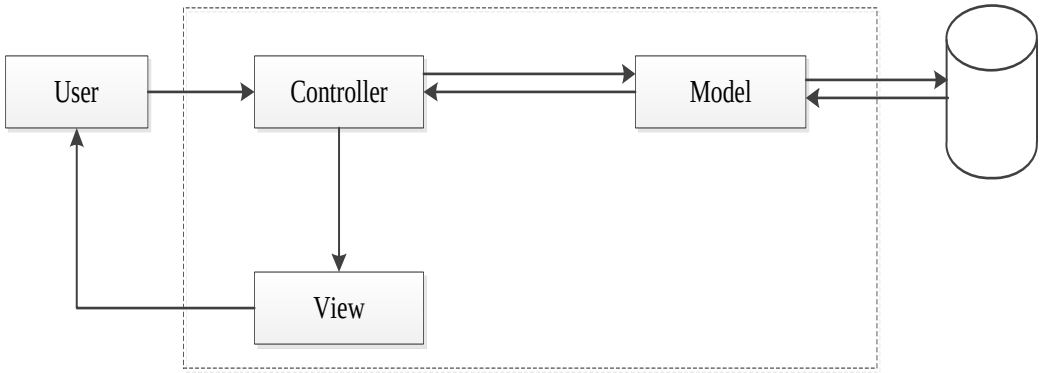
XAMPP merupakan gabungan dari beberapa software maka kali ini kami akan menjelaskan satu – persatu fungsi atau kegunaannya, tentunya berhubungan dengan dunia developer web. Sebagai software yang cross platform tentunya dimaksudkan agar semua orang dapat menggunakannya. XAMPP juga menunjang beberapa Bahasa pemrograman khusus dalam *website* yakni PHP, MySQL dan Perl. PHP merupakan suatu Bahasa yang sering digunakan oleh programmer khusus *Back end* karena memang lebih mengutamakan logika dibanding tampilan, beda halnya dengan HTML atau CSS.

Oleh karena itu script PHP tidak akan terlihat dalam tampilan *website*. MySQL merupakan suatu software yang digunakan untuk mengelola SQL (Structured Query Language). Bahasa ini biasa digunakan untuk keperluan *database* khusus pada *website*. Pengelolaan *database* yang dimaksudkan adalah untuk menambah data, mengubah, menghapus dan lain – lain. Keberadaan MySQL juga biasanya identic dengan Bahasa PHP. Sedangkan phpMyAdmin merupakan suatu software khusus untuk mengelola administrasi MySQL. Jika pada Htdocs menyimpan file – file tampilan web anda maka di phpMyAdmin ini terdapat semua *database* yang anda gunakan untuk keperluan *website* (Riyadi, 2022).

2.7 Laravel

Laravel adalah web framework PHP yang bersifat open source dan gratis yang dibuat oleh Taylor Otweel yang dapat digunakan dalam mengembangkan webapplications dengan menggunakan aristektur MVC (*Model-View-Controller*). Framework Laravel mudah dipahami dan memudahkan dalam hal authentication, routing, sessionmanager, caching, dan beberapa kegunaan lain dari komponen –

komponen di Laravel. Laravel juga menyediakan fitur seperti *database* migration dan integrasi unit testing support yang memudahkan developeruntuk membangun aplikasi yang kompleks (Somya and Nathanael, 2019).



Gambar 2. 1 Arsitektur *Model-View-Controller* (MVC)

Berdasarkan arsitektur tersebut diketahui konsep *Model-View-Controller* yaitu:

1. *Model*

   Model merupakan suatu fungsi yang digunakan mengelola *database* pada sistem untuk menangani struktur data dari *controller*.
2. *View*

   View merupakan bagian untuk mengelola tampilan dari *website* atau dapat disebut sebagai *user interfac* yang diatur bagian *controller.*
3. *Controller*

   Controller merupakan kunci dalam konsep MVC dengan fungsi untuk menghubungkan *model* dengan *view.*

