**SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK**

**(CHATBOT TELEGRAM DENGAN AUTO-RESPONSE**

**DAN SENTIMENT ANALYSIS BERBASIS WEB)**

**TUGAS AKHIR**

MATA KULIAH : PEMROGRAMAN WEB LANJUT



**DISUSUN OLEH :**

ANGGOTA 1 – MARIO PARASIAN GABEMA : 222310069

ANGGOTA 2 – MUHAMMAD FIRDAUS : 212310036

ANGGOTA 3 – RIO OCTAVIANUS : 222310014

ANGGOTA 4 – SARLA SYAKIRA THALIB : 222310058

**PROGRAM STUDI SARJANA TEKNOLOGI INFORMASI**

**FAKULTAS INFORMATIKA DAN PARIWISATA**

**INSTITUT BISNIS DAN INFORMATIKA KESATUAN**

**BOGOR**

**2025**

**DAFTAR ISI**

[1. Pendahuluan 3](#_Toc22395)

[1.1 Tujuan 3](#_Toc22396)

[1.2 Ruang lingkup 3](#_Toc22397)

[1.3 Konvensi Dokumen 4](#_Toc22398)

[1.4 Target Audience 5](#_Toc22399)

[1.5 Referensi 6](#_Toc22400)

[2. Deskripsi Umum Sistem 7](#_Toc22401)

[2.1 Proses Bisnis 7](#_Toc22402)

[2.2 Karakteristik Pengguna 8](#_Toc22403)

[3. Desain dan Perancangan 8](#_Toc22404)

[3.1 Use Case Diagram 8](#_Toc22405)

[3.2 Entity Relationship Diagram 8](#_Toc22406)

[3.3 Activity Diagram 8](#_Toc22407)

[3.4 User Interface 9](#_Toc22408)

[4. Tahap Pengembangan 10](#_Toc22409)

[4.1 Alat dan Bahan 10](#_Toc22410)

[4.2 Bahasa Pemrograman 12](#_Toc22411)

[5. Pengujian Sistem 12](#_Toc22412)

# Pendahuluan

## Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SRS) ini disusun untuk menjelaskan secara terperinci kebutuhan dan fungsi dari aplikasi Chatbot Telegram dengan Auto-Response dan Sentiment Analysis Berbasis Web. Tujuan dari dokumen ini adalah sebagai acuan utama bagi pengembang, penguji, dan pemangku kepentingan dalam memahami sistem yang akan dibangun, baik dari sisi fungsionalitas, antarmuka pengguna, maupun batasan teknis.

Aplikasi ini dirancang untuk mempermudah pengguna dalam mengelola dan memantau percakapan yang terjadi melalui platform Telegram secara otomatis. Dengan fitur auto-response dan analisis sentimen, chatbot mampu merespons pesan pengguna secara cerdas dan mengklasifikasikan pesan berdasarkan nuansa

emosionalnya (positif, negatif, atau netral). Secara khusus, tujuan dari aplikasi ini adalah:

* Memberikan respon otomatis terhadap pesan yang masuk ke akun Telegram pengguna.
* Mengelompokkan pesan berdasarkan analisis sentimen menggunakan algoritma natural language processing.
* Menyediakan tampilan antarmuka web yang memudahkan pengguna dalam mengelola kontak dan percakapan.
* Memungkinkan penyimpanan data pesan dan kontak ke dalam basis data untuk kebutuhan riwayat dan pelaporan.

Dengan adanya dokumen ini, diharapkan proses pengembangan perangkat lunak dapat dilakukan secara sistematis dan sesuai dengan harapan pengguna akhir.

