

BAB II

LANDASAN TEORI

2.1 Pengertian *Website*

Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara dan atau gabungan dari semuanya, baik bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaring-jaringan halaman. Website merupakan kumpulan halaman yang menampilkan informasi data teks, data gambar, data animasi, suara, video, atau gabungan dari semuanya. Website merupakan teknologi komunikasi yang saat ini mengalami perkembangan sangat pesat. Sebagian besar aspek kehidupan manusia telah dibantu dengan adanya komputer. Salah satu manfaat komputer dalam kehidupan manusia adalah mengelola, mengakses dan menyebarkan informasi (Bangun et al., 2020).

Website adalah sesuatu yang dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar, data animasi, suara, video dan gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangun yang saling terkait, dimana masing-masing dihubungkan dengan jaringan-jaringan halaman. World wide web atau sering dikenal sebagai web adalah suatu layanan sajian informasi yang menggunakan konsep hyperlink (tautan), yang memudahkan surfer (sebutan para pemakai komputer yang melakukan browsing atau penelusuran informasi melalui internet).

Keistimewaan inilah yang telah menjadikan web sebagai service yang paling cepat pertumbuhannya (Irmayani & Munandar, 2020).

World Wide Web (www), lebih dikenal dengan *web* yang merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke *internet* dengan fasilitas *hypertext* untuk menampilkan data berupa *text*, gambar, suara animasi dan data multimedia lainnya. Sehingga *web* pada awalnya adalah ruang informasi dalam *internet* dengan menggunakan teknologi *hypertext* pemakai dituntun menemukan informasi dengan mengikuti link yang disediakan dalam dokumen *web* yang ditampilkan dalam *web browser*.

2.1.1 Jenis Website

Jenis-jenis *website* terdiri dari 3 macam yaitu :

1. Website statis

Website statis adalah jenis *website* yang kontennya tetap dan tidak berubah-ubah, kecuali dilakukan pengeditan langsung oleh pengelola *website* melalui kode sumber. Setiap halaman pada *website statis* ditulis secara manual dalam *HTML* dan *CSS*, serta tidak bergantung pada *database* atau skrip pemrosesan di sisi server. *Website statis* merupakan website yang dibangun dengan teknologi *HTML* dan *CSS*, yang kontennya tidak berubah secara otomatis dan hanya bisa diperbarui melalui modifikasi langsung pada file sumber.

2. Website Dinamis

Web dinamis adalah *web* yang menampilkan informasi serta dapat berinteraksi dengan pengguna.. *Web* yang dinamis memungkinkan pengguna untuk berinteraksi menggunakan *form* sehingga dapat memolah informasi yang ditampilkan. *Web dinamis* bersifat *interaktif*, tidak kaku dan terlihat lebih indah.

Sehingga dapat disimpulkan bahwa kelebihan dari aplikasi *web* adalah dapat diakses kapan pun dan darimana pun selama ada *internet*. Dan dapat diakses hanya dengan menggunakan *web browser* perlu menginstal, tidak perlu mnginstal aplikasi *web* itu sendiri adalah antarmuka yang dapat dibuat terbatas sesuai spesifikasi standar untuk membuat dokumen. *Web* dan keterbatasan kemampuan *web browser* untuk menampilkannya. Dan terbatasnya kecepatan internet mungkin membuat respon aplikasi menjadi lambat.

3. **Website Interaktif**

Website interaktif adalah jenis *website* yang memungkinkan pengguna untuk berinteraksi secara langsung dengan sistem melalui berbagai fitur seperti *form input*, *upload file*, pencarian data, dan respon dinamis dari sistem.

Website interaktif merupakan aplikasi *web* yang menyediakan interaksi dua arah antara pengguna dan sistem melalui elemen-elemen dinamis seperti formulir, animasi, dan database. Contoh: situs media social, *forum online*, *blog*, dan lain-lain.

2.1.2 **Manfaat Website**

Berikut beberapa manfaat *website* diantaranya adalah

1. *Website* dapat menjadi sarana informasi yang cepat dan mudah.
2. *Website* dapat menjadi sarana edukasi masyarakat.
3. *Website* dapat menjadi sarana pemasaran dan promosi bisnis yang efektif dengan jangkauan terluas.

2.1.3 **Tahapan Membangun Website**

Berikut ini merupakan tahapan dalam membangun sebuah situs *website* yang biasa diterapkan:

1. Rekayasa dan pemodelan sistem informasi

Pada tahap ini dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset kebutuhan *software* tersebut. Pandangan sistem ini penting ketika *software* harus berhubungan dengan elemen-elemen lain seperti *software*, manusia, dan *database*. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak.

