

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Aplikasi**

Aplikasi adalah suatu program yang dibangun untuk mengerjakan tugas tertentu dengan memproses input yang masuk menjadi output. Aplikasi merupakan program yang menjadi perantara antara manusia dengan komputer. Aplikasi dapat diartikan sebagai suatu program berbentuk perangkat lunak yang berjalan pada suatu sistem tertentu yang berguna untuk membantu berbagai kegiatan yang dilakukan oleh manusia [11]. Adanya aplikasi memudahkan manusia untuk memberikan perintah terhadap komputer yang hanya mengerti bahasa mesin.

#### **2.2 Pemrograman Web**

Pemrograman adalah proses atau aktivitas merancang, membangun, memperbaiki, dan mengembangkan program komputer dengan menggunakan bahasa pemrograman. Pemrograman memiliki fungsi untuk memuat program yang mampu membantu pekerjaan manusia dengan memproses perintah penggunanya.

Web merupakan kata yang diambil dari kalimat world wide web. World wide web adalah layanan internet yang paling populer saat ini dan dikenal juga sebagai halaman-halaman website yang dapat saling terkoneksi satu dengan lainnya [12]. Untuk mengakses web, diperlukan adanya web browser yang terkoneksi dengan layanan internet. Dengan menggunakan web browser, pengguna dapat mengeksplorasi, mengakses, dan berinteraksi dengan berbagai halaman web yang mencakup beragam konten seperti teks, gambar, video, dan elemen multimedia lainnya.

Web dapat berfungsi sebagai sistem informasi global yang memfasilitasi pertukaran informasi di seluruh dunia. Web memungkinkan pengguna untuk mengakses dan menavigasi informasi menggunakan browser melalui protokol HTTP (Hypertext Transfer Protocol). Informasi yang dimuat pada halaman-halaman web merupakan file yang berisi kumpulan sintaks HTML. Sintaks-sintaks HTML berfungsi untuk membuat dan merancang struktur halaman web sehingga data atau informasi yang dimuat dapat terlihat oleh pengguna.

Pemrograman web adalah proses pembuatan program yang hasilnya dapat dilihat melalui web browser. Tujuan dari pemrograman web adalah membuat halaman web yang interaktif dan dapat diakses oleh pengguna melalui browser. Pemrograman web dapat dibagi menjadi 2 bagian, diantaranya adalah sebagai berikut [12] :

### **2.2.1 Client Side Scripting**

Client side scripting merupakan jenis skrip yang dijalankan pada sisi client. Pemrograman ini akan berfokus pada bagaimana data dan informasi yang ada pada server dapat dilihat dan berinteraksi dengan sisi client. Client side scripting bertugas untuk mengenali semua perintah yang dilakukan pengguna pada sisi client sehingga perintah yang ada dapat diteruskan ke server untuk diproses lebih lanjut.

### **2.2.2 Server Side Scripting**

Server side scripting merupakan jenis skrip yang dijalankan pada sisi server. Pemrograman ini berfokus pada bagaimana permintaan yang diminta oleh pengguna dari sisi client dapat dilaksanakan. Server side scripting bertugas untuk menerima dan memproses perintah dari sisi client serta mengirimkan hasil pemrosesan untuk selanjutnya ditampilkan ke sisi client. Pemrograman web jenis ini tidak berinteraksi langsung dengan pengguna tanpa adanya sisi client.

## **2.3 Speech to Text**

Speech-to-text (STT) merupakan salah satu teknologi yang muncul seiring perkembangan teknologi yang semakin canggih. STT adalah teknologi yang mengonversi ucapan atau kata-kata yang diucapkan secara lisan menjadi teks tertulis. Tujuan dari teknologi ini adalah untuk memungkinkan komunikasi yang lebih efektif, seperti yang dijelaskan oleh I Komang Setia Buana [6]. Terutama dalam situasi di mana penulisan manual atau penggunaan keyboard kurang efisien.

Proses STT melibatkan penggunaan algoritma dan model bahasa yang canggih untuk mengenali dan mentranskripsikan ucapan menjadi teks. Aplikasi umum dari STT melibatkan sistem voice recognition pada perangkat mobile, asisten virtual, pengenalan ucapan dalam aplikasi keamanan, serta integrasi dengan perangkat lunak atau sistem lain yang memerlukan input teks.

## 2.4 Speech Recognition

Speech recognition, atau pengenalan ucapan, adalah teknologi yang memungkinkan komputer atau sistem untuk mengidentifikasi, memahami, dan mengartikan ucapan manusia menjadi teks tertulis atau perintah suara. Kata-kata yang diucapkan diubah bentuknya menjadi sinyal digital dengan cara mengubah gelombang suara menjadi sekumpulan angka yang kemudian disesuaikan dengan kode-kode tertentu untuk mengidentifikasi kata-kata tersebut [7].

Teknologi ini digunakan dalam berbagai aplikasi, termasuk asisten virtual, sistem kontrol suara pada perangkat elektronik, dan transkripsi otomatis pada layanan konferensi atau media. Speech recognition juga dapat menjadi fitur penting dalam pengembangan teknologi berbasis suara, memfasilitasi interaksi manusia dengan perangkat dan aplikasi tanpa memerlukan input tertulis atau fisik.

Saat ini banyak pengoalahan suara yang dimanfaatkan untuk mempermudah kehidupan manusia. Suara yang dihasilkan manusia diolah dan dikonversi sehingga responden yang dalam hal ini merupakan komputer, dapat mengerti dan memproses perintah sesuai dengan apa yang diucapkan. Akibat adanya perkembangan teknologi yang semakin pesat, banyak algoritma yang dapat dimanfaatkan dalam membantu manusia dalam membangun sistem yang lebih canggih.

Untuk mengidentifikasi kata yang diucapkan, teknologi speech recognition menggabungkan ilmu linguistik dan komputer untuk mengubahnya menjadi teks. Terdapat tiga tahap proses umum bagaimana komputer dapat mengubah sinyal suara ke dalam bentuk teks. Ketiga metode tersebut adalah preprocessing, feature extraction, dan pengklasifikasi.

