BAB II
TINJAUAN PUSTAKA

2.1 Teori Umum

Dalam teori ini, meliputi penjelasan tentang Pengertian Komputer, Perangkat Lunak (*Software*), dan Basis Data (*Database*), dan Sistem.

2.1.1 Pengertian Komputer

(Harmayani *et al*. 2021: 1) menjelaskan bahwa Komputer adalah suatu perangkat elektronik yang dapat digunakan untuk mengolah data sesuai dengan prosedur yang telah dirumuskan sebelumnya sehingga menghasilkan informasi bermanfaat bagi penggunanya.

(Paray Theo Lonando and Arif Fadillah 2023: 54) menjelaskan bahwa Komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas, yaitu menerima *input*, memproses *input* sesuai dengan instruksi yang diberikan, menyimpan perintah-perintah dan hasil pengolahannya, serta menyediakan *output* dalam bentuk informasi.

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa komputer adalah sebuah alat komunikasi yang dapat mengelola data dan informasi yang sangat dibutuhkan secara otomatis

2.1.2 Pengertian Perangkat Lunak (*Software*)

Menurut (Harmayani *et al*. 2021: 43), perangkat lunak atau *software* komputer merupakan sekumpulan data elektronik, yang tersimpan dan kemudian dikendalikan oleh perangkat komputer.

Menurut (Zalukhu *et al*. 2023: 62), perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*).

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa perangkat lunak atau *software* adalah sekumpulan data elektronik yang tersimpan dan dikendalikan oleh perangkat komputer, serta program komputer yang terasosiasi dengan dokumentasi perangkat.

2.1.3 Pengertian Basis Data (*Database*)

Menurut (Pahlevi *et al*. 2018: 19), basis data adalah sekumpulan data yang saling berhubungan secara logis dan terorganisir dengan baik. Kumpulan data tersebut yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Kumpulan dokumen/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

(Titus *et al*. 2019: 72) Menjelaskan bahwa Basis Data ialah koleksi terpadu antar data yang saling berkaitan yang berguna untuk memenuhi setiap kebutuhan informasi dalam suatu instansi. Setiap masing-masing tabel di dalam *database* memiliki fungsi sebagai penyimpan data-data yang saling berhubungan antar tabel.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Ada banyak cara untuk mengorganisasi skema atau memodelkan struktur basis data. Model yang umum digunakan sekarang adalah model relasional yang mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom saling berhubungan di mana setiap tabel terdiri dari baris dan kolom.

Dalam model ini, hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel. Model yang lain seperti model herarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel. Pada komputer, basis data disimpan dalam perangkat hardware penyimpan, dan dengan *software* tertentu dimanipulasi untuk kepentingan atau keguanaan tertentu. Hubungan atau relasi data biasanya ditunjukkan dengan kunci (*key*) dari tiap file yang ada. Data dalam basis data bersifat *integrated* dan *shared*:

a. Terpadu (*integrated*), berkas-berkas data yang ada pada basis data saling terkait (terjadi dependensi data);
b. Berbagi data (*shared*), data yang sama dapat digunakan sejumlah pengguna dalam waktu yang bersamaan atau yang disebut sebagai sistem *multiuser*.

Dalam implementasinya, untuk memudahkan dalam mengakses data, data disusun dalam suatu struktur logis yang menjelaskan bahwa:

a. Kumpulan tabel menyusun basis data.

b. Tabel tersusun atas sejumlah *record*.

c. Sebuah *record* mengandung sejumlah *field*.

d. Sebuah *field* disimpan dalam bentuk kumpulan bit.

Istilah secara umum tabel yang baik memiliki beberapa karakteristik sebagai berikut:

a. Memiliki kolom (*field*) dan baris data (*record*).

b. Nama kolom bersifat unik (tidak ada yang sama).

c. Sebuah kolom (*field*) hanya memiliki satu tipe data.

d. Antar kolom tidak terpisah kolom kosong atau tanpa nama kolom.

e. *Header* (judul kolom) hanya terdiri dari satu baris dan ditempatkan pada baris pertama.

f. Antar baris data tidak terpisah baris kosong ataupun baris subtotal.

Dalam pengembangan suatu sistem, basis data memiliki peran yang penting dan sangat diperlukan. Berikut merupakan alasan perlunya basis data:

1. Salah satu komponen penting dalam suatu sistem informasi adalah basis data, karena basis data merupakan dasar dalam menyediakan informasi.
2. Informasi dikatakan lebih bernilai jika memiliki manfaat yang efektif dibandingkan dengan biaya dalam mendapatkannya. Dalam hal ini, basis data akan menentukan kualitas informasi.
3. Basis data mengurangi duplikasi data (*data redudancy*).
4. Basis data dapat mengurangi pemborosan tempat simpanan luar.
5. Data akan dapat saling berelasi dengan menerapkan sistem basis data.

Dari pernyataan di atas dapat disimpulkan bahwa basis data (*database*) adalah kumpulan dari data-data terstruktur yang disimpan secara sistematis di dalam suatu sistem komputer.

2.1.4 Pengertian Sistem

Menurut (Sangga Rasefta and Esabella 2020: 52), Sistem adalah kumpulan komponen yang saling terkait satu dengan yang lainnya kegiatan pokok untuk mencapai suatu tujuan tertentu.

Menurut (Devianty *et al*. 2021: 101), Sistem merupakan sekumpulan komponen-komponen atau jaringan kerja dari prosedur-prosedur yang saling

berkaitan dan saling bekerja sama membentuk suatu jaringan kerja untuk mencapai sasaran atau tujuan tertentu.

