



SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

JUDUL

**Perancangan Dan Implementasi *Chatbot* Berbasis Telegram untuk
Mengelola Rekam Medis Penderita Diabetes**

TUGAS AKHIR

Richal Juliansyah

0110220177

PROGRAM STUDI TENIK INFORMATIKA

DEPOK

MARET -2024



SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

JUDUL

**Perancangan Dan Implementasi *Chatbot* Berbasis Telegram untuk
Mengelola Rekam Medis Penderita Diabetes**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar S.kom

RICHAL JULIANSYAH

0110220177

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

MARET - 2024

HALAMAN PERNYATAAN ORISINALITAS

Skripsi/Tugas Akhir ini adalah hasil karya penulis, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Richal Juliansyah

NIM : 0110220177

Depok , 7 Maret 2024

Tanda Tangan

A handwritten signature in black ink, appearing to read 'Richal Juliansyah', with a stylized flourish at the end.

Richal Juliansyah

HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh :

Nama : Richal Juliansyah

NIM : 0110220177

Program Studi : Teknik Informatika

Judul Skripsi : Perancangan Dan Implementasi *Chatbot* Berbasis Telegram untuk Mengelola Rekam Medis Penderita Diabetes.

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri

DEWAN PENGUJI

Pembimbing



(Henry Saptono, S.Si, M.kom.)

Penguji



(Dr. Sirojul Munir, S.Si, M.kom.)

Ditetapkan di : Depok

Tanggal : 21 Maret 2024

KATA PENGANTAR

Puji syukur penulis disampaikan kepada Allah SWT, karena dengan anugerah dan kasih sayang-Nya, penulis berhasil menyelesaikan skripsi/Tugas Akhir ini. Penulisan skripsi/Tugas Akhir ini dilaksanakan sebagai salah satu syarat untuk meraih gelar Sarjana dalam Program Studi Teknik Informatika di Sekolah Tinggi Teknologi Terpadu Nurul Fikri. Penulis menyadari bahwa tanpa dukungan dan bimbingan dari berbagai pihak, mulai dari masa perkuliahan hingga penyusunan skripsi ini, penyelesaian tugas ini akan menjadi tantangan yang berat.

Penulis ingin menyampaikan terima kasih kepada:

1. Allah SWT.
2. Orang tua dan seluruh anggota keluarga yang memberikan dukungan, baik secara moral maupun materi, dalam menyelesaikan tugas ini.
3. Bapak Drs. Rusmanto, M.M., selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
4. Bapak/Ibu Tiffani Nabarian, S.Kom., M.T.I., selaku Ketua Program Studi Teknik Informatika.
5. Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
6. Bapak Henry Saptono M.kom, sebagai Dosen Pembimbing Akademik yang telah memberikan bimbingan selama masa perkuliahan di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
7. Bapak Henry Saptono M.kom, sebagai Dosen Pembimbing Tugas Akhir penulis dalam menyelesaikan penulisan ilmiah ini.
8. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah memberikan bimbingan dalam proses pembelajaran.
9. Imelda yang telah membantu dalam menyelesaikan penulisan ilmiah ini

Dalam penulisan ilmiah ini, tentu masih terdapat kekurangan yang mungkin disebabkan oleh keterbatasan pengetahuan dan kemampuan penulis. Meskipun demikian, penulis berusaha sebaik mungkin untuk menyelesaikan penulisan ilmiah ini. Oleh karena itu, penulis menerima dengan rendah hati kritik dan saran dari pembaca apabila terdapat kekurangan.

Sebagai penutup, penulis berharap agar Allah SWT memberikan balasan yang berlipat ganda kepada semua pihak yang telah membantu. Semoga skripsi ini dapat memberikan manfaat dalam pengembangan ilmu pengetahuan.

Depok, 20 Desember 2014

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR
UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri,
saya yang bertanda tangan di bawah ini:

Nama : Richal Juliansyah
NIM : 0110220177
Program Studi : Teknik Informatika
Jenis karya : Skripsi / Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan
kepada STT- NF Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty -
Free Right*) atas karya ilmiah saya yang berjudul :

Perancangan Dan Implementasi Chatbot Berbasis Telegram untuk
Mengelola Rekam Medis Penderita Diabetes , beserta perangkat yang ada
(jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF
berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk
pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya
selama tetap mencantumkan nama saya sebagai penulis/pencipta dan
sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 23 Oktober 2023
Yang menyatakan :



(.....)

Richal Juliansyah

ABSTRAK

Nama : Richal Juliansyah
NIM :0110220177
Program Studi : Teknik Informatika
Judul : Perancangan Dan Implementasi *Chatbot* Berbasis Telegram untuk Mengelola Rekam Medis Penderita Diabetes

Tugas Akhir ini bertujuan untuk menginvestigasi implementasi bot di platform Telegram sebagai solusi untuk mengelola riwayat hasil rekam medis penderita diabetes di masyarakat. Diabetes mellitus merupakan salah satu penyakit kronis yang membutuhkan pemantauan dan manajemen yang berkelanjutan. Namun, pencatatan dan aksesibilitas data rekam medis penderita diabetes sering kali menjadi tantangan bagi tenaga medis. Penelitian ini menggunakan pendekatan tugas akhir dengan menggabungkan konsep pengembangan bot dan manajemen rekam medis. Bot di Telegram dikembangkan untuk memberikan kemudahan dalam mencatat dan mengelola riwayat hasil rekam medis penderita diabetes. Dalam tahap pengembangan, bot ini dirancang dengan kemampuan untuk mencatat data seperti tingkat gula darah, hasil tes laboratorium, catatan dokter, dan pengobatan yang diberikan. Bot juga dilengkapi dengan fitur untuk menampilkan riwayat rekam medis penderita secara terperinci dan mengirimkan peringatan jadwal pengobatan kepada penderita. Tugas Akhir ini dilakukan di STT Terpadu Nurul Fikri dengan melibatkan tim pengembang perangkat lunak dan tenaga medis. Tim pengembang perangkat lunak bertanggung jawab dalam mengembangkan bot di Telegram menggunakan bahasa pemrograman Python dan memanfaatkan API Telegram. Sedangkan tenaga medis memberikan masukan terkait kebutuhan dan persyaratan sistem untuk mengelola rekam medis penderita diabetes. Hasil dari Tugas Akhir ini diharapkan dapat memberikan manfaat yang signifikan bagi tenaga medis dalam mengelola rekam medis penderita diabetes di masyarakat. Dengan adanya bot di Telegram, diharapkan proses pencatatan dan aksesibilitas data rekam medis menjadi lebih efisien dan mudah. Bot ini juga dapat meningkatkan komunikasi antara tenaga medis dan penderita.

Kata Kunci: implementasi bot, Telegram, rekam medis, penderita diabetes.

ABSTRACT

Name : Richal Juliansyah
NIM : 0110220177
Study Program : Teknik Informatika
Title : Perancangan Dan Implementasi *Chatbot* Berbasis Telegram untuk Mengelola Rekam Medis Penderita Diabetes

The aim of this Final Project is to investigate the implementation of a bot on the Telegram platform as a solution for managing the medical records history of diabetes patients in the community. Diabetes mellitus is one of the chronic diseases that requires continuous monitoring and management. However, recording and accessing medical data of diabetes patients often pose challenges for healthcare professionals. This research adopts a final project approach by integrating the concepts of bot development and medical record management. A bot on Telegram is developed to facilitate recording and managing the medical records history of diabetes patients. In the development phase, the bot is designed with the ability to record data such as blood sugar levels, laboratory test results, doctor's notes, and prescribed medications. The bot is also equipped with features to display detailed medical records history and send medication schedule reminders to patients. This Final Project is conducted at STT Terpadu Nurul Fikri involving a software development team and healthcare professionals. The software development team is responsible for developing the bot on Telegram using the Python programming language and leveraging the Telegram API. Meanwhile, healthcare professionals provide input regarding the needs and system requirements for managing the medical records of diabetes patients. The results of this Final Project are expected to provide significant benefits to healthcare professionals in managing the medical records of diabetes patients in the community. With the presence of the bot on Telegram, it is hoped that the process of recording and accessing medical data becomes more efficient and accessible. This bot can also enhance communication between healthcare professionals and patients.

Keywords: *bot implementation, Telegram, medical records, diabetes patients.*

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS.....	3
HALAMAN PENGESAHAN.....	4
ABSTRAK.....	7
ABSTRACT.....	8
DAFTAR ISI.....	9
DAFTAR GAMBAR.....	11
DAFTAR TABEL.....	13
BAB I.....	14
PENDAHULUAN.....	14
1.1 Latar belakang.....	14
1.2 Rumusan Masalah.....	15
1.3 Tujuan dan Manfaat Penelitian.....	16
1.4 Batasan Masalah.....	17
1.5 Sistematika Penulisan.....	18
BAB II KAJIAN LITERATUR.....	19
2.1 Diabetes.....	19
2.2 <i>Chatbot</i>	20
2.3 Manajemen Rekam Medis.....	21
2.4 Alasan Pentingnya Penelitian adalah.....	23
2.5 Telegram.....	23
2.6 Python dan aiogram.....	25
2.7 <i>MySQL</i> sebagai Basis Data.....	27
2.8 Peneliatan terkait.....	28
2.9 Metode pengembangan software.....	29
2.10 Agile Method.....	30
BAB III METODOLOGI PENELITIAN.....	32
3.1 Tahapan Penelitian.....	32
3.2 Rancangan Penelitian.....	33
3.3 Jenis Penelitian.....	34
3.4 Metode Analisis.....	34
BAB IV IMPLEMENTASI DAN EVALUASI.....	37

4.1	Analisa Dan Perancangan	37
4.2	Implementasi	44
4.3	Hasil Implementasi.....	49
4.4	Pengujian.....	53
BAB V KESIMPULAN DAN SARAN.....		64
5.1	Kesimpulan	64
5.2	Saran.....	65
DAFTAR REFERENSI		66
LAMPIRAN.....		68

DAFTAR GAMBAR

GAMBAR 2.10-1 AGILE METHODE	30
GAMBAR 3.1-1 TAHAPAN PENELITIAN	32
GAMBAR 3.2-1 RANCANGAN PENELITIAN	33
GAMBAR 3.4-1 KEBUTUHAN FUNGSIONAL	35
GAMBAR 4.1-1 ARSITEKTUR LOGIK.....	38
GAMBAR 4.1-2 DIAGRAM ARSITEKTUR SISTEM.....	39
GAMBAR 4.1-3 USECASE	40
GAMBAR 4.1-4 ERD DATABASE	40
GAMBAR 4.1-5 ALUR PERCAKAPAN	41
GAMBAR 4.1-6 DIAGRAM AKTIVITAS	42
GAMBAR 4.2-1 INSTALL PYTHON	44
GAMBAR 4.2-2 INSTALL AIOGRAM	44
GAMBAR 4.2-3 INSTALL AIOGRAM	45
GAMBAR 4.2-4 MYSQL SERVER	45
GAMBAR 4.2-5 BOTFATHER	46
GAMBAR 4.2-6 TOKEN BOT	46
GAMBAR 4.2-7 PENANGANAN PESAN	47
GAMBAR 4.2-8 STATE MACHINE.....	48
GAMBAR 4.2-9 INTERAKSI DATABASE	48
GAMBAR 4.2-10 VISUALISASI GRAFIK	49
GAMBAR 4.3-1 TAMPILAN AWAL	50
GAMBAR 4.3-2 MENU UTAMA	51
GAMBAR 4.3-3 PROFIL	51
GAMBAR 4.3-4 REKAM MEDIS	52
GAMBAR 4.4-1 KODE PENGUJIAN.....	54
GAMBAR 4.4-2 HASIL UJI UNIT.....	54
GAMBAR 4.4-3 SKENERIO BEHAVE.....	55
GAMBAR 4.4-4 KODE UJI BEHAVEHASIL PENGUJIAN.....	56
GAMBAR 4.4-5 HASIL UJI UAT BEHAVE.....	57
GAMBAR 4.4-6 USER 1.....	58