2. Planning (Perencanaan)

Pada tahap ini yaitu untuk menentukan tujuan dari *software* yang akan dibuat, melakukan analisis kebutuhan dan pengumpulan data yang diperlukan. Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada kebutuhan *software*. Untuk memahami sifat program yang dibangun, analisis harus memahami domain informasi, tingkah laku, unjuk kerja, dan *interface* yang diperlukan.

3. Designing (desain)

Desain *software* sebenarnya merupakan proses multi langkah yang berfokus pada empat atribut yang berbeda, yaitu struktur data, arsitektur *software*, representasi *interface*, dan detail (*algoritma*) *procedural*. Jadi pada tahap ini proses mendesain harus berdasarkan kebutuhan sehingga sesuai dengan yang diharapkan sebelum memulai pengkodean.

4. Scripting (pemograman)

Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Dalam tahap ini adanya pembuatan kode. Jika desain dilakukan secara lengkap, pembuatan kode dapat diselesaikan secara mekanis.

5. *Testing* (Pengujian)

Proses pengujian berfokus pada *logika internal software* untuk memastikan bahwa semua pernyataan sudah diuji, dan pada fungsi eksternal yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa dengan input yang terbatas akan didapatkan hasil actual yang sesuai dengan yang dibutuhkan.

6. *Maintenance* (pemeliharaan)

Software mungkin akan mengalami perubahan setelah diserahkan kepada pelanggan. Perubahan bisa terjadi karena kesalahan-kesalahan tertentu, karena *software* harus diubah untuk mengakomodasikan perubahan-perubahan di dalam lingkungan eksternalnya, atau karena pelanggan perlu melakukan pengembangan fungsional atau unjuk kerja. Pemeliharaan *software* mengaplikasikan lagi setiap fase program sebelumnya dan tidak dilakukan dengan membuat yang baru.

2.2 Sistem Informasi

Sistem informasi merupakan suatu rangkaian prosedur yang digunakan untuk mengumpulkan, menyimpan, memproses, dan menganalisis data besar sehingga menghasilkan informasi bermakna bagi pengambilan keputusan dan mendukung interaksi antar elemen organisasi. Sistem Informasi merupakan penyajian informasi yang dikomputerisasi dengan baik dan benar. Informasi ibarat darah yang mengalir didalam tubuh manusia, maksud dari kalimat tersebut yaitu bahwa informasi sangat penting pada suatu organisasi(Yanris & Sihombing, 2020).

Definisi sistem informasi adalah sebuah sistem yang terdiri atas rangkaian subsistem informasi terhadap pengolahan data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan, selain itu definisi dari "sistem Informasi adalah kumpulan dari sub-sub sistem baik fisik maupun non fisik yang saling berhubungan satu sama dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data mencapai satu tujuan yaitu mengolah data menjadi informasi yang berguna(Purnama et al., 2020). Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu.

Sementara itu, dari perspektif sosioteknis, sebuah sistem informasi melibatkan aspek **struktur**—meliputi tugas, peran, dan teknologi—serta aspek **proses** dalam pengelolaan informasi (Wikipedia, *Information system*, 2025). **Data** adalah kenyataan yang menggambarkan suatu kejadian-kejadian dalam kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat yang tertentu. Untuk memperoleh informasi yang berguna, tindakan yang pertama adalah mengumpulkan data, kemudian mengolahnya sehingga menjadi informasi. Dari data-data tersebut informasi yang didapatkan lebih terarah dan penting karena telah dilalui berbagai tahap dalam pengolahannya diantaranya yaitu pengumpulan data, data apa yang terkumpul dan menemukan informasi yang diperlukan.

2.2.2 Karakteristik Sistem

Sebuah sistem mempunyai karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bias dikatakan sebagai suatu sistem.

Karakteristik- karakteristik tersebut adalah:

1. **Komponen Sistem (*Components*)**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi dan bekerjasama membentuk satu kesatuan. Komponen- komponen sistem tersebut dapat berupa subsistem atau bagian-bagian dari sistem yang mempunyai sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. **Batasan sistem (*Boundary*)**

Batasan sistem merupakan daerah yang membatasi antara sistem yang dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini menunjukkan ruang lingkup dari sistem itu sendiri.

3. **Lingkungan luar sistem (*Environtment*)**

Lingkungan luar dari sistem merupakan apapun yang ada di luar lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut.

4. **Penghubung sistem (*Interface*)**

Penghubung sistem atau interface merupakan media yang menghubungkan sistem dengan subsistem yang lainnya.

5. **Masukan sistem (*Input*)**

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan sistem dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). *Maintenance input* merupakan energy yang dimasukkan agar sistem tersebut dapat

beroperasi. Sedangkan, signal input adalah energy yang diproses untuk menghasilkan keluaran.