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa sistem adalah suatu kumpulan komponen atau unsur yang saling terkait dan saling mempengaruhi satu sama lain dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu.

2.2 Teori Khusus

Dalam teori ini, meliputi penjelasan tentang Pengertian *Data Flow Diagram (DFD), Flowchart, Blockchart, Entity Relationship Diagram (ERD)*, Kamus Data *(Data Dictionary),* Metode Pengembangan Sistem Menggunakan *Waterfall*, dan Metode Pengujian.

2.2.1 Pengertian *Data Flow Diagram* (DFD)

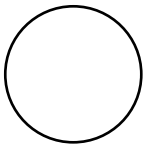
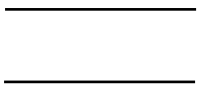
(Yusdistira 2020: 2) menjelaskan bahwa *Data Flow Diagram* (DFD) merupakan suatu model yang dapat memberikan suatu tampilan secara visual, yang mana pada model tersebut dapat menggambarkan suatu aliran data maupun informasi pada sebuah sistem.

Menurut Kristanto dalam (Yunita *et al*. 2023: 241), *Data Flow Diagram* (DFD) disebut juga dengan Diagram Arus Data (DAD). DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data, kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut, interaksi antara data yang tersimpan, dan proses yang dikenakan pada data tersebut.

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa *Data Flow Diagram* adalah suatu representasi grafis yang digunakan untuk menggambarkan aliran data dan informasi dalam sebuah sistem.

Adapun simbol-simbol yang sering digunakan dalam *Data Flow Diagram* (DFD) dapat dilihat pada tabel berikut ini.

Tabel 2.1 Simbol *Data Flow Diagram* (DFD)

No	Simbol	Nama	Keterangan
1.		Proses (*Process*)	Proses (*Process*) atau fungsi atau prosedur pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur.
2.		Penyimpanan (*Data Store*)	Data store adalah file untuk menyimpan data yang digunakan untuk proses selanjutnya. Dapat dikatakan juga, sama seperti basis data (*database*).
3.		Entitas Luar (*External Entity*).	Entitas luar merupakan individu, divisi, atau perusahaan yang berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data.
4.		Aliran Data (*Data Flow*)	*Data flow* adalah arus data yang mengalir antara entitas, proses, dan data *store*.

Sumber: Sukamto (2022: 91-93)

DFD dipopulerkan Oleh DeMacro & Yordan dan Gane & Sarson dengan menggunakan pendekatan Metoda Analisis Sistem Terstruktur. DFD merupakan teknik model proses untuk mengorganisasikan struktur dan alir data di dalam sistem. Dalam laporan ini akan digunakan model DeMarco/Yordan. Berikut adalah langkah-langkah dalam membuat *Data Flow Diagram*.

1. Memahami sistem yang akan dikembangkan dilakukan dengan cara wawancara kepada bagian yang mengunakan serta memproses data.
2. Tentukan batas sistem yang akan digambar, batasan meliputi data yang masuk dan informasi yang akan keluar dari sistem.
3. Setelah diketahui batasan sistem dapat dikembangkan ke dalam diagram konteks. Diagram konteks adalah diagram tingkat atas, merupakan diagram dari sebuah sistem yang menggambarkan aliran-aliran data yang masuk dan keluar dari sistem.
4. Membuat dekomposisi fungsional untuk memudahkan kita dalam memgambarkan DFD pada level-level berikutnya.

5. Beri nomor urut pada setiap proses dan membuat diagram secara rinci.

2.2.2 Pengertian *Flowchart*

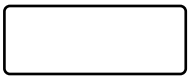
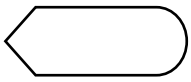
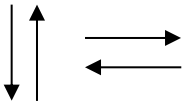
Menurut (Zalukhu *et al*. 2023: 62), *Flowchart* adalah gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut.

Menurut (Zalukhu *et al*. 2023: 62), *Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program.

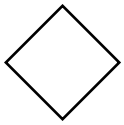
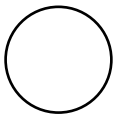
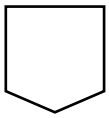
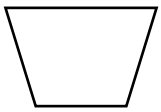
Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa *Flowchart* merupakan penggambaran langkah-langkah atau urutan prosedur dalam bentuk notasi-notasi tertentu.

Adapun simbol-simbol yang sering digunakan dalam *Flowchart* dapat dilihat pada tabel berikut ini.

Tabel 2.2 Simbol *Flowchart*

No	Simbol	Nama	Keterangan
1.		Terminator	Terminator berfungsi sebagai awal atau akhir dari sebuah sistem.
2.		Proses (*Process*)	Simbol yang menyatakan suatu proses yang dilakukan oleh komputer.
3.		*Input* / *Output*.	Simbol yang menyatakan proses *input* atau *output* tanpa tergantung peralatan.
4.		*Display*	Simbol yang menyatakan layar peraga (monitor).
5.		Aliran (*Flow)*	Simbol yang menyatakan aliran / alur dari sebuah sistem.

Lanjutan Tabel 2.2 Simbol *Flowchart*

6.		Keputusan (*Decision*)	Simbol yang menghasilkan dua kemungkinan jawaban.
7.		*On Page Connector*	Simbol yang berfungsi menghubungkan bagian *Flowchart* yang terdapat pada satu halaman.
8.		*Off Page Connector*	Simbol yang berfungsi menghubungkan bagian *Flowchart* yang terdapat pada halaman berbeda.
9.		Dokumen *(Document)*	Simbol untuk mencetak keluaran dokumen berupa kertas.
10.		*Predefined Process*	Simbol yang menyatakan proses permulaan penjalanan sub program.
11.		*Manual Operation*	Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.