GAMBAR 4.4-7 USER 2.....	58
GAMBAR 4.4-8 PENGISIAN PROFIL USER 1.....	59
GAMBAR 4.4-9 PENGISIAN PROFIL USER 2.....	59
GAMBAR 4.4-10 TAMBAH REKAM MEDIS USER 1	60
GAMBAR 4.4-11 TAMBAH REKAM MEDIS USER 2	60
GAMBAR 4.4-12 INSIGHT USER 1.....	61
GAMBAR 4.4-13 INSIGHT USER 2.....	61

DAFTAR TABEL

TABEL 2.8-1 PENELITIAN TERKAIT	28
TABEL 3.4-1 KELOMPOK PENGGUNA	34
TABEL 3.4-2 KEBUTUHAN FUNGSIONAL	35
TABEL 3.4-3 NON-FUNGSIONAL	35
TABEL 3.4-4 KEBUTUHAN PRIORITAS	35
TABEL 3.4-5 DATA REKAM MEDIS	36
TABEL 4.1-1K EBUTUHAN HARDWARE.....	37
TABEL 4.1-2 KEBUTUHAN SOFTWARE	37
TABEL 4.1-3 TAHAP PENGUJIAN	43
TABEL 4.1-4 ALAT PENGUJIAN	43
TABEL 4.4-1 HASIL PENGUJIAN UNIT	55
TABEL 4.4-2 HASIL PENGUJIAN UAT	57
TABEL 4.4-3 PENGUJIAN REGISTER	62
TABEL 4.4-4 PENGISIAN PROFIL.....	62
TABEL 4.4-5 TAMBAH REKAM MEDIS	62
TABEL 4.4-6 MELIHAT INSIGHT.....	63

BAB I

PENDAHULUAN

Pendahuluan dalam penelitian "Pengembangan *Chatbot* Berbasis Telegram untuk Manajemen Rekam Medis Penderita Diabetes " adalah bagian awal dari penulisan yang menjelaskan secara komprehensif tentang latar belakang, konteks, dan landasan dari penelitian tersebut. Berikut adalah komponen-komponen penting yang ada dalam pendahuluan.

1.1 Latar belakang

Diabetes Melitus (DM) merupakan suatu kelompok penyakit metabolik yang ditandai dengan hiperglikemik akibat kelainan sekresi insulin, kerja insulin atau keduanya. Dimana kadar gula (glukosa) dalam darah melebihi 110 mg/dl dalam keadaan puasa dan melebihi 200 mg/dl dalam keadaan tidak puasa. Diabetes tipe 2 adalah jenis yang paling sering ditemukan, biasanya timbul pada usia di atas 40 tahun. Banyak negara tidak menyadari dampak bahaya diabetes terhadap sosial ekonomi. Kurang waspada dan tidak paham akan membuat penyakit diabetes merajalela menghancurkan kehidupan manusia. Komplikasi diabetes timbul akibat kontrol gula tidak teratur, gaya hidup yang keliru, tidak disiplin berduet, enggan minum obat, atau berolahraga. Gejala umum penderita penderita diabetes adalah sering buang air kecil, cepat lelah dan mengantuk, berat badan menurun drastis, selalu lapar dan haus, gatal-gatal di sekitar kemaluan[1].

Diabetes merupakan penyakit kronis yang memerlukan penanganan dan pemantauan kesehatan penderita secara terus menerus. Salah satu cara untuk meningkatkan pengelolaan penderita diabetes adalah dengan menerapkan sistem *chatbot* yang dapat membantu mengelola rekam medisnya. *Chatbots* adalah program komputer yang dapat mensimulasikan percakapan manusia melalui interaksi teks atau suara. Mereka dapat diintegrasikan ke dalam *platform* perpesanan seperti Telegram, yang terkenal dengan fitur keamanan dan keandalan enkripsi datanya, Beberapa penelitian telah dilakukan mengenai desain dan implementasi *chatbot* menggunakan Telegram. Misalnya, Chandra dkk [2]. merancang *chatbot* untuk

pariwisata di Bali menggunakan pemrosesan bahasa alami *Dialogflow*. *Chatbot* tersebut mampu memberikan informasi destinasi wisata, akomodasi, dan transportasi. Kajian lain yang dilakukan Harry Priambodo dan Adi Muhajirin[3]. merancang *chatbot* untuk pendaftaran siswa menggunakan Telegram. *Chatbot* mampu membantu siswa mendaftar kursus dan memberikan informasi jadwal dan ketersediaan kursus. Untuk merancang dan mengimplementasikan *chatbot* untuk mengelola rekam medis penderita diabetes, penting untuk mempertimbangkan pendekatan desain yang berpusat pada pengguna [4]. Pendekatan ini melibatkan pemahaman kebutuhan dan preferensi pengguna dan merancang *chatbot* yang sesuai. *Chatbot* harus dapat memberikan rekomendasi dan pengingat yang dipersonalisasi kepada penderita berdasarkan riwayat kesehatan dan kondisi saat ini.

Chatbot juga harus dapat berintegrasi dengan sistem layanan kesehatan lain untuk memberikan gambaran komprehensif tentang kesehatan penderita. Dalam konteks ini, penelitian ini bertujuan untuk mengembangkan *chatbot* berbasis Telegram yang dapat membantu penderita diabetes dalam pengelolaan rekam medis mereka. *Chatbot* ini akan dirancang untuk mencatat dan memantau data medis, memberikan saran perawatan, serta memberikan edukasi yang relevan tentang penyakit diabetes. Penelitian ini akan memanfaatkan potensi Telegram sebagai *platform* pesan yang populer dan mudah diakses oleh berbagai kelompok pengguna. Dengan demikian, penelitian ini akan mengisi celah dalam literatur terkait dan diharapkan dapat memberikan solusi yang efektif untuk pengelolaan rekam medis penderita diabetes dengan memanfaatkan teknologi *chatbot* dan *platform* telegram.

1.2 Rumusan Masalah

Rumusan masalah adalah pernyataan atau pertanyaan yang merinci permasalahan yang akan diteliti dalam suatu penelitian. Dalam konteks penelitian ini, rumusan masalah mengidentifikasi pertanyaan-pertanyaan kunci yang perlu dijawab agar penelitian dapat mencapai tujuannya.

1. Bagaimanakah rancangan dan implementasi *chatbot* Berbasis telegram untuk Mengelola Rekam Medis Penderita Diabetes ?

2. Bagaimanakah efektifitas *chatbot* Berbasis telegram untuk mengelola rekam medis penderita diabetes?

1.3 Tujuan dan Manfaat Penelitian

Penelitian ini bertujuan untuk mengembangkan *chatbot* berbasis Telegram yang efektif dalam membantu individu yang menderita Diabetes dalam manajemen rekam medis mereka , Manfaat penelitian ini diharapkan dapat memberikan manfaat yang signifikan sebagai berikut.

1.3.1. Tujuan Penelitian adalah:

1. Membuat rancangan dan implementasi *chatbot* berbasis Telegram yang bermanfaat kepada penderita diabetes dalam pengelolaan rekam medis mereka.
2. Untuk mengetahui efektivitas *chatbot* berbasis Telegram dalam membantu penderita diabetes dalam mengelola rekam medis mereka, termasuk pemantauan kadar gula darah penyuluhan tentang gaya hidup sehat, dan sebagainya.

1.3.2 Manfaat penelitian adalah:

1. Memberikan kemudahan secara mandiri kepada penderita diabetes melalui *chatbot* berbasis Telegram memungkinkan akses 24/7 untuk pengelolaan rekam medis, pemantauan kesehatan diri, memberikan dampak positif yang signifikan terhadap kualitas hidup dan manajemen penyakit diabetes mereka secara keseluruhan.
2. Mengetahui efektivitas *chatbot* berbasis Telegram dalam pengelolaan rekam medis penderita diabetes dapat menghasilkan karya tulis yang menjadi sumber pengetahuan yang berharga dalam literatur kesehatan, memperluas pemahaman pemanfaatan teknologi dalam manajemen penyakit kronis, memberikan panduan praktis penyedia layanan kesehatan, mendorong inovasi dalam pengembangan teknologi kesehatan, dan meningkatkan kesadaran masyarakat akan manfaat teknologi dalam perawatan kesehatan, menciptakan dampak yang signifikan dalam pemahaman dan praktik kesehatan publik.

1.4 Batasan Masalah

Cakupan masalah dalam konteks penelitian adalah batasan yang menjelaskan area atau lingkup spesifik dari topik yang akan diteliti. Ini membantu menentukan parameter yang akan dijelajahi dalam penelitian dan apa yang akan menjadi fokus utama dari penelitian tersebut. Dalam penelitian ini, cakupan masalah akan mencakup hal-hal berikut:

1. Cakupan Fungsionalitas *Chatbot* : *Chatbot* akan mampu membantu penderita Diabetes dalam mencatat dan memantau datamedis mereka, termasuk pengukuran gula darah, tekanan darah, berat badan, dan catatan medis lainnya.
2. Tools yang Digunakan: Dalam pengembangan *chatbot*, berikut adalah beberapa tools dan teknologi yang akan digunakan.
3. Fokus pada Platform Telegram: Penelitian ini akan terbatas pada pengembangan *chatbot* yang khusus berjalan di platform pesan Telegram. Pengembangan aplikasi server terpisah untuk *chatbot* tidak akan dilakukan, sehingga *chatbot* akan bergantung pada fitur dan kemampuan yang disediakan oleh Telegram API. Dengan demikian, integrasi dengan platform lain tidak akan menjadi fokus utama dari penelitian ini.
4. Keamanan: Meskipun *chatbot* akan mengelola data medis penderita, namun karena keterbatasan keamanan pada platform Telegram, data sensitif seperti informasi medis akan disimpan dengan enkripsi yang sesuai dan hanya dapat diakses oleh penderita yang bersangkutan dan penyedia layanan kesehatan yang berwenang. Selain itu, upaya akan dilakukan untuk memastikan bahwa *chatbot* memiliki lapisan keamanan yang memadai untuk melindungi informasi sensitif pengguna.