## Ruang lingkup

Aplikasi Chatbot Telegram dengan Auto-Response dan Sentiment Analysis

Berbasis Web merupakan sistem berbasis web yang terintegrasi dengan platform Telegram, dirancang untuk memberikan kemudahan dalam mengelola komunikasi pengguna dengan fitur respons otomatis dan analisis sentimen. Ruang lingkup dari aplikasi ini mencakup:

* Integrasi dengan Telegram API: Aplikasi menggunakan akun Telegram pengguna yang terhubung dengan bot untuk menerima dan mengirim pesan.
* Auto-Response: Sistem mampu membalas pesan masuk secara otomatis berdasarkan kata kunci tertentu yang telah didefinisikan.
* Analisis Sentimen: Setiap pesan yang diterima akan dianalisis secara otomatis untuk menentukan kategorinya (positif, negatif, atau netral) menggunakan algoritma analisis sentimen.
* Manajemen Kontak: Pengguna dapat menambahkan, mengubah nama, atau menghapus kontak yang terhubung dengan bot.
* Antarmuka Web: Disediakan dashboard berbasis React.js untuk

menampilkan daftar kontak, riwayat percakapan, dan hasil analisis sentimen secara real-time.

* Penyimpanan Data: Seluruh data percakapan dan kontak disimpan secara permanen di dalam basis data MySQL.
* Socket.io Real-time: Pesan-pesan masuk dan keluar dapat ditampilkan secara langsung tanpa perlu reload halaman.

Batasan dari sistem ini adalah pengguna harus memiliki koneksi internet aktif dan akun Telegram yang terverifikasi. Sistem ini juga belum mendukung multiple users secara bersamaan satu instansi bot hanya diperuntukkan untuk satu pengguna.

## Konvensi Dokumen

Bagian ini menjelaskan istilah, akronim, dan singkatan yang digunakan dalam dokumen *Software Requirements Specification* (SRS) ini. Penulisan istilah mengacu pada standarisasi dokumentasi perangkat lunak IEEE 29148:2018 untuk memastikan konsistensi dan kemudahan pemahaman bagi seluruh pihak yang terlibat, baik dari kalangan teknis maupun non-teknis. Adapun daftar istilah dan konvensi yang digunakan dapat dilihat pada tabel berikut:

Tabel 1. Konvensi Dokumen

|  |  |
| --- | --- |
| **Istilah / Akronim** | **Definisi** |
| SRS | *Software Requirements Specification*, yaitu dokumen yang mendeskripsikan kebutuhan perangkat lunak secara rinci. |
| Chatbot | Program komputer yang dirancang untuk mensimulasikan otomatis. percakapan manusia secara |
| Telegram | Telegram adalah aplikasi pesaninstan berbasis cloud yang memungkinkan pengiriman teks, gambar, video, dan dokumen. |
| Bot | Bot adalah program otomatis yang menjalankan perintah tertentu tanpa campur tangan manusia secara langsung. |
| Auto-Response | Auto-response adalah sistem balasan otomatis yang dikirim oleh bot atau aplikasi ketika menerima pesan atau input tertentu. |
| Sentiment Analysis | Sentiment analysis adalah proses analisis teks (misalnya chat, komentar, ulasan) untuk mengetahui emosi atau opini pengguna, apakah bersifat positif, negatif, atau netral. |
| Node.js | Node.js adalah runtime environment berbasis JavaScript yang digunakan untuk menjalankan kode JavaScript di luar browser. |
| MySQL | MySQL adalah sistem manajemen basis data relasional (RDBMS) yang menyimpan dan mengelola data dalam bentuk tabel. |
| REST API | REST API (Representational State Transfer Application Programming Interface) adalah standar komunikasi antara client dan server, yang memungkinkan frontend dan backend berinteraksi. |
| Socket.io | Socket.io adalah library JavaScript untuk komunikasi real-time antara client dan server menggunakan WebSocket. |
| Frontend | Frontend adalah bagian antarmuka pengguna (UI) dari sebuah aplikasi atau website — yang dilihat dan diakses langsung oleh user, seperti halaman login, form, tampilan chat, tombol, dan lainnya. |
| Backend | Backend adalah bagian "belakang layar" dari aplikasi yang mengatur logika bisnis, database, autentikasi, API, dan lainnya. |
| UI (User Interface) | UI adalah desain visual dari antarmuka pengguna, mencakup tata letak, warna, tombol, ikon, teks, dan elemen grafis lainnya. |
| IEEE 29148:2018 | Standar internasional yang digunakan sebagai pedoman pembuatan dokumen kebutuhan perangkat lunak. |