Sumber: Iswandy dalam Indrianto *et al*., (2019: 102)

2.2.3 Pengertian *Blockchart*

Menurut (Zahra. M *et al*. 2023: 330), *Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

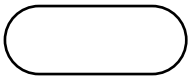
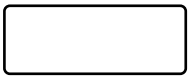
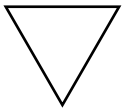
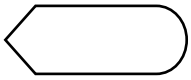
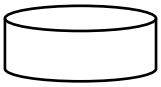
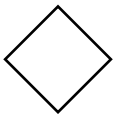
Menurut (Lestari 2021: 39), Diagram blok *(Blockchart Diagram*) berfungsi memodelkan masukan, keluaran, referensi, master, proses maupun transaksi dalam simbol-simbol tertentu.

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa *Blockchart* berfungsi untuk menjelaskan alur masukan, keluaran, dan proses dari sebuah sistem dengan simbol-simbol tertentu. *Blockchart* juga dapat memudahkan pengguna dalam memahami alur dari sebuah sistem.

Adapun simbol-simbol yang sering digunakan dalam *Blockchart* dapat dilihat pada

tabel berikut ini.

Tabel 2.3 Simbol *Blockchart*

No	Simbol	Nama	Keterangan
1.		Terminator	Terminator berfungsi sebagai awal atau akhir dari sebuah sistem.
2.		Proses (*Process*)	Simbol yang menyatakan suatu proses yang dilakukan oleh komputer.
3.		*Offline Storage*	Simbol yang menyatakan dokumen yang akan diarsipkan secara manual.
4.		*Display*	Simbol yang menyatakan layar peraga (monitor).
5.		*Data Storage*	Simbol yang menyatakan suatu tempat peyimpanan data.
6.		Keputusan (*Decision*)	Simbol yang menunjukan kondisi tertentu dan menghasilkan dua kemungkinan jawaban.
7.		Dokumen (*Document*)	Simbol untuk mencetak keluaran dokumen berupa kertas.
8.		*Multi Document*	Simbol untuk mencetak lebih dari 1 keluaran dokumen berupa kertas.
9.		*Manual Input*	Simbol untuk pemasukan data secara manual melalui keyboard.

Sumber: Kristanto (2018: 76)

2.2.4 Pengertian *Entity Relationship Diagram* (ERD)

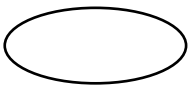
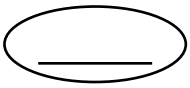
(Khomsi Pane *et al*. 2022: 91) menjelaskan bahwa *Entity Relationship Diagram* (ERD) adalah suatu bentuk diagram yang menjelaskan hubungan antar objek - objek data yang mempunyai hubungan antar relasi.

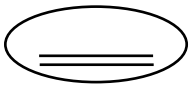
Menurut Sunanta dikutip dalam (Putra *et al*. 2019: 139), *Entity Relationship Diagram* (ERD) merupakan suatu model data yang dikembangkan berdasarkan objek. Dan digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis.

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa *Entity Relationship Diagram* (ERD) merupakan model data yang berbasis objek dan representasi grafis dari logika *database*. ERD mencakup deskripsi detail tentang entitas, hubungan dan batasan dalam suatu sistem *database*.

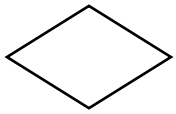
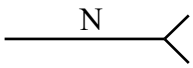
Adapun simbol-simbol yang sering digunakan dalam *Entity Relationship* Diagram (ERD) dapat dilihat pada tabel berikut ini.

Tabel 2.4 Simbol *Entity Relationship Diagram* (ERD)

No	Simbol	Nama	Keterangan
1.		Entitas (*Entity*)	Entitas merupakan benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh sistem atau aplikasi komputer.
2.		Atribut	*Field* atau kolom data yang butuh disimpan dalam suatu entitas.
3.		Atribut Kunci Primer	*Field* atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan. Kunci Primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut bersifat unik.

4.		Atribut Multi Nilai	*Field* atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

Lanjutan Tabel 2.4 Simbol *Entity Relationship Diagram* (ERD)

5.		Relasi (*Relation*)	Simbol ini digunakan untuk menunjukan hubungan yang ada antara entitas yang satu dan entitas yang lainnya. Biasanya diawali dengan kata kerja.
6.		Asosiasi	Penghubung antara relasi dan entitas di mana kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian.

Sumber: Rosa A.S & M. Shalahuddin (2018: 50-51)

2.2.5 Pengertian Kamus Data (*Data Dictionary*)

Menurut (Yunita *et al*. 2023: 173), Kamus data merupakan suatu penjelasan tertulis secara lengkap dan jelas tentang suatu data dalam *database* sesuai dengan sistem yang dibuat. Kamus data dibuat untuk mengurangi duplikasi data.

Menurut (Premana 2019: 56), Kamus data (*Data Dictionary*) adalah penjabaran dari aliran-aliran data yang ada di dalam sebuah data flow diagram. Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa kamus data merupakan suatu dokumen yang penting dalam pengelolaan data dan pengembangan sistem informasi karena memberikan panduan yang jelas tentang struktur data serta aliran-aliran data yang terjadi dalam sistem.

Adapun simbol-simbol yang sering digunakan dalam kamus data dapat dilihat pada table berikut.

