

## **BAB II**

### **KAJIAN LITERATUR**

#### **A. Perancangan**

Perancangan dapat di artikan sebagai “pembuatan sistem baru atau bentuk lain berupa pengembangan dari sistem yang sudah ada sebelumnya hingga menjadi sebuah sistem yang lebih baik lagi”. (Kurniawan dan Romzi, 2022:3).

Perancangan aplikasi adalah “proses untuk sesuatu yang dikerjakan dengan yang bervariasi untuk membuat tampilan antarmuka program”. (Aziz, dkk., 2020:1-5).

Berdasarkan kedua penjelasan sebelumnya, penulis menyimpulkan bahwa perancangan merupakan pendekatan terstruktur untuk menciptakan sistem baru atau meningkatkan sistem yang sudah ada menjadi lebih baik.

#### **B. Sistem Informasi**

Sistem informasi adalah “suatu perkembangan teknologi yang sangat membantu aktivitas sebuah organisasi”. (Nasution, dkk., 2022: 5893-5896).

Sistem informasi adalah suatu kombinasi teratur apa pun baik dari *people*, *hardware*, *software*, maupun *database* yang mengumpulkan, mengubah, dan menyebarkan informasi di dalam suatu bentuk organisasi. (Fitriana dan Kristannia 2021)

Dari pernyataan tersebut, dapat disimpulkan bahwa Sistem Informasi adalah fondasi yang mendukung kelancaran pelaksanaan aktivitas sehari-hari, operasional, manajerial, dan strategis dalam suatu organisasi. Tujuan utamanya adalah memenuhi kebutuhan organisasi dalam mengelola operasinya secara efisien dengan menyediakan laporan yang relevan dan penting.

#### **C. Inventaris Aset**

Inventaris aset adalah kegiatan pengelolaan persediaan segala sesuatu yang memiliki nilai finansial, berlaku untuk perorangan, perusahaan dan pemerintahan (Usnaini, dkk., 2021: 36-56).

Passaribu (2021:229-241) Inventaris aset adalah “suatu kegiatan pencatatan aset atau barang sekaligus pengelolaan data aset yang dimiliki organisasi secara profesional demi kelancaran operasionalnya”.

Dari pernyataan di atas dapat disimpulkan inventaris aset adalah sesuatu yang bernilai ekonomis dari pemanfaatan/pengoperasian yang menghasilkan pendapatan dan siklus umurnya Panjang.

#### **D. Aset**

Aset dapat diartikan sebagai “sesuatu aset yang memiliki nilai ekonomi, nilai komersial ataupun nilai tukar yang dimiliki oleh suatu individu atau instansi”. (Sihombing dan Yanris, 2020:12-15). Sedangkan Aset adalah “kekayaan yang dimiliki perusahaan yang diharapkan agar dapat memberikan manfaat di masa yang akan datang”. (Farizki, dkk., 2021:17-22).

Dapat ditarik kesimpulan bahwa aset adalah sesuatu yang memiliki nilai ekonomi atau manfaat yang dapat diukur dan dimiliki oleh individu, perusahaan, atau entitas lainnya. Tujuan aset juga dapat melibatkan perlindungan terhadap nilai investasi, pertumbuhan nilai dari waktu ke waktu, atau memberikan dukungan finansial di masa depan.

#### **E. Website**

*Website* dapat diartikan “kumpulan informasi/kumpulan *page* yang biasa diakses lewat jalur internet”. (Romadhon, dkk., 2021:31).

Rochmawati (2019) Situs web atau *Website* merupakan “kumpulan informasi yang terdiri dari halaman web yang saling terhubung satu sama lain yang disediakan secara perorangan, kelompok, atau pun organisasi”.

Menurut penjelasan tersebut, *Website* dapat diartikan sebagai kumpulan halaman atau informasi yang tersedia untuk diakses melalui internet. Dengan konektivitas internet, individu memiliki kemampuan untuk mengaksesnya dari berbagai lokasi dan waktu. Secara teknis, *Website* merupakan gabungan halaman web yang terhubung dalam suatu domain atau sub domain spesifik.

