

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Menurut penelitian dari Iski Zaliman, dkk. Pada tahun 2022 dengan judul “ Sistem Informasi Buku Induk Siswa Pada SMA Negeri 1 Kelekar ” yang dilakukan di SMA Negeri 1 Kelekar, penelitian ini menjelaskan bahwa perlu dibutuhkannya suatu sistem informasi berbasis *web* untuk mengelola data siswa. Permasalahan yang menjadi dibentuknya gagasan tersebut, karena informasi hasil pengolahan data tidak berkualitas baik, prosesnya masih ditulis tangan (manual), lambat bahkan membuat kesulitan jika buku tersebut hilang atau terkena bencana alam, dan juga laporan tentang identitas siswa sering terlambat atau tidak tepat waktu dikarenakan proses

Dengan tujuan dibentuknya Sistem Informasi Buku Induk Siswa Berbasis *Web* diharapkan dapat lebih mudah diakses, memberikan manfaat bagi pihak sekolah khususnya bagian Tenaga Administrasi Sekolah agar bisa menggunakan sistem informasi tersebut sesuai dengan kebutuhan yang akan datang. Segala fitur menu-menu yang tersedia dalam sistem ini untuk memudahkan proses pencatatan dan pelaporan buku induk siswa, agar bisa dapat dipercepat dalam pencarian data siswa pada SMA Negeri 1 Kelekar dalam penggunaan proses buku induk siswa yang ada tersedia dapat berjalan dengan baik dan lancar.

2.2 Landasan Teori

2.2.1 Buku Induk Siswa

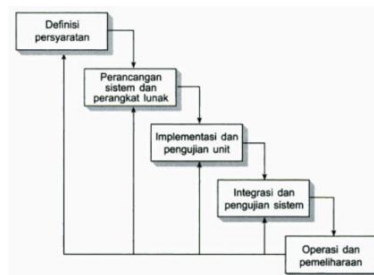


Gambar 2.1 Buku Induk Siswa

Buku Induk menurut Ulyani dalam (ULYANI, 2019), Buku induk Siswa adalah kumpulan daftar nama siswa sepanjang masa dari sekolah tersebut. Setiap siswa harus dicatat dalam buku besar yang biasa disebut buku induk siswa atau buku pokok. Catatan dalam buku induk siswa harus lengkap yang meliputi data dan identitas siswa. Data-data tersebut diambil dari formulir pada saat pendaftaran Penerimaan Peserta Didik Baru (PPDB).

Selain dari identitas siswa yang meliputi nama orang tua dll., dalam buku induk juga berisi nomor induk siswa, nomor induk siswa nasional, nomor kode sekolah, keterangan orangtua/wali siswa, dan prestasi belajar siswa (daftar nilai raport) dari tahun ke tahun selama siswa tersebut belajar di sekolah. Catatan dalam buku induk harus jelas, dan ini merupakan tanggung jawab Kepala Sekolah yang penggarapannya bisa diserahkan kepada pegawai sekolah.

2.2.2 Metode *Waterfall*



Gambar 2.2 Metode *Waterfall*

Metode *Waterfall* menurut Sanubari dkk (2020) adalah suatu metode pengembangan *software* dengan cara pendekatan sistematis secara berurutan.

Metode ini melibatkan tahapan-tahapan seperti analisis kebutuhan, perancangan, implementasi, pengujian, pengiriman, dan pemeliharaan, yang dilakukan secara berurutan dan linear. Meskipun cocok untuk proyek-proyek dengan persyaratan yang stabil dan jelas, metode ini kurang fleksibel dalam menghadapi perubahan kebutuhan yang mungkin terjadi selama proses pengembangan.

2.2.3 *Web Server*



Web Server

Gambar 2.3 *Icon Web Server*

Menurut Abdulloh (2018:4) “*Web server* merupakan perangkat lunak yang berfungsi untuk menerima permintaan (*request*) melalui

protocol HTTP atau *HTTPS* dari *client* kemudian mengirimkan kembali dalam bentuk halaman-halaman web”.

Web server juga dapat melakukan berbagai tugas lainnya, seperti pemrosesan formulir, otentikasi pengguna, dan mengelola akses ke berbagai bagian dari situs *web*. Contoh *web server* yang populer termasuk *Apache HTTP Server*, *Nginx*, dan *Microsoft Internet Information Services (IIS)*.