2.8 Flutter

Menurut Anon dalam penelitian (Tjandra & Chandra, 2020), Flutter *adalah Mobile App SDK* (*Software Development Kit*) untuk membuat aplikasi Android dan iOS dari satu codebase dengan performa tinggi. Tujuannya adalah memungkinkan pengembang untuk menghadirkan aplikasi berkinerja tinggi yang terasa alami pada platform yang berbeda , Flutter dibuat menggunakan bahasa C,C++, Skia dan Dart, Flutter terdiri dari dua bagian penting yaitu SDK (Perangkat Pengembangan

Perangkat Lunak): Kumpulan alat yang akan membantu mengembangkan aplikasi. Ini termasuk alat untuk mengkompilasi kode ke dalam kode native (kode untuk iOS dan Android) dan Framework (Perpustakaan antar muka pengguna berdasarkan widget): Kumpulan elemen antar muka pengguna (tombol, *input* teks, slider, dan sebagainya) yang dapat di personalisasi untuk kebutuhan aplikasi.

Widget adalah blok bangunan dasar dari antarmuka pengguna aplikasi Flutter. Setiap widget adalah deklarasi bagian dari antarmuka pengguna yang tidak dapat diubah. Tidak seperti kerangka kerja lain yang memisahkan tampilan, pengontrol tampilan, tata letak, dan properti lainnya, Flutter memiliki model objek yang konsisten dan terpadu: widget.Widget dapat menentukan beberapa hal diantaranya elemen struktural (seperti tombol atau menu), elemen gaya (seperti font atau skema warna), aspek tata letak (seperti bantalan), Widget membentuk hierarki berdasarkan komposisi. Setiap widget bersarang di dalam, dan mewarisi properti dari, induknya. Bahasa yang digunakan dalam pemograman framework flutter adalah Dart, sebagai dijelaskan berikut:

2.8.1 Dart

Dart merupakan bahasa pemrograman general-purpose yang dirancang oleh Lars Bak dan Kasper Lund. Bahasa pemrograman ini dikembangkan sebagai bahasa pemrograman aplikasi yang dapat dengan mudah untuk dipelajari dan disebarkan. Bahasa pemrograman besutan Google ini dapat digunakan untuk mengembangkan berbagai macam platform termasuk di dalamnya adalah web, aplikasi *mobile*, *server*, dan perangkat yang mengusung teknologi Internet of Things. Bahasa pemrograman tersebut dapat digunakan untuk mengembangkan aplikasi untuk dijalankan pada berbagai macam peramban modern.

Dart juga dapat digunakan untuk mengembangkan aplikasi dari codebase tunggal menjadi aplikasi Android maupun iOS. Dart dapat digunakan secara bebas oleh para developer, karena bahasa ini dirilis secara open-source oleh Google di bawah lisensi BSD. Bahasa pemrograman Dart merupakan bahasa pemrograman berbasis class dan 11 berorientasi terhadap obyek dengan menggunakan sitaks bahasa pemrograman C.

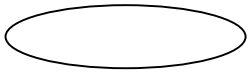
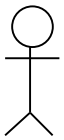
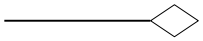
2.9 Alat Pengembang Sistem (Unified Modelling Language)

Alat pengembang sistem merupakan konsep desain yang digunakan untuk menggambarkan sistem dengan menggunakan diagram. Penyesuaian alat yang digunakan harus sesuai dengan metode pengembangan yang dilakukan salah satunya adalah penerapan *Unified Modelling Language*. Menurut (A.S. & Shalahuddin, 2019) UML *(unified Modelling Language)* adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML *(Unified Modelling Language).*

2.9.1 *Use Case Diagram*

Menurut (A.S. and Shalahuddin, 2019) *Use Case* adalah *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada tabel 2.1 berikut ini:

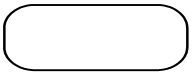
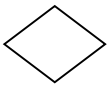
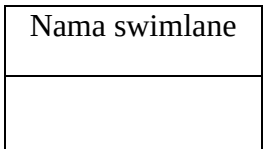
Tabel 2. 1 Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.		Usecase Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal *frase* nama *Use Case.*
2.		Aktor Aktor seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi/association merupakan komunikasi antara aktor dan *Use Case* yang berpartisipasi pada *Use Case* atau *Use Case* memiliki interaksi dengan aktor.
4.		Generalisasi *(generalization)* merupakan hubungan (umum – khusus) antara dua buah *Use Case* dimana fungsi yang satu adalah fungsi yang lebih umum
5.	<< Include >> 	Include berarti *Use Case* yang ditambahkan akan dipanggil saat *Use Case* tambahan dijalankan.
6.	<<Extend>> 	Ekstensi *(extend)* merupakan *Use Case* tambahan ke sebuah *Use Case* yang ditambahkan dapat berdiri sendiri walau tanpa *Use Case* tambahan itu.