#### **F. Internet**

Internet adalah suatu jaringan komunikasi yang memiliki fungsi untuk menghubungkan antara satu media elektronik dengan media elektronik yang lain dengan cepat dan tepat. (Maharani, dkk., 2021:1-7).

Internet dapat diartikan sebagai “sebuah jaringan komunikasi *public* dan global yang menyediakan koneksi langsung kepada siapa saja melalui *Local area network* (LAN) dan *Internet Service Provider* (ISP)” (Susana, dkk., 2022:2).

Dari kedua penjelasan tersebut, dapat diambil kesimpulan bahwa internet adalah sebuah infrastruktur komunikasi global dan publik yang memfasilitasi interaksi langsung antara individu dan organisasi melalui *Local Area Network* (LAN) dan penyedia layanan *Internet Service Provider* (ISP). Hal ini memungkinkan terjalinnya interaksi antara konsumen, pemasar, dan berbagai perusahaan lainnya, menciptakan sebuah jaringan global yang luas dan terkoneksi.

### **G. Web Browser**

Suryana (2021) menyimpulkan bahwa "*Web Browser* adalah Sebuah *software* aplikasi yang digunakan untuk menerima, menampilkan, dan menerjemahkan informasi dari *world wide web*".

*Web Browser* dapat diartikan sebagai "aplikasi yang digunakan untuk menampilkan halaman web untuk proses pengolahan informasi, pengambilan dan penyajian informasi pada *Website*" (Saputra, dkk., 2021:58-70).

Dari kedua uraian tersebut, disimpulkan bahwa *web browser* adalah sebuah program atau aplikasi yang memungkinkan pengguna untuk menampilkan dan mengakses halaman web. Beberapa contoh *web browser* termasuk *Google Chrome*, *Microsoft Edge*, *Mozilla Firefox*, *Opera Mini*, dan *UC Browser*.

### **H. SDLC (System Development Life Cycle)**

*System Development Life Cycle* (SDLC) adalah aktivitas seperti mendefinisikan, mengembangkan, menguji, mengirim, mengoperasikan, dan memelihara perangkat lunak atau sebuah sistem informasi. (Nova. dkk., 2022:139-148).

Perdana dan Suharni (2021:481-489) menarik kesimpulan sebagai berikut:

SDLC adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik).

Dari kedua uraian tersebut, dapat disimpulkan bahwa yakni sebuah rangkaian metodologi proses untuk menciptakan, mengembangkan, dan menjaga perangkat lunak. Dalam SDLC, terdapat beberapa tahap yang saling terkait, seperti perencanaan, analisis, desain, implementasi, pengujian, dan pemeliharaan. Setiap tahapan bertujuan untuk memastikan bahwa perangkat

lunak yang dihasilkan memenuhi kebutuhan pengguna, spesifikasi, dan standar kualitas yang telah ditetapkan. Cara-cara khusus yang diterapkan dalam SDLC dapat bervariasi tergantung pada kebutuhan proyek dan preferensi organisasi yang bersangkutan.

- *Requirement*

*Requirement* adalah sebuah pernyataan tentang apa yang harus diberikan atau bagaimana penggunaan suatu produk. Istilahnya “kebutuhan” biasanya mengacu pada sesuatu yang diperlukan atau diinginkan untuk mencapai tujuan tertentu atau untuk memenuhi standar tertentu. *Requirement* harus spesifik, tidak ambigu, dan jelas agar dapat dilaksanakan dan diukur secara efektif.

- *Design*

*Design* adalah proses kreatif merencanakan dan merancang sesuatu yang umumnya berfungsi dan tidak ada sebelumnya dalam rangka menyelesaikan suatu masalah agar memiliki nilai lebih dan menjadi lebih bermanfaat bagi penggunanya. Oleh karena itu, *design* melibatkan perencanaan dan perancangan untuk membuat suatu benda, baik dari segi tampilan maupun fungsinya.

- Implementasi

Implementasi juga dikenal sebagai pengembangan, adalah proses atau kegiatan yang bertujuan untuk mengembangkan, memperbaiki, atau meningkatkan suatu. Dalam teknologi informasi, pengembangan mengacu pada upaya untuk menciptakan, memperbaiki, dan mengembangkan sistem, *software*, atau aplikasi yang digunakan dalam komputer atau perangkat digital lainnya.