2.2.4 HTML (*HyperText Markup Language*)



Gambar 2.4 HTML

Menurut Abdulloh (2018:7) “*HTML* merupakan singkatan dari *HyperText Markup Language* yaitu bahasa standar *web* yang dikelola penggunaannya oleh *W3C (World Wide Web Consortium)* berupa tag-tag yang menyusun setiap elemen dari *website*. membuat halaman *website* yang dapat diakses atau ditampilkan menggunakan *Web Browser*”.

Dengan *HTML*, dapat menentukan bagaimana konten tersebut akan ditampilkan dan diatur di dalam *browser web*. *HTML* menggunakan tag-tag (elemen) untuk menandai awal dan akhir dari suatu elemen, serta memberikan instruksi kepada *browser* tentang bagaimana menampilkan elemen tersebut. Misalnya, tag `<p>`

digunakan untuk menandai awal dari sebuah paragraf, sedangkan tag `</p>` digunakan untuk menandai akhir dari paragraf tersebut.

2.2.5 CSS (*Cascading Style Sheets*)



Gambar 2.5 CSS

Menurut Yudhanto & Prasetyo (2019:6) “CSS adalah *Cascading Stylesheet*, yaitu bahasa yang digunakan untuk *HTML* agar menjadi lebih bagus dan efektif dalam tampilan.

Dengan menggunakan CSS, dapat memisahkan antara struktur konten (yang ditentukan dengan *HTML*) dan tampilan atau gaya dari konten, memungkinkan pengembang *web* untuk membuat situs *web* yang lebih dinamis, konsisten, dan mudah dikelola. CSS juga mendukung konsep "*cascading*", di mana gaya dapat diwariskan dari satu elemen ke elemen lain dalam dokumen, memungkinkan untuk membuat perubahan tampilan secara konsisten dengan mengedit hanya beberapa aturan gaya.

2.2.6 Javascript



Gambar 2.6 Javascript

Bahasa pemrograman yang bersifat *client side* yang pemrosesnya dilakukan oleh *client*, sering digunakan pada *web browser* untuk menciptakan halaman web yang menarik.

Menurut Abdulloh (2018:193) “*JavaScript* merupakan bahasa pemrograman *web* yang pemrosesnya dilakukan di sisi *client*”.

Dengan *JavaScript*, pengembang *web* dapat menambahkan fungsi interaktif seperti validasi formulir, efek visual, animasi, dan banyak lagi, yang meningkatkan pengalaman pengguna dalam menjelajahi halaman *web*. *JavaScript* juga memungkinkan pengembang untuk berinteraksi dengan *API (Application Programming Interface)* untuk mengakses dan mengubah konten halaman *web* secara dinamis. Dengan demikian, *JavaScript* merupakan salah satu komponen penting dalam pengembangan *web* modern yang memungkinkan pembuatan aplikasi *web* yang lebih dinamis dan responsif.

2.2.7 JQuery



Gambar 2.7 JQuery

Menurut Abdulloh (2018:233) “*JQuery* merupakan salah satu dari sekian banyak *JavaScript library*, yaitu kumpulan fungsi *JavaScript* yang siap pakai, sehingga mempermudah dan mempercepat dalam membuat kode *JavaScript*”.

jQuery menyediakan berbagai fungsi dan fitur yang dapat digunakan untuk memanipulasi elemen *HTML*, menangani event, membuat efek animasi, dan berinteraksi dengan server menggunakan *AJAX* dengan lebih mudah dan efisien daripada menggunakan JavaScript murni.

2.2.8 PHP (*Hypertext Preprocessor*)

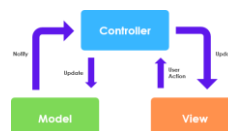


Gambar 2.8 PHP

Menurut Yudhanto & Prasetyo (2019:9) “*PHP* atau *Hypertext Preprocessor* adalah Bahasa pemrograman *script server side* yang sengaja dirancang lebih cenderung untuk membuat dan mengembangkan *web*”.

PHP dapat digunakan untuk berbagai macam tugas dalam pengembangan *web*, seperti mengolah formulir, berinteraksi dengan database, mengelola sesi pengguna, dan banyak lagi. Dengan *PHP*, pengembang *web* dapat membuat aplikasi *web* yang kompleks dan fungsional dengan mudah.

2.2.9 Konsep MVC (*Model View Controller*)



Gambar 2. 9 MVC

Menurut Yudhanto & Prasetyo (2019:8) “*Model View Controller*

atau *MVC* adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*)”.

a. *Model*

Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data, seperti memasukkan data ke basis data, pembaharuan data, dan lain-lain.