1.5 Sistematika Penulisan

Sistematika Penulisan adalah susunan atau pengaturan struktur penulisan suatu karya ilmiah atau laporan penelitian. Sistematika penulisan membantu pembaca atau peneliti lain untuk memahami bagaimana informasi disusun dan menyajikan topik secara terstruktur. Dalam penulisan karya ilmiah, sistematika penulisan umumnya mencakup bab-bab atau bagian-bagian yang memadukan elemen-elemen berikut:

1. BAB 1: Pendahuluan Pada bab ini, akan diberikan latar belakang mengenai masalah manajemen rekam medis penderita Diabetes
2. BAB 2: Kajian Literatur Bab ini akan mengulas secara mendalam tentang Diabetes sebagai kondisi medis yang menjadi fokus penelitian.
3. BAB 3: Metodologi Bab ini akan membahas desain penelitian yang digunakan dalam penelitian ini. Penjelasan akan mencakup jenis penelitian, alat dan sumber daya yang digunakan, serta proses pengembangan
4. BAB 4: Implementasi dan Evaluasi Bab ini akan fokus pada tahap implementasi *chatbot* dalam konteks praktis. Proses pengujian dan evaluasi *chatbot* akan dijelaskan secara mendalam.
5. BAB 5: Kesimpulan dan Saran Bab ini akan memberikan kesimpulan utama dari penelitian, termasuk hasil-hasil penelitian dan implematasinya dalam manajemen rekam medis penderita Diabetes.

BAB II

KAJIAN LITERATUR

Bab kajian literatur ini mendalam menggali beragam aspek terkait pengembangan *chatbot* untuk manajemen rekam medis penderita Diabetes. Dalam bab ini, akan diuraikan berbagai aspek penting yang menjadi landasan teoritis dan pemahaman yang mendalam terhadap topik penelitian.

2.1 Diabetes

Diabetes adalah penyakit kronis yang ditandai dengan tingginya kadar gula darah atau glukosa dalam tubuh [5]. Glukosa merupakan sumber energi utama bagi sel tubuh manusia. Pada penderita diabetes, glukosa tersebut tidak dapat digunakan oleh tubuh karena kurangnya insulin atau resistensi insulin [6]. Diabetes dibagi menjadi beberapa tipe utama, yaitu:

1. Diabetes Tipe 1: Terjadi ketika sistem kekebalan tubuh menyerang dan menghancurkan sel-sel pankreas yang memproduksi insulin. Ini mengharuskan penderita Diabetes Tipe 1 untuk menerima insulin secara teratur [6].
2. Diabetes Tipe 2: Bentuk yang paling umum dari diabetes. Ini terjadi ketika tubuh tidak dapat menggunakan insulin dengan efisien (resistensi insulin) atau tidak memproduksi cukup insulin [6].
3. Diabetes Gestasional: Terjadi selama kehamilan dan biasanya menghilang setelah melahirkan. Namun, wanita yang pernah mengalami diabetes gestasional memiliki risiko lebih tinggi mengembangkan diabetes Tipe 2 di kemudian hari [6].
4. Diabetes Lainnya: Ada beberapa jenis diabetes lain yang kurang umum, termasuk diabetes monogenik (diabetes yang disebabkan oleh mutasi genetik), diabetes sekunder (diabetes yang muncul sebagai akibat penyakit lain), dan lain-lain [6].

Pengelolaan diabetes yang efektif sangat penting, yang melibatkan pemantauan gula darah, penggunaan obat-obatan, diet yang seimbang, dan gaya hidup sehat [6]. Dalam konteks penelitian ini, pengelolaan rekam medis penderita diabetes menjadi faktor

penting karena membantu penderita dan tenaga medis dalam pengelolaan yang lebih baik[5] .

2.2 *Chatbot*

Chatbot adalah program komputer yang dirancang untuk meniru percakapan manusia. *Chatbot* dapat digunakan dalam berbagai bidang, termasuk kesehatan. Namun, penggunaan *chatbot* dalam Telegram untuk kesehatan masih tergolong baru dan belum banyak diteliti. Salah satu penelitian yang membahas penggunaan *chatbot* pada masa pandemi COVID-19 adalah oleh Sugiono [9] . Penelitian ini menunjukkan bahwa *chatbot* dapat membantu dalam menyebarkan informasi resmi kepada penggunanya melalui teknologi kecerdasan buatan. *Chatbot* juga dapat membantu dalam memberikan informasi kesehatan yang akurat dan terpercaya kepada masyarakat luas. Selain itu, *chatbot* juga dapat membantu dalam mengumpulkan umpan balik dari penderita tentang pengalaman mereka dengan layanan kesehatan [9]. Saat ini, belum ada penelitian yang secara khusus membahas penggunaan *chatbot* dalam Telegram untuk rekam medis diabetes. Oleh karena itu, perlu dilakukan penelitian lebih lanjut untuk mengevaluasi efektivitas penggunaan *chatbot* dalam Telegram untuk rekam medis diabetes.

Namun, penggunaan *chatbot* dalam Telegram untuk kesehatan masih tergolong baru dan belum banyak diteliti. Salah satu penelitian yang membahas penggunaan *chatbot* pada masa pandemi COVID-19 adalah oleh Sugiono [9]. Penelitian ini menunjukkan bahwa *chatbot* dapat membantu dalam menyebarkan informasi resmi kepada penggunanya melalui teknologi kecerdasan buatan. *Chatbot* juga dapat membantu dalam memberikan informasi kesehatan yang akurat dan terpercaya kepada masyarakat luas. Selain itu, *chatbot* juga dapat membantu dalam mengumpulkan umpan balik dari penderita tentang pengalaman mereka dengan layanan kesehatan [9]. Saat ini, belum ada penelitian yang secara khusus membahas penggunaan *chatbot* dalam Telegram untuk rekam medis diabetes. Oleh karena itu, perlu dilakukan penelitian lebih lanjut untuk mengevaluasi efektivitas penggunaan *chatbot* dalam Telegram untuk rekam medis diabetes.

2.3 Manajemen Rekam Medis

Manajemen rekam medis memainkan peran penting dalam pengelolaan diabetes. Rekam medis penderita diabetes harus mencakup informasi tentang riwayat kesehatan penderita, hasil tes laboratorium, dan catatan medis lainnya [7]. Dalam penelitian ini, *chatbot* akan membantu penderita dalam pencatatan dan pemantauan data kesehatan mereka [7]. Dengan demikian, *chatbot* dapat membantu meningkatkan kualitas manajemen rekam medis penderita diabetes [7]. Selain itu, *chatbot* juga dapat membantu penderita dalam memahami kondisi mereka dan memberikan edukasi yang relevan tentang diabetes [8]. Dengan tingkat personalisasi dan kustomisasi yang ditingkatkan, *chatbot* dapat memahami kebutuhan dan preferensi individu penderita, dan memberikan informasi yang sesuai dengan situasi pribadi mereka [8]. Penelitian ini akan mengukur efektivitas *chatbot* dalam membantu penderita diabetes dengan mengukur tingkat penggunaan *chatbot*, perbaikan dalam pemahaman penderita tentang diabetes, dan kemampuan penderita dalam mencatat dan memantau data kesehatan mereka [10]. Ini adalah dasar pemahaman mengenai mengapa pengelolaan rekam medis menjadi fokus penelitian.

1. Pemantauan Kesehatan Penderita: Rekam medis memungkinkan tenaga medis untuk memantau kondisi kesehatan penderita diabetes secara teratur. Data medis seperti tingkat gula darah, tekanan darah, berat badan, dan sejarah pengobatan dicatat dan dipantau.
2. Diagnosis dan Pengobatan: Rekam medis adalah sumber informasi kunci yang digunakan oleh dokter untuk mendiagnosis kondisi penderita. Data dalam rekam medis membantu dokter untuk memahami perkembangan penyakit dan meresepkan pengobatan yang sesuai.
3. Pengelolaan Pengobatan: Rekam medis mencatat riwayat pengobatan penderita diabetes, termasuk jenis obat yang digunakan, dosis, dan jadwal pengobatan. Hal ini memastikan penderita mendapatkan pengobatan yang konsisten dan sesuai.
4. Pemantauan Gula Darah: Data gula darah yang tercatat dalam rekam medis membantu penderita dan dokter untuk memahami bagaimana makanan,

olahraga, dan obat-obatan memengaruhi kadar gula darah. Ini memungkinkan penyesuaian yang tepat dalam pengobatan dan pola hidup.

5. Pemantauan Berat Badan dan Tekanan Darah: Penderita diabetes sering perlu memantau berat badan dan tekanan darah mereka. Rekam medis mencatat perkembangan ini seiring waktu dan membantu penderita dan dokter mengidentifikasi perubahan yang mungkin memerlukan tindakan.
6. Pemantauan Komplikasi: Diabetes dapat menyebabkan komplikasi serius seperti kerusakan mata, gangguan saraf, masalah jantung, dan lainnya. Rekam medis mencatat gejala dan hasil tes yang berkaitan dengan komplikasi ini, memungkinkan deteksi dini dan pengelolaan yang tepat.
7. Kerjasama Antar Tenaga Medis: Rekam medis memungkinkan berbagai tenaga medis yang terlibat dalam perawatan penderita diabetes, termasuk dokter umum, ahli endokrinologi, ahli gizi, dan perawat, untuk berbagi informasi dan bekerja sama dalam perawatan penderita.
8. Pendidikan Penderita: Rekam medis dapat berfungsi sebagai alat pendidikan bagi penderita diabetes. Penderita dapat melihat bagaimana keputusan tentang makanan, olahraga, dan pengobatan memengaruhi kondisi mereka.
9. Pemantauan Jangka Panjang: Rekam medis berfungsi sebagai catatan jangka panjang yang membantu penderita dan tenaga medis dalam melihat perkembangan sepanjang waktu. Hal ini memungkinkan perencanaan perawatan jangka panjang yang efektif.
10. Kepatuhan Penderita: Rekam medis dapat membantu memantau tingkat kepatuhan penderita terhadap rekomendasi perawatan. Penderita dan dokter dapat bekerja sama untuk mengidentifikasi faktor-faktor yang mungkin memengaruhi kepatuhan dan mencari solusi.
11. Perencanaan Perawatan: Rekam medis membantu dalam perencanaan perawatan jangka panjang dan perencanaan tindakan yang sesuai. Dokter dapat merencanakan kunjungan berdasarkan data rekam medis.

Manajemen rekam medis yang efektif dalam diabetes membantu meningkatkan perawatan dan kualitas hidup penderita. Dengan bantuan teknologi seperti *chatbot*

berbasis Telegram yang mengelola rekam medis, penderita dapat lebih mudah memantau dan mengelola data kesehatan mereka, sementara tenaga medis dapat memiliki akses yang lebih baik ke informasi penting untuk memberikan perawatan yang efektif.