- Testing

Testing adalah proses menganalisis entitas aplikasi untuk mengevaluasi fiturnya dan menemukan perbedaan antara keadaan saat ini dengan keadaan yang diinginkan.

- Pengujian

Semua proses yang dilakukan untuk mendapatkan *software* dan *hardware* baru dan berfungsi dengan baik di lingkungannya, termasuk instalasi, konfigurasi, pengoperasian, pengujian, dan membuat perubahan yang diperlukan, termasuk dalam istilah “Development”.

- *maintenance*

*maintenance* adalah proses menjaga dan memelihara fasilitas dan peralatan dengan melakukan perbaikan, penyesuaian dan pergantian yang

diperlukan untuk memastikan bahwa operasi sistem berjalan dengan baik sesuai rencana. Logo *SDLC* dapat dilihat pada gambar 1.



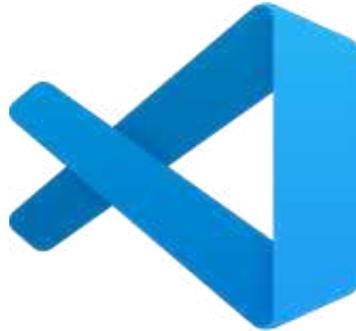
Gambar 1. Logo *SDLC* (Sumber: Penulis)

### I. *VS Code (Visual Studio Code)*

*VS Code* dapat diartikan “sebuah teks editor ringan dan andal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux*, *Mac*, dan *Windows*” (Permana dan Romadlon, 2019:155).

*Visual Studio Code* adalah editor kode sumber yang ringan namun kuat yang berjalan di *desktop* dan tersedia untuk *Windows*, *macOS*, dan *Linux* (Kurniawan dan Kurniawan, 2020:1-9).

Dari penjelasan tersebut, dapat ditarik kesimpulan bahwa *Visual Studio Code (VS Code)* adalah sebuah perangkat lunak penyunting kode sumber yang dibuat oleh *Microsoft*. Program ini dapat dijalankan pada sistem operasi *Windows*, *macOS*, dan *Linux*. *VS Code* berfungsi pada komputer *desktop* dan mendukung berbagai bahasa pemrograman dan lingkungan *runtime*, termasuk *JavaScript*, *TypeScript*, dan *Node.js*. Selain itu, *VS Code* juga menyediakan dukungan luas untuk ekosistem pengembangan yang mencakup bahasa seperti *C++*, *C#*, *Java*, *Python*, *PHP*, *Go*, dan *.NET*. Logo *VS Code* dapat dilihat pada gambar 2.



Gambar 2. Logo *Visual Studio Code* (Sumber: <https://code.visualstudio.com/>)

### **J. HTML (*Hypertext Markup Language*)**

Mariko (2019:80-91) menyimpulkan bahwa “*Hypertext Markup Language* (HTML) adalah bahasa standar yang digunakan untuk menampilkan konten pada halaman *Website*”.

HTML dapat diartikan sebagai “bahasa pemrograman yang digunakan untuk pembuatan halaman *Website* agar dapat menampilkan berbagai informasi baik tulisan maupun gambar pada sebuah *web browser*” (Setiawan, dkk., 2019).

Berdasarkan kedua pernyataan tersebut, dapat disimpulkan bahwa HTML adalah bahasa *markup* yang digunakan untuk membuat halaman web. Dengan menggunakan HTML, fungsi HTML adalah untuk mengelola sekumpulan data dan informasi sehingga dokumen dapat diakses dan ditampilkan di Internet melalui layanan web. Sebagai fondasi sebuah halaman web, pengembang menggunakan kode HTML untuk mendesain elemen halaman web di *browser*, seperti teks, *hyperlink*, dan *file media*.

### **K. CodeIgniter**

Maydianto (2021) menyimpulkan bahwa “*CodeIgniter* adalah sebuah aplikasi berbasis *web networking* yang sifatnya terbuka (*open source*) yang dapat digunakan dalam merancang sebuah aplikasi web yang dinamis dengan cepat dan mudah”.