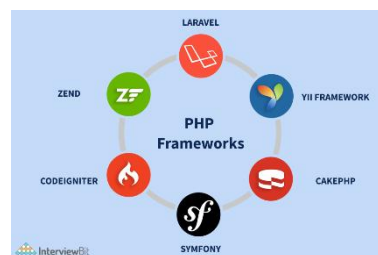
b. *View*

View adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.

c. *Controller*

Controller merupakan bagian yang menjembatani model dan *view*. *Controller* berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman *web*.

2.2.10 Framework



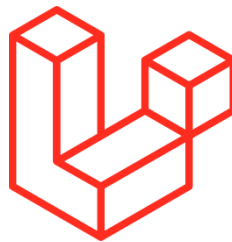
Gambar 2.10 *Framework PHP*

Menurut Yudhanto & Prasetyo (2019:10) “*Framework* adalah komponen pemrograman yang siap digunakan ulang kapan saja sehingga *programmer* tidak harus membuat *script* yang sama untuk

tugas yang sama”.

Framework menyediakan struktur dan standar yang sudah ditentukan untuk memudahkan pengembangan aplikasi. Dengan menggunakan *framework*, pengembang dapat menghemat waktu dan usaha dalam pengembangan aplikasi, karena mereka dapat memanfaatkan kembali komponen-komponen yang sudah ada dan teruji. Selain itu, *framework* juga membantu dalam meningkatkan konsistensi, keamanan, dan skalabilitas aplikasi yang dikembangkan.

2.2.11 *Laravel*



Gambar 2.11 *Laravel*

Laravel mempunyai keunggulan yang tidak dimiliki oleh *PHP Framework* lain. Menurut (Zanin & Wernke, 2019) dalam artikelnya yang berjudul “A Comparative study of *PHP frameworks performance*”, *laravel* mempunyai keunggulan dibandingkan dengan *Symfony* dan *CodeIgniter*. *Laravel* mempunyai permintaan per detik (*request per second*) tertinggi dibandingkan *Symfony* dan *CodeIgniter*. *Laravel* juga mempunyai penggunaan memori (*memory usage*) terendah dibandingkan *Symfony* dan *CodeIgniter*.

Selain dua hal itu, *laravel* juga unggul pada waktu respon (*response time*). *Laravel* mencatat *response time* terendah

dibandingkan dengan *Symfony* dan *CodeIgniter*. Namun, *laravel* mempunyai kekurangan dibandingkan kedua *framework* lain, yaitu dalam hal jumlah file (*numbers of file*).

2.2.12 *Bootstrap*



Gambar 2.12 *Bootstrap*

Menurut Abdullah (2018:261) “*Bootstrap* merupakan salah satu *Framework CSS* paling populer dari sekian banyak *framework CSS* yang ada”.

Bootstrap menyediakan kumpulan komponen dan gaya *CSS* yang siap pakai untuk mempercepat pengembangan antarmuka pengguna yang responsif dan menarik. Dengan menggunakan *Bootstrap*, pengembang *web* dapat dengan mudah membuat tata letak responsif, navigasi, formulir, tombol, dan elemen antarmuka pengguna lainnya tanpa harus menulis kode *CSS* tambahan. *Bootstrap* juga menyediakan dukungan untuk *grid system* yang fleksibel, yang memungkinkan pengembang untuk dengan mudah mengatur tata letak halaman *web* mereka untuk berbagai perangkat dan ukuran layar.

2.2.13 AdminLTE



Gambar 2.13 AdminLTE

AdminLTE adalah salah satu *template* yang sering digunakan oleh *web developer* sebagai *template backend* pada proyek yang sering dikerjakan. Jadi *admin LTE* ini adalah sebuah *dashboard Administrator* dibuat menggunakan *bootstrap* yang merupakan *framework css* yang paling banyak digunakan.

AdminLTE menyediakan berbagai komponen siap pakai untuk membangun antarmuka admin yang responsif dan menarik, termasuk grafik, tabel, formulir, dan elemen UI lainnya. Dengan menggunakan *AdminLTE*, pengembang dapat mempercepat pengembangan *backend* aplikasi *web* dengan menghemat waktu dalam desain dan pengembangan antarmuka pengguna.

2.2.14 Basis Data



Gambar 2.14 Database

Basis data kumpulan data yang tersimpan didalam komputer yang digunakan suatu program untuk memperoleh informasi dari basis data

tersebut.

