

## **PENGEMBANGAN SISTEM MANAJEMEN WHATSAPP API (*APPLICATION PROGRAMMING INTERFACE*) DENGAN MENGGUNAKAN WHATSAPP-WEB.JS**

*Development of WhatsApp API (Application Programming Interface) Management System using  
Whatsapp-web.js*

**<sup>1</sup>Faisal Tifta Zany, <sup>2</sup>Amru Sujud, <sup>3</sup>Juanda Nurgaza**  
Prodi S1 Informatika, Fakultas Ilmu Komputer, Universitas Ubudiyah Indonesia  
Corresponding Author : [juanda.nurgaza@gmail.com](mailto:juanda.nurgaza@gmail.com)

### **Abstrak**

WhatsApp sebagai salah satu Platform Aplikasi Pesan Instant yang paling banyak digunakan saat ini memiliki potensi bagus sebagai media penyebar informasi yang paling efektif. Pemanfaatan WhatsApp dapat digunakan dalam berbagai kebutuhan baik bagi pelaku bisnis maupun organisasi pemerintahan. Kebutuhan - kebutuhan tersebut meliputi, media iklan, media pembelajaran, media sosial, notifikasi *One-Time Password (OTP)*, notifikasi pemberitahuan pajak atau tagihan pajak, Layanan *front desk online* dan notifikasi agenda kegiatan. Pada penelitian ini digunakan Metode Air Terjun (*waterfall*) dalam pengembangan perangkat lunak dimana telah dilakukan pengembangan pada *Back End* WhatsApp API dengan menggunakan Restful (*Representational State Transfer API*). Pengembangan ini menggunakan kode sumber terbuka (*open source*) whatsapp-web.js untuk mengimplementasikan fitur-fitur yang ada pada WhatsApp. Penelitian ini telah berhasil mengembangkan Sistem Manajemen WhatsApp API yang dapat digunakan sebagai penghubung antar aplikasi dalam rangka memenuhi kebutuhan-kebutuhan yang telah dijelaskan sebelumnya. Adapun *End Point* utama yang telah dikembangkan sebagai representasi fitur pengiriman pesan yang ada pada WhatsApp adalah pengiriman pesan teks, pengiriman pesan gambar, file dan lokasi serta beberapa fitur manajemen user lainnya.

**Kata kunci:** WhatsApp, API, Restful, Whatsapp-web.js, Back End

### **Abstract**

WhatsApp as one of the most widely used Instant Messaging Application Platforms today has great potential as the most effective information dissemination media. WhatsApp can be used for various needs for both business people and government organizations. These needs include advertising media, learning media, social media, *One-Time Password (OTP)* notifications, tax notification or tax bill notifications, online front desk services and activity agenda notifications. This study uses the waterfall method in software development where WhatsApp API Back End Development has been carried out using Restful (*Representational State Transfer API*). This development uses the opensource code Library Whatsapp-web.js to implement the features available on WhatsApp. This study has succeeded in developing the WhatsApp API Management System that can be used as a liaison between applications in order to meet the needs that have been explained previously. The main endpoints that have been developed as a representation of the messaging features available on WhatsApp are sending text messages, sending image messages, files and locations as well as several other user management features.

**Keywords:** WhatsApp, API, Restful, Whatsapp-web.js, Back End

---

## PENDAHULUAN

Diawali dengan perkembangan teknologi GSM (*Global System for Mobile Communications*), Pesan Teks mulai dikenal oleh hampir sebagian besar masyarakat dunia. Pada tahun 1993 perusahaan Nokia yang saat itu berlokasi di Finlandia pertama kali membuat ponsel yang dilengkapi kemampuan melakukan pengiriman Pesan Teks atau *Short Message Service* (SMS). Walau pada awalnya tidak begitu banyak digunakan namun sejalan dengan kebutuhan komunikasi yang makin tinggi SMS makin banyak dikenal dan digunakan oleh hampir sebagian orang. Jika dilihat pada pengguna ponsel di Amerika Serikat pada 1995 rata-rata hanya mengirim 0,4 pesan per bulan. Namun seiring waktu angka tersebut naik, pada tahun 2000 pengguna ponsel Amerika Serikat rata-rata mengirim 35 pesan per bulan.

Pada akhir tahun 2000, pengguna Inggris mengirim rata-rata 35 pesan teks per bulan dan pada tahun 2010, dilaporkan bahwa 6,1 triliun pesan dikirim secara global. Ini menunjukkan bahwa penggunaan SMS sudah kian masif digunakan.

Pesan Teks dalam perkembangannya memiliki banyak keterbatasan walaupun sudah banyak digunakan. Di dalam berkomunikasi masyarakat tidak hanya menginginkan agar pesan mereka dapat tersampaikan dengan baik tetapi juga layaknya surat biasa yang di dalamnya terdapat banyak lampiran berkas lain seperti foto atau dokumen – dokumen tambahan lainnya, pesan dapat dikirimkan berupa kumpulan informasi berupa teks dan lampiran tambahan tersebut [1].

Masyarakat saat ini juga telah secara luas mengenal dan menggunakan layanan aplikasi pesan instant. Pesan Instan dikenal tidak hanya sebagai media untuk bertukar informasi secara cepat tetapi juga merupakan sarana yang dapat mendukung dalam berbagai aspek kebutuhan seperti, media iklan, media pembelajaran, media sosial, notifikasi *One Time Password* (OTP), notifikasi pemberitahuan pajak atau tagihan pajak, layanan *front desk online* dan notifikasi agenda kegiatan. Walaupun pesan teks yang telah dijelaskan sebelumnya tidak memiliki keterkaitan latar belakang pengembangannya dengan pesan instant ini, namun penggunaannya sudah memiliki peran sebagai

alternatif pengganti dari teks ke multimedia bahkan menjadi aplikasi yang paling digemari oleh kalangan anak – anak remaja hingga orang dewasa.