*CodeIgniter* dapat diartikan sebagai “kerangka kerja pengembangan aplikasi PHP berdasarkan arsitektur yang terstruktur”. (Syaebani, dkk., 2021:34).

Dari kedua pernyataan tersebut, dapat disimpulkan bahwa *CodeIgniter* merupakan *Framework* PHP yang fokus pada struktur dan organisasi yang terstruktur. *Framework* ini mempermudah pembuatan aplikasi PHP yang dinamis dengan menyediakan berbagai alat bantu, seperti *helper* dan *pustaka*. *CodeIgniter* dirancang untuk mempercepat proses pengembangan proyek dengan

menyediakan berbagai pustaka yang mendukung kegiatan umum dalam pengembangan aplikasi web. *CodeIgniter* berperan dalam pengembangan web dengan tujuan menyederhanakan proses penguraian program yang kompleks menjadi bagian yang lebih mudah dikelola. Melalui arsitektur *MVC*, tanggung jawab dibagi secara terpisah antara Model (bertugas untuk mengelola data), *View* (mengatur tampilan yang dipresentasikan), dan *Controller* (mengelola interaksi dan logika). Hal ini meningkatkan struktur, pemeliharaan, dan kemungkinan penggunaan kembali kode. Logo *CodeIgniter* dapat dilihat pada gambar 3.



Gambar 3. Logo CodeIgniter (Sumber: <https://codeigniter.com/>)

#### L. PHP (*Hypertext Preprocessor*)

PHP adalah bahasa pemrograman untuk dijalankan melalui halaman web, umumnya digunakan untuk mengolah informasi di internet. (Hidayah, dkk., 2019:41-52).

PHP (*Hypertext Preprocessor*) dapat diartikan sebagai “suatu bahasa pemrograman yang digunakan *user* untuk membangun sebuah web saat ini dan dapat digunakan secara gratis” (Lubis, dkk., 2023).

Dari kedua uraian tersebut, dapat disimpulkan bahwa PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman sisi server yang kuat dan fleksibel untuk pengembangan web. Karena kemudahannya dalam penggunaan



Gambar 4. Logo PHP (Sumber: <https://www.php.net/>)

dan bersifat *open-source*, PHP menjadi pilihan yang populer bagi pengembang web. Logo PHP dapat dilihat pada gambar 4.

### **M. CSS (*Style Sheet Language*)**

Fataha (2022) menyimpulkan bahwa “CSS adalah bahasa *Cascading Style Sheet* dan biasanya digunakan untuk mengatur tampilan elemen yang tertulis dalam bahasa markup, seperti HTML”.

CSS (*Style Sheet Language*) dapat diartikan sebagai “salah satu bahasa desain web yang mengontrol format tampilan sebuah halaman web yang ditulis dengan menggunakan penanda markup *language*”. (Parjito, dkk., 2022:356).

Dari kedua pernyataan tersebut, dapat disimpulkan bahwa CSS (*Cascading Style Sheets*) adalah bahasa yang diciptakan secara khusus untuk mengendalikan penampilan visual dan pemformatan halaman web.

CSS (*Cascading Style Sheets*) memungkinkan pemisahan antara konten dan aspek visual halaman web, yang memudahkan dalam pemeliharaan dan perbaruan situs. Sebagai contoh, Anda bisa merubah gaya *font* atau warna *header* pada keseluruhan situs hanya dengan memperbarui kode CSS, tanpa perlu mengedit HTML untuk setiap *header* secara terpisah. Dengan berbagai atribut yang disediakan oleh CSS untuk mengatur tampilan halaman web, seperti *font*, warna teks, latar belakang, *margin*, *padding*, dan lainnya, Anda dapat menciptakan desain visual yang seragam dan menarik untuk situs Anda.

### **N. XAMPP**

XAMPP dapat diartikan Sebagai “aplikasi web server instan dan lengkap dikarenakan segala yang dibutuhkan untuk membuat sebuah situs web dengan *Content Management System (Joomla)* bisa dicoba di dalam aplikasi ini”. (Hidayah, dkk., 2019:41-52).