2.9.2 *Activity Diagram*

Menurut (A.S. and Shalahuddin, 2019) *Activity* diagram adalah *Activity* Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Activity Diagram* dapat dilihat pada tabel 2.2 berikut ini :

Tabel 2. 2 Simbol *Activity* Diagram

No.	Simbol	Keterangan
1.		Status awal aktivitas sitem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan *(Decision)* merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan *(Join)* merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.9.3 *Class Diagram*

Menurut (A.S. and Shalahuddin, 2019) *Class Diagram* adalah *Class Diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada tabel 2.3 berikut ini :

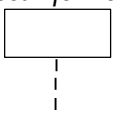
Tabel 2. 3 Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.	Nama_kelas +Atribute +Operasi	Kelas pada struktur sistem**.**
2.	Antar Muka/*Interface* Nama_*Interface*	Sama dengan konsep *interface* dalam pemrograman berorientasi objek.
3.	Asosiasi / Asociation	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	Asosiasi Berarah / Directed Association	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Agregasi / aggregation	Relasi antar kelas dengan maksna semua bagian *(whole-part)*

2.9.4 *Sequence Diagram*

Merupakan diagram rangkaian menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada eksekusi sebuah use-case atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima diantara objek dan dalam sekuensi. Berikut penjelesan dari simbol *Sequence Diagram* pada tabel 2. 4 :

Tabel 2. 4 Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	*Object lifeline* 	Penggambarkan panjang kehidupan suatu objek selama scenario sedang di buat contohnya
2.	*Activation* 	Dimana proses sedang dilakukan oleh object atau class untuk memenuhi pesan atau perintah
3.	*Message*	Sebuah anak panah yang mengindikasikan pesan diantara objek. Dan objek dapat mengirimkan pesan ke dirinya sendiri

2.10 Rapid Application Development (RAD)

Rapid Application Development (RAD) merupakan metode pengembangan sistem informasi dengan waktu singkat, sehingga dinilai tepat digunakan dalam pengembangan sistem. RAD menggunakan metode iteratif (berulang) dalam mengembangkan sistem dimana *working model* (model bekerja) sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan (*requirement*) pengguna dan selanjutnya disingkirkan. Dibandingkan dengan metode lain seperti Waterfall, keunggulan dari metode RAD adalah kecepatan, ketepatan dan biaya yang relatif lebih rendah.

Namun bukan berarti metode pengembangan lain tidak lebih baik dari metode RAD (Widiyanto, 2018). *Rapi*d *Application Development* (RAD), metode ini digunakan karena metode RAD dapat mempersingkat waktu dalam pembangunan sistem informasi dari pada metode tradisional lainnya (Aini, Wicaksono, dan Arwani 2019), juga dengan menggunakan metode RAD sistem yang dikembangkan lebih cepat tersampaikan ke *User* dan tidak diperlukan dalam menunggu fitur yang lain terselesaikan (Widiyanto, 2018). Terdapat 3 tahapan dari metode RAD seperti pada gambar 2. 2 :

Gambar 2. 2 *Rapid Application Development* (RAD)

Sumber: (Kendall, 2010)

a. Proses Perencanaan Syarat-Syarat/Kebutuhan

Tahap ini merupakan hal yang paling penting karena diperlukan adanya keterlibatan antara kedua belah pihak yaitu calon pengguna dan analis. Calon pengguna dan Analis melakukan pertemuan untuk mendiskusikan kebutuhan sistem dan kebutuhan informasi untuk mencapai tujuan yang diinginkan.

b. Proses Desain Sistem/*Workshop* Desain

Proses desain sistem dilakukan dengan melibatkan keaktifan calon pengguna dalam melakukan konfirmasi dan perbaikan terhadap kesesuaian desain antara calon pengguna dan analis. Seorang calon pengguna dapat memberikan masukan dan komentar apabila terdapat ketidaksesuaian pada desain serta melakukan perancangan sistem yang mengacu pada kebutuhan sistem yang sudah dibuat pada tahap sebelumnya. Keluaran dari tahap ini adalah spesifikasi sistem yang meliputi organisasi sistem secara umum, struktur data, dan lain-lain.