Pada tahun 2024, jumlah total orang yang menggunakan aplikasi pesan instant ini telah melampaui angka 5 miliar. Angka ini mencakup berbagai platform, mulai dari yang terkenal seperti WhatsApp, Facebook Messenger, WeChat, dan Telegram hingga aplikasi yang lebih spesifik secara regional yang melayani preferensi budaya dan bahasa yang berbeda. Tercatat 2,4 miliar pengguna aktif bulan dengan WhatsApp sebagai pengguna terbanyak [1]. Hal ini menunjukkan bahwa saat ini hampir sebagian besar orang di seluruh dunia memiliki akun WhatsApp. Dengan demikian, Aplikasi Whatsapp merupakan aplikasi berbasis internet yang paling populer yang dapat digunakan sebagai media komunikasi. Selain mudah dan populer, aplikasi ini didesain agar setiap pengguna dapat saling berbagi informasi dan berbagai konten sesuai dengan fitur pendukungnya [2].

Whatsapp menjadi salah satu media sosial yang paling populer yang digunakan oleh masyarakat Indonesia. Penggunaan whatsapp sebagai media sosial hampir sering digunakan orang dalam kehidupan sehari-hari [3].

Saat ini, ada banyak aplikasi yang memanfaatkan API (*Application Programming Interface*) dalam rangka mendukung proses bisnis yang berlaku pada suatu organisasi/badan hukum/Perusahaan. API sendiri merupakan sebuah aplikasi antarmuka pemrograman yang biasanya digunakan pada dua sistem komputer yang berbeda untuk memberikan jaminan keamanan saat bertukar dan berbagi informasi melalui jaringan atau koneksi internet (<https://course-net.com/blog/apa-itu-restful-apis-pengertian-panduan-cara-kerjanya/>). Salah satu jenis API saat ini yang paling banyak diimplementasikan dan sederhana dalam pengembangannya adalah Restful (*Representational State Transfer Application Programming Interface*) yang dapat dimanfaatkan dalam rangka menghubungkan 2 aplikasi/layanan yang berbeda dalam melakukan pertukaran informasi, hal ini sangatlah ideal bagi aplikasi apa pun untuk memanfaatkan fitur – fitur yang ada pada WhatsApp.

Pemanfaatan itu dapat berupa automatisasi pesan instant yang terintegrasi dengan aplikasi apa pun seperti yang telah dijelaskan sebelumnya. WhatsApp API memungkinkan pelaku bisnis untuk mengintegrasikan fungsionalitas perpesanan WhatsApp ke dalam sistem mereka. Artinya, pelaku bisnis dapat menggunakan WhatsApp untuk mengirim dan menerima pesan dengan pelanggannya sama seperti aplikasi perpesanan lainnya. API ini memberi pelaku bisnis cara yang aman dan andal untuk berkomunikasi dengan pelanggan melalui WhatsApp, memungkinkan otomatisasi layanan pelanggan. Untuk mengakses layanan resmi WhatsApp API, pengguna harus membayar dengan harga tertentu. Dengan kata lain, WhatsApp API resmi adalah layanan berbayar.

Oleh karenanya, peneliti menganggap diperlukan suatu pengembangan sebuah sistem manajemen dengan basis backend yang baik dan andal yang dapat menyajikan manajemen layanan API dan dapat diintegrasikan terhadap aplikasi apa pun dalam rangka memanfaatkan fitur layanan pesan instant pada Whatsapp secara mandiri atau tanpa menggunakan layanan berbayar.

## I. STUDI PUSTAKA

### 1. *Library Puppeteer*

Puppeteer adalah *library* Node.js yang dikembangkan oleh Google yang memungkinkan pengguna mengontrol Chrome tanpa kepala (*headless/no visible UI*) melalui DevTools Protocol. Puppeteer adalah utilitas atau alat bantu untuk mengotomatisasi pengujian aplikasi berbasis web dengan menggunakan Chrome tanpa kepala atau perangkat Chromebit, tanpa memerlukan ekstensi browser atau Selenium WebDriver atau PhantomJS. Sebuah contoh sederhana menjalankan Library ini adalah sebagai berikut :

```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://google.com');
  await page.screenshot({path: 'google.png'});
});
```

```
await browser.close();
})();
```

Program diatas akan menjalankan chromium secara headless, lalu membuka halaman web google.com dan melakukan tangkapan layar yang selanjutnya filenya disimpan dengan nama google.png.

### 2. *Library Whatsapp-web.js*

Prinsip kerja *library* ini adalah menjalankan aplikasi Whatsapp Web pada *browser* dan mengelola aplikasi tersebut dengan menggunakan *instant object* dari *library* Puppeteer. Whatsapp API klien terkoneksi melalui Whatsapp Web Browser dan mengakses fungsi – fungsi internal yang dimiliki oleh *library* Puppeteer tersebut [4]. Dengan demikian developer memiliki akses pada hampir semua fitur – fitur yang terdapat pada Whatsapp Web. Namun terdapat hal yang harus diketahui bahwa mengakses fitur – fitur Whatsapp Web dengan memanfaatkan *library* Puppeteer berpotensi terjadi pemblokiran terhadap penggunaan akun maupun aplikasi whatsapp web tersebut. Hal ini dikarenakan Whatsapp tidak mengizinkan pengguna melakukan pengiriman pesan *bots* atau pesan otomatis yang dilakukan secara berulang atau adanya akses tidak resmi pada platform mereka. *library* whatsapp-web.js sendiri merupakan alat bantu yang tidak dirilis oleh Whatsapp melainkan oleh individu yang kemudian dikembangkan lebih lanjut oleh komunitas tertentu yang bersifat sumber terbuka (*Open Source*). Oleh karenanya, *library* whatsapp-web.js adalah *library* tidak resmi (*unofficial*) namun telah digunakan secara luas dan sukses oleh banyak pengembang. Berikut adalah contoh sederhana untuk menjalankan klien whatsapp dengan proses scan qr dan *stand by* apabila menerima pesan dengan teks “!ping” maka secara otomatis akan membalas “pong” kepada pengirim.

```
const { Client } = require('whatsapp-web.js');

const client = new Client();
client.on('qr', (qr) => {
  console.log('QR RECEIVED', qr);
});
client.on('ready', () => {
  console.log('Client is ready!');
```

```
});
client.on('message', msg => {
  if (msg.body == '!ping') {
    msg.reply('pong');
  }
});
client.initialize();
```