6. Keluaran sistem (*Output*)

Keluaran sistem adalah hasil energy yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan yang berguna bagi subsistem yang lain.

7. Pengolah sistem (*Proses*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti. Hal ini karena sasaran sangat berguna untuk menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya(Irmayani & Zega, 2022).

2.3 Perancangan Sistem

Perancangan sistem adalah merancang atau mendesain suatu system yang baik yang isinya adalah langkah-langkah operasi dalam proses pengolahan data dan proses prosedur-prosedur untuk mendukung operasi sistem. Tujuan dari perancangan sistem adalah untuk memenuhi kebutuhan para pemakai sistem serta memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada programmer dan ahli-ahli yang terlibat didalam. Tujuan pada tahap perancangan sistem adalah untuk memenuhi kebutuhan kepada para pemakai, serta memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada

pemrogram komputer dan ahli teknik lain yang terlibat (Irmayani, Nasution, et al., 2021).

2.3.1 Konsep Dasar Sistem

Suatu sistem terdiri dari sistem-sistem bagian (*subsystems*). Masing-masing subsistem terdiri dari subsistem-subsistem yang lebih kecil lagi atau terdiri dari komponen-komponen. Interaksi dari subsistem-subsistem sedemikian rupa, sehingga dicapai suatu kesatuan yang terpadu atau terintegrasi (*integrated*). Sistem adalah suatu kumpulan objek-objek yang saling berhubungan dan berinteraksi satu sama lain serta menjadi satu kesatuan untuk mencapai suatu tujuan. Karakteristik sistem terdiri dari komponen sistem, batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem, pengolahan sistem, dan sasaran sistem (Purnama et al., 2021).

Keterpaduan sistem ini memungkinkan terciptanya kerjasama untuk menghasilkan informasi yang cepat, tepat dan akurat. Pendekatan sistem yang lebih menekankan pada prosedur, mendefinisikan sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan. Berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.

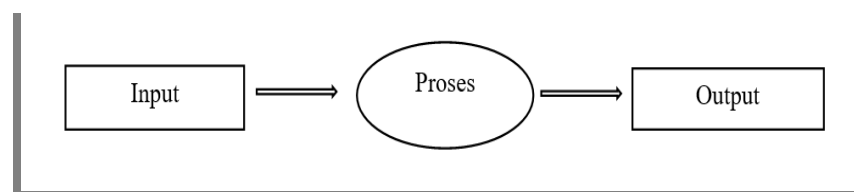
Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan operasi di dalam sistem. Pendekatan sistem yang lebih menekankan pada prosedur, mendefinisikan sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan. Berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.

Definisi prosedur sebagai berikut: Suatu prosedur adalah urutan-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang

harus dikerjakan, siapa (*who*) yang mengerjakannya, kapan (*when*) dikerjakan dan bagaimana (*how*) mengerjakannya Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai berikut ini: Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencaai suatu tujuan tertentu.

2.3.2. Bentuk Umum Sistem

Bentuk umum sistem dari suatu sistem terdiri atas masukan (*Input*), proses dan keluaran (*Output*), dalam bentuk umum sistem ini terdapat satu atau lebih masukan yang akan diproses dan akan menghasilkan suatu keluaran.



Gambar 2. 1 Bentuk Umum Sistem

2.3.3 Elemen Sistem

Semua sistem meliputi tiga elemen utama yaitu input, transformasi dan output. Sebagian sistem dapat mengendalikan operasi mereka sendiri yang disebut sebagai sistem lingkaran tertutup (*closed-loop system*). Sistem lingkaran tertutup mencakup suatu mekanisme kontrol, tujuan dan lingkaran umpan balik (*feedback loop*) disamping tiga elemen utama. Sistem yang tidak memiliki kemampuan pengendalian disebut sistem lingkaran terbuka (*open-loop system*), dalam arti mereka berhubungan dengan lingkungan mereka. Perusahaan adalah suatu contoh sistem terbuka dan sistem lingkaran tertutup.

2.3.4 Klasifikasi sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya, Menurut.

1. Sistem diklasifikasikan sebagai sistem abstrak dan sistem fisik. Sistem abstrak adalah sistem yang berupa pemikiran atau ide - ide yang tidak tampak secara fisik. Sistem fisik adalah sistem yang ada secara fisik.
2. Sistem diklasifikasikan sebagai sistem alamiah dan sistem buatan manusia. Sistem alamiah adalah sistem yang terjadi karena proses alam tidak dibuat oleh manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia.
3. Sistem diklasifikasikan sebagai sistem tertentu dan sistem tak tentu. Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung probabilitas.
4. Sistem diklasifikasikan sebagai sistem tertutup dan sistem terbuka. Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem terbuka adalah sistem yang hubungan dan terpengaruh dengan lingkungan luarnya.