Dalam transisi sinyal suara menjadi teks, masukan sinyal suara akan melalui tahapan pre-processing terlebih dahulu agar sinyal suara yang menjadi input menjadi layak untuk dilanjutkan ke proses selanjutnya. Data yang sudah melalui tahapan pre-processing selanjutnya akan diekstraksi dan dilakukan perbandingan tingkat penekanan suara yang selanjutnya dicocokkan dengan database tersedia.

## 2.5 Machine Learning

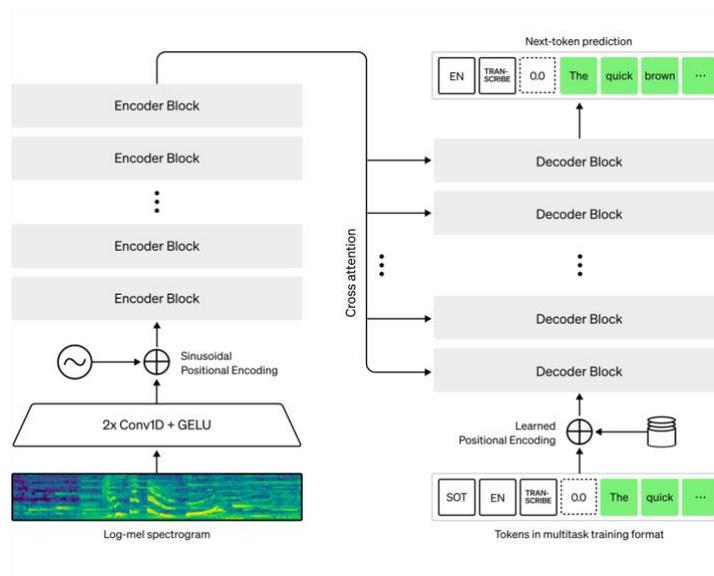
Machine learning adalah cabang dari kecerdasan buatan yang fokus pada pengembangan algoritma yang memungkinkan sistem untuk belajar dari data dan

mengidentifikasi pola atau membuat keputusan. Machine Learning adalah pendekatan dalam kecerdasan buatan yang bertujuan untuk meniru perilaku manusia dalam menyelesaikan masalah atau melakukan otomatisasi. Dua aplikasi utama ML adalah klasifikasi dan prediksi.

Ciri khas ML adalah proses pelatihan atau training, yang memerlukan data untuk dipelajari, dikenal sebagai data training. Klasifikasi adalah metode di mana mesin mengkategorikan objek berdasarkan karakteristik tertentu, mirip dengan cara manusia membedakan satu benda dari yang lain. Prediksi atau regresi digunakan untuk menebak keluaran dari suatu data masukan berdasarkan data yang telah dipelajari selama training. Metode ML populer termasuk Sistem Pengambil Keputusan, Support Vector Machine (SVM), dan Neural Network [13].

## **2.6 Whisper Model Machine Learning**

Whisper model machine learning merupakan model untuk pengenalan suara yang dibangun untuk mentranskrip audio dalam berbagai bahasa. Model ini dilatih dengan 680.000 jam data yang terdiri dari berbagai macam bahasa. Model ini dikembangkan oleh OpenAI untuk meningkatkan kemampuan teknologi pengolahan bahasa alami (NLP). Whisper dilatih (training) dengan dataset yang mencakup berbagai jenis suara dan kondisi audio. Model ini bersifat opensource untuk mendukung penelitian lebih lanjut mengenai pemrosesan bahasa alami. Gambar 2.1 berikut merupakan arsitektur yang digunakan dalam model machine learning Whisper.



Gambar 2.1 Proses Transkrip pada Whisper Model

Sumber gambar : <https://openai.com/index/whisper/>

Arsitektur whisper diimplementasikan sebagai encoder-decoder Transformer. Model ini bekerja dengan membagi audio yang dimasukkan ke dalam beberapa potongan kemudian diproses menjadi representasi visual spektrum frekuensi audio oleh encoder. Dari hasil pemrosesan tersebut, decoder akan melakukan prediksi teks yang ditambah dengan token khusus seperti identifikasi bahasa, timestamp tingkat frasa, dan berbagai macam transkrip bahasa [14].

## 2.7 Huggingface

Hugging Face adalah platform open source machine learning, yang memungkinkan penggunanya membuat, melatih, dan menerapkan model machine learning khususnya dalam pemrosesan bahasa alami (NLP). Huggingface memanfaatkan Transformers untuk menyediakan model-model siap pakai untuk berbagai tugas seperti menerjemahkan teks, menganalisis sentimen, dan menjawab pertanyaan. Dengan adanya transformers pada huggingface ini memudahkan penggunanya untuk menambah atau menggunakan machine learning yang tersedia tanpa harus membangun model dari awal.

Selain itu, Hugging Face juga memiliki platform di mana pengguna bisa berbagi model, dataset, dan hasil eksperimen. Platform ini memudahkan para penggunanya untuk bekerja sama, berbagi sumber daya, dan belajar dari satu sama

lain. Pengguna bisa menemukan model yang sudah ada, menyesuaikannya dengan kebutuhan mereka, atau melatih model baru dengan data yang tersedia di platform tersebut.

Melakukan *training* pada model machine learning yang kompleks memerlukan banyak keahlian dan sumber daya. Dengan hal demikian, penggunaan ulang model yang sudah ada merupakan hal yang menguntungkan. Adanya huggingface ini dapat memfasilitasi kebutuhan tersebut [15].

Huggingface tidak hanya menyediakan model machine learning saja, namun juga terdapat dataset, space, dan libraries untuk membantu dalam menggunakan model-model yang tersedia. Huggingface juga bekerjasama dengan beberapa perusahaan seperti amazon untuk mendukung pelatihan (training), inferensi (inference), deploy, dan demonstrasi model [16].