Constructor Client juga dapat dibuat dengan melakukan inisiasi parameter authStrategy sebagai contoh :

```
const client = new Client({
  authStrategy: new LocalAuth({
    clientId: clientId,
    dataPath: './session.json',
  })
});
```

Contoh diatas menjelaskan bahwa data session disimpan pada folder lokal session.json dan diberi nama sama dengan nilai parameter clientId.

Fitur – fitur yang didukung dapat dilihat pada Tabel 1 berikut :

Tabel 1. Fitur yang didukung whatsapp-web.js

Fitur	Status
Multi Device	<input checked="" type="checkbox"/>
Send messages	<input checked="" type="checkbox"/>
Receive messages	<input checked="" type="checkbox"/>
Send media (images/audio/documents)	<input checked="" type="checkbox"/>
Send media (video)	<input checked="" type="checkbox"/> <a href="#">(requires Google Chrome)</a>
Send stickers	<input checked="" type="checkbox"/>
Receive media (images/audio/video/documents)	<input checked="" type="checkbox"/>
Send contact cards	<input checked="" type="checkbox"/>
Send location	<input checked="" type="checkbox"/>
Send buttons	<input checked="" type="checkbox"/> <a href="#">(DEPRECATE D)</a>
Send lists	<input checked="" type="checkbox"/> <a href="#">(DEPRECATE D)</a>
Receive location	<input checked="" type="checkbox"/>

- Message replies
- Join groups by invite
- Get invite for group
- Modify group info (subject, description)
- Modify group settings (send messages, edit info)
- Add group participants
- Kick group participants
- Promote/demote group participants
- Mention users
- Mention groups
- Mute/unmute chats
- Block/unblock contacts
- Get contact info
- Get profile pictures
- Set user status message
- React to messages
- Create polls
- Vote in polls
- Communities
- Channels

Fitur yang paling sering digunakan dalam whatsapp-web.js diimplementasikan pada Class Client dengan fungsi sebagai berikut :

```
/**
 * Send a message to a specific chatId
 * @param {string} chatId
 * @param
 {string|MessageMedia|Location|Poll|Contact|Array<
 Contact>|Buttons|List} content
 * @param {MessageSendOptions} [options] -
 Options used when sending the message
 *
 * @returns {Promise<Message>} Message that
 was just sent
 */
async sendMessage(chatId, content, options = {})
```

dimana :

- nilai chatId adalah nomor telp dengan format : (YY) + 8XXXXXXXXX (Kode Indonesia : 62)

- nilai content adalah pesan teks yang ingin dikirimkan
- options adalah variable tambahan

### 3. JSON Web Token (JWT)

JSON Web Token (JWT) adalah keamanan berbasis JSON token encoding yang memungkinkan identitas dan keamanan informasi untuk dibagikan di seluruh domain keamanan [5]. JSON Web Token atau biasa disebut dengan JWT adalah standar terbuka (RFC 7519) yang mendefinisikan cara untuk mentransmisikan informasi dengan aman antar pihak sebagai objek JSON. Informasi ini dapat diverifikasi dan dipercaya karena ditandatangani secara digital. JWT dapat ditandatangani menggunakan sebuah secret (dengan algoritma HMAC) atau pasangan kunci publik/pribadi menggunakan RSA atau ECDSA. Sederhananya JWT merupakan Web Token yang berupa JSON dan umumnya digunakan untuk proses *authentication*. Dimana untuk mendapatkan suatu data didalam server kita harus menyertakan token tersebut sebagai kuncinya. Kelebihan yang ditawarkan oleh JWT adalah stateless. Stateless JWT memungkinkan pengguna untuk berbagi, menggunakan Kembali dan control akses ke sumberdaya [6].

JWT digunakan pada kondisi :

- *Authentication*,  
 Kebutuhan ini dilakukan ketika pengguna melakukan authentication dan mendapatkan token, maka setiap permintaan berikutnya akan menyertakan token tersebut, dan memungkinkan pengguna untuk mengakses route, service, dan resources yang diizinkan
- *Pertukaran Informasi*  
 JWT merupakan cara yang baik untuk mengirimkan informasi antar pihak dengan aman. Dengan token yang sudah ditandatangani dengan algoritma RSA, maka kita bisa tahu siapa yang melakukan request tersebut.

### 4. PM2

PM2 merupakan manajer proses produksi Node.js yang memungkinkan pengguna untuk menjalankan aplikasi Node.js dalam lingkungan produksi. Kelebihan PM2 terletak pada kemampuannya dalam mengelola proses, memantau performa, dan memperbarui aplikasi tanpa adanya *downtime*.