## Target Audience

Dokumen ini ditujukan bagi berbagai pihak yang memiliki keterlibatan langsung maupun tidak langsung dalam proses pengembangan sistem aplikasi. Setiap pihak memiliki kepentingan dan fokus tersendiri terhadap isi dokumen ini, sehingga penyusunan dilakukan dengan memperhatikan keterbacaan dan struktur yang sistematis. Pihak-pihak yang menjadi sasaran utama dari dokumen ini meliputi:

1. Pengembang:

Dokumen ini menjadi pedoman utama dalam proses implementasi sistem, mulai dari perancangan database, alur logika aplikasi, hingga antarmuka pengguna.

1. Dosen Pembimbing:

Digunakan sebagai bahan evaluasi terhadap pemahaman mahasiswa mengenai dokumentasi kebutuhan perangkat lunak serta penerapan teknologi yang digunakan.

1. Pengguna Sistem :

Meskipun tidak terlibat langsung dalam pengembangan teknis, pengguna akhir dapat memahami gambaran umum fitur dan tujuan sistem melalui dokumen ini.

Dengan adanya dokumentasi ini, diharapkan seluruh pihak memiliki pemahaman yang sama mengenai sistem yang akan dibangun, sehingga dapat mengurangi risiko miskomunikasi serta meningkatkan efektivitas pengembangan perangkat lunak.

## Referensi

Dokumen ini disusun dengan mengacu pada berbagai referensi teknis, akademik, dan standar dokumentasi perangkat lunak untuk memastikan kualitas, akurasi, dan keselarasan dengan praktik terbaik dalam pengembangan sistem.

Berikut adalah referensi yang digunakan:

Tabel 2. Referensi

|  |  |  |
| --- | --- | --- |
| **No** | **Referensi** | **Keterangan** |
| **1** | IEEE Std 29148-2018 | Standar internasional untuk dokumentasi Software  Requirements Specification (SRS). |
| **2** | React Documentation | sebuah **library JavaScript** yang digunakan untuk membangun **UI (User Interface)** yang interaktif dan efisien di aplikasi web. |
| **3** | Express Documentation | framework backend berbasis **Node.js** yang digunakan untuk membangun **web server dan REST API** secara cepat dan ringan. |
| **4** | Tailwindcss Docs | Dokumentasi CSS framework untuk pembuatan tampilan antarmuka responsif. |
| **5** | MySQL Documentation | Panduan penggunaan system manajemen basis data MySQL |

# Deskripsi Umum Sistem

## Proses Bisnis

Sisi Aplikasi Chatbot Telegram dengan Auto-Response dan Sentiment Analysis Berbasis Web dirancang untuk mempermudah pengguna dalam berkomunikasi melalui Telegram, sekaligus memantau dan menganalisis isi pesan yang diterima dengan pendekatan berbasis sentimen. Proses bisnis utama dari sistem ini terdiri dari beberapa tahapan:

1. Interaksi Awal Pengguna dengan Bot Telegram

Pengguna mengirimkan pesan ke bot Telegram yang telah dikonfigurasi. Sistem akan memverifikasi apakah pengirim termasuk dalam daftar kontak yang terdaftar di database. Jika tidak, pesan akan diabaikan demi menjaga keamanan dan relevansi data.

1. Penyimpanan & Analisis Pesan Masuk

Pesan yang diterima dari kontak terdaftar akan langsung disimpan ke dalam basis data. Selanjutnya, pesan akan dianalisis menggunakan Sentiment Analysis untuk mengklasifikasikan isi pesan menjadi tiga kategori: positif, negatif, atau netral.

1. Respons Otomatis oleh Chatbot

Berdasarkan kata kunci yang terdeteksi dalam pesan, sistem akan mencocokkan dengan daftar respons otomatis (auto-responses) yang telah ditentukan. Jika ada kecocokan, chatbot akan mengirimkan balasan yang sesuai secara otomatis ke pengguna melalui Telegram.