2.4 Bahasa Pemrograman

Bahasa pemrograman adalah sistem notasi atau sintaksis yang digunakan oleh programmer untuk menuliskan instruksi yang dapat dieksekusi oleh komputer. Bahasa ini menjadi alat komunikasi antara manusia dan mesin, sehingga memungkinkan komputer menjalankan tugas-tugas tertentu seperti

pengolahan data, perhitungan logika, manipulasi file, dan pengembangan aplikasi. Menurut (Sari & Widodo, 2021), bahasa pemrograman adalah serangkaian aturan sintaks dan semantik yang dirancang untuk mengontrol perilaku mesin komputer melalui instruksi logis yang dapat dipahami dan diterjemahkan oleh *compiler* atau *interpreter*. Bahasa pemrograman dapat diklasifikasikan menjadi beberapa jenis berdasarkan level abstraksi dan paradigma pemrogramannya,

2.4.1 Fungsi Bahasa Pemrograman

1. Mengontrol komputer Menjalankan *instruksi logis* dan *matematis*.
2. Membangun perangkat lunak Aplikasi *web*, *mobile*, *desktop*, dan *sistem tertanam*.
3. Otomatisasi tugas Mengurangi pekerjaan manual dan meningkatkan efisiensi.
4. Simulasi dan pemodelan Dalam dunia akademik dan industri.
5. Interaksi manusia-mesin Menyediakan antarmuka pengguna yang dapat digunakan.

2.4.2 HTML (*HyperText Markup Language*)

HTML (HyperText Markup Language) merupakan bahasa markah standar yang digunakan untuk membuat dan menyusun halaman *web*. *HTML* berfungsi sebagai kerangka dasar dalam pengembangan halaman *web* dengan mendefinisikan struktur konten seperti teks, gambar, tautan, tabel, dan elemen-elemen lainnya yang ditampilkan dalam *browser*.

HTML terdiri dari elemen-elemen (*tags*) yang digunakan untuk mendefinisikan berbagai bagian dari halaman *web*. *Elemen HTML* dibuka dengan

tanda kurung sudut (< >) dan sebagian besar memiliki tag penutup (</ >). Contoh elemen dasar *HTML* adalah <html>, <head>, <body>, <h1> hingga <h6>, <p>, <a>, dan .



Gambar 2.2 *HTML*

HTML tidak bersifat *case-sensitive*, namun penulisan tag dalam huruf kecil lebih disarankan agar konsisten dan sesuai standar W3C. Dalam praktik pengembangan *web* modern, *HTML* digunakan sebagai pondasi utama, sedangkan aspek desain diserahkan pada *CSS*, dan interaktivitas pada *JavaScript*. Kelebihan *HTML* yaitu:

1. Mudah Dipelajari dan Digunakan

HTML memiliki sintaks yang sederhana dan mudah dimengerti oleh pemula. Bahkan tanpa latar belakang pemrograman, seseorang dapat mulai membuat halaman *web* dasar hanya dengan teks editor.

2. Didukung oleh Semua *Browser*

Semua *browser* modern (*Google Chrome*, *Firefox*, *Safari*, *Edge*, dll.) mendukung *HTML*. Ini menjamin bahwa halaman yang dibuat dengan *HTML* dapat ditampilkan di berbagai platform dan perangkat.

3. Gratis dan Open Source

HTML adalah teknologi terbuka dan gratis digunakan. Tidak ada biaya lisensi atau perangkat khusus yang dibutuhkan.

4. Struktur Standar Web

HTML adalah standar utama untuk membangun struktur halaman *web*. Bahasa ini menyediakan kerangka dasar untuk teks, gambar, tautan, video, dan *elemen interaktif* lainnya.

5. Kombinasi Kuat dengan CSS & JavaScript

HTML bekerja sangat baik bila dipadukan dengan *CSS* untuk desain tampilan dan *JavaScript* untuk interaktivitas. Ketiganya menjadi "tulang punggung" pengembangan *web* modern.

6. Ringan dan Cepat Dimuat

File *HTML* berukuran kecil dan cepat dimuat oleh *browser*, sehingga memberikan performa yang baik terutama pada koneksi internet lambat.

7. Kompatibel dengan Perangkat Mobile

HTML5 mendukung desain responsif dan fitur-fitur *mobile-friendly*, sehingga halaman *web* dapat ditampilkan dengan baik di berbagai ukuran layar.

8. Fitur Semantik (*HTML5*)

HTML5 menghadirkan elemen semantik seperti `<header>`, `<footer>`, `<article>`, `<section>` yang membantu strukturisasi konten lebih bermakna dan meningkatkan aksesibilitas serta *SEO*.