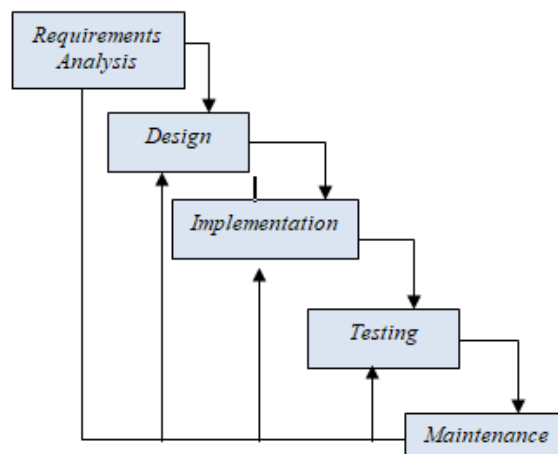
Fitur-fitur Utama PM2:

1. Manajemen Proses yang Efisien: Memulai, menghentikan, dan mengelola proses aplikasi Node.js dengan mudah.
2. Pemantauan Performa Aplikasi: Statistik performa seperti penggunaan CPU, memori, dan log aplikasi yang terkumpul.
3. Pemulihan Otomatis saat Terjadi Kegagalan: Memulai ulang aplikasi secara otomatis saat terjadi crash.
4. Skalabilitas dengan Fitur Clustering: Menjalankan beberapa instance aplikasi untuk menyeimbangkan beban.
5. Manajemen Log yang Efisien: Akses log dari setiap proses yang berjalan secara terstruktur.

## II. METODE

### 1. Tahapan Penelitian

Penelitian ini mengadopsi Metode Air Terjun untuk pengembangan perangkat lunak. Langkah-langkah untuk melakukan metode ini ditunjukkan pada Gambar 1.



Gambar 1. Metode Siklus Waterfall

#### 1. Requirements Analysis

Tahapan ini melakukan identifikasi komponen *hardware*, *software* yang diperlukan dalam pembuatan Whatsapp API. Identifikasi bertujuan menentukan metode yang efektif dan efisien dalam membuat sebuah manajemen Whatsapp API. Metode yang digunakan adalah implementasi *backend* Restfull API.

## 2. Design

Pembuatan model sesuai dengan identifikasi metode yang telah analisa pada tahap sebelumnya berupa aktor yang berperan dan aliran data.

## 3. Implementation

Pengaturan Lingkungan Pengembangan (Environment Set Up) berupa instalasi nodejs, instalasi paket library, instalasi PM2 dan Pengkodean.

## 4. Testing

Ujicoba Autentikasi dan Otorisasi terhadap user yang didaftarkan sesuai dengan grup user. Ujicoba dilakukan pada seluruh end point dan memastikan hasil yang diperoleh sudah sesuai dengan yang sebenarnya.

## 5. Maintenance

Pada fase ini, dilakukan pemeliharaan pada sistem manajemen whatsapp api dan memastikan fungsionalitas sistem berfungsi dengan baik.

## 2. Instrument Penelitian

Alat dan bahan yang digunakan dalam proses pembuatan sistem informasi ini adalah sebagai berikut :

### A. Perangkat Keras

Satu unit perangkat komputer dan smartphone dengan spesifikasi cukup digunakan dalam penelitian ini. Perangkat unit komputer yang digunakan adalah dengan processor minimal core i3, RAM minimal 4Gb sementara Smartphone menggunakan android minimal 5.0.

### B. Perangkat Lunak

#### 1. Visual Studio Code

Visual Studio Code merupakan teks editor yang digunakan dalam proses pengkodean (coding) aplikasi.

#### 2. Node js

Pada penelitian ini menggunakan Node JS versi 20.15.0 dimana dipastikan agar dapat mendukung seluruh Library yang digunakan.

#### 3. SQL Server 2008

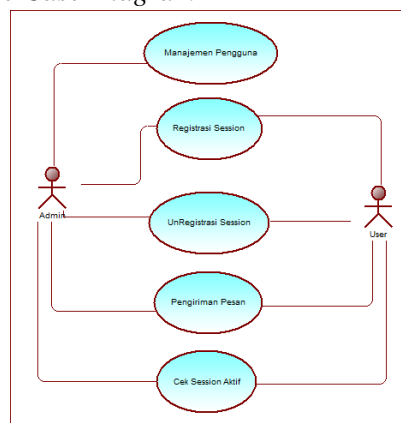
Penggunaan database diperlukan didalam melakukan penyimpanan data agar sistem manajemen user dapat dikelola dengan baik.

## 3. Analisis Sistem

Pengembangan difokuskan pada implementasi Whatsapp API dengan menggunakan library whatsapp-web.js. Penggunaan library membutuhkan beberapa analisa dalam memastikan dependensi dan kompatibilitas versi antara satu dan lainnya dapat berjalan dengan baik. Selain itu, dilakukan pengelompokan otorisasi hak akses terhadap grup yang ada. Pengembangan juga memastikan aspek keamanan dapat terjamin ketika akses pada semua end point dilakukan.

## 4. Perancangan Sistem

### 1. Use Case Diagram

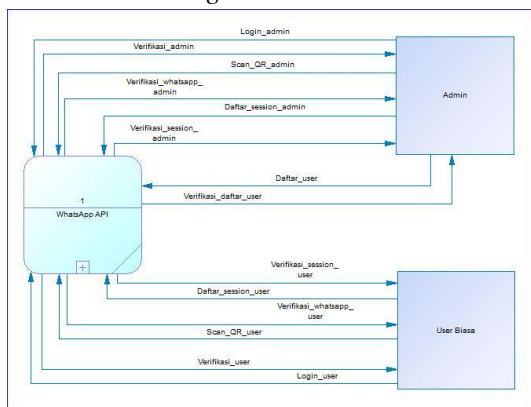


Gambar 2. Use Case Diagram

Pada Use Case ini terdapat hubungan 2 aktor yang saling berinteraksi terhadap Sistem Manajemen Whatsapp API ini, yaitu Admin dan User biasa. Admin memiliki otoritas lebih banyak dibanding User biasa, dimana hanya Admin yang dapat melakukan pengelolaan manajemen pengguna.