XAMPP adalah *tool* yang menyediakan paket perangkat lunak dalam satu buah paket. (Novendri, dkk., 2019)

XAMPP, yang dikembangkan oleh *Apache Friends*, adalah paket perangkat lunak yang memungkinkan pembuatan lingkungan pengembangan web lokal di komputer pengguna. Ini disingkat dari "*Cross-Platform (X), Apache (A), MySQL (M), PHP (P), dan Perl (P)*". Di dalamnya, terdapat *integrasi server Apache, MySQL, PHP, dan Perl*. Komponen-komponen ini *esensial* dalam proses pengembangan dan pengujian *situs web* serta aplikasi web.

## O. MySQL

MySQL adalah salah satu jenis *database* server yang sangat populer, hal ini disebabkan karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *datasenya*. (Wananjar dan Susanti, 2021:97-105).

Putri (2023) MySQL adalah sistem manajemen basis data *open-source* yang sangat fleksibel dan aman.

Dari kedua uraian tersebut, bisa disimpulkan bahwa MySQL adalah sistem manajemen *database relasional (RDBMS)* berbasis SQL yang bersifat *open source*. Jika DBMS adalah sistem manajemen *database* secara umum, RDBMS merupakan sistem pengelolaan *database* berdasarkan data yang saling berkaitan (memiliki relasi) dalam satu kelompok. Lebih mudahnya, MySQL merupakan sebuah *software* yang berfungsi untuk membuat dan mengelola berbagai informasi yang ada pada *database* di dalam sebuah server dengan bahasa pemrograman SQL. Logo MySQL dapat dilihat pada gambar 5.



Gambar 5. Logo MySQL (Sumber: <https://www.mysql.com/>)

## P. ERD (*Entity Relationship Diagram*)

ERD adalah “sebagai alat bantu dalam pembuatan *database* dan memberikan gambaran bagaimana kerja *database* yang akan dibuat”. (Felix, dkk., 2021).

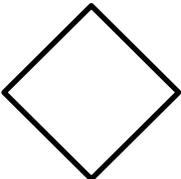
ERD adalah “suatu gambaran grafis yang mewakili logika *database* secara lengkap dan mendetail, di mana antar *database* membentuk entitas yang terhubung satu sama lain”. (Kalua, dkk., 2023:24-32).

Kesimpulan dari kedua pernyataan tersebut adalah bahwa *Entity Relationship Diagram (ERD)* merupakan alat bantu visual yang penting dalam desain dan manajemen basis data. ERD memberikan gambaran menyeluruh tentang struktur basis data, termasuk *entitas*, atribut, dan hubungan antar entitas

tersebut. Pada tahap perencanaan, penggunaan *Entity Relationship Diagram* (ERD) membantu desainer basis data dalam menyusun penyimpanan data secara strategis, membangun koneksi data, dan meningkatkan efisiensi proses bisnis. Lebih lanjut, ERD memperkuat komunikasi antara tim pengembang dan pemangku kepentingan, sehingga memastikan pemahaman yang sama tentang desain basis data di semua pihak yang terlibat.

*Entity Relationship Diagram* (ERD) berperan sebagai instrumen analisis untuk mengenali data dan kebutuhan proses bisnis, memastikan bahwa persyaratan tersebut dipenuhi dalam struktur *database*. Selain itu, ERD berfungsi sebagai dokumen penting yang dapat digunakan sebagai rujukan di masa depan, mempermudah proses penyesuaian atau penambahan elemen baru ke dalam *database*. Terakhir, ERD membantu dalam mencegah kesalahan desain dengan memfasilitasi identifikasi dan pencegahan kesalahan, sehingga mengurangi kemungkinan kesalahan selama fase implementasi *database*. Oleh karena itu, *diagram Entity Relationship* (ERD) memiliki peranan yang krusial dalam menjamin efektivitas, organisasi, dan orientasi pada pengguna dari struktur *database*. Simbol *Entity Relationship Diagram* (ERD) terdiri dari 4 simbol yang akan diuraikan pada tabel 1.