9. Mendukung Multimedia Langsung

HTML5 memungkinkan integrasi langsung elemen multimedia seperti video dan audio tanpa *plugin* tambahan (seperti *Flash*).

10. Kompatibel dengan Teknologi Baru

HTML dapat digunakan bersama teknologi *web* lainnya seperti *WebAssembly*, *Progressive Web Apps (PWA)*, *WebGL*, dan lainnya.

2.4.3 CSS (*Cascading Style Sheets*)

CSS (Cascading Style Sheets) adalah bahasa pemrograman desain (*style language*) yang digunakan untuk mengatur tampilan visual dan tata letak halaman *web*. *CSS* berfungsi memisahkan konten (yang ditulis dalam *HTML*) dari aspek presentasi seperti warna, *font*, *margin*, *padding*, dan posisi elemen.

CSS merupakan teknologi pelengkap *HTML* yang sangat penting dalam pengembangan *web* modern karena memudahkan pengaturan tampilan antar halaman secara konsisten dan efisien. Dengan *CSS*, pengembang dapat membuat desain yang lebih menarik dan responsive.



Gambar 2.3 CSS

Jenis Penulisan CSS

1. *Inline CSS* Ditulis langsung dalam atribut *HTML* (*style*="...").
2. *Internal CSS* Ditulis dalam tag `<style>` di bagian `<head>`.

3. *External CSS* Ditulis dalam file terpisah dengan ekstensi *.css* dan dihubungkan dengan tag *<link>*.

Penggunaan *External CSS* dinilai lebih efisien dalam proyek besar karena file *CSS* dapat digunakan bersama oleh banyak halaman *HTML*. Menurut (Handayani et al.2023), Penggunaan *CSS* eksternal sangat dianjurkan dalam pengembangan *web* skala besar karena meningkatkan efisiensi, konsistensi desain, serta memudahkan pemeliharaan kode.

Perkembangan *CSS* Versi terbaru, *CSS3*, memperkenalkan banyak fitur baru seperti animasi, transformasi *2D* dan *3D*, efek transisi, serta *layout modern* seperti *Flexbox* dan *Grid*. *CSS3* juga mendukung media *queries* yang memungkinkan tampilan *web* menyesuaikan secara otomatis terhadap ukuran layar perangkat (*responsif*). *CSS3* telah meningkatkan kemampuan desain antarmuka pengguna dengan menyediakan berbagai fitur modern seperti animasi, *shadow*, dan *layout* responsif yang sangat dibutuhkan dalam pengembangan *web* saat ini.

Manfaat *CSS* dalam Pengembangan *Web*:

1. Memisahkan desain dari struktur
2. Meningkatkan kecepatan pemuatan halaman
3. Memudahkan pengelolaan dan perubahan tampilan
4. Mendukung desain yang konsisten di seluruh halaman
5. Membuat *website* lebih responsif dan ramah pengguna

2.4.4 *Python*

Python adalah salah satu bahasa pemrograman tingkat tinggi yang bersifat *general-purpose*, *open-source*, dan mudah dibaca. Bahasa ini mendukung

berbagai paradigma pemrograman, seperti prosedural, berorientasi objek, dan fungsional. *Python* banyak digunakan dalam berbagai bidang, seperti pengembangan *web*, *data science*, *machine learning*, automasi sistem, dan pengolahan data.

Menurut Fitria dan Nugraha (2022), *Python* menjadi populer karena sintaksisnya yang sederhana dan efisien, serta memiliki dokumentasi dan komunitas yang luas. Selain itu, *Python* menyediakan berbagai pustaka (*library*) dan *framework* yang mendukung pengembangan aplikasi modern, seperti *Flask*, *Django*, *NumPy*, *Pandas*, dan *TensorFlow*.

Python dirancang untuk meningkatkan produktivitas programmer serta memudahkan proses *debugging* dan pemeliharaan kode. Hal ini sejalan dengan pendapat Hidayat dan Firmansyah (2020) yang menyatakan bahwa *Python* mendukung pengembangan perangkat lunak dengan struktur kode yang bersih, modular, dan mudah dikembangkan.



Gambar 2.4 Python

2.4.5 *Flask* Sebagai Web Server

Flask adalah sebuah *microframework* berbasis bahasa pemrograman *Python* yang digunakan untuk membangun aplikasi *web*. *Framework* ini ringan,

fleksibel, dan mudah dipahami, sehingga sangat cocok digunakan oleh pemula maupun pengembang profesional yang menginginkan kendali penuh terhadap aplikasi *web* yang dibangun. *Flask* dikembangkan berdasarkan *Werkzeug* (sebagai *web server gateway*) dan *Jinja2* (sebagai *template engine*).