## 2.8 GPS

GPS singkatan dari Global Positioning System, adalah sistem navigasi satelit yang memberikan informasi lokasi dan waktu di seluruh dunia. Sistem ini terdiri dari jaringan satelit yang mengorbit bumi dan penerima GPS yang terpasang pada perangkat elektronik seperti smartphone, kendaraan, atau perangkat GPS khusus. Penerima GPS kemudian menggunakan data sinyal tersebut untuk menghitung lokasi, kecepatan, dan elevasi pengguna. Komponen yang terdapat di dalam GPS terdiri dari unit antena sebagai penangkap sinyal satelit, unit display sebagai penampil informasi hasil pembacaan, unit receiver sebagai penerima sinyal satelit untuk memperkirakan posisi [17].

Dengan menggunakan data dari sejumlah satelit, GPS dapat memberikan informasi yang sangat akurat tentang posisi dan pergerakan seseorang di permukaan bumi. Menurut Setiawan dan Sari [18], tujuan utama dari GPS adalah memberikan informasi lokasi dengan cepat, tepat, dan akurat. Sebagai bentuk pemanfaatan teknologi tersebut, GPS sudah banyak digunakan dalam aspek kehidupan sehari-hari seperti membantu orang untuk menentukan rute perjalanan, melacak aktivitas olahraga, atau memberikan bantuan darurat dengan menentukan lokasi seseorang.

Terdapat berbagai cara untuk mendapatkan informasi mengenai titik koordinat dari perangkat yang memiliki GPS. Salah satu cara yang dapat digunakan adalah

dengan menggunakan Geolocation API yang disediakan dalam bahasa Javascript. Pengembang tidak perlu melakukan instalasi apapun terlebih dahulu untuk menggunakannya. Berikut ini merupakan contoh pemanfaatan Geolocation API untuk mendapatkan data koordinat perangkat yang sedang digunakan.

Tabel 2.1 Script Geolocation API

```
function success(position) {
  doSomething(position.coords.latitude, position.coords.longitude);
}

function error() {
  alert("Sorry, no position available.");
}

const options = {
  enableHighAccuracy: true,
  maximumAge: 30000,
  timeout: 27000,
};

const watchID = navigator.geolocation.watchPosition(success, error, options);
```

## 2.9 Location Based Service

Location-Based Service (LBS) adalah jenis layanan informasi atau aplikasi yang menyediakan fungsi berdasarkan lokasi fisik pengguna atau perangkat mereka. Menurut Febrianto [19], Layanan Berbasis Lokasi (LBS) merupakan bentuk pemanfaatan data perangkat mobile melalui penyedia layanan jaringan komunikasi, yang memungkinkan aplikasi untuk menyediakan layanan berdasarkan lokasi pengguna.

LBS memanfaatkan teknologi lokasi seperti GPS (Global Positioning System) untuk menjalankan layanannya. Khukuh dan Chanifah pada tinjauan literturnya menyebutkan bahwa teknologi GPS menjadi salah satu teknologi yang sering digunakan sebagai elemen posisi dalam menerapkan teknologi Location-Based Service (LBS) [20]. Untuk menjalankan LBS, terdapat beberapa komponen yang harus ada untuk mendukung keberlangsungan layanan [20]. Komponen-komponen tersebut adalah *mobile devices* yang mejadi komponen paling penting, kemudian *communication network* yang menjadi penghubung antara komponen satu dengan komponen lain, komponen *positioning* yang menjadi informasi lokasi pengguna, *services and application provider* yang menjadi navigator bagi pengguna atau dapat

disebut juga sebagai penyedia layanan, dan terakhir adalah data and content provider untuk penyimpanan data.

## 2.10 Mapbox

Mapbox merupakan salah satu API Maps yang memiliki fitur seperti Google Maps API yang dapat dimanfaatkan untuk styling dan background peta. Platform ini dapat digunakan secara fleksibel untuk menyesuaikan kebutuhan. Mapbox membebaskan pengguna untuk menggunakan berbagai styling dan opsi latar belakang sehingga memungkinkan untuk membuat peta yang fungsional dan menarik.

Mapbox muncul seiring perkembangan teknologi informasi untuk mendukung visualisasi peta sehingga dapat membantu pengembang untuk membangun aplikasi sesuai kebutuhan spesifik yang diperlukan. Data yang digunakan pada mapbox sebagian besar berasal dari beberapa sumber seperti OpenStreetMap, USGS, Landsat, dan OpenAdress.

Mapbox dapat diimplementasikan ke dalam beberapa framework dan bahasa pemrograman seperti JavaScript, Python, dan Swift yang memudahkan pengembang dalam mengimplementasikan peta ke dalam aplikasi web dan mobile. Selain untuk menampilkan peta, mapbox juga memiliki beberapa layanan seperti geocoding dan rute perjalanan. Geocoding merupakan konversi antara alamat dan titik koordinat. Sedangkan fitur rute perjalanan dapat digunakan untuk memberikan penunjuk arah dari satu lokasi ke lokasi lain [21].

Terdapat beberapa cara dalam mengimplementasikan Mapbox. Beberapa cara dapat dilakukan dengan menginstall library mapbox GL JS atau melakukan HTTP request langsung dengan API Key yang didapatkan setelah mendaftarkan akun. Berikut ini adalah contoh implementasi penggunaan Mapbox GL JS untuk menampilkan peta pada halaman web menggunakan React JS.

Tabel 2.2 Script Mapbox GL JS

```
import React, { useEffect, useRef } from 'react';
import mapboxgl from 'mapbox-gl';

import 'mapbox-gl/dist/mapbox-gl.css';

const MapboxExample = () => {
  const mapContainerRef = useRef();
```

```

const mapRef = useRef();

useEffect(() => {
  mapboxgl.accessToken = 'pk ';

  mapRef.current = new mapboxgl.Map({
    container: mapContainerRef.current,
    center: [-74.5, 40], // starting position [lng, lat]
    zoom: 9 // starting zoom
  });
});

return (
  <div
    style={{ height: '100%' }}
    ref={mapContainerRef}
    className="map-container"
  />
);
};

export default MapboxExample;

```

## 2.11 Library Leaflet

Leaflet adalah sebuah library JavaScript open-source yang diciptakan oleh Vladimir Agafonkin. Leaflet mudah untuk diimplementasikan di berbagai platform, baik web maupun mobile, sehingga memudahkan pengembang dalam menciptakan peta interaktif yang responsif dan efisien di berbagai perangkat.