2.4 Alasan Pentingnya Penelitian adalah

1. Telegram sebagai *Platform* Pesan Populer: Telegram memiliki jutaan pengguna di seluruh dunia, dan penggunaannya terus berkembang. Memanfaatkan *platform* ini untuk pengelolaan rekam medis dapat memperluas akses penderita diabetes terhadap layanan kesehatan.
2. Kemudahan Penggunaan: *Chatbot* dirancang untuk menjadi *user-friendly* dan mudah digunakan oleh berbagai kelompok usia. Mereka dapat memudahkan penderita diabetes dalam mencatat dan memantau data medis mereka tanpa memerlukan pengetahuan teknis yang tinggi.
3. Potensi untuk Peningkatan Kualitas Hidup: *Chatbot* dapat memberikan informasi yang akurat, membantu penderita diabetes mengelola data medis mereka dengan lebih baik, dan meningkatkan pemahaman mereka tentang perawatan yang diperlukan. Ini dapat berkontribusi pada peningkatan kualitas hidup penderita diabetes .
4. Pengembangan Selanjutnya: Penelitian ini dapat membuka pintu untuk pengembangan *chatbot* yang lebih canggih dalam pengelolaan penyakit kronis lainnya. Ini dapat memiliki dampak positif yang lebih besar dalam dunia kesehatan.

2.5 Telegram

Telegram adalah salah satu *platform* pesan populer yang telah menjadi tempat yang baik untuk mengembangkan berbagai jenis aplikasi, termasuk *chatbot* [11]. Berikut adalah penjelasan lebih rinci tentang Telegram sebagai *platform* untuk pengembangan *chatbot*:

1. Bot API Telegram: Telegram menyediakan API yang kuat untuk pengembangan *chatbot*, yang dikenal sebagai Bot API. API ini memungkinkan

pengembang untuk membuat bot yang bisa menerima pesan dari pengguna, merespons dengan pesan yang sudah diprogram, mengirim multimedia seperti gambar dan dokumen, dan melakukan banyak tugas lainnya. API ini mudah digunakan dan sangat fleksibel dalam pengembangan bot.

2. **Fitur Kaya:** Telegram menyediakan berbagai fitur yang dapat dimanfaatkan oleh *chatbot*, seperti keyboard inline, tampilan dokumen yang kaya, pesan suara, dan banyak lagi. Ini memungkinkan pengembang untuk menciptakan pengalaman yang interaktif dan kaya dalam *chatbot* mereka.
3. **Notifikasi Real-time:** Telegram mendukung notifikasi real-time, sehingga *chatbot* dapat memberikan informasi dan pemberitahuan segera kepada pengguna. Ini penting dalam konteks manajemen rekam medis dan pengingat perawatan medis.
4. **Kompatibilitas Multiplatform:** Telegram dapat digunakan di berbagai *platform*, termasuk iOS, Android, Windows, macOS, dan dalam bentuk web. *Chatbot* yang dibangun di Telegram dapat diakses oleh pengguna dari berbagai perangkat.
5. **Ketersediaan API Terbuka:** Telegram menyediakan API terbuka yang memungkinkan pengembang untuk memanfaatkan fitur Telegram dalam pengembangan *chatbot*. API ini memungkinkan pengembang untuk mengintegrasikan berbagai layanan dan fungsi dalam bot mereka.

Dengan fitur-fitur keamanan, API yang kuat, kemudahan penggunaan, dan berbagai fitur interaktif, Telegram menjadi pilihan yang populer untuk mengembangkan *chatbot*, terutama dalam konteks layanan kesehatan dan manajemen rekam medis. *Platform* ini memberikan banyak fleksibilitas dalam merancang bot yang sesuai dengan kebutuhan Anda.

2.6 Python dan aiogram

Python adalah bahasa pemrograman populer yang digunakan di berbagai bidang, termasuk perawatan kesehatan, dan AIogram adalah kerangka kerja Python untuk membuat bot Telegram. Menggunakan Python sebagai bahasa pemrograman untuk mengembangkan *chatbot* Telegram adalah pilihan yang populer dan beralasan, dan kerangka kerja AIogram merupakan alat yang berguna dalam pengembangan *chatbot* Telegram dengan Python. Berikut penjelasan lebih rinci mengenai kedua komponen ini. Analisis citra medis Python banyak digunakan dalam analisis citra medis, termasuk segmentasi, registrasi, dan klasifikasi. Pustaka Python seperti SimpleITK, OpenCV, dan scikit-image menyediakan alat untuk pemrosesan dan analisis gambar.

2.5.1 Python

Python menonjol sebagai pilihan utama untuk pengembangan *chatbot* karena sejumlah keunggulan yang dimilikinya. Pertama-tama, kemudahan pemahaman sintaksis Python merupakan aspek yang sangat menguntungkan, terutama bagi pengembang pemula dan berpengalaman. Dengan sintaksis yang mudah dibaca dan dipahami, tim pengembangan dapat lebih fokus pada logika bisnis dan fungsi *chatbot*, tanpa terlalu terbebani oleh kompleksitas sintaksis yang tinggi[12]. Hal ini mempercepat proses pengembangan dan memungkinkan penggunaan waktu yang lebih efisien dalam menghasilkan solusi yang berkualitas.

Selain itu, Python memiliki ekosistem yang kaya dengan banyak modul dan pustaka yang siap pakai, termasuk yang secara khusus dikembangkan untuk pengembangan *chatbot*. Keanekaragaman ini tidak hanya mempercepat proses pengembangan dengan menyediakan solusi siap pakai, tetapi juga mengurangi waktu yang diperlukan untuk mengimplementasikan fitur-fitur tambahan, memberikan fleksibilitas yang lebih besar dalam merancang *chatbot* sesuai kebutuhan. Terakhir, kemudahan integrasi Python dengan berbagai layanan dan API, termasuk API Telegram, merupakan aspek penting lainnya yang membuatnya menjadi pilihan yang sangat menarik. Dukungan yang baik untuk integrasi ini memungkinkan *chatbot* untuk berinteraksi dengan berbagai ekosistem dan platform lainnya dengan lancar,

meningkatkan cakupan dan fungsionalitasnya secara signifikan. Dengan demikian, kombinasi dari kemudahan pemahaman, keberagaman modul, dan kemudahan integrasi membuat Python menjadi pilihan yang sangat cocok untuk pengembangan *chatbot* yang efektif dan efisien.

2.5.2 AIogram

AIogram adalah kerangka kerja Python khusus untuk mengembangkan *chatbot* di Telegram. Berikut adalah penjelasan mengenai AIogram:

1. Asynchronous Programming: AIogram didasarkan pada konsep pemrograman asinkron (*async/await*) yang memungkinkan *chatbot* untuk merespons dengan cepat terhadap permintaan pengguna dan melakukan operasi yang panjang tanpa menghentikan eksekusi program. Ini sangat penting dalam *chatbot* yang harus menangani banyak permintaan secara bersamaan.
2. Interaksi dengan API Telegram: AIogram menyediakan API yang memudahkan pengembang untuk berinteraksi dengan API Telegram. Hal ini mencakup mengirim dan menerima pesan, membuat keyboard inline, mengelola grup, dan melakukan banyak tugas lainnya yang dibutuhkan oleh *chatbot*.
3. Manajemen Peristiwa: AIogram memungkinkan Anda untuk mengelola peristiwa dan interaksi yang terjadi dalam *chatbot*, seperti tindakan pengguna, pesan masuk, dan perubahan keanggotaan grup. Anda dapat merespons peristiwa ini dengan logika khusus dalam kode *chatbot* Anda.
4. Dokumentasi yang Kuat: AIogram memiliki dokumentasi yang lengkap dan jelas, yang membuatnya mudah untuk memulai pengembangan *chatbot* Telegram dengan Python. Dokumentasi ini mencakup contoh kode dan panduan langkah-demi-langkah.
5. Pengembangan Aktif: AIogram adalah proyek open-source dengan pengembangan aktif oleh komunitas. Ini berarti Anda dapat mengharapkan pembaruan dan perbaikan terus-menerus. Secara keseluruhan, Python dan AIogram dapat digunakan dalam berbagai aplikasi perawatan kesehatan, termasuk analisis citra medis, sistem EHR, telemedis, dan penelitian medis.

2.7 *MySQL* sebagai Basis Data

MySQL adalah sistem manajemen basis data (DBMS) populer yang digunakan untuk menyimpan data terstruktur. Berikut beberapa alasan mengapa *MySQL* merupakan pilihan yang baik untuk menyimpan rekam medis:

1. Kecepatan : *MySQL* dikenal dengan kinerjanya yang cepat dan dapat menangani data dalam jumlah besar secara efisien [13] .Hal ini penting untuk pengelolaan rekam medis yang bisa memuat banyak data.
2. Keandalan : *MySQL* merupakan DBMS handal yang banyak digunakan di berbagai industri, termasuk kesehatan [14]. Ia memiliki rekam jejak stabilitas dan keamanan yang terbukti, yang penting untuk menyimpan informasi medis sensitif.
3. Fleksibilitas : *MySQL* adalah DBMS fleksibel yang dapat disesuaikan untuk memenuhi kebutuhan spesifik. Ini mendukung berbagai tipe data dan dapat menangani pertanyaan dan transaksi yang kompleks [15] .Hal ini penting untuk mengelola rekam medis, yang dapat berisi berbagai jenis data dan memerlukan pertanyaan yang kompleks.
4. Structured Query Language (SQL) : *MySQL* menggunakan SQL, bahasa komputer standar yang digunakan untuk mengelola (mengakses dan memanipulasi) sistem *database* [16]. SQL adalah sintaks sederhana yang berisi perintah untuk pengambilan dan modifikasi data, atau biasa disebut query. SQL memiliki lima bagian dalam pengolahan data, yaitu pengambilan data, definisi data, manipulasi data, pengendalian data, dan bahasa transaksi data. SQL banyak digunakan dalam perawatan kesehatan untuk mengelola rekam medis.

Secara keseluruhan, *MySQL* adalah pilihan yang baik untuk menyimpan rekam medis karena kecepatan, keandalan, fleksibilitas, dan dukungan untuk SQL. Namun, penting untuk memastikan bahwa setiap aplikasi layanan kesehatan yang dikembangkan menggunakan *MySQL* mematuhi peraturan dan standar yang relevan untuk privasi dan keamanan data.

2.8 Peneliatan terkait

Penelitian terkait merujuk pada penelitian atau studi yang memiliki relevansi atau hubungan dengan topik atau subjek tertentu yang sedang diteliti. Penelitian terkait adalah sumber informasi yang dapat digunakan untuk memahami konteks, teori, metode, temuan, atau masalah yang terkait dengan penelitian utama yang sedang dilakukan. Berikut ini adalah tabel yang mencakup penelitian terkait dalam konteks perancangan *chatbot* untuk rekam medis diabetes di Telegram:

Tabel 2.8-1 Penelitian Terkait

No	Nama dan Tahun	Judul	Topik	Subjek
1	H. Priambodo and A. Muhajirin, 2022	Perancangan <i>Chatbot</i> Pendaftaran Siswa Dengan Telegram BOT	Information Technology, <i>Chatbots</i>	Merancang <i>chatbot</i> untuk pendaftaran siswa menggunakan Telegram BOT sebagai <i>platform</i> komunikasi.
2	D. Ismawati and I. Prasetyo, 2020	The Development of Telegram BOT Through Short Story	Information Technology, <i>Chatbots</i>	Mengembangkan <i>chatbot</i> menggunakan Telegram dengan pendekatan berdasarkan cerita pendek.
3	S. Sugiono, 2021	Pemanfaatan <i>Chatbot</i> Pada Masa Pandemi Covid-19 : Kajian Fenomena Society 5.0	Information Technology, <i>Chatbots</i> , COVID-19	Mempelajari penggunaan <i>chatbot</i> dalam situasi darurat atau krisis kesehatan selama pandemi COVID-19.