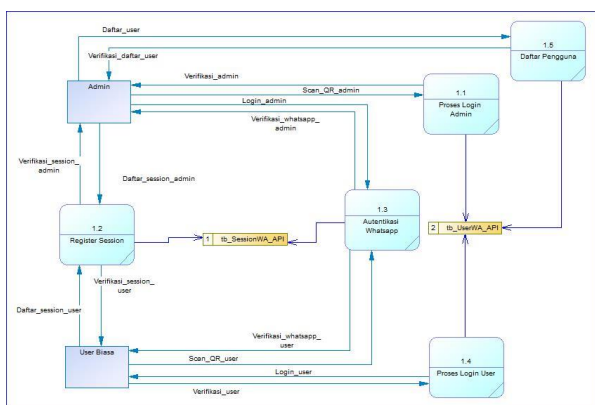


## 2. Data Flow Diagram



Gambar 3. Context Diagram

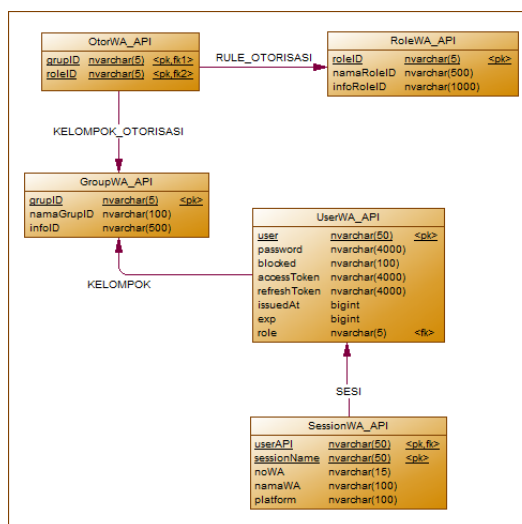
Pada *Context Diagram* menjelaskan proses aliran data yang dilakukan oleh 2 pengguna, yaitu Admin dan User biasa. Proses pendaftaran pengguna hanya bisa dilakukan oleh Admin selain itu diakses oleh User biasa.



Gambar 4. Data Flow Diagram (DFD) Level 1

Pada DFD Level 1 ini menjelaskan rincian dari aliran data sebelumnya. Semua aliran data yang terjadi tersimpan dan diakses pada 2 tabel yaitu : `tb_SessionWA_API` dan `tb_UserWA_API`.

## 3. Entity Relationship Diagram (ERD Physical Model)



Gambar 5. Entity Relationship Diagram

Terdapat 5 entitas pada *Entity Relationship Diagram*, yaitu UserWA\_API, GroupWA\_API, RoleWA\_API, OtorWA\_API dan SessionWA\_API. UserWA\_API berupa data seluruh pengguna yang sudah didaftarkan. GroupWA\_API adalah kelompok grup yang dapat mengakses API. RoleWA\_API merupakan seluruh daftar akses end point yang ada. OtorWA\_API adalah pembatasan akses end point terhadap masing – masing grup dan SessionWA\_API adalah daftar pengguna dengan *session* yang telah didaftarkan.

## III. HASIL DAN PEMBAHASAN

### 1. Alur Pengembangan

Implementasi library `whatsapp-web.js` membutuhkan eksplorasi yang lebih lanjut dari sekedar contoh sederhana terdahulu. Penambahan dan pengembangan lanjutan yang telah dilakukan adalah sebagai berikut :

- Inisialisasi Paket *Library* Pendukung  
 Adapun *Library* yang digunakan dalam pengembangan ini adalah sebagai berikut :

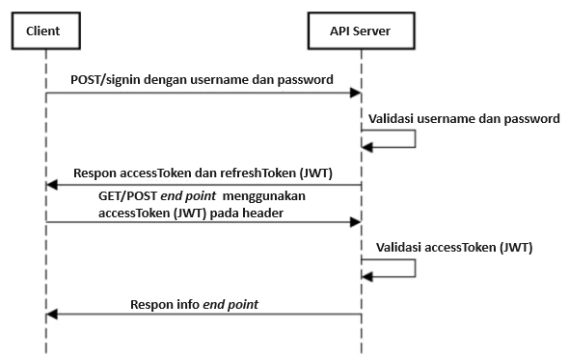
Tabel 2. Paket Library Pendukung

Nama <i>Library</i>	Fungsi
<code>bcrypt@5.1.1</code>	Utilitas algoritma password hashing
<code>cors@2.8.5</code>	Pengelolaan Cross-Origin Resource Sharing

dotenv@16.3.1	Modul untuk memuat variabel lingkungan dari file .env ke process.env
express@4.18.2	Framework aplikasi web Node.js yang minimal dan fleksibel untuk menjalankan aplikasi web dan seluler
fs@0.0.1-security	Modul yang menyediakan akses ke file system
jsonwebtoken@9.0.2	Modul implementasi JWT
path@0.12.7	Modul akses path (lokasi file) ke Sistem Operasi
<a href="#">qrcode@1.5.3</a>	Modul konversi string menjadi QR Code <i>image</i>
rimraf@5.0.5	Modul utilitas Perintah UNIX rm -rf untuk node dalam implementasi lintas platform
<a href="#">sequelize@6.35.2</a>	Modul koneksi dan akses database
whatsapp-web.js@1.23.1-alpha.5	Modul utama akses fitur WhatsApp Web. Didalamnya sudah termasuk Library Puppeteer

- Pengembangan Aspek Keamanan/Autentikasi  
 Sistem API ini dibangun dengan menggunakan autentikasi JWT dimana setiap akses end point pengguna diharuskan untuk menyertakan Token pada headernya. Token ini memiliki masa kadaluarsa yang telah di tentukan dari Sistem API. Apa bila Token sudah melewati masa kadaluarsa maka akses pada setiap end point akan ditolak. Token JWT diperoleh ketika pengguna melakukan login menggunakan end point manage/loginUserAPI. Respon dari end point ini akan berisi variabel tokenRefresh dan variable tokenAccess. Variabel tokenRefresh

digunakan apabila tokenAccess sudah kadaluarsa dan ingin diperbaharui. Pembaharuan ini dilakukan pada end point manage/refreshTokenAPI yang sekanjutnya respon akan menghasilkan tokenAccess yang baru. Variabel tokenAccess digunakan apa bila pengguna ingin melakukan akses pada setiap end point yang ada kecuali end point manage/loginUserAPI dan manage/refreshTokenAPI. Berikut diagram sekuensial yang menjelaskan proses autentikasi dengan menggunakan JWT :