Tabel 1. Simbol *Entity Relationship Diagram* (ERD)

No.	Simbol	Nama Simbol	Fungsi
1		<i>Entitas</i>	Adalah dapat diidentifikasi berdasarkan data yang terkumpul, membedakannya dari entitas lain yang terkait.
2		<i>Relationship</i>	Yaitu representasi dari keterkaitan antara dua entitas atau lebih, yang memperlihatkan bagaimana entitas saling berhubungan dalam suatu basis data.
3		Atribut	Atribut merupakan ciri atau kualitas dari suatu entitas yang memberikan tambahan informasi mengenai entitas tersebut.

No.	Simbol	Nama Simbol	Fungsi
4		Alur	Dalam Diagram Entitas-Relasi (ERD), garis digunakan untuk mengaitkan entitas dengan atributnya atau untuk menghubungkan entitas satu dengan yang lain melalui relasi. Garis ini menggambarkan hubungan dan arus informasi antara komponen dalam diagram.

(Sumber: Ilham, 2023).

### Q. *Data Flow Diagram* (DFD)

Purnama (2021:1-6) menarik kesimpulan sebagai berikut:

*Data flow* diagram ini adalah suatu *network* yang menggambarkan suatu sistem automat/komputerisasi, manual isasi, atau gabungan dari keduanya, yang penggambarannya disusun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan mainnya.

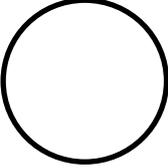
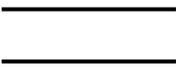
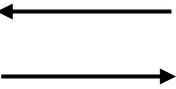
Irawan (2019:13-20) menarik kesimpulan sebagai berikut:

DFD (*Data Flow Diagram*) adalah model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan ke mana tujuan data yang keluar dari sistem, di mana data disimpan, proses apa yang dihasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang akan dikenakan pada data tersebut.

Dari kedua pernyataan tersebut dapat diambil kesimpulan *Data Flow Diagram* (DFD) adalah representasi grafis yang menggambarkan pergerakan data dalam suatu sistem atau proses bisnis. DFD memberikan gambaran yang jelas tentang sumber dan tujuan setiap aliran data, bersama dengan operasi spesifik yang dilakukan pada data tersebut.

Selain itu, DFD juga menggambarkan bagaimana data dimanipulasi dan interaksi antara data dan prosedur yang sedang berlangsung. DFD memberikan pemahaman yang komprehensif tentang komponen penting dari sistem, interkoneksi mereka, dan memastikan aliran data yang optimal di dalam sistem. Menjamin fungsionalitas optimal dari sistem sangat penting selama proses analisis dan desain. Simbol-simbol dalam *Entity Relationship Diagram* (ERD) terdiri dari 4 jenis simbol yang akan dijelaskan dalam tabel.

Tabel 2. Simbol *Data Flow Diagram* (DFD)

No.	Simbol	Nama Simbol	Keterangan
1		<i>Process</i>	Menunjukkan tentang bagaimana input berubah menjadi output, di mana bisa saja banyak input menghasilkan satu output, atau sebaliknya.
2		<i>Terminator</i>	Menggambarkan entitas luar yang sistemnya berinteraksi dengannya.
3		Penyimpanan	Untuk menggambarkan atau merepresentasikan sekumpulan data atau paket data.
4		Aliran	Mengilustrasikan pergerakan paket data atau informasi dari satu bagian sistem ke tempat penyimpanan data.

(Sumber: Pratama dan Sariana, 2019:4).