Dengan merujuk pada penelitian terkait yang paling relevan untuk pengembangan bot, pengembang dapat memperoleh wawasan yang mendalam tentang berbagai aspek krusial seperti keamanan data, penyimpanan data, implementasi bot, dan penggunaan

bot dalam situasi darurat kesehatan. Informasi yang terdapat dalam penelitian tersebut menjadi landasan yang berharga dalam merancang bot yang efektif dan aman untuk pengelolaan rekam medis diabetes melalui platform Telegram.

Dengan mempelajari hasil dari penelitian sebelumnya, pengembang dapat mengidentifikasi praktik terbaik, mengevaluasi risiko, dan merancang solusi yang memperhitungkan kebutuhan kesehatan serta privasi pengguna secara teliti. Ini akan membantu dalam menciptakan bot yang memberikan manfaat maksimal bagi pengguna, sambil tetap memastikan keamanan dan kerahasiaan informasi medis mereka terjaga dengan baik.

2.9 Metode pengembangan software

Metodologi pengembangan perangkat lunak adalah pendekatan sistematis yang digunakan untuk merencanakan, merancang, mengimplementasikan, menguji, dan memelihara perangkat lunak. Beberapa metodologi pengembangan perangkat lunak yang umum digunakan antara lain:

1. Waterfall (Air Terjun): Metodologi pengembangan perangkat lunak yang linear, di mana setiap fase pengembangan dilakukan secara berurutan dari tahap analisis, desain, implementasi, pengujian, hingga pemeliharaan.
2. Agile: Pendekatan pengembangan perangkat lunak yang iteratif dan inkremental, di mana pengembangan dilakukan dalam siklus pendek yang disebut sprint. Tim bekerja secara kolaboratif dan fleksibel, dengan fokus pada pengiriman produk yang berfungsi dan bernilai tinggi dalam waktu singkat.
3. Scrum: Salah satu kerangka kerja Agile yang paling populer, di mana pengembangan dilakukan dalam iterasi pendek yang disebut sprint. Tim bekerja dalam tim terorganisir kecil (tim Scrum) dan melakukan pertemuan harian untuk menyinkronkan aktivitas.

Setiap metodologi memiliki kelebihan dan kelemahan tertentu, dan pemilihan metodologi tergantung pada karakteristik proyek, kebutuhan bisnis, dan preferensi tim pengembangan. Dalam praktiknya, seringkali metodologi pengembangan perangkat lunak dapat disesuaikan atau dikombinasikan dengan kebutuhan spesifik proyek.

2.10 Agile Method



Gambar 2.10-1 Agile Methode

Metodologi *Agile* adalah pendekatan pengembangan perangkat lunak yang fokus pada kolaborasi, fleksibilitas, dan iterasi. Dalam pengembangan *chatbot*, prinsip-prinsip *Agile* termasuk pembagian proyek ke dalam iterasi (*sprint*), prioritas nilai untuk fitur, responsif terhadap perubahan, pengujian terus-menerus, dan keterlibatan pengguna akhir yang merujuk pada gambar *agile methode*. Dengan ini, Anda dapat mengembangkan *chatbot* yang lebih responsif terhadap perubahan kebutuhan dan prioritas pengguna.

1. Kolaborasi Tim: Fokus pada kerja sama tim.
2. Pengembangan Iteratif: Proyek dibagi ke dalam iterasi mandiri.
3. Prioritas Nilai: Fitur prioritas berdasarkan nilai pengguna.
4. Responsif terhadap Perubahan: Siap merespons perubahan.
5. Pelibatan Pengguna Akhir: Pengguna terlibat dalam evaluasi produk.
6. Pengujian Terus-Menerus: Pengujian berkelanjutan secara terus menerus
7. Siklus Retrospektif: Evaluasi dan perbaikan setelah setiap iterasi.
8. Pemantauan Kinerja: Pantau kemajuan dan performa.
9. Kualitas Utama: Fokus pada kualitas perangkat lunak.
10. Deliver Early and Often: Sering memberikan produk yang berguna.

11. Kendali Terpusat pada Tim: Tim memiliki otonomi dalam mengelola pekerjaan.

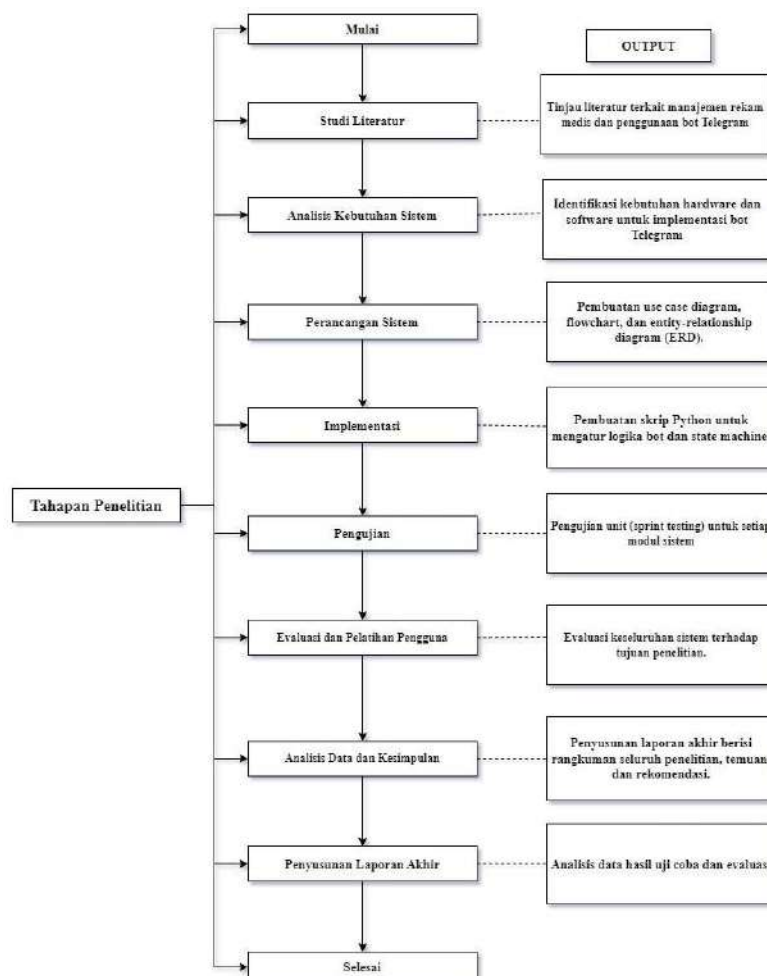
Metodologi Agile dapat digunakan dalam berbagai konteks pengembangan perangkat lunak, termasuk pengembangan *chatbot* dalam kesehatan. Dengan pendekatan ini, Anda dapat lebih responsif terhadap perubahan dalam kebutuhan pengguna, mengurangi risiko pengembangan, dan meningkatkan kepuasan pengguna akhir.

BAB III METODOLOGI PENELITIAN

Metodologi adalah pendekatan dan metode yang digunakan dalam penelitian untuk merencanakan, melaksanakan, dan menganalisis data dengan tujuan mencapai hasil penelitian yang valid dan konsisten. Ini mencakup pemilihan metode pengumpulan data, kerangka kerja teoritis, dan teknik analisis data yang digunakan.

3.1 Tahapan Penelitian

Tahapan penelitian adalah serangkaian langkah yang harus diikuti dalam menjalankan suatu penelitian. Berikut adalah tahapan penelitian yang umumnya diterapkan:

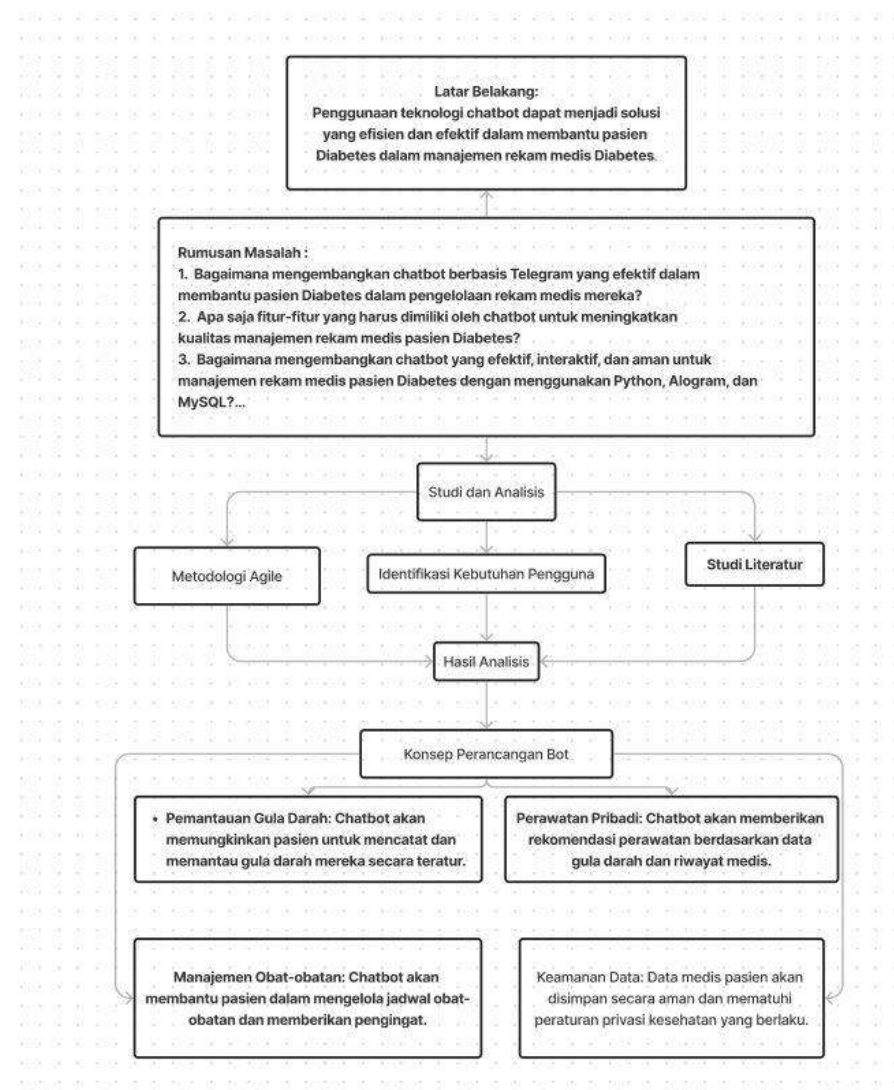


Gambar 3.1-1 Tahapan Penelitian

Pada gambar tahapan penelitian , dalam penelitian pengembangan *chatbot* untuk manajemen rekam medis penderita Diabetes, Setiap tahapan ini memiliki tujuan dan output khusus yang berkontribusi pada pengembangan *chatbot* dan dokumentasi penelitian secara keseluruhan. Dengan mengikuti tahapan ini dengan cermat, Anda akan dapat menjalankan penelitian dengan efektif dan meraih hasil yang valid dan bermanfaat.