Tabel 2.5 Simbol Kamus Data

No	Simbol	Keterangan

1.	=	Disusun atau terdiri atas
2.	+	Dan
3.	()	Opsional

Lanjutan Tabel 2.5 Simbol Kamus Data

4.	[]	Memilih salah satu alternatif
5.	**	Komentar
6.	@	Identifikasi atribut kunci
7.	|	Pemisah alternatif simbol

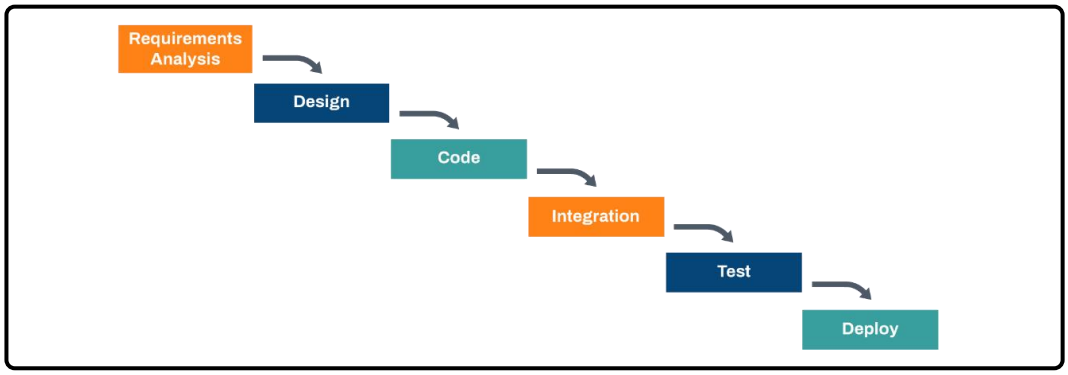
Sumber: Rosa A.S & M. Shalahuddin (2018: 74)

2.2.6 Metode Pengembangan Sistem

Metode pengembangan sistem yang penulis gunakan adalah metode *waterfall.* Menurut (Paray Theo Lonando and Arif Fadillah 2023: 54), Metode *waterfall* adalah rangkaian pada saat proses pengembangan *software* secara berurutan, dimana kemajuan akan tetap terus mengalir kebawah melalui proses perencanaan, pemodelan, implementasi dan pengujian.

Menurut (Anggraini *et al*. 2023: 88), Model *waterfall* adalah model yang paling banyak digunakan untuk tahap pengembangan. Model *waterfall* atau model air terjun sering juga disebut model sekuensial linier (*squential linear*) atau alur hidup klasik (*classic cycle*).

Pengembangan sistem dengan menggunakan metode *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari Analisis, Desain, Pengodean, Pengujian dan Tahap Pendukung (*support*). Berikut ini adalah gambar ilustrasi model *waterfall.*



Gambar 2.1 Metode *Waterfall*

Sumber: Sukamto & Shalahuddin (2016)

1) **Analisis kebutuhan perangkat lunak**

   Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan.

2) **Desain perangkat lunak**

   Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3) **Pembuatan kode program**

   Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakanimplementasi dari tahap *design* yang secara teknis nantinya dikerjakan oleh *programmer*.

4) **Pengujian Aplikasi**

   Pengujian fokus pada perangkat lunak dari segi lojik dan fungsional serta memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk

meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5) Pendukung

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.2.7 Metode Pengujian

Pengujian sistem merupakan proses pengeksekusian sistem perangkat lunak untuk menentukan apakah sistem perangkat lunak tersebut cocok dengan spesifikasi sistem dan berjalan dengan lingkungan yang diinginkan. Pengujian sistem sering diasosiasikan dengan pencarian *bug*, ketidaksempurnaan program, kesalahan pada baris program yang menyebabkan kegagalan pada eksekusi sistem perangkat lunak. Pengujian yang dilakukan pada penelitian ini adalah metode pengujian langsung yaitu dengan menggunakan pengujian *White-Box* dan *Black-Box*. Menurut (Wahyudi and Deni Ahmad 2023: 25), *Black-Box* adalah adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak.

2.3 Teori Judul

Dalam teori ini, meliputi penjelasan tentang Pengertian Platform, Inovasi, Keselamatan dan Kesehatan Kerja (K3), CV (*Commanditaire Venootschap*), serta Platform Digital untuk Meningkatkan Kinerja Karyawan: Inovasi Keselamatan dan Kesehatan Kerja (K3) di CV. UI3 Media *Advertising* & *Interior*.

2.3.1 Pengertian Platform Digital

(Alghozi *et al*. 2021: 141 – 42), menyatakan bahwa Platform Digital adalah salah satu media yang biasanya melalui internet dan fungsinya sebagai tempat atau wadah untuk berbagi informasi dalam bentuk teks, gambar, suara, video, dll.

Menurut (Eka Yuda Wibawa 2021: 78), Platform digital adalah sekumpulan *software* yang membentuk suatu sistem tertentu. *Software* ini bisa dibuka pada PC atau

sistem android.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa platform digital merupakan sebuah media melalui internet yang berfungsi sebagai tempat untuk berbagi informasi dalam berbagai bentuk. Platform digital memiliki berbagai fungsi yang melampaui sekadar berbagi informasi. Mereka menjadi pusat interaksi, kolaborasi, dan transaksi di era modern ini. Platform digital dapat mengubah cara individu dan organisasi berkomunikasi dan beroperasi, memberikan kemudahan akses, efisiensi, dan jangkauan yang lebih luas.

2.3.2 Pengertian Kinerja

Kinerja merupakan suatu fungsi dari motivasi dan kemampuan. Untuk menyelesaikan tugas atau pekerjaan seseorang sepatutnya memiliki derajat kesediaan dan tingkat kemampuan tertentu. Kesediaan dan keterampilan seseorang tidak cukup efektif untuk mengerjakan sesuatu tanpa pemahaman yang jelas tentang apa yang akan dikerjakan dan bagaimana mengerjakannya (Tesmanto and Rina 2022: 2)

Konsep kinerja merupakan singkatan dan kinetika energi kerja yang dalam bahasa inggris adalah *performance*. Kata *performance* sering di indonesiakan sebagai performa. Kinerja adalah keluaran yang dihasilkan fungsi-fungsi atau indikator-indikator suatu pekerjaan atau suatu profesi dalam waktu tertentu (Tesmanto and Rina 2022: 2).