c. Proses Implementasi

Proses implementasi adalah proses pengembangan aplikasi dengan mengimplementasikan apa saja yang sudah disepakati berdasarkan kebutuhan dan desain sistem. Kemudian pengujian dilakukan untuk memeriksa dan

menguji apakah kebutuhan pada aplikasi sudah sesuai atau belum. Apabila sudah sesuai maka aplikasi dapat digunakansebagaimana mestinya. Apabila masih terdapat kekurangan, maka dilakukan evaluasi dan pebaikan sampai aplikasi benar-benar berjalan sesuai kebutuhan.

2.11 Black Box Testing

Menurut (Agus Rahardi and Muhammad Fauzan Azima, 2019) mengatakan *black box testing* dilakukan dengan memastikan bahwa semua bagian telah diuji dan memastikan *input* yang diberikan memberikan hasil *output* yang sesuai dengan harapan. Pengujian sistem adalah proses untuk mengecek apakah suatu perangkat lunak yang dihasilkan sudah dapat dijalankan sesuai standar atau belum.

Pengujian sistem dapat menggunakan metode *black box testing*, yang berfokus pada pengujian persyaratan fungsional perangkat lunak, karena untuk mendapatkan serangkaian kondisi *input* yang sesuai dengan persyaratan fungsional suatu program. Pengujian sistem dapat dilakukan dengan pengecekan proses sebagai berikut :

1. Pengecekan *input*, meliputi kelengkapan item-item *input*, kemudahan pengoperasian, kemudahan manipulasi data, dan pengendalian kesalahan.
2. Pengecekan proses, dilakukan dengan pengecekan *output* program.
3. Pengecekan *output*, meliputi pengecekan terhadap format dan bentukbentuk laporan.

2.12 Tinjauan Penelitian Terdahulu

Teori atau hasil penelitian sebelumnya diperlukan dan dapat digunakan sebagai data untuk mendukung penelitian ini. Perbedaan dari beberapa penelitian terkait penilaian ini dapat dilihat pada tabel 2.5 dibawah ini :

Tabel 2. 5 Tinjauan Penelitian Terdahulu

Tahun	Nama	Judul		Hasil
2016	Arhandi	Pengembangan Sistem Informasi Perijinan Tenaga Kesehatan Dengan Menggunakan Metode *Back end* Dan Front End	1.	kecepatan waktu respon aplikasi yang signifikan dengan waktu respon 114 mili detik untuk permintaan *login* dan 47 mili detik untuk permintaan *get*.
2019	Roedyanto	Aplikasi *Mobile* Untuk Sistem Presensi Dosen dan Karyawan Universitas Kristen Petra	1.	Aplikasi dapat digunakan dengan mudah (user friendly) oleh dosen dan karyawan tetap Universitas Kristen Petra dengan rata-rata nilai 3.7 pada hasil kuisioner yang dibagikan.
			2.	Aplikasi dapat melakukan kegiatan presensi seperti pengecekkan presensi kehadiran, pengajuan presensi manual, pengajuan izin datang terlambat atau pulang cepat, dan pengajuan pencatatan penugasan.
			3.	Aplikasi dapat menyimpan bukti penugasan berupa file dalam bentuk jpeg
2020	Ma'ruf	Penerapan Framework Laravel Pada Aplikasi *HRIS* (*Human Resource Information System*)	1.	Aplikasi *HRIS* (*Human Resource Information System*) membuat proses perhitungan gaji karyawan menjadi lebih cepat karena aplikasi dapat menghitung gaji karyawan secara keseluruhan.

			2.	Pengajuan cuti yang dibuat karyawan menjadi lebih mudah, karena karyawan tinggal mengajukan cuti pada sistem dan atasan akan melakukan persetujuan jika memungkinkan.
			3.	Aplikasi *HRIS* (*Human Resource Information System*) dapat mengolah data pegawai dengan baik karena setiap pegawai memiliki hak akses untuk merubah dan melihat data pegawai tersebut. Saran