3.2 Rancangan Penelitian

Rancangan penelitian adalah rencana sistematis yang merinci langkah-langkah yang akan diambil untuk menjalankan penelitian dengan tujuan mencapai hasil penelitian yang valid dan relevan. Rancangan penelitian berikut :



Gambar 3.2-1 Rancangan Penelitian

Dalam gambar yang dijelaskan pada Rancangan Penelitian pengembangan *chatbot* untuk manajemen rekam medis penderita Diabetes, latar belakangnya adalah kebutuhan akan solusi efisien dalam pemantauan dan pengelolaan penyakit ini. Rumusan masalah melibatkan pengembangan *chatbot* yang efektif, interaktif, dan aman dengan menggunakan Python, AIogram, dan *MySQL*.

3.3 Jenis Penelitian

Penelitian Pengembangan (*Developmental Research*) dalam konteks merancang dan mengimplementasikan *chatbot* berbasis Telegram untuk manajemen rekam medis penderita diabetes adalah suatu pendekatan penelitian yang berorientasi pada pengembangan solusi inovatif untuk meningkatkan layanan kesehatan. Proses ini melibatkan serangkaian langkah-langkah yang cermat dan terstruktur untuk memastikan bahwa solusi yang dihasilkan memenuhi kebutuhan unik penderita diabetes dan dapat diintegrasikan secara efektif dalam lingkungan pelayanan kesehatan yang ada.

3.4 Metode Analisis

Metode analisis dalam konteks pengembangan *chatbot* manajemen rekam medis penderita Diabetes dapat mencakup analisis berbagai aspek, termasuk analisis kebutuhan pengguna, analisis data, serta evaluasi dan pengujian *chatbot*. Berikut adalah beberapa langkah metode analisis yang dapat digunakan.

3.4.1 Analisis Kebutuhan Pengguna

1. Kelompok Pengguna :

Tabel 3.4-1 Kelompok Pengguna

No	Kelompok Pengguna Utama	Deskripsi
1	Penderita Diabetes	Penderita yang menggunakan <i>chatbot</i> untuk mencatat dan mengelola rekam medis mereka.

Pada table tersebut menjelaskan penderita diabetes adalah individu dengan diabetes yang menggunakan *chatbot* untuk mencatat dan mengelola rekam medis, serta menerima rekomendasi berdasarkan data medis.

2. Kebutuhan Fungsional :

Tabel 3.4-2 Kebutuhan Fungsional

Kebutuhan	Deskripsi
Pengisian dan Pengelolaan Profil	Penderita dapat mengisi dan mengelola profil mereka.
Pencatatan Data Rekam Medis	Penderita dapat mencatat data gula darah, tekanan darah, dll.
Akses Mudah ke Data Rekam Medis Sebelumnya	Kemampuan untuk mengakses rekam medis sebelumnya untuk referensi.
Pembuatan Insight	<i>Chatbot</i> memberikan insight atau visualisasi data rekam medis.

Gambar 3.4-1 Kebutuhan fungsional

Pada tabel ini menjelaskan kebutuhan fungsional untuk *chatbot* mencakup pengisian dan pengelolaan profil penderita, pencatatan data rekam medis seperti gula darah dan tekanan darah, kemudahan akses ke data rekam medis sebelumnya, serta kemampuan *chatbot* untuk memberikan insight atau visualisasi dari data rekam medis yang telah dicatat, membantu penderita memahami pola kesehatan dan membuat keputusan yang lebih baik terkait perawatan mereka.

3. Kebutuhan Non-Fungsional:

Tabel 3.4-3 Non-Fungsional

No	Kebutuhan	Deskripsi
1	Antarmuka Pengguna yang Ramah	Antarmuka pengguna <i>chatbot</i> harus ramah dan mudah.

Kebutuhan non-fungsional utama adalah memastikan antarmuka pengguna *chatbot* yang ramah dan mudah digunakan, dengan desain intuitif dan responsif untuk memastikan pengalaman pengguna yang baik dan efisien.

4. Prioritaskan Kebutuhan :

Tabel 3.4-4 Kebutuhan Prioritas

No	Kebutuhan	Prioritas	Deskripsi Singkat
1	Pengisian dan Pengelolaan Profil	Tinggi	Fokus pada kemampuan pengguna untuk mengisi dan mengelola profil secara efisien.
2	Pencatatan Data Rekam Medis	Sedang	Kebutuhan untuk mencatat data medis penting dan memastikan keakuratan informasi.

3	Pembuatan Insight	Sedang	Pentingnya <i>chatbot</i> memberikan insight atau visualisasi dari data rekam medis.
---	-------------------	--------	--

Pada tabel diatas Dalam pengembangan *chatbot*, prioritas utama adalah pengisian dan pengelolaan profil, diikuti oleh pencatatan data rekam medis dan pembuatan insight yang memiliki prioritas sedang. Pencatatan data medis difokuskan untuk memastikan keakuratan informasi, sementara pembuatan insight menekankan pentingnya *chatbot* memberikan pemahaman tambahan atau visualisasi data rekam medis untuk mendukung pengguna dalam pengambilan keputusan terkait kesehatan.

3.4.2 Analisis Data Rekam Medis

proses analisis data yang terdapat dalam rekam medis pasien. Data rekam medis merupakan informasi yang sangat berharga bagi tenaga medis dalam memahami kondisi kesehatan seorang pasien serta dalam merencanakan perawatan yang tepat.

Tabel 3.4-5 Data rekam medis

Jenis Data Rekam Medis	Deskripsi
Gula Darah Sewaktu	Representasi tingkat glukosa dalam darah pada waktu tertentu.
Gula Darah Puasa	Kadar glukosa dalam darah setelah puasa selama semalam.
Tekanan Darah Sistolik	Tekanan darah saat jantung berkontraksi.
Tekanan Darah Diastolik	Tekanan darah saat jantung beristirahat di antara kontraksi.
Catatan	Catatan tambahan atau informasi khusus terkait rekam medis.

Pada table diatas menjelaskan jenis data rekam medis yang disediakan meliputi gula darah sewaktu dan gula darah puasa untuk mengukur kadar glukosa, serta tekanan darah sistolik dan diastolik untuk mencatat tekanan saat jantung berkontraksi dan beristirahat. Selain itu, terdapat juga catatan medis yang berisi informasi tambahan atau penting terkait kesehatan penderita.

BAB IV

IMPLEMENTASI DAN EVALUASI

Implementasi adalah fase di mana konsep-konsep dan desain yang telah dirancang di tahap sebelumnya (analisis dan desain) diterjemahkan menjadi kode yang dapat dijalankan. Untuk bot Telegram ini, implementasi melibatkan penerapan logika bisnis, koneksi ke *database*, penggunaan API Telegram (dengan menggunakan *library aiogram*), dan penyusunan struktur program secara keseluruhan.

4.1 Analisa Dan Perancangan

Analisis dan perancangan sistem adalah dua tahap awal dalam siklus pengembangan perangkat lunak yang membentuk dasar untuk menghasilkan solusi perangkat lunak yang efektif dan efisien.

4.1.1 Analisa Kebutuhan Sistem

Analisis kebutuhan sistem mencakup kebutuhan hardware dan software yang diperlukan untuk menjalankan aplikasi *chatbot* berbasis Telegram untuk manajemen rekam medis penderita diabetes.

1. Kebutuhan *Hardware* :

Tabel 4.1-1K ebutuhan hardware

No.	Komponen	Spesifikasi
1	<i>Server</i>	Minimal CPU: 2 GHz, RAM: 8 GB
2	<i>Storage</i>	Minimal 50 GB untuk penyimpanan data

Keterangan Tabel kebutuhan hardware Server di sini mencakup server tempat aplikasi di-host dan menyimpan basis data.

2. Kebutuhan *Software* :

Tabel 4.1-2 Kebutuhan Software

No.	Perangkat Lunak	Versi Minimum	Keterangan
1	<i>Python</i>	3.6	Bahasa pemrograman utama untuk pengembangan bot.

2	<i>MySQL</i>	5.7	Sistem manajemen basis data untuk menyimpan data.
3	<i>Telegram API</i>	-	Akses <i>API</i> Telegram untuk mengintegrasikan bot.
4	<i>Aioogram Library</i>	2.14.2	Library Python untuk pengembangan bot Telegram.
5	<i>Matplotlib</i>	3.4.3	Library untuk membuat grafik dan visualisasi data.
6	<i>MySQL Connector</i>	8.0.26	Koneksi antara aplikasi <i>Python</i> dan <i>MySQL Database</i> .

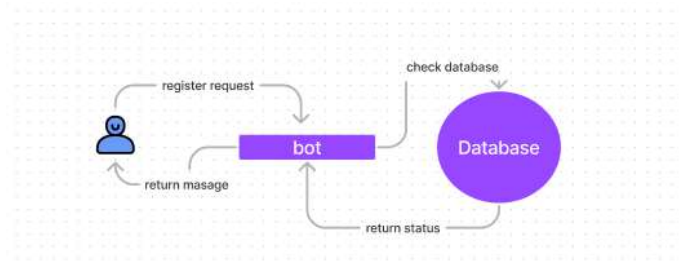
Keterangan pada Tabel kebutuhan software menjelaskan Versi perangkat lunak di atas adalah versi minimum yang diuji dan dipastikan kompatibel dengan aplikasi.

4.1.2 Perancangan Sistem

Perancangan sistem ini mencakup rancangan dan arsitektur secara logic dan fisik dari aplikasi *chatbot* rekam medis penderita diabetes.

1. Desain Arsitektur Logik

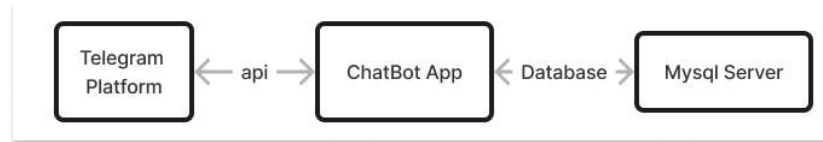
Struktur *Chatbot*: *Chatbot* menggunakan framework *aioogram* untuk interaksi dengan *platform* Telegram



Gambar 4.1-1 Arsitektur logik

Pada gambar diatas desain arsitektur logik Bot meminta pengguna untuk melakukan registrasi melalui web terlebih dahulu sebelum dapat menggunakan fitur-fitur lainnya. Proses registrasi dilakukan di luar Telegram, dan informasi pengguna yang dimasukkan akan disimpan dalam basis data *MySQL*. Terdapat dua state groups: satu untuk pengisian/edit profil dan satu lagi untuk pengisian rekam medis.