Basis data umumnya terdiri dari tabel-tabel yang saling terkait, yang menyimpan informasi dalam bentuk *record* atau baris. Program-program komputer dapat menggunakan bahasa khusus, seperti *SQL* (*Structured Query Language*), untuk berinteraksi dengan basis data dan melakukan berbagai operasi, seperti menambahkan data baru, mengubah data yang sudah ada, menghapus data, dan mengambil data untuk ditampilkan atau diproses lebih lanjut. Basis data digunakan untuk menyimpan informasi yang penting bagi suatu organisasi atau aplikasi, seperti informasi pelanggan, transaksi, produk, dan lainnya.

2.2.15 MySQL



Gambar 2.15 MySQL

Menurut Nugroho (2019:133) “*MySQL* merupakan *database* yang paling digemari dikalangan *Programmer Web*, dengan alasan bahwa program ini merupakan *database* yang sangat kuat dan cukup stabil untuk digunakan sebagai media penyimpanan data”.

MySQL adalah salah satu sistem manajemen basis data relasional (*RDBMS*) yang paling populer di dunia. *MySQL* banyak digunakan dalam pengembangan aplikasi web karena kemudahan penggunaannya, kinerja yang baik, dan dukungan yang luas dari komunitas pengembang. *MySQL* mendukung bahasa *SQL* standar untuk mengelola

dan mengambil data dari basis data, serta memiliki fitur-fitur seperti transaksi, indeks, dan keamanan yang memungkinkan pengembang untuk membangun aplikasi web yang andal dan efisien.

2.2.16 *phpMyAdmin*



Gambar 2.16 *phpMyAdmin*

phpMyAdmin adalah perangkat lunak gratis yang ditulis dalam *PHP*, dimaksudkan untuk menangani administrasi *MySQL* melalui *Web*. *phpMyAdmin* mendukung berbagai operasi di *MySQL* dan *MariaDB*. Operasi yang sering digunakan (mengelola *database*, tabel, kolom, relasi, indeks, pengguna, izin, dll) dapat dilakukan melalui antarmuka pengguna, sementara Anda masih memiliki kemampuan untuk langsung mengeksekusi pernyataan *SQL* apa pun.

2.2.17 *Laragon*



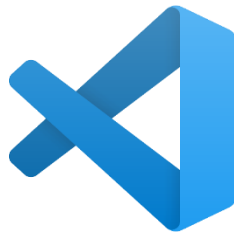
Gambar 2.17 *Laragon*

Laragon adalah lingkungan pengembangan universal yang portabel, terisolasi, cepat & kuat untuk *PHP*, *Node.js*, *Python*, *Java*, *Go*, *Ruby*. Ini cepat, ringan, mudah digunakan dan mudah diperluas.

Laragon sangat bagus untuk membangun dan mengelola aplikasi

web modern. Ini berfokus pada kinerja, dirancang berdasarkan stabilitas, kesederhanaan, fleksibilitas, dan kebebasan.

2.2.18 *Visual Studio Code*



Gambar 2.18 *Visual Studio Code*

Visual Studio Code / *VSCode* merupakan suatu program yang tidak asing lagi bagi para *programmer*. *VSCode* adalah sebuah teks editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi multiplatform, artinya tersedia juga untuk versi *Linux*, *Mac*, dan *Windows*.

Kelebihan *VSCode* antara lain adalah kemampuannya dalam menyediakan berbagai fitur yang mendukung pengembangan perangkat lunak, seperti *syntax highlighting*, *code completion*, *debugging*, dan integrasi dengan berbagai tools pengembangan lainnya. Dengan fitur-fitur yang dimilikinya, *VSCode* telah menjadi salah satu pilihan utama bagi banyak *programmer* dalam mengembangkan berbagai jenis aplikasi.

2.2.19 *Git*

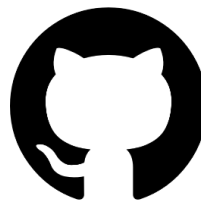


Gambar 2.19 *Git*

Salah satu *version control system* populer yang digunakan para *developer* untuk mengembangkan *software* secara bersama-sama. Dengan adanya *GIT* kita dapat melakukan *programming* dengan kolaborasi di dunia nyata dan di mana lokasinya yang harus sudah terhubung dengan *internet*. Pemanfaat *GIT* ini terdistribusi bukan tersentralisasi.