1. Pengelolaan Kontak dan Pesan melalui Web App

Admin atau pengguna internal dapat mengakses panel berbasis web untuk:

* + Menambahkan, mengubah, atau menghapus kontak
  + Melihat riwayat percakapan per kontak
  + Mengirim pesan secara manual dari antarmuka web
  + Melihat hasil analisis sentimen dari setiap pesan

1. Notifikasi Real-time via WebSocket

Setiap pesan masuk dan balasan bot akan ditampilkan secara real-time di halaman web melalui integrasi Socket.io, sehingga admin dapat langsung memantau percakapan dan aktivitas bot.

## Karakteristik Pengguna

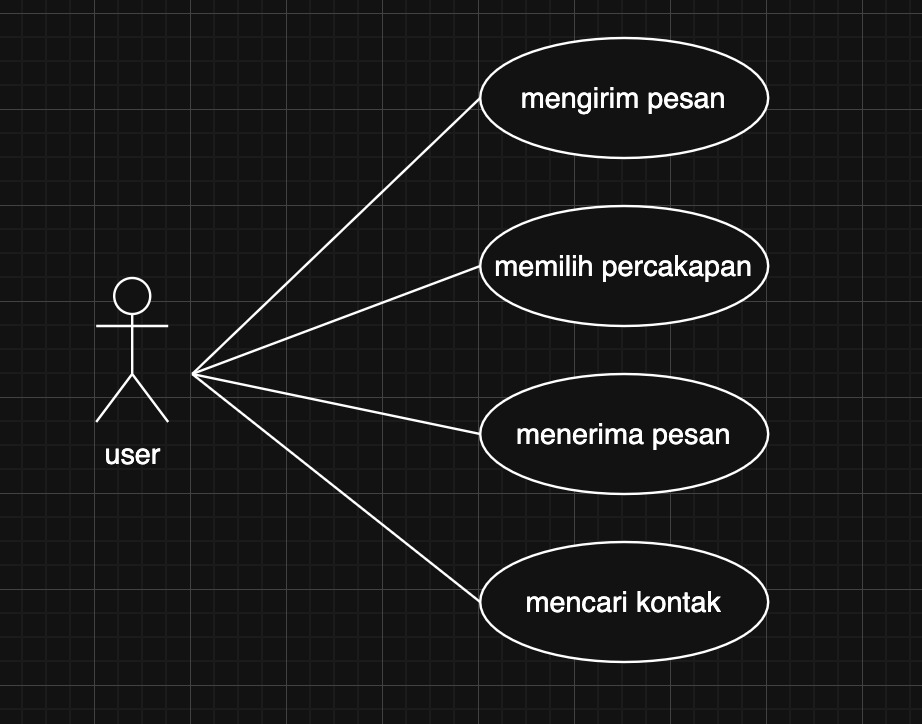
Sistem Pengguna dari sistem ini adalah individu umum yang menggunakan aplikasi Telegram untuk berkomunikasi dengan chatbot yang telah disediakan. Mereka merupakan pengguna akhir yang tidak memiliki akses terhadap sisi pengelolaan atau administrasi sistem. Karakteristik utama pengguna ini adalah tidak memerlukan kemampuan teknis atau latar belakang teknologi informasi untuk menggunakan sistem. Cukup dengan memiliki akun Telegram dan bisa

mengoperasikan aplikasi tersebut secara dasar, pengguna sudah dapat berinteraksi dengan chatbot.