Berdasarkan definisi diatas, kinerja dapat disimpulkan sebagai hasil dari sinergi antara motivasi, kemampuan, dan pemahaman seseorang terhadap tugas yang diberikan. Kinerja mencakup evaluasi terhadap output atau hasil dari sebuah pekerjaan atau profesi dalam jangka waktu tertentu. Dengan demikian, kinerja tidak hanya mengukur hasil akhir tetapi juga proses dan kualitas yang mendasarinya.

2.3.3 Pengertian Inovasi

Menurut (Suja'i 2021: 79), inovasi adalah ide, benda, kejadian, atau metode yang dianggap atau diamati sebagai sesuatu yang baru oleh seseorang atau sekelompok orang. Inovasi ini dapat berupa penemuan atau pengetahuan baru yang bertujuan untuk memecahkan masalah tertentu.

Menurut (Suja'i 2021), inovasi dapat menjadi positif atau negatif. Inovasi positif

didefinisikan sebagai proses membuat perubahan terhadap sesuatu dengan memperkenalkan sesuatu yang baru dan memberikan nilai. Inovasi negatif menyebabkan hilangnya kepercayaan karena tidak memiliki nilai tambah.

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa inovasi adalah suatu konsep dinamis yang mencakup pengenalan ide, benda, kejadian, atau metode baru yang bertujuan untuk memecahkan masalah tertentu. Inovasi memerlukan kreativitas dan keberanian untuk mencoba hal-hal baru.

2.3.4 Pengertian Keselamatan dan Kesehatan Kerja (K3)

Menurut (Setiyadi and Mufarokhah 2023: 207), Keselamatan dan Kesehatan Kerja (K3) meupakan sebuah program yang dibuat pekerja ataupun pengusaha sebagai upaya mengantisipasi adanya kecelakaan akibat kerja serta penyakit akibat kerja dengan cara mengetahui hal yang berpotensi menimbulkan kecelakaan serta penyakit akibat kerja serta tindakan antisipatif jika terjadi kecelakaan serta penyakit akibat kerja.

Menurut (Darmayani *et al*. 2023: 22), Keselamatan dan Kesehatan Kerja atau yang biasa disebut K3 merupakan suatu elemen dalam sistem ketenagakerjaan yang memiliki peran penting dalam keberlangsungan roda ekonomi di tempat kerja.

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa Keselamatan dan Kesehatan Kerja (K3) adalah sebuah program yang dilakukan baik oleh pekerja maupun pengusaha sebagai langkah preventif untuk menghindari terjadinya kecelakaan dan penyakit yang disebabkan oleh aktivitas kerja. Program K3 ini tidak hanya bertujuan untuk melindungi pekerja dari risiko kecelakaan dan penyakit, tetapi juga untuk memastikan kelangsungan produktivitas di tempat kerja, serta perusahaan dapat menciptakan lingkungan kerja yang lebih aman dan sehat.

2.3.5 Pengertian CV (*Commanditaire Venootschap*)

Menurut (Maharni Partyani 2019: 1), *Commanditaire Venootschap* atau seringkali disebut CV adalah suatu perseroan terbatas untuk menjalankan suatu perusahaan yang dibentuk antara satu orang atau beberapa orang persero yang secara tanggung jawab memikul tanggung jawab untuk seluruhnya pada satu pihak, dan satu orang atau lebih sebagai pelepas uang (*geldschieter*) pada pihak yang lain.

Menurut (Maharni Partyani 2019: 71), *Commanditaire Vennootschap* (CV) adalah suatu persekutuan yang didirikan oleh seorang atau beberapa orangyang mempercayakan uang atau barang kepada seorang atau beberapa orang yang menjalankan perusahaan dan

bertindak sebagai pemimpin.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa CV merupakan persekutuan di mana pemilik modal mempercayakan modalnya kepada pengelola perusahaan yang bertanggung jawab atas jalannya usaha.

2.3.6 Pengertian Platform Digital untuk Meningkatkan Kinerja Karyawan: Inovasi Keselamatan dan Kesehatan Kerja (K3) di CV. UI3 Media *Advertising* & *Interior*

Platform digital untuk meningkatkan kinerja Karyawan dengan inovasi keselamatan dan kesehatan kerja (K3) di CV. UI3 Media Advertising & Interior adalah sebuah sistem yang memfasilitasi pemantauan serta pelaporan penggunaan Alat Pelindung Diri (APD).

Platform ini dirancang untuk meningkatkan kesadaran Karyawan terhadap pentingnya K3 dengan menyediakan informasi dan pelatihan yang relevan. Melalui platform ini, Karyawan dapat dengan mudah melaporkan kondisi K3, termasuk penggunaan APD dan insiden terkait keselamatan kerja serta dapat memastikan bahwa standar keselamatan dan kesehatan kerja dipatuhi secara konsisten. Selain itu, platform ini memungkinkan manajemen untuk memantau kepatuhan terhadap protokol K3 secara *real-time*.