2. Arsitektur sistem



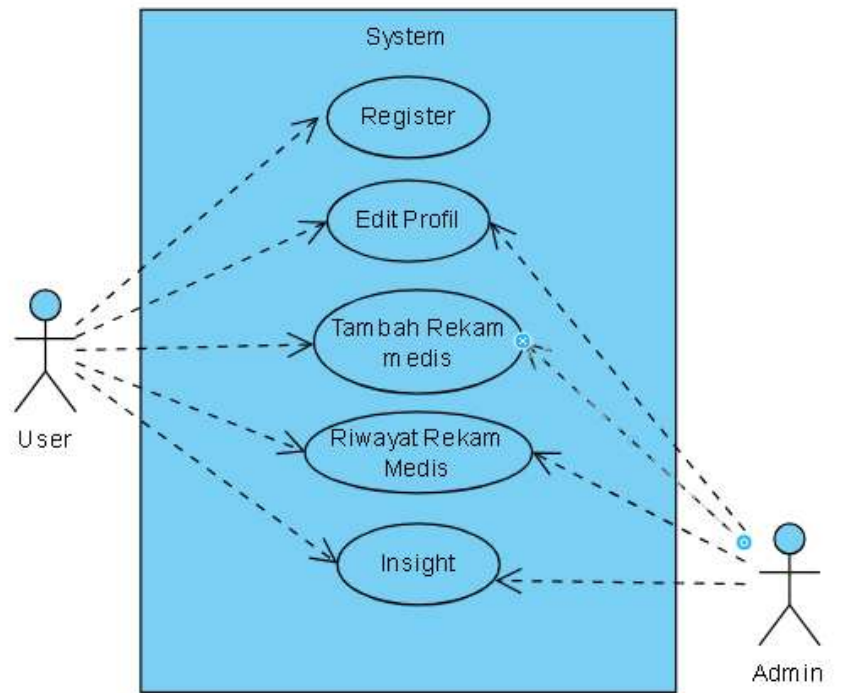
Gambar 4.1-2 Diagram arsitektur sistem

Diagram ini memberikan gambar diagram arsitektur sistem visual tentang interaksi antara *Telegram Platform*, *Chatbot Application* (menggunakan aiogram), dan *MySQL Server* yang menyimpan data pengguna.

Perancangan sistem ini bertujuan untuk memberikan pengalaman pengguna yang baik dalam pengelolaan profil dan rekam medis mereka melalui *chatbot*. Dengan proses registrasi yang terintegrasi dengan sistem eksternal dan kemampuan untuk mengakses dan mengelola informasi profil dan rekam medis melalui Telegram, diharapkan pengguna akan merasa lebih nyaman dan mudah dalam mengelola informasi kesehatan mereka. Dengan demikian, sistem ini diharapkan dapat meningkatkan kualitas layanan bagi penderita diabetes dalam manajemen dan pemantauan kondisi kesehatan mereka melalui platform *chatbot* yang intuitif dan efisien.

4.1.3 Use Case Diagram

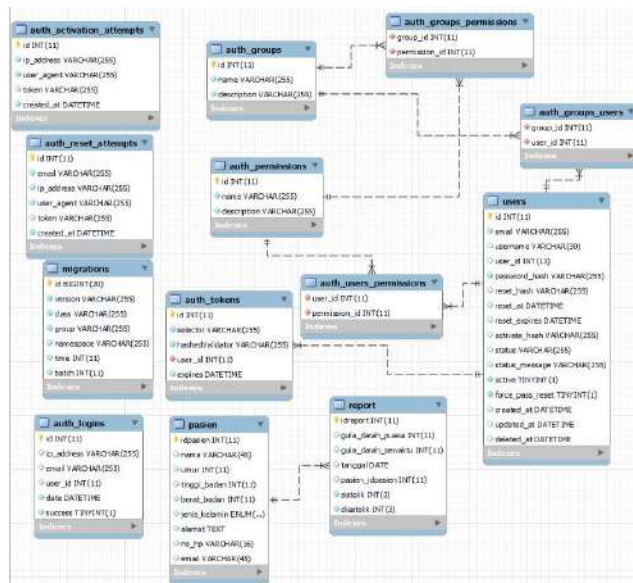
Use case diagram merupakan alat visual yang mendeskripsikan bagaimana sistem berinteraksi dengan pengguna atau entitas lain. Berikut adalah beberapa skenario use case untuk sistem rekam medis di bot Telegram:



Gambar 4.1-3 Usecase

Diagram use case pada gambar usecase di atas mencerminkan interaksi antara pengguna dan sistem rekam medis melalui berbagai fungsi yang tersedia.

4.1.4 Entity-Relationship Diagram (ERD)



Gambar 4.1-4 Erd Database

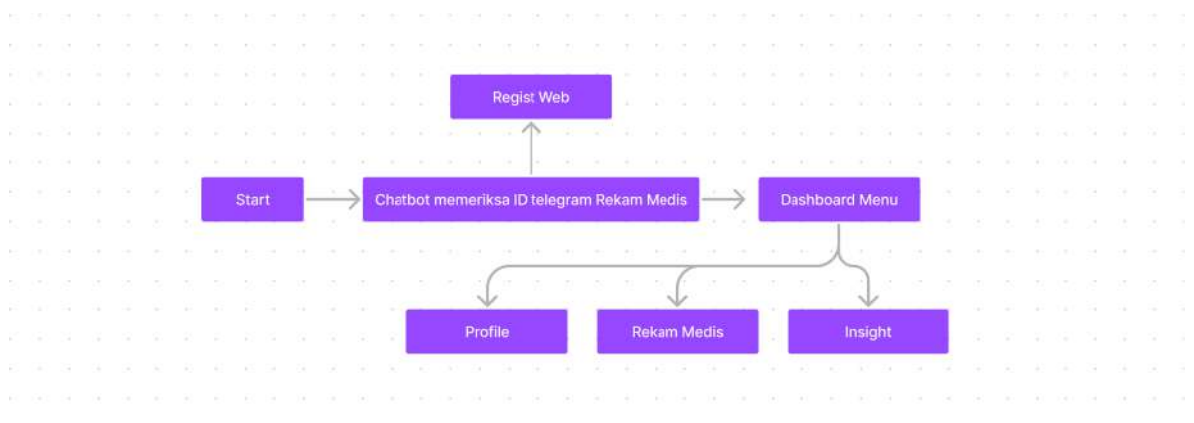
ERD pada gambar ERD menggambarkan entitas dan hubungan antara entitas dalam basis data. Keterangan ERD:

1. Tabel utama: users, profile, rekam_medis.
2. Hubungan antara tabel menggambarkan keterkaitan data antara pengguna, profil, dan rekam medis.

Dengan struktur database ini, *chatbot* Telegram dapat efisien menyimpan dan mengelola informasi profil pengguna serta rekam medis yang terkait. Setiap pengguna memiliki profil penderita yang unik, dan dapat memiliki banyak rekam medis yang tercatat. Hal ini memungkinkan bot untuk memberikan layanan yang tepat dan personal kepada pengguna sesuai dengan kebutuhan kesehatan mereka.

4.1.5 Alur Percakapan

Chatbot Telegram yang dirancang khusus untuk memfasilitasi manajemen profil pengguna dan rekam medis mereka. Bot ini memungkinkan pengguna untuk dengan mudah mengelola informasi pribadi mereka, merekam data medis seperti gula darah dan tekanan darah, serta mendapatkan wawasan tentang kondisi kesehatan mereka melalui grafik dan analisis yang disediakan oleh *chatbot*.



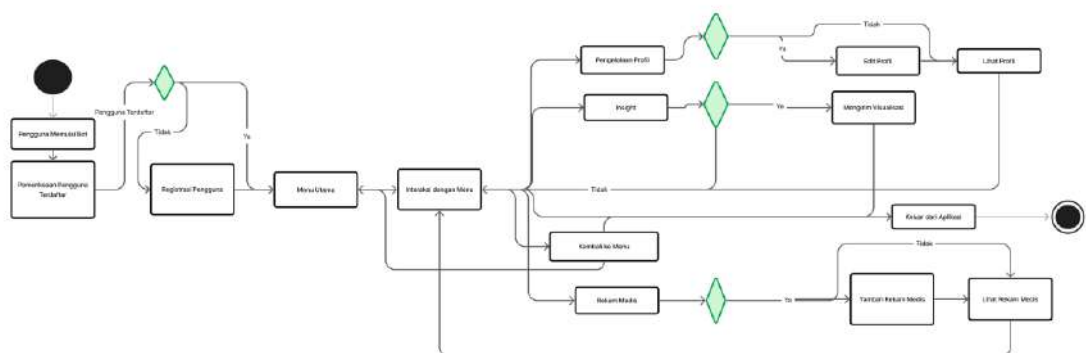
Gambar 4.1-5 Alur Percakapan

Pertama, pengguna memulai interaksi dengan bot dan diberi opsi untuk mengelola profil, merekam data medis, atau mendapatkan wawasan kesehatan. Jika pengguna memilih untuk mengedit profil, mereka diminta untuk memasukkan informasi pribadi seperti nama, usia, dan alamat, yang kemudian disimpan oleh bot.

Selanjutnya, pengguna dapat membuat rekam medis baru dengan memasukkan data seperti gula darah dan tekanan darah, atau melihat dan menghapus rekam medis yang ada. Terakhir, dengan opsi Insight, pengguna dapat melihat grafik yang menampilkan tren kesehatan mereka berdasarkan data rekam medis yang telah disimpan, memberikan pemahaman yang lebih baik tentang kondisi kesehatan mereka. Dengan alur percakapan ini, pengguna dapat dengan mudah mengelola profil dan data medis mereka serta mendapatkan wawasan yang berharga tentang kesehatan mereka melalui interaksi yang intuitif dengan bot.

4.1.6 Activity Diagram

Diagram aktivitas (Activity Diagram) adalah jenis diagram dalam UML yang menggambarkan alur kerja atau aktivitas yang terjadi dalam suatu proses atau skenario. Berikut adalah diagram untuk aktivitas utama dari *chatbot* telegram rekam medis diabetes :



Gambar 4.1-6 Diagram Aktivitas

Diagram ini menggambarkan alur dasar dari program berdasarkan input pengguna dan interaksi dengan database. Dalam setiap langkah, program menanggapi perintah pengguna, memperbarui basis data sesuai, dan memberikan balasan atau tindakan yang sesuai.

4.1.7 Metode Pengujian

Pengujian sistem akan dilakukan dengan pendekatan *Agile*, memastikan bahwa setiap perubahan atau penambahan fitur diuji secara menyeluruh selama proses pengembangan. Berikut adalah perincian perancangan pengujian dengan metode

Agile. Pengujian akan mencakup berbagai aspek fungsional dan non-fungsional dari sistem, memastikan kualitas keseluruhan produk. Prosedur pengujian akan dibagi menjadi beberapa tahap yang sesuai dengan metodologi *Agile*.

Tabel 4.1-3 Tahap Pengujian

Pendekatan Pengujian	Deskripsi
Pengujian Unit	Setiap unit fungsionalitas dalam fitur Telegram diuji secara terpisah menggunakan alat seperti Pytest.
Pengujian Penerimaan Pengguna (UAT)	Pengujian melibatkan pengguna aktual atau representatif dengan skenario pengujian realistis menggunakan alat BDD seperti behave.
Pengujian dengan Berbagai ID Pengguna	Dilakukan untuk mengevaluasi respons terhadap interaksi dari berbagai pengguna, termasuk interaksi antara pengguna dan bot Telegram jika berlaku.