Banyak fitur leaflet yang dapat digunakan pengembang untuk mengembangkan aplikasi yang berhubungan dengan pemetaan. Leaflet dapat digunakan dengan platform lainnya. Leaflet memiliki kemampuan yang kuat untuk menginterpretasi peta dalam berbagai sistem koordinat. Hal tersebut menjadikan Leaflet sebagai library yang fleksibel bagi pengembang yang membutuhkan solusi pemetaan yang dapat disesuaikan dengan berbagai kebutuhan [22].

Terdapat beberapa fungsi yang ada pada leaflet. Berikut ini merupakan salah satu contoh penggunaan fungsi `lat.lng` untuk merepresentasikan poin geografis dalam bentuk latitude dan longitude.

Tabel 2.3 Script Leaflet

```
L.latLng(<Number> latitude, <Number> longitude, <Number> altitude?)
```

## 2.12 Web Socket

Web socket merupakan protokol komunikasi antara server dan klien secara real time. Komunikasi ini dilakukan melalui koneksi TCP yang menyebabkan adanya

komunikasi data secara *real-time*. WebSocket dirancang untuk teknologi komunikasi *full-duplex* yang memiliki arti bahwa data dapat dikirim dan diterima secara bersamaan dengan menggunakan HTTP sebagai transport layer untuk memanfaatkan infrastruktur yang sudah ada seperti proxy, filtering, dan autentikasi. WebSocket secara utama memberikan solusi yang efektif dalam membangun komunikasi full-duplex antara web browser dan web server.

Koneksi websocket dimulai dari permintaan koneksi dari klien ke server melalui HTTP untuk memulai proses koneksi. Setelah itu, pesan akan dikirimkan melalui TCP. Jika server mendukung WebSocket, server akan mengirimkan pesan persetujuan melalui HTTP dan koneksi pun dimulai. Setelah koneksi terjalin, klien dan server mulai menggunakan protokol WebSocket untuk komunikasi. Koneksi antara kedua node ini berlanjut hingga koneksi berakhir.

Pesan yang dikirim melalui WebSocket bisa berisi teks atau data biner. Header pesan dibuat sekecil mungkin untuk meminimalkan penggunaan bandwidth. Pesan juga bisa dibagi menjadi beberapa frame untuk dikirim secara berurutan, yang dikenal sebagai streaming. Koneksi WebSocket dapat diatur baik dalam bentuk tidak terenkripsi maupun terenkripsi menggunakan TLS, seperti pada HTTP. Karena WebSocket menggunakan port yang sama dengan HTTP dan HTTPS (80 dan 443 secara default), protokol ini dapat dikenali oleh firewall dan proxy, sehingga dapat berfungsi tanpa perlu mengubah pengaturan HTTP yang sudah ada [23].

### **2.13 Socket Io**

Socket.IO adalah sebuah library JavaScript yang memungkinkan komunikasi real-time dua arah antara klien dan server. Komunikasi antara server dan klien dilakukan dengan menggunakan protocol web socket. Socket.IO membuat proses pengaturan koneksi terus-menerus antara aplikasi web dan server. Data dapat dikirim dan diterima secara tanpa perlu melakukan permintaan HTTP secara terus menerus. Socket.IO biasanya digunakan dalam aplikasi chat, notifikasi langsung dan aplikasi yang memerlukan update data secara cepat.

Socket.IO terbagi menjadi 2 bagian yaitu bagian server dan klien. Pada sisi server, Socket.IO banyak digunakan dengan Node.js, dan di sisi klien, dapat

diintegrasikan dengan aplikasi berbasis JavaScript di browser. Socket IO membantu sisi server untuk saling mengirim pesan dengan klien dalam bentuk *event*. *Event* ini yang akan membawa data antara server dan klien [24].

Penggunaan Socket.IO dilakukan dengan menginstall library ke dalam lingkungan yang akan diimplementasikan. Berikut ini merupakan contoh penggunaan Socket.IO pada bagian server.

Tabel 2.4 Script Socket.IO pada Sisi Server

```
const http = require('http');
const socketIo = require('socket.io');
const server = http.createServer();
const io = socketIo(server);

io.on('connection', (socket) => {
  console.log('A user connected');

  socket.on('message', (msg) => {
    console.log('Message: ' + msg);
    io.emit('message', msg);
  });
  socket.on('disconnect', () => {
    console.log('User disconnected');
  });
});

server.listen(3000, () => {
  console.log('Listening on *:3000');
});
```

Agar komunikasi real-time dapat berjalan, Socket.IO juga perlu diimplementasikan pada sisi *client*. Berikut ini merupakan contoh penggunaan Socket.IO pada sisi *client* yang akan berinteraksi dengan server.

Tabel 2.5 Script Socket.IO pada Sisi Client

```
<script src="/socket.io/socket.io.js"></script>
<script>
  const socket = io();

  const input = document.getElementById('input');
  const sendBtn = document.getElementById('sendBtn');
  const messages = document.getElementById('messages');

  sendBtn.addEventListener('click', () => {
    const msg = input.value;
    socket.emit('message', msg);
    input.value = '';
  });

  socket.on('message', (msg) => {
    const item = document.createElement('li');
    item.textContent = msg;
    messages.appendChild(item);
  });
</script>
```

```

    });
  </script>
</script>

```

## 2.14 Service Worker

Service worker merupakan sebuah skrip yang dijalankan oleh browser pada latar belakang dan terpisah dari halaman web. Service worker dapat membantu aplikasi web dalam mengirim notifikasi. Service Worker tidak dapat mengakses objek yang ada pada halaman web secara langsung.