Flask merupakan *framework* minimalis namun sangat powerfull, karena memberikan kebebasan kepada pengembang untuk menentukan struktur dan komponen yang digunakan, tanpa dibatasi oleh arsitektur tertentu seperti pada *framework* besar. Hal ini memungkinkan pengembang untuk membangun aplikasi *web* yang ringan namun tetap mampu menangani berbagai fitur fungsional secara efisien.

Flask mendukung penggunaan *blueprint*, *routing*, *template rendering*, serta dapat dengan mudah dikombinasikan dengan basis data seperti *SQLite*, *MySQL*, atau *PostgreSQL*. *Framework* ini juga mendukung ekstensi pihak ketiga seperti *Flask-Login* untuk autentikasi, *Flask-WTF* untuk *validasi form*, dan *Flask-RESTful* untuk pengembangan *API* berbasis *REST*.

Sebagaimana dijelaskan oleh (Nugroho&Setiawan, 2020), *Flask* memungkinkan pengembangan *web* secara modular dan cepat, karena tidak memerlukan banyak konfigurasi awal. Mereka menambahkan bahwa *Flask* sangat cocok digunakan untuk pengembangan sistem informasi skala kecil hingga menengah, serta *prototipe* aplikasi *web* yang cepat.

Flask juga banyak digunakan dalam lingkungan akademik dan penelitian karena kesederhanaannya dan dukungan pustaka *Python* yang luas. *Framework* ini dapat dijalankan di berbagai server dan sistem operasi, serta *kompatibel* dengan berbagai teknologi *frontend* seperti *HTML*, *CSS*, dan *JavaScript*.



Gambar 2.5 flask

2.4.6 HTTP

Menurut *Chaffey* bahwa server *HTTP* umumnya digunakan untuk melayani dokumen *hypertext*, karena *HTTP* adalah protokol dengan *overhead* yang sangat rendah sehingga pada kenyataanya, navigasi informasi dapat ditambahkan langsung ke dalam dokumen dan dengan demikian protokolnya sendiri tidak harus mendukung navigasi secara penuh.

2.5 Use case diagram

Diagram yang menggambarkan actor, use case dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah use case digambarkan sebagai elips horizontal dalam suatu diagram UML use case(Irmayani, Siregar, et al., 2021). *Use Case Diagram* merupakan serangkaian yang saling terkait dan membentuk sebuah sistem secara sistematis yang dikontrol oleh seorang actor. *Use Case Diagram* menggambarkan hubungan atau interaksi

tentang fungsi-fungsi yang ada dalam sistem.(Martasya Berkat Silaen et al., 2024)

Berikut adalah *use case diagram* dari penelitian ini :

Tabel 2.1 Simbol pada *Use Case Diagram*

Simbol	Nama	Keterangan
	Actor (Aktor)	Menunjukkan pihak yang berinteraksi dengan sistem (misal: pengguna, admin).
	Use Case (Kasus Penggunaan)	Menunjukkan fungsi atau layanan yang disediakan oleh sistem.
	Association (Garis Asosiasi)	Menyatakan hubungan antara aktor dengan use case yang digunakan.
 atau 	Include	Menyatakan bahwa satu use case selalu menyertakan use case lain.
 atau 	Extend	Menyatakan bahwa satu use case bisa memperluas use case lain jika diperlukan.
	Boundary (Sistem)	Digambarkan sebagai kotak besar, menunjukkan batas dari sistem.

Use Case Diagram sangat berguna pada tahap awal perancangan perangkat lunak karena:

1. Menunjukkan apa yang dilakukan sistem (bukan bagaimana caranya).
2. Mempermudah komunikasi antara *developer* dan pengguna.
3. Menjadi acuan dalam membuat desain dan pengujian.

2.5.1 *Sequence Diagram*

Sequence Diagram menggambarkan interaksi antar objek dalam sistem dalam bentuk urutan pesan yang dikirimkan sesuai dengan skenario yang terjadi di sistem tersebut(Susanto & Pratama,2020). *Sequence Diagram* merupakan

penggambaran interaksi antar objek didalam maupun disekitar sistem yang kegunaannya untuk menunjukkan rangkaian pesan yang dikirim melalui beberapa objek, dan juga berinteraksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem *Sequence Diagram Login*(Listia et al., 2020).