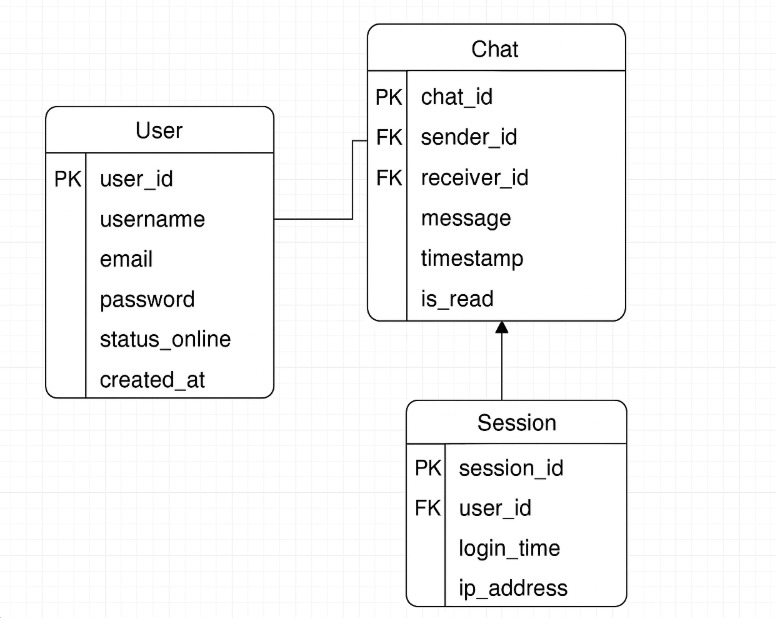
Pengguna cukup mengetik pesan seperti biasa dalam bentuk percakapan sehari-hari. Sistem akan merespons secara otomatis berdasarkan kata kunci tertentu yang telah dipetakan, dan akan melakukan analisis sentimen dari setiap pesan yang diterima. Tujuan dari pengalaman pengguna yang sederhana ini adalah untuk memberikan layanan berbasis chatbot yang informatif, responsif, dan mudah diakses kapan pun melalui platform Telegram tanpa memerlukan proses instalasi tambahan.

# Desain dan Perancangan

## Use Case Diagram

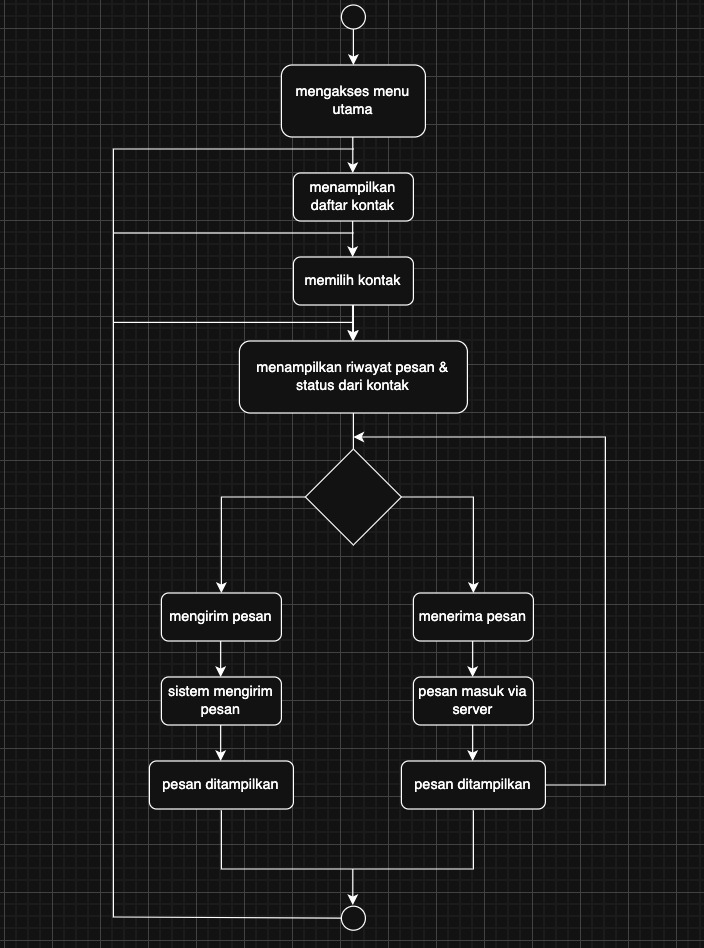


## Entity Relationship Diagram



menggambarkan struktur data dalam sistem website messenger, mencakup entitas User, Chat, dan Session yang saling terhubung melalui relasi pengiriman pesan dan riwayat login.