Proses implementasi service worker diawali dengan mendaftarkannya terlebih dahulu pada JavaScript yang ada pada halaman web. Jika proses pendaftaran berhasil, browser akan melakukan instalasi service worker. Terdapat beberapa aset statik yang disimpan dalam cache saat instalasi service worker dilakukan. Saat semua file berhasil disimpan, maka proses instalasi akan berhasil. Namun jika cache gagal disimpan, maka service worker gagal melakukan instalasi dan tidak dapat digunakan.

Setelah instalasi berhasil, tahap pengaktifan service worker dimulai, di mana cache lama dapat dikelola kembali. Setelah tahap pengaktifan berhasil dilakukan, service worker dapat mengontrol semua halaman yang berada dalam cakupannya. Ketika service worker aktif, ada dua kondisi yang bisa terjadi. Pertama adalah kondisi saat service worker dihentikan untuk menghemat memori. Kedua adalah kondisi saat service worker menangani fetch dan message event yang terjadi saat ada permintaan jaringan atau pesan yang dikirim dari halaman web [25].

Penggunaan service worker tidak perlu menggunakan instalasi apapun sebelumnya. Namun service worker perlu melakukan pendaftaran yang dilakukan dari sisi *client* terlebih dahulu. Berikut ini merupakan contoh daftar kode yang didaftarkan ke dalam service worker.

Tabel 2.6 Script File Service Worker

```

const addResourcesToCache = async (resources) => {
  const cache = await caches.open("v1");
  await cache.addAll(resources);
};

self.addEventListener("install", (event) => {
  event.waitUntil(
    addResourcesToCache([
      "/",

```

```

    "/index.html",
    "/style.css",
    "/app.js",
    "/image-list.js",
    "/star-wars-logo.jpg",
    "/gallery/bountyHunters.jpg",
    "/gallery/myLittleVader.jpg",
    "/gallery/snowTroopers.jpg",
  ],
);
});

```

Barisan kode di atas merupakan cache yang akan didaftarkan ke dalam service worker pada web browser. Cache ini tidak dapat didaftarkan dengan sendirinya, tetapi perlu ada fungsi yang akan mendaftarkan cache tersebut. Berikut ini merupakan contoh fungsi yang akan mendaftarkan cache di atas ke dalam service worker.

Tabel 2.7 Script File Sisi Client untuk Mendaftarkan File Service Worker

```

const registerServiceWorker = async () => {
  if ("serviceWorker" in navigator) {
    try {
      const registration = await navigator.serviceWorker.register("/sw.js", {
        scope: "/",
      });
      if (registration.installing) {
        console.log("Service worker installing");
      } else if (registration.waiting) {
        console.log("Service worker installed");
      } else if (registration.active) {
        console.log("Service worker active");
      }
    } catch (error) {
      console.error(`Registration failed with ${error}`);
    }
  }
};
registerServiceWorker();

```

## 2.15 Push Notification

Push notification merupakan teknologi yang digunakan untuk mengirim pesan berupa notifikasi ke pengguna. Push notification merupakan teknologi yang termasuk ke dalam model komunikasi internet. Informasi dikirim ke klien tanpa perlu melakukan permintaan atau persetujuan dari server terlebih dahulu. Push notification secara otomatis dapat mengirimkan pesan atau informasi ke pengguna ketika terdapat pembaruan tertentu sehingga pengguna dapat menerima informasi secara langsung.

Push notification sangat berguna dalam berbagai aplikasi, seperti aplikasi berita, media sosial, atau layanan pesan instan yang memudahkan pengguna mendapatkan informasi terkini tanpa harus secara aktif mencari atau membuka aplikasi. Teknologi ini memungkinkan pengguna tetap terhubung dengan konten atau layanan penting, meskipun tidak sedang menggunakan aplikasi. Dengan push notification, server dapat mengirimkan pemberitahuan tentang pesan baru, pembaruan konten, penawaran khusus, atau peringatan lainnya langsung ke perangkat pengguna. Hal ini membuat interaksi antara aplikasi dan pengguna menjadi lebih dinamis dan efektif, serta meningkatkan keterlibatan pengguna dengan aplikasi tersebut [26].

Berikut ini merupakan contoh barisan kode sederhana penggunaan push notification dalam sistem.

Tabel 2.8 Script Push Notification

```
<button onclick="notifyMe()">Notify me!</button>

function notifyMe() {
  if (!("Notification" in window)) {
    alert("This browser does not support desktop notification");
  } else if (Notification.permission === "granted") {
    const notification = new Notification("Hi there!");
  } else if (Notification.permission !== "denied") {
    Notification.requestPermission().then((permission) => {
      if (permission === "granted") {
        const notification = new Notification("Hi there!");
      }
    });
  }
}
```

Push notification dapat digunakan dalam komponen html dan javascript sehingga tidak memerlukan instalasi apapun namun terdapat beberapa batasan yang perlu diperhatikan. Penggunaan push notification dalam android perlu memerlukan service worker dan web push. Berikut ini contoh penggunaan push notification dengan service worker.

Tabel 2.9 Script Push Notification dengan Service Worker

```
navigator.serviceWorker.register("sw.js");

function showNotification() {
  Notification.requestPermission().then((result) => {
    if (result === "granted") {
      navigator.serviceWorker.ready.then((registration) => {
        registration.showNotification("Vibration Sample", {
```

```
body: "Buzz! Buzz!",  
icon: "../images/touch/chrome-touch-icon-192x192.png",  
vibrate: [200, 100, 200, 100, 200, 100, 200],  
tag: "vibration-sample",  
});  
});  
}  
});  
}
```

## 2.16 Web Push

Web Push Notification merupakan pemberitahuan yang dapat dikirim ke pengguna melalui web desktop dan web seluler. Layanan webpush diperlukan dalam penerapan push notification [27]. Web Push adalah teknologi yang memungkinkan pengiriman push notification melalui browser web, bahkan ketika pengguna tidak sedang membuka situs web yang terkait. Hubungan antara push notification dan web push adalah web push merupakan implementasi spesifik dari push notification pada web.