Gambar 6. Diagram Sekuensial Autentikasi JWT

- Pengembangan Pendaftaran Pengguna dan Grup Otorisasi  
 Setiap kelompok pengguna baik Admin maupun User biasa memiliki akses tertentu terhadap end point yang ada. Pengelompokan akses tersebut ditentukan pada tabel OtorWA\_API. Pengelompokan akses berdasarkan pengelolaan pengguna dan session dibedakan atas tabel berikut :

Tabel 3. Pengelompokan Akses End Point berdasarkan Pengelolaan Pengguna dan Session

Kelompok Pengguna	End Point
Admin	manage/loginUserAPI
Admin	session/isSessionActive
Admin	manage/refreshTokenAPI
Admin	manage/listUserAPI
Admin	manage/createUserAPI
Admin	manage/deleteUserAPI
Admin	manage/blockUserAPI



Admin	manage/logoutUserAPI
Admin	session/getAllSessions
User Biasa	manage/loginUserAPI
User Biasa	manage/isSessionActive
User Biasa	manage/refreshTokenAPI
User Biasa	manage/logoutUserAPI

- Pengembangan Multi Session

Akses pengguna yang sama pada WhatsApp secara normal mengizinkan 5 pengguna termasuk login pada perangkat *smartphone/pc*. Hal ini tentu membutuhkan pengembangan secara lebih khusus agar API dapat mengakomodir hal yang sama dengan penggunaan WhatsApp. Oleh karenanya, dibutuhkan pembuatan Class baru untuk dapat menyimpan kumpulan instance dari implementasi sebuah Whatsapp client seperti yang sudah dijelaskan sebelumnya. Adapun langkah – langkah yang telah dilakukan adalah :

1. Membuat Class *whatsappWebSession* yang berfungsi mengimplementasikan *single* WhatsApp dengan membuat constructor parameter berupa ID, sebagai ID unik penaman folder session dan parameter Client menggunakan LocalAuth
2. Membuat Class *WhatsappSessionManager* yang berfungsi menyimpan semua instance object dari Class *whatsappWebSession* pada langkah 1 atau *single* WhatsApp. Pada Class *WhatsappSessionManager* ini perlu di buat beberapa *methods* yang dapat membaca semua folder session dan menyimpannya sebagai *list* dari *object instant* *whatsappWebSession*.

2. End Point dan Ujicoba

Ada sebanyak 24 End Point yang telah diimplementasikan dengan menggunakan metode Restful API.

Tabel 4. Daftar End Point

URI	Meth od	Entit as	Hasil
manage/listUserAPI	POST	Admin	Respon JSON berhasil

			menampil kan seluruh Daftar User dengan Session dan Token
manage/loginUserAPI	POST	User , Adm in	Respon JSON berhasil menampil kan User dengan Grup dan Token
manage/logoutUserAPI	DEL	User , Adm in	Respon JSON berhasil menampil kan Logout berhasil
manage/deleteUserAPI	DEL	Adm in	Respon JSON berhasil menampil kkan User telah dihapus
manage/blockUserAPI	POST	Adm in	Respon JSON berhasil menampil kan user telah diblok
manage/createUserAPI	POST	Adm in	Respon JSON berhasil menampil kan nama user dan token
manage/refreshTokenAPI	GET	User ,	Respon JSON

		Admin	berhasil menampilkan Token Akses dan Token user yang diproses
session/start	POST	User, Admin	Respon JSON berhasil menampilkan session instance berhasil dibuat
session/getQR	POST	User, Admin	Respon JSON berhasil menampilkan gambar QR Code untuk melakukan registrasi WA
session/stop	POST	User, Admin	Respon JSON berhasil menampilkan session telah dihapus
session/getAllSessions	POST	Admin	Respon JSON berhasil menampilkan daftar semua session yang aktif
session/isSessionActive	POST	User,	Respon JSON

		Admin	berhasil menampilkan status session aktif atau tidak
group	GET	User, Admin	Respon JSON berhasil menampilkan daftar grup yang terdaftar pada no kontak pada session tersebut
group/:Nama Group/media	POST	User, Admin	Respon JSON berhasil menampilkan status pesan gambar/file ke grup berhasil dikirimkan
group/:Nama Group	POST	User, Admin	Respon JSON berhasil menampilkan status pesan teks ke grup berhasil dikirimkan
contacts/:phone	GET	User, Admin	Respon JSON berhasil menampilkan apakah No. Kontak

			yang di cek merupakan data kontak dari No. Kontak yang diregistrasi pada session tersebut
contacts/:phone/is_registered	GET	User , Admin	Respons JSON berhasil menampilkan status apakah No. Kontak yang ingin dicek tersebut terdaftar sebagai pengguna WhatsApp
contacts/:phone/profile_picture	GET	User , Admin	Response JSON berhasil menampilkan link download gambar profile dari No. Kontak yang di cek
chat	GET	User , Admin	Response JSON berhasil menampilkan chat terakhir dari semua