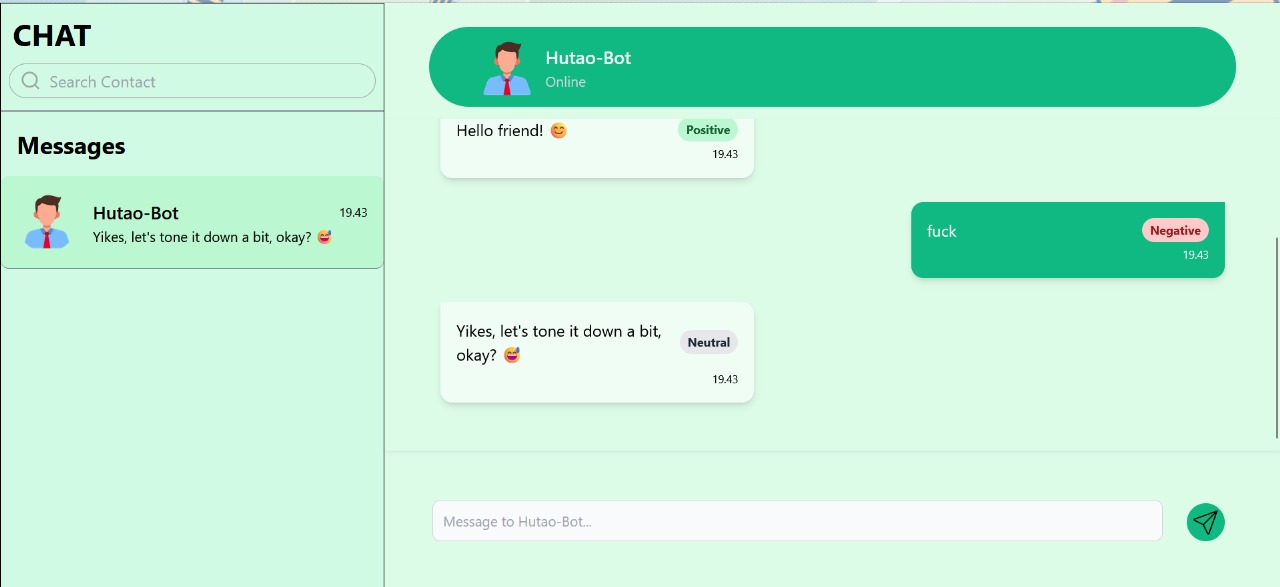
## Activity Diagram



menunjukkan alur aktivitas pengguna dari mulai login, melihat daftar chat, memilih kontak, mengirim dan menerima pesan, hingga logout, sehingga memberikan gambaran proses bisnis sistem secara menyeluruh dan runtut.

## User Interface

a. Main Page



Gambar 1. Main Page

Gambar di atas menunjukkan antarmuka utama dari aplikasi *Chatbot Telegram dengan Auto-Response dan Sentiment Analysis Berbasis Web*. Aplikasi ini terdiri dari tiga bagian utama, yaitu daftar kontak di sisi kiri, jendela obrolan di bagian tengah, dan kolom input pesan di bagian bawah.

Pada bagian kiri, pengguna dapat menambahkan kontak baru dengan memasukkan ID Telegram dan nama pengguna yang ingin ditambahkan, kemudian menekan tombol Simpan. Terdapat pula daftar kontak yang telah tersimpan, yang ditampilkan dalam bentuk kartu kontak lengkap dengan inisial dan nama.

Bagian tengah adalah area obrolan antara pengguna dan chatbot. Setiap pesan yang dikirim dan diterima ditampilkan dengan warna dan posisi berbeda. Balon pesan berwarna biru menunjukkan pesan dari chatbot (mahesa\_bot), sedangkan balon berwarna abu-abu berasal dari pengguna. Di bawah setiap pesan terdapat label sentimen yang menunjukkan apakah pesan tersebut bernada Positif, Netral, atau Negatif, yang ditentukan melalui proses *sentiment analysis* secara otomatis.