Tabel 2.2 Simbol pada *Sequence Diagram*

Simbol	Nama	Keterangan
	Lifeline (Objek/Aktor)	Menunjukkan entitas (aktor atau objek) yang terlibat dalam skenario. Ditandai dengan kotak kecil di atas dan garis vertikal ke bawah.
	Message (Pesan)	Panah horizontal dari satu lifeline ke lifeline lain. Menunjukkan komunikasi atau pemanggilan metode antar objek.
	Self-Message	Panah dari dan ke objek yang sama, menunjukkan objek memanggil dirinya sendiri.
	Activation (Kotak Aktivasi)	Kotak tipis vertikal di atas lifeline saat objek sedang melakukan aktivitas atau proses.
	X (Penghentian/Destruction)	Menunjukkan objek dihentikan (destroyed), ditandai dengan simbol X di bagian bawah lifeline.
[]	Syarat (Condition)	Digunakan dalam pesan bersyarat (opsional), ditulis di atas garis pesan, contoh: [valid] atau [gagal login].
	Loop (Perulangan)	Dinyatakan dalam kotak persegi panjang dengan tulisan loop dan batas perulangan, misalnya: i=1 to n.

2.5.2 Activity Diagram

Activity Diagram adalah salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk memodelkan alur kerja (*workflow*) atau aktivitas dalam suatu sistem, baik dari sisi pengguna maupun sistem itu sendiri. Diagram ini sangat efektif untuk menggambarkan urutan aktivitas, percabangan, dan sinkronisasi proses dalam sistem.

Tabel 2.3 Simbol pada *Activity Diagram*

Simbol	Nama	Fungsi / Keterangan
	Initial Node (Titik Awal)	Titik awal dari proses aktivitas. Ditandai dengan bulatan hitam.
	Activity (Aktivitas)	Menunjukkan suatu proses atau langkah kerja. Digambarkan dengan persegi panjang dengan sudut tumpul.
	Decision (Keputusan)	Menandakan adanya percabangan (if/else, validasi). Digambarkan sebagai belah ketupat (diamond).
	Final Node (Titik Akhir)	Titik akhir dari proses. Digambarkan sebagai bulatan dengan garis luar tebal.
	Control Flow (Alur Proses)	Menunjukkan urutan antar aktivitas, digambarkan dengan panah.
	Fork / Join Node	Menyatakan proses paralel atau sinkronisasi proses. Digambarkan dengan garis tebal horizontal/vertikal.

Komponen Activity Diagram

1. Initial Node

Initial Node merupakan simbol yang menandai titik awal dimulainya seluruh proses dalam sebuah *activity diagram*. Simbol ini digambarkan dengan sebuah lingkaran hitam penuh (*solid black circle*). *Initial Node* berfungsi sebagai titik inisiasi dari alur kerja sistem, di mana aktivitas pertama akan dijalankan setelah titik ini. Pada umumnya, dalam satu diagram hanya terdapat satu *Initial Node* yang merepresentasikan dimulainya alur logika dalam sistem atau proses tertentu.

2. Activity/Action Node

Activity atau *Action Node* menggambarkan suatu aktivitas atau langkah spesifik yang dilakukan dalam proses sistem. Setiap *Action Node* menunjukkan satu aksi atau tugas yang dapat dilakukan oleh pengguna, sistem, atau aktor lain dalam alur kerja. Simbol dari *node* ini berbentuk persegi panjang dengan sudut membulat. *Action Node* merupakan komponen utama dalam *activity diagram*

karena mencerminkan aktivitas-aktivitas yang saling terhubung satu sama lain dan dijalankan secara berurutan atau berdasarkan kondisi tertentu.

3. *Decision Node*

Decision Node adalah titik percabangan alur berdasarkan kondisi atau keputusan logis tertentu. Simbolnya berbentuk belah ketupat (seperti simbol pada *flowchart*). Pada titik ini, sistem akan melakukan evaluasi terhadap suatu kondisi, kemudian memilih salah satu jalur dari beberapa kemungkinan jalur yang tersedia. *Decision Node* selalu memiliki satu alur masuk dan dua atau lebih alur keluar, yang masing-masing disertai kondisi logis yang menentukan arah jalur tersebut.

4. *Merge Node*

Merge Node digunakan untuk menggabungkan beberapa jalur proses menjadi satu alur tunggal. Simbolnya juga berbentuk belah ketupat, seperti *Decision Node*, namun digunakan untuk tujuan yang berbeda. Jika *Decision Node* memecah jalur berdasarkan kondisi, maka *Merge Node* menyatukan kembali alur yang telah bercabang. *Merge Node* memungkinkan sistem untuk kembali ke satu jalur proses setelah sebelumnya memiliki beberapa kemungkinan jalur yang berbeda akibat keputusan logis.

5. *Fork dan Join Node*

Fork dan Join Node berfungsi untuk mengatur proses paralel dalam sistem. *Fork Node* digunakan untuk memecah satu alur proses menjadi beberapa jalur paralel yang berjalan secara bersamaan. Simbolnya berupa garis tebal horizontal atau vertikal dengan satu alur masuk dan beberapa alur keluar. Sebaliknya, *Join Node* digunakan untuk menyatukan kembali beberapa jalur paralel menjadi satu alur tunggal. Simbolnya juga berupa garis tebal namun dengan beberapa alur

masuk dan satu alur keluar, *Fork* dan *Join* sangat penting dalam menggambarkan aktivitas sistem yang dapat terjadi secara bersamaan maupun disinkronkan kembali.