2.4 Teori Program

Dalam teori ini, meliputi penjelasan tentang Pengertian *Visual Studio Code*, XAMPP, *MySQL*, *Bootstrap*, PHP, HTML, CSS, dan *Javascript.*

2.4.1 Pengertian *Visual Studio Code*

Menurut (Ningsih *et al*. 2022: 95), *Visual Studio Code* adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung berbagai bahasa pemrograman dengan bantuan *plugin* yang dapat dipasang via *marketplace Visual Studio Code* seperti: C++, C#, *Python*, *Go*, *Java*, PHP, dst.

Visual Studio Code merupakan pilihan utama untuk pengembangan sistem karena ekstensibilitasnya yang luas, dukungan yang komprehensif untuk berbagai bahasa pemrograman, serta kinerja yang ringan dan responsif di berbagai platform. Dengan ribuan ekstensi yang tersedia, pengguna dapat menyesuaikan lingkungan pengembangan sesuai

kebutuhan, sementara dukungan untuk banyak bahasa pemrograman memungkinkan fleksibilitas dalam pengembangan sistem.

2.4.2 Pengertian XAMPP

Menurut (Ningsih *et al*. 2022), XAMPP merupakan aplikasi atau perangkat lunak bebas atau yang dapat berfungsi sebagai server yang berdiri sendiri (*localhost*) dan dapat mendukung banyak sistem operasi, XAMPP terdiri dari beberapa program yaitu, *Apache* HTTP Server, MySQL *database*, dan bahasa yang ditulis dapat diterjemahkan dengan bahasa pemograman PHP dan *Perl*.

Enterprise dalam (Arisantoso *et al*. 2023: 45), mengatakan bahwa XAMPP merupakan server yang paling banyak digunakan. Fiturnya lengkap, mudah digunakan programmer PHP pemula karena yang perlu gunakan hanyalah menjalankan salah satu module bernama *Apache* yang dapat memproses PHP.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa XAMPP adalah aplikasi atau perangkat lunak bebas yang berfungsi sebagai *server* lokal yang berdiri sendiri (*localhost*) dan mendukung berbagai sistem operasi untuk mengembangkan dan menguji situs *website* secara *offline*.

XAMPP merupakan *software* yang dikembangkan oleh sekelompok tim *Apache Friend* pada 2002 dan bisa didapatkan secara gratis dengan label *General Public License* (GNU). Sebagai *software open source* berbasis *web server* yang gratis, XAMPP ini memiliki berbagai program dan mendukung berbagai sistem operasi yang umum digunakan, seperti Linux, Windows, MacOS, dan Solaris. Aplikasi ini berfungsi sebagai server lokal yang sudah mencakup program *Apache, MySQL*, dan PHP.

XAMPP disebut sebagai *standalone server* atau server yang dapat berdiri sendiri sehingga memudahkan pengguna saat menjalankan proses pengeditan, desain, dan pengembangan sistem atau aplikasi. Penggunaan XAMPP dapat dikatakan mampu menghemat anggaran karena dapat menggantikan peran *web hosting* dengan cara menyimpan file *website* ke dalam *localhost* agar bisa dihubungkan melalui browser. Kepanjangan XAMPP merupakan program yang tersedia pada *software* ini.

1. **X (*Cross Platform*)**

   Merupakan kode penanda untuk *software cross platform* atau yang bisa berjalan di banyak sistem operasi, seperti Windows, Linux, dan MacOS.

2. **A (*Apache*)**

Berarti *web server* gratis (*opensource*) yang bisa dikembangkan oleh banyak orang untuk menciptakan halaman *web*. Dengan menggunakan *web server*, pengguna dapat menjalankan file yang berisi bahasa pemrograman PHP pada *localhost*.

3. **M (*MySQL/MariaDB*)**

   MySQL adalah salah satu aplikasi *database server* dengan Bahasa pemrograman *structured query language* (SQL) yang berfungsi untuk mengelola data secara terstruktur dan sistematis. Misalnya, mengolah, mengedit, dan menghapus daftar melalui *database*. *MySQL* juga dapat digunakan di *localhost* tanpa memerlukan sambungan internet. Sedangkan, *MariaDB* merupakan sistem manajemen *database* yang merupakan bagian dari pengembangan mandiri *MySQL*.

4. **P (PHP)**

   Hypertext Preprocessor (PHP) merupakan bahasa pemrograman khusus berbasis *web* untuk kebutuhan pada sisi server atau *back end* sehingga dapat digunakan untuk membuat halaman *web* menjadi lebih dinamis dengan menerapkan *server-side scripting*. Selain itu, PHP juga mendukung manajemen sistem pada *Oracle*, *PostgreSQL*, *Microsoft Access*, dan lain sebagainya.

5. **P (*Perl*)**

   Perl adalah bahasa pemrograman yang dapat digunakan untuk segala kebutuhan dan dapat berjalan di berbagai sistem operasi sehingga sangat fleksibel dan banyak digunakan. Bahasa pemrograman ini juga berfungsi sebagai penunjuk eksistensi dari PHP. *Perl* banyak digunakan untuk keperluan pengembangan aplikasi, *web server*, hingga sistem berbasis *Content Management System* (CMS), seperti *WordPress*.

Adapun komponen-komponen penting yang terdapat pada XAMPP adalah sebagai berikut.

1. ***Control Panel***, merupakan layanan yang digunakan untuk mengelola XAMPP. Melalui *Control Panel*, pengguna dapat memulai atau menghentikan layanan seperti *Apache*, *MySQL*, *FileZilla*, *Mercury*, dan *Tomcat*. Ini juga menyediakan akses cepat ke berbagai alat dan pengaturan yang diperlukan untuk mengelola server lokal.
2. **HTDocs**, merupakan komponen XAMPP yang berbentuk folder dan berfungsi sebagai tempat penyimpanan file *website*. Semua file dan folder yang ditempatkan di dalam folder HTDocs akan diakses melalui *server web Apache* yang berjalan di

XAMPP. Ini adalah root direktori untuk semua aplikasi *web* yang dikembangkan dan diuji secara lokal.