Dari contoh percakapan yang ditampilkan, dapat dilihat bahwa chatbot merespons pertanyaan umum seperti "halo", "apa hobi kamu", dan "apa pekerjaan kamu". Sistem juga mampu mendeteksi kata-kata kasar atau tidak sopan dan memberi label Negatif, seperti yang terlihat pada pesan “motherfucker”. Ini menunjukkan bahwa fitur analisis sentimen berjalan dengan baik dan memberikan nilai tambah dalam memahami emosi atau niat pengguna saat berinteraksi dengan sistem.

# Tahap Pengembangan

## Alat dan Bahan

Pengembangan sistem ini memanfaatkan berbagai perangkat lunak dan sumber daya sebagai alat bantu untuk mendukung proses perancangan, pengkodean, dan pengujian sistem. Adapun alat dan bahan yang digunakan antara

lain:

Tabel 3. Alat dan Bahan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama** | | **Jenis** | **Keterangan** |
| 1. | Laptop/PC | | Perangkat  Keras | Digunakan untuk coding, debugging, dan menjalankan server/backend. |
| 2. | Visual Studi  Code | o | Perangkat  Lunak | Code editor untuk menulis dan mengelola kode proyek frontend dan backend. |
| 3. | Node.js |  | Perangkat  Lunak | Runtime JavaScript untuk menjalankan backend (Express & Socket.io). |
| 4. | React.js |  | Perangkat  Lunak | Framework frontend untuk membangun antarmuka pengguna web. |
| 5. | MySQL |  | Perangkat  Lunak | Database untuk menyimpan data kontak dan pesan. |
| 6. | Telegram  Desktop  Mobile App | / | Perangkat  Lunak | Untuk menguji bot Telegram secara langsung. |
| 7. | Bot Telegram  & API Token | | Layanan API | Diperlukan untuk menghubungkan bot dengan akun Telegram pengguna. |
| 8. | GramJS | | Library | Library untuk komunikasi dengan Telegram secara langsung melalui MTProto. |
| 9. | Socket.IO | | Library | Digunakan untuk komunikasi real-time antara frontend dan backend. |
| 10. | Sentiment.js | | Library | Untuk melakukan analisis sentimen pada setiap pesan yang masuk. |
| 11. | Postman | | Perangkat  Lunak | Alat bantu untuk menguji endpoint API secara manual. |
| 12. | Git & GitHub | | Versi Kontrol | Digunakan untuk manajemen versi dan kolaborasi proyek. |
| 13. | dotenv | | Library | Untuk mengelola variabel lingkungan (.env) dalam proyek backend. |
| 14. | **Axios** | | Library | Digunakan untuk melakukan request HTTP di sisi frontend. |

## Bahasa Pemrograman

Dalam pengembangan aplikasi ini, digunakan kombinasi beberapa bahasa pemrograman yang saling melengkapi antara sisi frontend, backend, dan integrasi dengan API Telegram. Pertama, JavaScript digunakan sebagai bahasa utama untuk pengembangan aplikasi baik di sisi frontend maupun backend. Pada sisi frontend, JavaScript dikombinasikan dengan React.js untuk membangun antarmuka pengguna (UI) yang interaktif, responsif, dan modular. React memungkinkan pembuatan komponen yang dapat digunakan kembali, memudahkan pengelolaan tampilan kontak, jendela obrolan, dan input pesan.

Di sisi backend, JavaScript digunakan bersama Node.js untuk membangun server. Dengan memanfaatkan Express.js, backend mampu menangani permintaan HTTP dari frontend seperti pengelolaan kontak dan pesan. Selain itu, integrasi Socket.IO juga ditulis menggunakan JavaScript untuk mendukung komunikasi *real-time* antara pengguna dan bot. Untuk kebutuhan interaksi dengan Telegram, digunakan GramJS, sebuah library Node.js yang mendukung protokol MTProto Telegram, memungkinkan aplikasi membaca dan mengirim pesan secara langsung melalui akun pengguna (bukan hanya bot API). Di sisi database, perintah-perintah SQL digunakan untuk mengelola penyimpanan data dalam MySQL, mulai dari penyimpanan kontak, pesan, hingga data analisis sentimen.