Git menggunakan pendekatan terdistribusi, yang berarti setiap pengembang memiliki salinan lengkap dari repositori proyek, bukan hanya salinan kerja. Hal ini memungkinkan pengembang untuk bekerja secara independen pada cabang (*branch*) yang berbeda dari repositori dan kemudian menggabungkan perubahan mereka ke cabang utama (*master*) proyek.

2.2.20 *Github*



Gambar 2.20 *Github*

GitHub adalah situs web dan layanan *cloud* yang bisa membantu para pengguna untuk menyimpan, mengelola dan mengembangkan. Di dalam *GitHub* kita bisa mengupload file, membuat file yang mana file-nya bisa dikelola, dengan *version control system* punya github.

GitHub menyediakan fitur-fitur kolaborasi yang memungkinkan pengembang bekerja sama dalam sebuah proyek, seperti pelacakan isu (*issue tracking*), permintaan tarik (*pull requests*), dan diskusi. Dengan *GitHub*, pengembang dapat mengakses proyek mereka dari berbagai tempat dan berkolaborasi dengan pengembang lain di seluruh dunia.

2.2.21 *Composer*



Gambar 2.21 *Composer*

Definisi *composer* menurut (Yosef Murya, 2016) “*Composer* adalah *dependency manager for PHP*, digunakan untuk mempermudah dalam instalasi aplikasi web berbasis *PHP*”.

Composer adalah alat yang sangat berguna dalam pengembangan aplikasi web *PHP* modern karena dapat digunakan untuk mengelola *dependensi* (*library* atau paket lainnya) yang diperlukan oleh proyek *PHP*. Dengan *Composer*, pengembang dapat dengan mudah menambahkan, menghapus, atau memperbarui *dependensi* proyek mereka dengan cepat dan efisien. Ini memungkinkan pengembang

untuk fokus pada pengembangan fungsionalitas aplikasi tanpa harus khawatir tentang manajemen dependensi secara manual.

2.2.22 *Web Browser*



Gambar 2.22 *Web Browser*

Web browser adalah peranti lunak yang dibuat dengan menyimpan informasi apa pun seperti *URL history*, *search keyword*, *timestamp*, *password*, dan segala sesuatu yang dilakukan pengguna saat menjelajah di internet.

Sedangkan Menurut Irawan “ *Web browser* adalah aplikasi perangkat lunak yang memungkinkan penggunanya untuk berinteraksi dengan teks, gambar, video, game, dan informasi lain yang terdapat pada halaman *web* yang terletak di *World Wide Web (WWW)* atau *Local Area Network (LAN)*”.

2.2.23 *UML (Unified Modeling Language)*

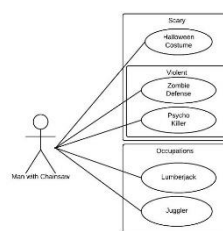


Gambar 2.23 *Unified Modeling Language*

UML (Unified Modeling Language) adalah sebuah bahasa yang digunakan untuk mendokumentasikan, merancang, dan memodelkan sistem perangkat lunak. *UML* menggunakan notasi grafis untuk

menggambarkan struktur dan perilaku dari sistem yang akan dibangun. *UML* memiliki berbagai jenis diagram yang masing-masing digunakan untuk tujuan tertentu, seperti diagram kelas untuk menggambarkan struktur kelas, diagram sekuen untuk menggambarkan interaksi antar objek, dan diagram aktivitas untuk menggambarkan alur kerja atau proses bisnis. *UML* banyak digunakan dalam pengembangan perangkat lunak karena dapat membantu pengembang dalam memahami, merancang, dan mengkomunikasikan desain sistem dengan lebih efektif.

2.2.24 *Usecase Diagram*



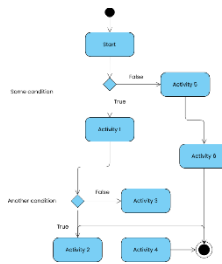
Gambar 2.24 *Usecase Diagram*

Usecase Diagram adalah diagram dalam pemodelan sistem yang menggambarkan interaksi antara aktor (pengguna atau entitas lain yang berinteraksi dengan sistem) dengan sistem untuk mencapai suatu tujuan tertentu. Diagram ini tidak menjelaskan urutan atau langkah-langkah teknis yang dilakukan dalam mencapai tujuan tersebut, tetapi lebih fokus pada apa yang dilakukan oleh aktor dan sistem dalam konteks penggunaan tertentu.

Usecase Diagram membantu dalam memvisualisasikan fungsionalitas sistem secara keseluruhan dan membantu dalam

pemahaman tentang bagaimana sistem berinteraksi dengan pengguna atau entitas lain dalam lingkungan tertentu.