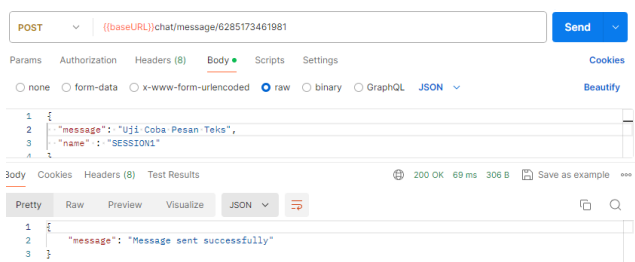
			kontak pribadi atau pun grup yang ada saat ini
chat/:phone	GET	User , Admin	Response JSON berhasil menampilkan chat terakhir dari No. Kontak yang di cek
chat/message/:phone	POST	User , Admin	Response JSON berhasil menampilkan pesan teks telah dikirim
chat/media/:phone	POST	User , Admin	Response JSON berhasil menampilkan pesan gambar/file telah dikirim
chat/location/:phone	POST	User , Admin	Response JSON berhasil menampilkan pesan lokasi telah dikirim

Berikut Ujicoba pengiriman pesan, yaitu :

1. Pengiriman Pesan Teks

Skenario Pengiriman dilakukan dengan menggunakan Postman kemudian hasil pesan di cek dengan menggunakan WhatsApp Web. Berikut No. Kontak Akun WhatsApp yang digunakan :

No. Pengirim : 6281360120053  
 No. Penerima : 6285173461981



Gambar 6. Kirim Pesan Teks dengan Postman

Hasil respon yang diterima menunjukkan pesan berhasil dikirim. Tampilan WhatsApp pada No. Penerima adalah sebagai berikut :

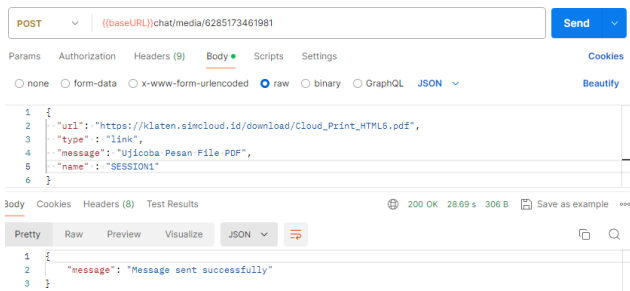


Gambar 7. Pesan Teks yang diterima

2. Pengiriman Pesan File

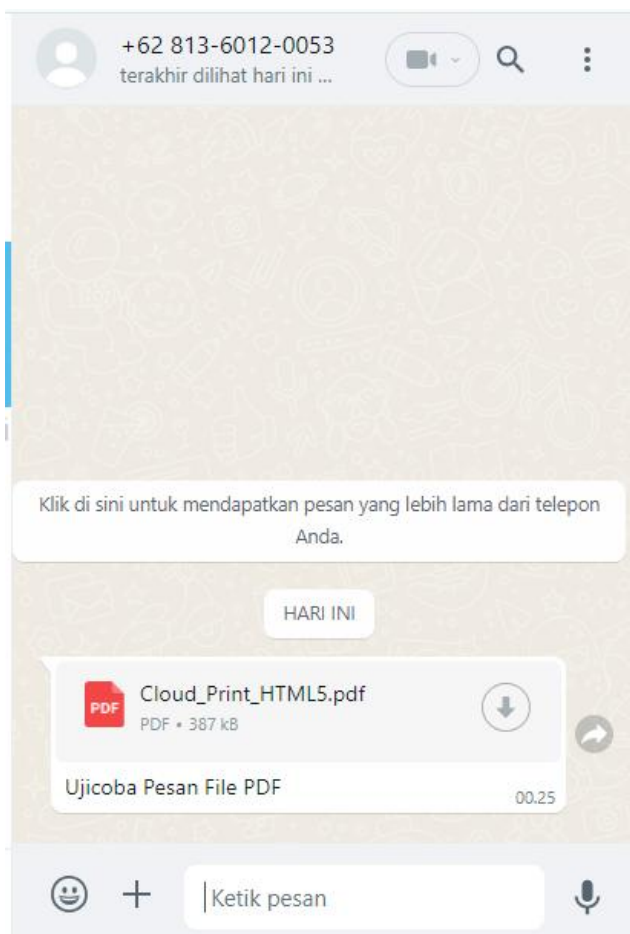
Pengiriman dilakukan dengan scenario yang sama pada ujicoba sebelumnya, namun

pengiriman ini ada penambahan lampiran file format PDF.



Gambar 8. Kirim Pesan File dengan Postman

Hasil respon pada Postman menunjukkan pesan berhasil dikirim. Untuk tampilan WhatsApp pada No. Penerima adalah sebagai berikut :