# Pengujian Sistem

Pengujian sistem merupakan tahap penting dalam proses pengembangan perangkat lunak untuk memastikan bahwa seluruh fungsi dan fitur dari sistem telah berjalan sesuai dengan kebutuhan pengguna dan spesifikasi yang telah ditentukan sebelumnya. Pengujian ini dilakukan untuk mendeteksi adanya kesalahan (*bug*), kekurangan, dan memastikan bahwa sistem berfungsi secara stabil, akurat, dan dapat digunakan dalam kondisi nyata.

Dalam proyek *Chatbot Telegram dengan Auto-Response dan Sentiment*

*Analysis Berbasis Web*, pengujian dilakukan pada fitur utama yang mencakup pengelolaan kontak, pengiriman dan penerimaan pesan, respon otomatis berdasarkan kata kunci, serta analisis sentimen. Metode pengujian yang digunakan adalah pengujian fungsional, di mana setiap fitur diuji berdasarkan input dan output yang diharapkan.

Tabel 3. Pengujian - Sisi Pengguna (User)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Fitur yang  Diuji | Input | Langkah  Pengujian | Hasil yang  Diharapkan | Status |
| 1 | Tambah  Kontak | user\_id & nama baru | Masukkan data dan klik  "Simpan" | Kontak tersimpan dan  tampil di daftar kontak | Berhasil |
| 2 | Tampilkan  Daftar  Kontak | null | Akses halaman utama | Semua kontak dari database ditampilkan | Berhasil |
| 3 | Edit Nama  Kontak | Nama baru | Klik ikon titik tiga > "Edit  Nama" >  Simpan perubahan | Nama kontak berubah di  database dan UI | Tidak Berhasil |
| 4 | Hapus  Kontak | null | Klik ikon titik tiga > "Hapus  Kontak" | Kontak terhapus dari database  dan UI | Tidak Berhasil |
| 5 | Kirim  Pesan | Teks pesan | Ketik pesan dan klik "Kirim" | Pesan terkirim ke pengguna Telegram &  tampil di UI | Berhasil |
| 6 | Menerima  Pesan  Telegram | Pesan dari kontak  Telegram | Kontak kirim pesan ke akun Telegram yang terhubung | Pesan ditampilkan di frontend | Berhasil |
| 7 | Respon  Otomatis | Pesan berisi kata kunci tertentu | Kirim pesan seperti "halo",  "fitur", dll | Bot membalas otomatis sesuai  respon yang  ditentukan | Berhasil |
| 8 | Analisis  Sentimen | Pesan teks (positif/ne gatif/netral  ) | Kirim pesan dengan nuansa berbeda | Sentimen dianalisis dan label ditampilkan  (Pos/Neg/Net) | Berhasil |
| 9 | Realtime  Chat | Kirim dan terima pesan | Komunikasi dua arah  berlangsung di  UI | Tampilan diperbarui secara langsung (tanpa reload) | Berhasil |

Berdasarkan hasil pengujian yang telah dilakukan terhadap sistem *Chatbot Telegram dengan Auto-Response dan Sentiment Analysis Berbasis Web*, dapat disimpulkan bahwa seluruh fitur utama aplikasi telah berjalan sesuai dengan spesifikasi kebutuhan yang dirancang sebelumnya. Fitur-fitur seperti penambahan kontak, pengiriman dan penerimaan pesan secara real-time, analisis sentimen, serta tanggapan otomatis dari chatbot telah diuji dan menunjukkan hasil yang sesuai harapan.

Seluruh skenario uji menghasilkan status Berhasil, yang berarti sistem dapat menangani input yang diberikan dengan benar serta memberikan keluaran yang sesuai. Hal ini menunjukkan bahwa sistem memiliki tingkat keandalan yang baik dan siap digunakan oleh pengguna dalam lingkungan operasional sebenarnya.

Dengan demikian, aplikasi ini telah memenuhi kriteria fungsionalitas dan performa yang ditetapkan, serta mampu memberikan pengalaman interaksi chatbot yang responsif dan informatif melalui platform Telegram.

**LAMPIRAN**

Link Repository Github: https://github.com/Rio-Octavianus/chat-message-fe-rio