Web Push bekerja dengan cara yang mirip dengan push notification pada aplikasi mobile. Ketika pengguna mengunjungi situs web yang mendukung Web Push, maka pengguna dapat memilih untuk menerima notifikasi dari situs tersebut pada permintaan akses penggunaan notifikasi. Setelah pengguna memberikan izin, situs web dapat mengirimkan notifikasi ke perangkat pengguna melalui browser, bahkan ketika tab atau jendela situs tersebut tidak sedang aktif.

Instalasi web push diperlukan untuk mengimplementasikan web push ke dalam sistem. Instalasi dilakukan dalam lingkup NPM dan perlu di *generate* terlebih dahulu untuk mendapatkan VAPID keys yang nantinya akan digunakan ke dalam sistem.

VAPID (Voluntary Application Server Identification) Keys adalah sebuah mekanisme yang digunakan dalam web push notifications untuk mengautentikasi server aplikasi yang mengirimkan pemberitahuan ke browser klien. Dengan VAPID, server menghasilkan sepasang kunci kriptografi, yaitu Public Key dan Private Key. Public Key digunakan oleh klien untuk berlangganan (subscribe) ke layanan push, sedangkan Private Key digunakan oleh server untuk menandatangani pesan push yang dikirimkan.

VAPID Keys digunakan untuk mengautentikasi server yang mengirim notifikasi push dengan menandatangani pesan menggunakan Private Key dan mengirimkannya bersama Public Key ke layanan push. Layanan push kemudian memverifikasi signature ini untuk memastikan bahwa pesan berasal dari server yang sah. VAPID memungkinkan server terotorisasi untuk mengirim notifikasi tanpa perlu pendaftaran ke layanan push tertentu. Berikut ini merupakan contoh hasil VAPID Keys yang didapat saat proses *generate* setelah instalasi berhasil dilakukan.

Tabel 2.10 VAPID Key dari Generate Web Push

```
{ "publicKey": "BGtkbcjr012YMoDuq2sCQeHlu47uPx3SHTgFKZFYiBW8Qr0D9vgyZSZPdw6_4ZFEI9Snk1VEAj2qTYI1I1YxBXE", "privateKey": "I0_d0vnesxbBSUm1Dd0KibGo6vEXRO-Vu88Q1S1m5j0" }
```

Berikut ini merupakan contoh penggunaan VAPID Keys dalam sisi server untuk mengirimkan pesan push notifikasi.

Tabel 2.11 Script Web Push

```
const webpush = require('web-push');

webpush.setGCMAPIKey('<Your GCM API Key Here>');
webpush.setVapidDetails(
  'mailto:example@yourdomain.org',
  vapidKeys.publicKey,
  vapidKeys.privateKey
);

const pushSubscription = {
  endpoint: '.....',
  keys: {
    auth: '.....',
    p256dh: '.....'
  }
};

webpush.sendNotification(pushSubscription, 'Your Push Payload Text');
```

Web Push menggunakan Service Worker sebagai perantara antara server dan browser. Service Worker, yang berjalan di latar belakang, menerima pesan dari server dan menampilkan push notification kepada pengguna. Hal ini dapat membantu membangun aplikasi dengan komunikasi real-time antara situs web dan pengguna tanpa memerlukan interaksi langsung dari pengguna.

## 2.17 Web Service

Web service merupakan suatu layanan yang memiliki kemampuan untuk mengumpulkan dan mengintegrasikan data yang dapat berinteraksi antar perangkat lunak [28]. Web service adalah layanan yang disediakan melalui internet, memungkinkan komunikasi dan pertukaran data antara berbagai sistem perangkat lunak atau aplikasi. Web service menyajikan endpoint dalam bentuk URL yang dapat dijangkau melalui jaringan internet dengan memanfaatkan format pertukaran data [29].

Kelebihan web service terletak pada kemampuannya untuk digunakan oleh banyak klien secara bersamaan, meskipun menggunakan bahasa pemrograman yang berbeda. Web service banyak digunakan karena fleksibilitasnya terhadap berbagai platform dan bahasa pemrograman [30].

Terdapat dua metode dalam web service, yaitu informasi Representational State Transfer (REST) dan Simple Object Access Protocol (SOAP) yang mengarah pada sumber daya dan pelayanan informasi. Meskipun metode SOAP telah diterapkan oleh banyak perusahaan, saat ini metode REST mendominasi dalam pengembangan layanan web karena dianggap lebih efisien baik dalam pengembangan maupun penggunaannya [31].

## 2.18 API (Application Programming Interface)

Application program interface (API) adalah fungsi yang dibangun untuk memungkinkan akses ke sistem lain sehingga suatu sistem dapat diakses oleh aplikasi atau sistem lainnya [32][33]. API bekerja sebagai penghubung atau perantara dari berbagai macam sistem sehingga antar sistem dapat melakukan komunikasi. API merupakan suatu sistem yang beroperasi di belakang layar untuk mengintegrasikan secara bersamaan informasi dari berbagai aplikasi [34]. Tujuan dari API adalah untuk mempermudah pemanfaatan suatu teknologi saat membuat perangkat lunak atau aplikasi bagi pengembang [35]. Web Service merupakan bentuk API yang memfasilitasi pengguna untuk mengambil data. Melalui arsitektur Representational State Transfer (ReST) yang beroperasi menggunakan Hypertext Transfer Protocol (HTTP), file Javascript Object Notation (JSON) yang akan disajikan kepada pengguna ketika mengakses API sebagai bentuk respon yang

diberikan[33]. Respon yang diberikan tidak selalu berupa JSON, namun bisa juga berupa Extensible Markup Language (XML) [36].