3. ***Config***, merupakan menu yang digunakan untuk melakukan konfigurasi dasar pada XAMPP. Pengguna dapat mengatur aplikasi editor teks dan browser yang akan digunakan secara *default* oleh XAMPP. *Config* juga memungkinkan pengguna untuk mengakses dan mengedit file konfigurasi utama seperti "httpd.conf" untuk Apache dan "my.ini" untuk *MySQL*.
4. ***PhpMyAdmin***, merupakan komponen XAMPP yang berfungsi untuk mengelola *database* melalui browser. Ini menyediakan antarmuka grafis yang memudahkan pengguna untuk membuat, mengedit, dan menghapus *database*, tabel, dan data tanpa perlu menulis *query SQL* secara manual. *PhpMyAdmin* juga mendukung berbagai operasi *database* lainnya seperti impor dan ekspor data.
5. **NetStat**, merupakan menu untuk memastikan *port* yang digunakan oleh XAMPP telah digunakan oleh aplikasi lain atau belum. Bila *port* standar seperti *port* 80 untuk *Apache* atau *port* 3306 untuk *MySQL* sudah digunakan oleh aplikasi lain, maka XAMPP tidak dapat berfungsi dan pengguna harus mengubah *port* tersebut melalui menu NetStat atau konfigurasi manual.

XAMPP diperlukan dalam pembuatan sistem berbasis *website* ini karena menyediakan lingkungan pengembangan lokal yang lengkap, mudah digunakan, dan fleksibel untuk mengembangkan, menguji, dan memelihara situs *web*. Dengan XAMPP, pengembang dapat memastikan bahwa *website* dapat berjalan dengan baik sebelum diunggah ke server produksi. Lingkungan pengembangan lokal ini juga memungkinkan pengembang untuk mengidentifikasi dan memperbaiki masalah dengan cepat, serta menghemat waktu selama proses pengembangan.

2.4.3 Pengertian *MySQL*

Menurut (Wahyudi and Deni Ahmad 2023), *MySQL* (*My Structure Query Language*) adalah sebuah program berbasis DOS yang bersifat *open source*. MySQL adalah produk yang berjalan pada platform baik windows maupun Linux. Selain itu, *MySQL* merupakan program pengakses *database* yang bersifat jaringan sehingga dapat digunakan untuk *multiuser* (banyak pengguna).

Menurut (Khomsi Pane *et al*. 2022), *MySQL* adalah DBMS yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi

MySQL adalah *database server* yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat Anda pakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada.

Menurut Wardhani dalam MySQL terdapat fitur-fitur yang sangat melekat pada perangkat lunak ini, yaitu:

1. *Relational Database System*.
2. Arsitektur *Client – Server*.
3. Mendukung *Sub Select*.
4. Mendukung *View*, *Stored Prosedured* (SP), *Triggers*, *Replication*, *Transaction*, dan *Foreign Key* (FK).
5. Tersedia fungsi GIS, bebas diunduh (*download*), stabil, tangguh, fleksibel, dan memiliki keamanan (security) yang baik.
6. Mengenal perintah *Structured Query Language* (SQL) standar.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa *MySQL* adalah sebuah sistem untuk menyimpan, mengelola, dan mengakses data secara efisien dalam aplikasi *web* dan berbagai sistem yang memerlukan penyimpanan data yang bersifat jaringan serta memungkinkan pengguna *multiuser*, dan dapat digunakan pengguna tanpa biaya lisensi.

MySQL digunakan dalam pembuatan sistem ini karena *MySQL* memiliki kelebihan sebagai berikut.

1. Berlisensi GPL dan dapat dijalankan di berbagai sistem operasi.
2. Dapat diintegrasikan dengan beberapa bahasa pemrograman.
3. Mendukung ODBC untuk sistem operasi windows.
4. Bisa dijalankan pada spesifikasi *hardware* yang rendah karena lebih hemat *resource memory* (dibandingkan *database* lain) sehingga mudah digunakan untuk bahan pembelajaran.
5. *MySQL* dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari 20 bahasa meskipun bahasa indonesia belum termasuk didalamnya.

2.4.4 Pengertian *Bootstrap*

Menurut (Firmansyah and Herman 2023: 64), *Bootstrap* adalah salah satu kerangka kerja dari CSS yang digunakan untuk membantu *developer* dalam mempermudah, mempercepat dan memperindah tampilan *website* tersebut.

Menurut (Suli and Nirsal 2023: 27), *Bootstrap* adalah *front-end framework* yang mengedepankan tampilan untuk *mobile device* guna mempercepat dan mempermudah pengembangan *website*.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa *Bootstrap* adalah sebuah kerangka kerja (*framework*) *front-end* CSS yang dirancang untuk memudahkan dan mempercepat pengembangan tampilan *website*. *Bootstrap* membantu pengembang dalam menciptakan desain yang menarik dan responsif secara efisien.

2.4.5 Pengertian PHP

Menurut (Ningsih *et al*. 2022: 32), *Hypertext Preprocessor* (PHP) merupakan *script* untuk pemrograman *script web server-side*, *script* yang membuat dokumen HTML secara *on the fly* atau dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks.

Menurut (Firmansyah and Herman 2023), PHP atau singkatan dari *Hypertext Preprocessor* merupakan salah satu bahasa pemrograman *open source* yang digunakan untuk komunikasi ke sisi server yang kemudian transfer hasilnya ke *client* yang melakukan permintaan.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa PHP adalah sebuah bahasa pemrograman skrip yang digunakan secara *server-side* untuk menghasilkan dokumen HTML secara dinamis.