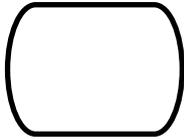
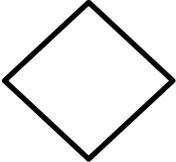
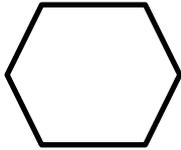
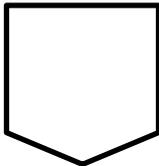
## R. *Flowchart*

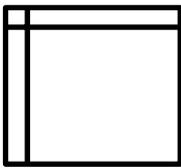
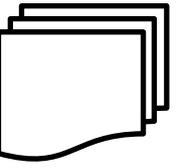
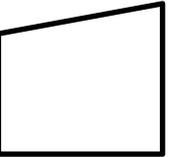
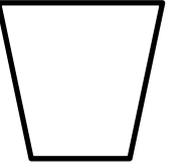
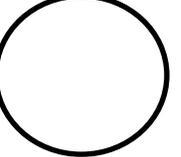
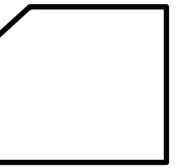
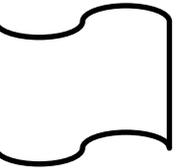
Fauzi (2020) *Flowchart* dapat diartikan sebagai “cara penulisan algoritma dengan menggunakan notasi grafis”.

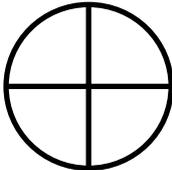
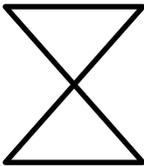
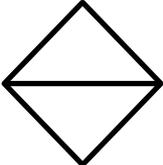
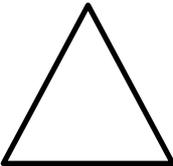
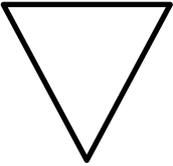
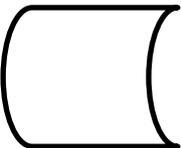
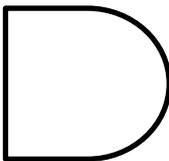
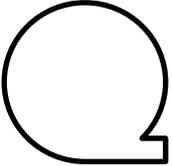
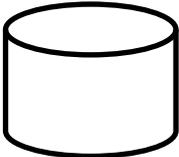
*Flowchart* dapat diartikan sebagai “gambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program”. (Zalukhu, dkk., 2023:63).

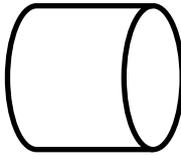
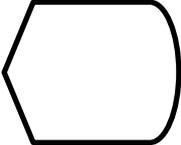
Dari kedua penjelasan tersebut, dapat disimpulkan bahwa *Flowchart* adalah representasi visual yang menggambarkan langkah-langkah dan urutan prosedur dalam sebuah program atau proses. *Flowchart* menggunakan simbol-simbol dan panah untuk mengilustrasikan pergerakan data saat melewati serangkaian instruksi atau pilihan menuju tujuan akhir. Ini mempermudah pemahaman dan analisis proses bagi pengembang, analis, serta pihak-pihak terkait lainnya, yang pada akhirnya meningkatkan efisiensi dan keterbacaan dalam memahami proses tersebut. *Flowchart* merupakan alat yang sangat bermanfaat dalam perencanaan, analisis, dan dokumentasi sistem dan proses. Simbol-simbol *Flowchart* dijelaskan secara detail dalam tabel 3.

Tabel 3. Simbol *Flowchart*

No.	Simbol	Nama Simbol	.Fungsi
1		<i>Terminator</i>	Ini digunakan untuk menunjukkan titik awal atau akhir dari suatu proses atau sistem.
2		<i>Decision</i>	Ini dipakai untuk menandai titik di mana suatu keputusan harus diambil.
3		<i>Flow Line</i>	Garis atau anak panah yang menghubungkan simbol-simbol berbeda dalam <i>Flowchart</i> .
4		<i>Preparation</i>	Simbol yang mencerminkan tahapan persiapan atau langkah awal yang perlu diselesaikan sebelum memulai proses utama..
5		<i>Data</i>	Ini dipakai untuk menandakan masukan atau keluaran data dalam suatu proses.
6		<i>Off-Page Connector</i>	Simbol yang menunjukkan titik di mana suatu proses melanjutkan ke halaman berikutnya atau komponen lain dari diagram aliran.
7		<i>Process</i>	Digunakan untuk menunjukkan tugas atau tindakan tertentu yang harus dilakukan selama proses.
8		<i>Predefined process</i>	Digunakan untuk menunjukkan aktivitas atau tindakan yang telah direncanakan dan ditetapkan sebelumnya.