### **2.19 JSON (JavaScript Object Notation)**

JSON (JavaScript Object Notation) adalah format pertukaran data ringan yang digunakan untuk mentransmisikan dan menyimpan data. Format ini bersifat independen terhadap bahasa pemrograman dan mudah untuk dibaca dan ditulis oleh manusia. JSON menggunakan struktur data yang mirip dengan objek dan array dalam JavaScript, yang melibatkan pasangan nama/kunci dan nilai [37].

Data dalam JSON direpresentasikan sebagai pasangan nama/nilai, nama adalah string dan nilai dapat berupa string, angka, objek, array, boolean, atau null. Struktur data yang fleksibel ini membuat JSON menjadi pilihan yang baik untuk pertukaran data antar aplikasi yang ditulis dalam berbagai bahasa pemrograman. JSON difokuskan pada bagaimana data direpresentasikan di halaman web [38].

### **2.20 Node JS**

Node.js adalah suatu platform atau lingkungan yang digunakan untuk menjalankan kode-kode yang ditulis dalam bahasa pemrograman JavaScript. Node.js memanfaatkan mesin JavaScript V8 yang dikembangkan oleh Google untuk menjalankan kode JavaScript dengan kecepatan tinggi. Tugas utama V8 adalah mengonversi kode JavaScript menjadi bytecode. Bytecode ini nantinya akan dieksekusi Node.js [39].

Kehadiran Node.js memungkinkan JavaScript untuk berperan sebagai backend, mengubah peran bahasa pemrograman yang sebelumnya hanya digunakan untuk membuat tampilan website lebih interaktif (frontend) menjadi kemampuan server (backend). Ini mencakup fungsi-fungsi seperti mengakses, menyimpan, mengelola, membaca, memperbarui, dan menghapus data yang disimpan dalam database [40].

Node.js awalnya digunakan untuk mengembangkan program yang berkaitan dengan jaringan komputer. Namun saat ini Node.js lebih banyak digunakan dalam pengembangan aplikasi berbasis web karena Node.js dapat memproses permintaan pada sisi klien dengan baik [39].

## 2.21 Express JS

Express Js adalah kerangka kerja web aplikasi yang dirancang untuk berjalan di atas platform Node.js, memberikan pengembang alat untuk membangun aplikasi web dan API dengan cepat dan efisien. Express Js merupakan kerangka kerja aplikasi web backend untuk membangun RESTful API [41]. Sebagai framework yang ringan, Express.js menyediakan dasar-dasar yang diperlukan untuk mengembangkan aplikasi web, seperti routing dan middleware. Namun, meskipun memiliki struktur minimalis, Express.js tetap sangat fleksibel, memungkinkan pengembang untuk menambahkan fitur dan fungsi sesuai dengan kebutuhan proyek mereka.

Keunggulan utama Express.js terletak pada kemampuannya untuk mempercepat proses pengembangan. Dengan menyediakan berbagai utilitas dasar dan mendukung penambahan middleware, Express.js membantu pengembang menghindari pekerjaan yang berulang dan membosankan. Hal ini sangat berguna untuk pengembangan aplikasi yang membutuhkan skalabilitas tinggi dan penanganan banyak permintaan secara bersamaan. Penggunaan Express.js juga memungkinkan integrasi yang mudah dengan berbagai database dan layanan lain, memperluas kemampuan aplikasi yang dibangun di atasnya.

Selain itu, karena Express.js dibangun di atas Node.js, ia memanfaatkan sepenuhnya keunggulan performa yang dimiliki oleh Node.js, seperti kemampuan pemrosesan asynchronous dan manajemen koneksi yang efisien. Ini menjadikan Express.js sangat cocok untuk aplikasi web modern yang membutuhkan responsivitas tinggi, seperti aplikasi real-time atau aplikasi yang melibatkan banyak interaksi pengguna. Dengan komunitas yang besar dan dukungan yang luas, Express.js terus berkembang menjadi salah satu kerangka kerja web yang paling populer di dunia pengembangan JavaScript.

## 2.22 React JS

React JS adalah library JavaScript yang dikembangkan untuk merancang antarmuka pengguna yang interaktif. React JS menggunakan konsep 'View' dari model MVC (Model-View-Controller). React JS dirancang oleh Jordan Walke, seorang teknisi perangkat lunak pada tahun 2011 dan dirilis untuk digunakan

secara umum pada tahun 2013. Tujuan dirancangnya untuk memberikan performa rendering yang fokus pada bagaimana sebuah komponen dapat dirender atau ditampilkan ke pengguna. Dengan penggunaan React JS untuk merancang tampilan pengguna, penggunaan komponen berulang ditemukan dapat memudahkan pengembang dalam merancang design antarmuka yang kaya.

React Js dapat diinstall pada berbagai sistem operasi seperti MacOS, Microsoft Windows, Unix, dll. Node JS dan NPM adalah platform yang diperlukan untuk mengembangkan aplikasi yang menggunakan React JS. NPM adalah package manager yang dapat menjalankan aplikasi pada port tertentu dan NPM juga digunakan saat melakukan instalasi library yang akan diperlukan dalam aplikasi React JS.

Terdapat beberapa keuntungan yang didapatkan dalam penggunaan React JS dalam membangun antarmuka pengguna. Beberapa keuntungan tersebut yang pertama adalah kemudahan dalam membangun antarmuka yang dinamis. Kemudian React JS ini bersifat open-source sehingga dapat digunakan oleh umum. Terakhir, React JS sangat berfokus pada komponen yang dapat digunakan berulang sehingga dapat mempersingkat waktu pengembangan [42].