Penggunaan PHP memungkinkan *web* dapat dibuat dinamis sehingga *maintenance web* menjadi lebih mudah dan efisien. PHP memiliki banyak kelebihan yang tidak dimiliki oleh bahasa pemrograman lainnya. PHP difokuskan pada pembuatan *script server-side* yang bisa melakukan hal seperti mengumpulkan data dari *form* dan menghasilkan isi halaman *web* dinamis, bahkan lebih daripada kemampuan CGI. PHP tidak terbatas pada hasil keluaran HTML, namun PHP juga

memiliki kemampuan untuk mengolah gambar, file PDF, dan *movie flash*. PHP juga dapat menghasilkan teks seperti XHTML dan file XML lainnya. Salah satu fitur yang dapat diandalkan oleh PHP adalah dukungan terhadap *database*, salah satunya adalah *MySQL*.

2.4.6 Pengertian HTML

Menurut (Noviantoro *et al*. 2022: 91), HTML (*Hyper Text Mark Up Language*) berfungsi untuk mempublikasi dokumen online, memungkinkan pengguna untuk mengakses dan berinteraksi dengan konten melalui peramban *web*. HTML menyediakan kerangka kerja dasar untuk mengatur teks, gambar, dan elemen lainnya dalam sebuah halaman *web*, yang kemudian dapat ditingkatkan dengan penggunaan CSS (*Cascading Style Sheets*) untuk *styling* dan *JavaScript* untuk interaktivitas.

Menurut (Firmansyah and Herman 2023: 364), HTML atau singkatan dari *HyperText Markup Language*) merupakan salah satu bahasa *markup* standar yang digunakan untuk membuat sebuah halaman *website*.

HTML berperan sebagai fondasi dari struktur dan tampilan halaman *website*, memungkinkan interaksi antara pengguna dan konten *website* melalui berbagai media. Setiap elemen dalam HTML memiliki peran tertentu, seperti “<header>” untuk *header* halaman, “<footer>” untuk *footer*, “<article>” untuk konten artikel, dan “<section>” untuk bagian spesifik dari dokumen. Penggunaan elemen-elemen ini membantu dalam pengaturan konten yang lebih terstruktur.

Dengan HTML, pengembang *web* dapat membuat halaman *web* yang informatif dan interaktif. HTML juga mendukung penyertaan multimedia seperti video dan audio, serta integrasi dengan teknologi *web* modern seperti API (*Application Programming Interface*) dan SVG (*Scalable Vector Graphics*).

Selain itu, HTML memungkinkan aksesibilitas yang lebih baik, memastikan bahwa konten dapat diakses oleh berbagai pengguna, termasuk mereka yang menggunakan alat bantu seperti *screen readers*. Ini membuat HTML menjadi komponen esensial dalam pembuatan *website* yang responsif dan ramah pengguna.

Secara keseluruhan, HTML adalah bahasa markup yang mendasari struktur dan tampilan halaman *website*, menyediakan fondasi yang kokoh untuk pengembangan *web* yang modern dan fungsional. HTML memungkinkan pengembang untuk menciptakan pengalaman pengguna yang dinamis dan beragam, memfasilitasi interaksi yang kaya antara pengguna dan konten *website* melalui berbagai media.

2.4.7 Pengertian CSS

Menurut (Firmansyah and Herman 2023: 364), CSS atau singkatan dari *Cascasding Style Sheet* merupakan salah satu bahasa *stylesheet* yang digunakan untuk memperindah atau mempercantik sebuah tampilan *website*. Biasanya CSS membantu HTML dalam proses memperindah sebuah *website*.

Menurut (Arisantoso *et al*. 2023: 85), CSS adalah seperangkat alat untuk membuat dan memodifikasi jenis huruf, warna, bentuk, dan elemen terkait gaya lainnya yang tidak dapat dilakukan menggunakan elemen HTML. CSS merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur. CSS bukan sebuah bahasa pemograman, melainkan *styles* seperti dalam aplikasi Microsoft Word yang dapat mengatur beberapa *style*.

Berdasarkan beberapa defisini diatas, dapat disimpulkan bahwa CSS adalah bahasa *stylesheet* yang digunakan untuk meningkatkan estetika dan tata letak sebuah halaman *website*. CSS memungkinkan pengembang web untuk mendefinisikan dan mengontrol elemen-elemen visual. Dengan menggunakan CSS, tampilan sebuah website menjadi lebih terstruktur, menarik, dan konsisten.

2.4.8 Pengertian *Javascript*

Menurut (Daniel Rudjiono and Heru Saputro 2021: 61), *Javascript* adalah bahasa pemrograman atau kode *script* yang diletakan bersama kode HTML ataupun secara terpisah dari HTML yang digunakan untuk membuat tampilan *website* lebih dinamis.

Menurut Arisantoso *et al*., (2023: 59), *Javascript* adalah bahasa pemrograman tingkat tinggi yang pertama kali dibuat untuk membuat *website* lebih hidup. *Javascript*, bersama dengan HTML dan CSS, adalah bahasa pemrograman yang paling umum digunakan untuk membuat aplikasi berbasis *website*. Bahasa ini dapat membuat *website*

lebih interaktif.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa *Javascript* adalah sebuah bahasa pemrograman yang berperan penting dalam pengembangan aplikasi *website*. Dalam konteks pengembangan *website*, *Javascript* bekerja bersama dengan HTML dan CSS untuk menciptakan interaktivitas dan dinamika yang meningkatkan pengalaman pengguna secara keseluruhan.