No.	Simbol	Nama Simbol	.Fungsi
9		<i>Internal storage</i>	Digunakan dalam diagram aliran untuk menunjukkan lokasi data atau informasi yang sedang disimpan dalam proses atau sistem.
10		<i>Document</i>	Digunakan untuk menunjukkan catatan atau dokumen yang dibuat, diperiksa, atau diolah sepanjang proses.
11		<i>Multidocument</i>	Ini digunakan untuk menunjukkan bagaimana beberapa dokumen atau catatan berinteraksi dalam suatu proses.
12		<i>Manual input</i>	Digunakan untuk menunjukkan proses di mana pengguna atau operator secara manual memasukkan data atau informasi.
13		<i>Manual operation</i>	Digunakan untuk menunjukkan proses yang dilakukan oleh pengguna atau operator secara manual dalam Flowchart.
14		<i>Connector</i>	Digunakan untuk menghubungkan dua bagian Flowchart yang berbeda atau pada halaman yang berbeda.
15		<i>Card</i>	Flowchart digunakan untuk menampilkan data atau informasi yang disimpan dalam bentuk kartu fisik.
16		<i>Punched tape</i>	Digunakan untuk menunjukkan langkah-langkah yang diambil untuk menyimpan atau mengolah data menggunakan media pita punch.
17		<i>Summing junction</i>	Digunakan untuk menunjukkan titik di mana beberapa aliran data atau data digabungkan atau dijumlahkan.

No.	Simbol	Nama Simbol	.Fungsi
18		<i>Or</i>	Menunjukkan titik keputusan di mana salah satu atau lebih kondisi harus dipenuhi.
19		<i>Collate</i>	Digunakan untuk menggambarkan proses pengelompokan atau pengurutan data atau informasi sesuai dengan kriteria tertentu.
20		<i>Sort</i>	Digunakan untuk menandai langkah-langkah di mana data diurutkan menurut kriteria tertentu.
21		<i>Exstrac</i>	Digunakan untuk menunjukkan proses ekstraksi atau pengambilan data atau informasi dari sumber yang sudah ada.
22		<i>Marge</i>	Digunakan untuk menggambarkan proses menggabungkan atau menyatukan data atau data dari berbagai alur atau sumber.
23		<i>Storage data</i>	Digunakan untuk menunjukkan lokasi data atau informasi yang disimpan dalam <i>Flowchart</i> .
24		<i>Delay</i>	Digunakan untuk menandai langkah yang tertunda atau waktu tunggu sebelum proses berikutnya dimulai.
25		<i>Sequential access storage</i>	Digunakan untuk mengidentifikasi langkah-langkah di mana data dapat diakses secara berurutan.
26		<i>Magnetic disc</i>	Digunakan untuk menandai proses di mana data dapat diakses atau disimpan dari disk magnetik.

No.	Simbol	Nama Simbol	.Fungsi
27		<i>Direct access storage</i>	Digunakan untuk menandai langkah-langkah di mana data dapat diakses secara langsung tanpa terlebih dahulu mengakses data lain.
28		<i>Display</i>	Menunjukkan proses tampilan atau output data atau informasi kepada pengguna atau sistem lain.

(Sumber: Penulis 2024).

### S. *Black Box Testing*

*Black Box* dapat diartikan sebagai “teknik pengujian tanpa mengacu pada struktur internal dari komponen atau sistem”. (Parlika, dkk., 2020:132-140).

*Black Box testing* adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja. (Putra, dkk., 2020:74-78).

Dari kedua pernyataan tersebut, dapat disimpulkan bahwa *Black Box testing* adalah Teknik pengujian perangkat lunak yang berfokus pada fungsi perangkat lunak dari sudut pandang pengguna, tanpa mengetahui struktur internal atau kode perangkat lunak. *Black Box testing* bertujuan untuk mengamati hasil input dan output dari perangkat lunak dan mengetahui apakah perangkat lunak beroperasi dengan baik dan optimal sesuai dengan persyaratan fungsional yang diharapkan.