## **2.23 UML (Unified Model Language)**

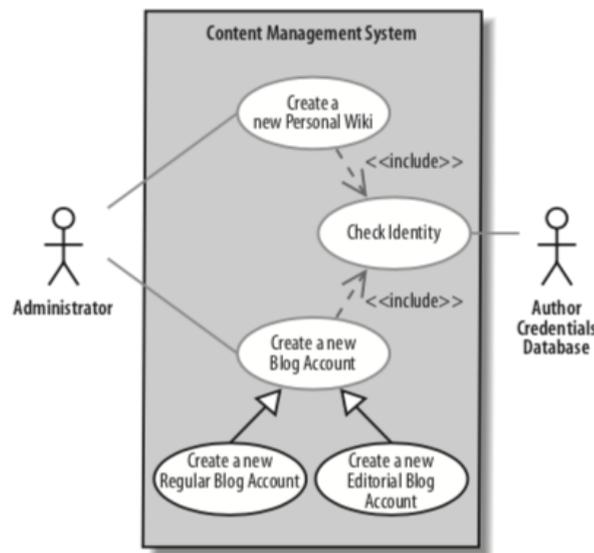
Unified Model Language adalah bahasa model standar pada pengembangan sistem dan perangkat lunak [43]. UML dapat membantu pengembang dalam mengatur sistem yang kompleks agar tampak sederhana sehingga pengembang dapat lebih fokus pada aspek penting dalam sistem yang akan dibangun. Adanya UML dalam proses pembangunan perangkat lunak dapat membantu pengembang untuk mengembangkan sistem yang sesuai dengan kebutuhan user, sehingga tujuan dari pembangunan perangkat lunak dapat tercapai dan kebutuhan user dapat terpenuhi.

### **2.23.1 Use Case Diagram**

Diagram use case adalah kebutuhan fungsional sistem yang dapat membantu pengembang dalam memahami kebutuhan-kebutuhan yang harus ada pada sistem. Pembuatan diagram use case merupakan tahapan awal yang penting dalam pengembangan sistem berorientasi objek. Penggunaan diagram use case ini

diharapkan dapat membantu pengembang dalam menyusun kebutuhan yang ada pada sebuah sistem.

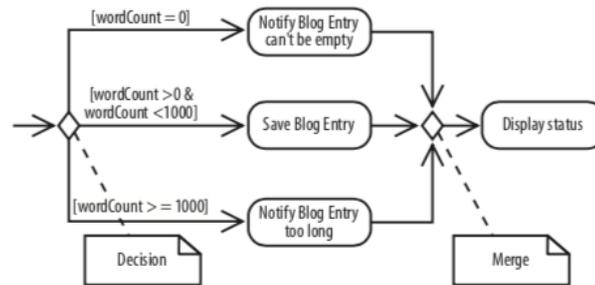
Terdapat dua komponen penting yang saling berinteraksi dalam sebuah diagram use case. Komponen tersebut adalah aktor dan use case. Aktor merepresentasikan orang-orang yang berinteraksi dengan sistem. Sedangkan use case merepresentasikan kebutuhan yang ada pada sistem. Setiap use case yang ada pada diagram menggambarkan kebutuhan spesifik yang berhubungan dengan aktor. Gambar 2.2 berikut merupakan contoh dari use case diagram.



Gambar 2.2 Contoh use case diagram

### 2.23.2 Activity Diagram

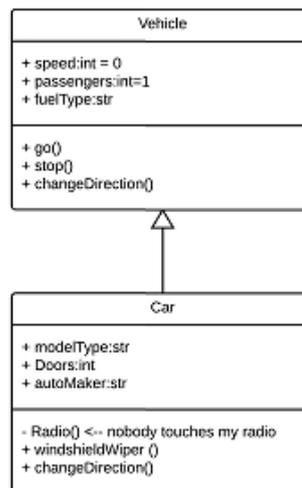
Activity diagram menggambarkan diagram alur aktivitas dalam sistem yang akan dibangun. Berbeda dengan usecase diagram yang merepresentasikan kebutuhan-kebutuhan pada sistem, activity diagram merepresentasikan bagaimana proses dari sebuah kebutuhan (*requirement*) dapat terpenuhi. Proses tersebut digambarkan dengan activity diagram dimulai dari bagaimana proses awal hingga suatu kebutuhan dapat tercapai. Gambar 2.3 berikut merupakan contoh activity diagram.



Gambar 2.3 Contoh activity diagram

### 2.23.3 Class Diagram

Class diagram merupakan diagram yang menggambarkan struktur dan deskripsi kelas, atribut, metode, serta hubungan antar objek digunakan dalam sistem yang akan dibangun. Diagram kelas ini menjelaskan hubungan apa yang terjadi antar kelasnya, bukan menjelaskan apa yang akan terjadi saat kelasnya berhubungan. Fungsi utama dari Class diagram adalah untuk merepresentasikan struktur sistem secara jelas. Gambar 2.4 berikut merupakan contoh dari class diagram.



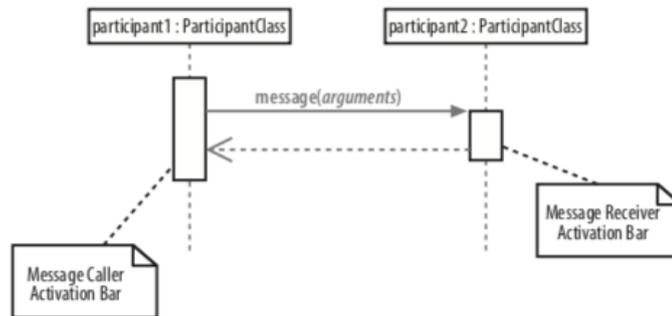
Gambar 2.4 Contoh class diagram

Sumber gambar : <https://www.lucidchart.com/pages/uml-class-diagram>

### 2.23.4 Sequence Diagram

Sequence diagram menggambarkan proses rinci bagaimana proses berjalan sesuai objek dan skenarionya sehingga suatu kebutuhan dapat terpenuhi. Interaksi antar objek yang ada dalam satu kebutuhan digambarkan secara terperinci, seperti pesan atau perintah apa yang dikirim dan kapan waktu pelaksanaannya. Diagram

sequence bertujuan untuk mengetahui urutan interaksi dari suatu proses sehingga output yang dihasilkan sesuai dengan kebutuhan. Gambar 2.5 berikut merupakan contoh dari sequence diagram.



Gambar 2.5 Contoh sequence diagram