6. *Final Node*

Final Node merupakan simbol yang menunjukkan akhir dari seluruh rangkaian aktivitas dalam sebuah *activity diagram*. Simbol ini digambarkan dengan lingkaran putih yang dikelilingi oleh lingkaran hitam (mirip simbol target). Ketika alur proses mencapai *Final Node*, maka seluruh aktivitas dianggap selesai. Tidak ada aktivitas lebih lanjut yang dilakukan setelah titik ini. *Final Node* penting untuk menandai titik terminasi dari alur proses yang digambarkan.

Fungsi Activity Diagram

1. Menggambarkan alur kerja atau proses sistem.
2. Memahami logika proses bisnis.
3. Menjadi dasar implementasi sistem.
4. Mempermudah komunikasi antara analis dan pengembang sistem

2.5.3 *Class diagram*

Class Diagram merupakan salah satu diagram utama dalam *Unified Modeling Language (UML)* yang digunakan untuk memodelkan struktur sistem berorientasi objek. Diagram ini menggambarkan kelas-kelas, atribut, metode, serta relasi antar kelas yang membentuk fondasi sistem perangkat lunak.

Tabel 2.4 Simbol pada *Class Diagram*

Simbol	Nama	Keterangan
	Class (Kelas)	Ditampilkan sebagai persegi panjang terbagi tiga bagian: 1. Nama kelas 2. Atribut 3. Metode (fungsi)
	Attribute (Atribut)	Properti atau variabel dari kelas. Biasanya ditulis dengan format: - nama: tipe data + = publik, - = privat
	Method (Metode)	Fungsi atau operasi dalam kelas. Format: + namaFungsi(parameter): tipe keluaran
	Association	Garis yang menghubungkan dua kelas, menunjukkan hubungan langsung.
	Aggregation	Menunjukkan bahwa suatu kelas memiliki kelas lain, tapi bisa berdiri sendiri. Digambarkan dengan diamond kosong.
	Composition	Seperti aggregation, tetapi objek yang dimiliki bergantung sepenuhnya pada kelas induknya (tidak bisa berdiri sendiri). Digambarkan dengan diamond berisi (hitam).
	Inheritance (Generalization)	Menunjukkan bahwa suatu kelas mewarisi atribut dan metode dari kelas lain. Digambarkan dengan panah segitiga putih.

Fungsi dan Manfaat *Class Diagram*

1. Menggambarkan struktur logis sistem perangkat lunak.
2. Menjadi dasar dalam pengkodean program berorientasi objek.
3. Menjelaskan hubungan dan ketergantungan antar komponen sistem.
4. Membantu dokumentasi dan komunikasi antar tim pengembang.

2.6 Konsep Dasar Informasi

Informasi adalah data yang telah diproses dan memiliki arti atau manfaat bagi penggunaannya. Informasi sebagai hasil dari pengolahan data dalam suatu organisasi sangatlah penting karena informasi merupakan landasan untuk mengambil suatu keputusan dan data merupakan sumber dari informasi. Konsep dasar informasi, adalah data yang diolah menjadi bentuk yang lebih berguna dan penting bagi yang menerimanya. Dari pengertian diatas penulis dapat menyimpulkan bahwa informasi merupakan suatu hasil (*output*) dari suatu data

yang diolah dengan metode pendekatan dan pengembangan tertentu (Susanto & Pratama, 2020).

Informasi berasal dari bahasa Perancis “*informacion*” dan bahasa latin “*informationem*” yang berarti garis besar, konsep, ide. Informasi merupakan data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya yang bermanfaat bagi pengambil keputusan saat mendatang (Nasution, 2022). Informasi adalah sesuatu keterkaitan antara data dan informasi sebagai entitas penting pembentuk sistem informasi. Data merupakan nilai, keadaan, atau sifat yang berdiri sendiri lepas dari konteks apapun. Sementara informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya. Sistem informasi adalah sistem yang menyediakan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerima. Informasi yang mempunyai nilai, biasanya karena rangkaian data yang tidak lengkap atau kadaluarsa (Pohan & Nasution, 2021).

2.7 Pendaftaran *online*

Pendaftaran *online* adalah proses pendaftaran yang dilakukan melalui jaringan internet menggunakan perangkat digital seperti komputer, *tablet*, atau *smartphone*. Sistem ini menggantikan proses manual dengan menyediakan *platform* digital yang memungkinkan pengguna mengisi data, mengunggah dokumen, dan menyelesaikan proses pendaftaran secara daring.