2.2.25 Activity Diagram



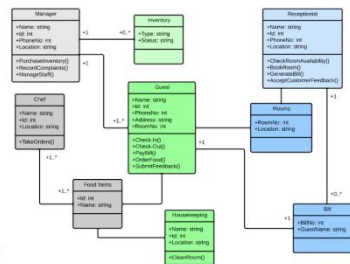
Gambar 2.25 Activity Diagram

Activity Diagram adalah salah satu jenis diagram dalam *UML* (*Unified Modeling Language*) yang digunakan untuk menggambarkan aktivitas atau alur kerja dari suatu proses bisnis, sistem, atau menu yang terdapat dalam perangkat lunak. Diagram ini menggunakan simbol-simbol seperti aktivitas (*rectangles*), garis-garis penghubung (*arrows*), dan penggabungan aktivitas (*diamonds*) untuk menggambarkan urutan aktivitas dalam suatu proses. *Activity Diagram* membantu dalam memvisualisasikan bagaimana suatu proses berjalan dari awal hingga akhir, serta interaksi antara aktivitas-aktivitas dalam proses tersebut.

Dalam *Activity Diagram*, yang digambarkan adalah aktivitas yang dilakukan oleh sistem atau proses, bukan aktivitas yang dilakukan oleh aktor. Aktor dalam konteks ini lebih berperan sebagai pemicu atau penerima dari aktivitas yang dilakukan oleh sistem. Dengan demikian, *Activity Diagram* membantu dalam pemahaman yang lebih baik tentang bagaimana proses bisnis atau sistem bekerja, sehingga memudahkan

dalam analisis, perancangan, dan pengembangan sistem yang efisien dan efektif.

2.2.26 Class Diagram



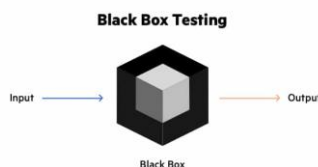
Gambar 2.26 Class Diagram

Class Diagram adalah salah satu jenis diagram dalam pemodelan sistem yang digunakan untuk menggambarkan struktur statis dari sistem atau perangkat lunak. Diagram ini menampilkan beberapa kelas beserta atribut-atribut dan metode-metode yang dimiliki oleh kelas-kelas tersebut, serta hubungan antara kelas-kelas tersebut. *Class Diagram* juga dapat menunjukkan keterkaitan antara objek-objek dalam database dengan mewakili kelas-kelas yang merepresentasikan entitas dalam database. Dengan menggunakan *Class Diagram*, pengembang dapat memahami struktur keseluruhan dari sistem atau perangkat lunak yang sedang dikembangkan, serta memodelkan entitas-entitas dan hubungan antar entitas dalam *database* dengan lebih jelas.

Class Diagram terdiri dari kelas-kelas, atribut-atribut, dan metode-metode. Kelas-kelas merepresentasikan entitas atau objek dalam sistem, sedangkan atribut-atribut adalah data yang dimiliki oleh kelas-kelas tersebut. Metode-metode adalah tindakan atau operasi yang

dapat dilakukan oleh kelas-kelas tersebut. Hubungan antar kelas dalam *Class Diagram* dapat ditunjukkan dengan menggunakan relasi seperti pewarisan (*inheritance*), asosiasi, dan agregasi. *Class Diagram* membantu pengembang dalam merancang struktur yang baik untuk sistem atau perangkat lunak yang akan dikembangkan, serta memudahkan dalam komunikasi antar anggota tim pengembangan.

2.2.27 Pengujian *Blackbox*



Gambar 2.27 Pengujian Blackbox

Pengujian *blackbox* adalah metode pengujian perangkat lunak di mana pengujian dilakukan tanpa memperhatikan struktur internal atau logika dari program yang diuji. Pengujian ini dilakukan dengan cara memasukkan input ke dalam sistem dan mengamati output yang dihasilkan, serta memeriksa apakah sistem berperilaku sesuai dengan spesifikasi yang telah ditentukan.

Dalam pengujian *blackbox*, penguji tidak perlu mengetahui bagaimana sistem diimplementasikan, namun fokus pada pengujian fungsionalitas dan kesesuaian dengan spesifikasi yang telah ditetapkan. Metode ini memungkinkan untuk mengidentifikasi kesalahan atau bug dalam sistem tanpa harus memahami kode atau struktur internalnya.