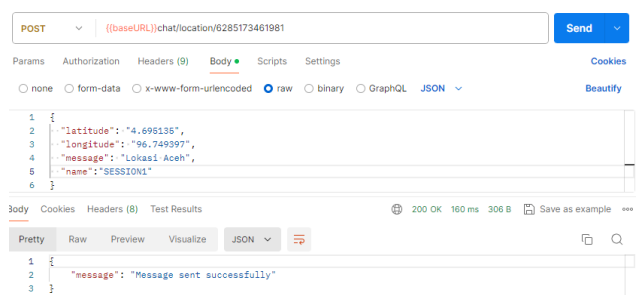


Gambar 9. Pesan File yang diterima

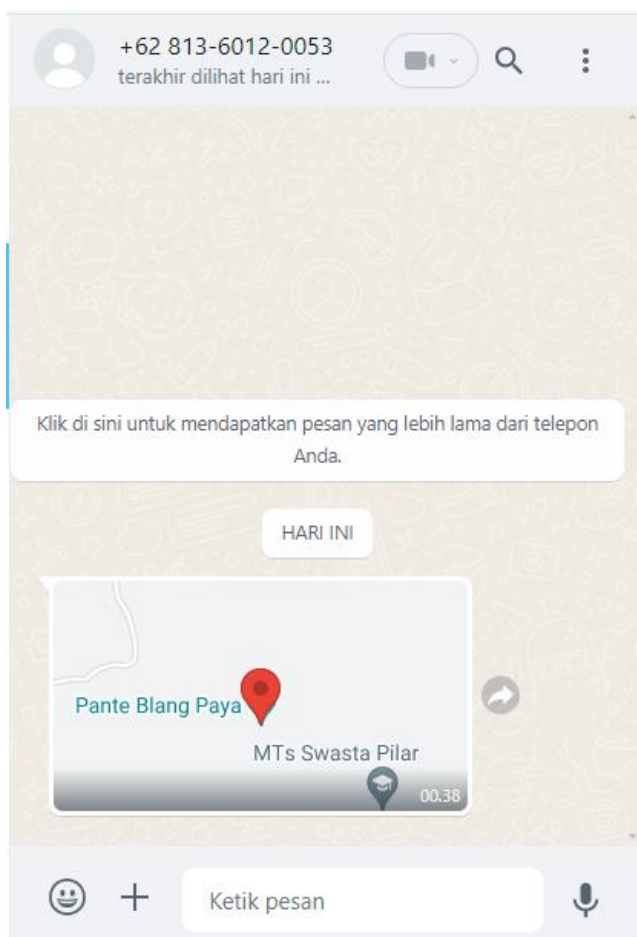
3. Pengiriman Pesan Lokasi

Ujicoba pesan lokasi masih menggunakan skenario yang sama seperti ujicoba

sebelumnya. Pengiriman berupa titik lokasi tertentu.



Gambar 10. Pesan Lokasi yang diterima Respon dari ujicoba Postman menunjukkan pengiriman lokasi telah berhasil dilakukan. Hasil yang diterima pada WhatsApp No. Penerima adalah sebagai berikut :



Terdapat beberapa end point lainnya yang sukses diujicobakan atau sudah sesuai dengan hasil yang diharapkan. Sistem Manajemen WhatsApp API ini secara keseluruhan sudah bisa digunakan dengan fitur

– fitur yang hampir mendekati Fitur yang ada pada Aplikasi WhatsApp. WhatsApp API yang di kembangkan sudah dapat digunakan dan diAplikasikan oleh pihak manapun yang ingin memanfaatkan WhatsApp untuk berbagai keperluan. Dengan demikian API WhatsApp ini memiliki fleksibilitas dan interoperabilitas untuk dapat diintegrasikan dengan Aplikasi dengan platform atau Bahasa pemrograman apa saja sehingga sangat terbuka bagi siapapun yang ingin memanfaatkan fitur – fitur yang ada. Sistem Manajemen WhatsApp API ini memiliki potensi yang besar untuk pengembangan lanjutan tidak hanya sebatas sarana pengiriman teks saja tetapi juga bisa sebagai media iklan, media pembelajaran, media sosial, notifikasi *One Time Password* (OTP), notifikasi pemberitahuan pajak atau tagihan pajak, layanan *front desk online* dan notifikasi agenda kegiatan seperti yang telah dijelaskan diawal.

#### IV. KESIMPULAN

Manajemen dan pengiriman pesan singkat telah berhasil diimplementasi dengan baik sesuai dengan hasil yang diharapkan. Sistem Manajemen WhatsApp API ini merupakan penelitian yang menerapkan metode restful API dengan melakukan implementasi Library Whatsapp-web.js. Pada aspek keamanan data Sistem Manajemen WhatsApp API ini menerapkan autentikasi JWT sehingga dapat mencegah dari akses – akses yang tidak berwenang mengingat No. Kontak Pengguna WhatsApp saat ini sangat rentan untuk disalah gunakan. Dari Sisi Otorisasi, kelompok pengguna juga merupakan bagian penting dari pengelolaan hak akses terhadap end point yang telah di kembangkan. Oleh karenanya tidak semua kelompok pengguna di berikan akses end point yang sama mengingat nantinya pengguna - pengguna tersebut bisa jadi adalah badan organisasi yang ingin memanfaatkan fitur – fitur sehingga tidak akan saling mengganggu privasi data akun – akun WhatsApp antar satu dan lainnya. Pengembangan dengan menggunakan Library whatsapp-web.js ini bersifat terbuka karena tidak berbayar dan adanya dukungan komunitas yang terus bertambah.

**REFERENSI**

- [1] Dharmik Joshi . (View July, 2024). Aplikasi Perpesanan Paling Populer di Tahun 2024 [Online]. Available: <https://controlhippo.com/blog/id/popular-messaging-apps/>
- [2] Rahartri., "WhatsApp" Media Komunikasi Efektif Masa Kini (Studi Kasus Pada Layanan Jasa Informasi Ilmiah di Kawasan Puspiptek). 2019. *Visi Pustaka*, 21(2), 147–156. <https://doi.org/10.37014/visipustaka.v21i2.552>
- [3] Firmansyah, F., Kejora, M. T. B., & Akil, A. Studi Analisis Pemanfaatan Whatsapp dalam Pembelajaran Daring Akidah Akhlak pada Siswa Madrasah Aliyah. *Edukatif : Jurnal Ilmu Pendidikan*. (2021). 3(5), 2886–2897. <https://doi.org/10.31004/edukatif.v3i5.1018>
- [4] Pedro S. Lopez. (View July 2024). *Whatsapp-web.js* [Online]. Available: <https://github.com/pedroslopez/whatsapp-web.js>.
- [5] M. B Jones, B. Campbell, and C. Mortimore. "JSON Web Token (JWT) Profile," *Internet Eng. Task Force*, no. 1, pp. 1-5, 2015.
- [6] N. Birwadkar, P. Malone, J. Hange, K. Doyle, E. Robson, D. Conway, S.Ivanov, L. Radziwonowicz, and R. Kleinfeld, "Personal Cloudlets: Implementing a User-Centric Datastore with Privacy Aware Access Control for Cloud for Cloud-based Data Platform," *IJARCCCE*, vol 4, no. 12, pp. 570-574, 2015.