Dengan mengadopsi pendekatan-pendekatan ini, fitur-fitur Telegram dapat diuji secara menyeluruh dan sesuai dengan harapan pengguna. Tabel yang memuat alat-alat pengujian yang dapat digunakan untuk masing-masing jenis pengujian :

Tabel 4.1-4 Alat pengujian

Jenis Pengujian	Alat Pengujiannya
Pengujian Unit	Pytest
UAT untuk Fitur Telegram	Behave
Pengujian dengan Berbagai ID Pengguna	Telegram

Dalam hal ini, *pytest* digunakan sebagai kerangka pengujian utama, sedangkan *Dispatcher* digunakan sebagai bagian dari pengujian integrasi, UAT, dan pengujian dengan berbagai ID pengguna. Kombinasi keduanya dapat memberikan cakupan pengujian yang lebih komprehensif dalam pengembangan *chatbot* untuk fitur-fitur Telegram.

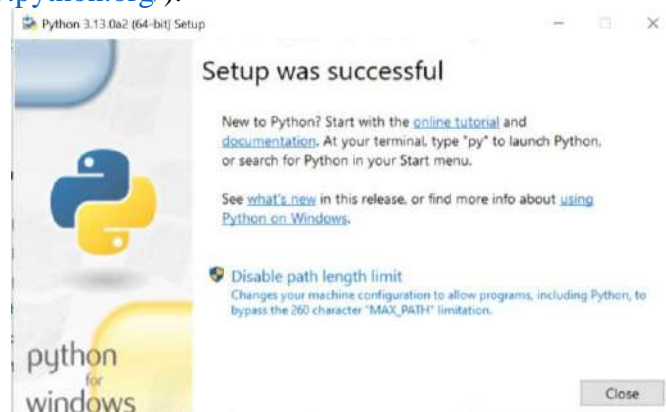
4.2 Implementasi

Implementasi sistem merupakan tahap eksekusi dari perancangan yang telah dilakukan sebelumnya. Langkah-langkah instalasi, konfigurasi, dan penerapan komponen sistem dijelaskan dengan rinci untuk memastikan keberhasilan implementasi

4.2.1 Instalasi dan Konfigurasi Python dan Libraries

Langkah pertama dalam pengembangan bot adalah menginstal dan mengonfigurasi Python serta perpustakaan yang diperlukan. Python adalah bahasa pemrograman utama yang digunakan, sedangkan perpustakaan seperti AIogram digunakan untuk berinteraksi dengan API Telegram. Ini adalah langkah awal penting sebelum mulai mengembangkan bot lebih lanjut.

1. Instalasi *Python*: Unduh dan instal versi terbaru *Python* dari situs resmi (<https://www.python.org/>).



Gambar 4.2-1 Install Python

Pada gambar di atas, berhasil diinstallnya Python dapat diidentifikasi melalui beberapa ciri. Kita dapat melihat pesan instalasi sukses tanpa adanya pesan kesalahan, memberikan indikasi bahwa Python telah terpasang dengan benar. Tampilan versi Python yang muncul juga menegaskan bahwa instalasi telah berhasil dilakukan.

2. AIogram Library: Gunakan manajer paket pip untuk menginstal AIogram dengan perintah `pip install aiogram`.

```
C:\Users\KHAFI COMP>pip install aiogram
```

Gambar 4.2-2 Install Aiogram

Setelah proses instalasi selesai, Anda dapat memulai pengembangan dengan AIOgram di proyek Python Anda. Untuk memeriksa versi AIOgram yang terinstal, Anda dapat menggunakan perintah berikut di terminal atau command prompt:

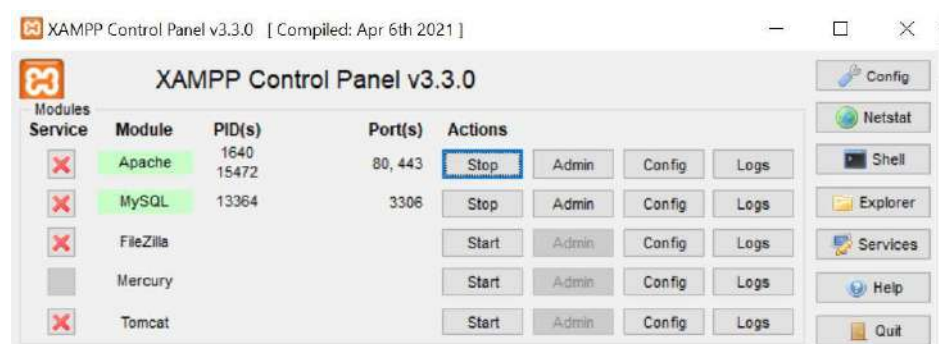
```
C:\Users\KHAFI COMP>pip show aiogram
Name: aiogram
Version: 3.2.0
Summary: Modern and fully asynchronous framework for Telegram Bot API
Home-page:
Author:
Author-email: Alex Root Junior <jroot.junior@gmail.com>
License:
Location: C:\Users\KHAFI COMP\AppData\Local\Programs\Python\Python311\Lib\site-packages
Requires: aiofiles, aiohttp, certifi, magic-filter, pydantic, typing-extensions
Required-by:
```

Gambar 4.2-3 Install Aiogram

Pada di atas Gambar ini akan menampilkan informasi terperinci tentang paket AIOgram yang terinstal, termasuk versi yang digunakan. Cari baris yang menyebutkan "Version" untuk mengetahui versi AIOgram yang sedang Anda gunakan.

4.2.2 Instalasi MySQL

Proses instalasi ini melibatkan mengunduh perangkat lunak *MySQL* dari situs resmi *MySQL* dan mengikuti panduan instalasi yang disediakan. Setelah instalasi selesai, pengguna dapat mulai menggunakan *MySQL* Database untuk menyimpan dan mengelola data yang diperlukan dalam pengembangan bot. Instalasi *MySQL* Server: Pasang *MySQL* Server dan ikuti panduan instalasi.

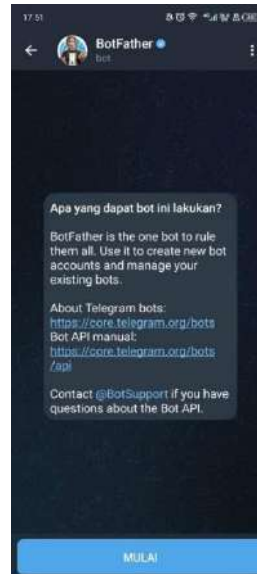


Gambar 4.2-4 Mysql server

Pada gambar diatas menunjukan bahwa *mysql* sudah berjalan. Konfigurasi *Database*: Buat *database* baru ("mydatabase3") dan atur pengguna serta hak aksesnya.

4.2.3 Telegram Bot

Untuk membuat bot Telegram, langkah pertama adalah menggunakan BotFather di aplikasi Telegram. BotFather adalah bot resmi Telegram yang memungkinkan pengguna untuk membuat dan mengelola bot mereka. Dengan menggunakan BotFather, pengguna dapat membuat bot baru dan memperoleh token bot yang diperlukan untuk menghubungkan bot dengan kode pengembangan. Gunakan BotFather di aplikasi Telegram untuk membuat bot dan dapatkan token bot:



Gambar 4.2-5 Botfather

Gambar ini menampilkan antarmuka Bot Father di Telegram, tempat di mana pengguna dapat dengan mudah membuat bot mereka sendiri dan mengakses token untuk integrasi, Ganti token bot pada kode dengan token yang diperoleh dari BotFather.



Gambar 4.2-6 Token bot

Pada Gambar diatas setelah membuat bot, BotFather akan memberikan token untuk bot tersebut. Token ini sangat penting untuk mengidentifikasi bot Anda.

4.2.1 Pemrograman dan penerapan

Pemrograman dan penerapan teknis dari bot Telegram untuk merekam dan mengelola data medis pengguna. Kita akan melihat lebih dalam tentang bagaimana bot ini diprogram dan diimplementasikan, termasuk penggunaan library dan teknologi tertentu.

1. Penanganan Pesan :

```
def menu_keyboard():
    keyboard = types.InlineKeyboardMarkup()
    keyboard.add(types.InlineKeyboardButton(text="Profil", callback_data="profile_menu"))
    keyboard.add(types.InlineKeyboardButton(text="Rekam Medis", callback_data="rekam_medis_menu"))
    keyboard.add(types.InlineKeyboardButton(text="Insight", callback_data="insight"))
    return keyboard

# Command /profile
@dp.callback_query_handler(text='profile_menu')
async def profile_menu(callback_query: types.CallbackQuery):
    await bot.send_message(callback_query.from_user.id, "Menu Profil", reply_markup=profile_keyboard())

# Keyboard untuk menu profil
def profile_keyboard():
    keyboard = types.InlineKeyboardMarkup()
    keyboard.add(types.InlineKeyboardButton(text="Edit/Isi Profile", callback_data="edit_profile"))
    keyboard.add(types.InlineKeyboardButton(text="Lihat Profile", callback_data="view_profile"))
    keyboard.add(types.InlineKeyboardButton(text="Kembali", callback_data="back_to_menu"))
    return keyboard

# Command /backtomenu
@dp.callback_query_handler(text='back_to_menu')
async def back_to_menu(callback_query: types.CallbackQuery):
    await bot.send_message(callback_query.from_user.id, "Kembali ke menu utama", reply_markup=menu_keyboard())
```

Gambar 4.2-7 Penanganan Pesan

Bot ini menggunakan aiogram untuk menangani pesan masuk dari pengguna. Berbagai jenis pesan, termasuk pesan teks dan pesan callback dari inline keyboard, ditangani dan diproses sesuai dengan fungsi-fungsi yang telah ditentukan.

2. State Machine :

```
# State untuk pengisian/edit profil
class ProfileStates(StatesGroup):
    NAMA = State()
    USIA = State()
    JENIS_KELAMIN = State()
    EMAIL = State()
    NOMOR_TELEPON = State()
    BERAT_BADAN = State()
    TINGGI_BADAN = State()
    ALAMAT = State()

# State untuk pengisian rekam medis
class RekamMedisStates(StatesGroup):
    GULA_DARAH_SEWAKTU = State()
    GULA_DARAH_PUASA = State()
    TEKANAN_DARAH_SISTOLIK = State()
    TEKANAN_DARAH_DIASTOLIK = State()
    DELETE_ID = State()
```

Gambar 4.2-8 State machine

Untuk mengelola proses yang melibatkan beberapa langkah, bot menggunakan konsep state machine yang disediakan oleh aiogram. Ini memungkinkan bot untuk mengatur keadaan atau state dari interaksi dengan pengguna, seperti saat pengguna mengisi formulir untuk mengedit profil atau menambahkan rekam medis baru.

3. Interaksi dengan Database :

```
Koneksi ke MySQL
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="mydatabase3"
)
```

Gambar 4.2-9 Interaksi Database

Bot ini berinteraksi dengan database *MySQL* untuk menyimpan dan mengambil data pengguna serta rekam medis mereka. Melalui koneksi *MySQL*, bot dapat melakukan operasi CRUD (Create, Read, Update, Delete) untuk mengelola data dengan efisien.

4. Visualisasi Grafik

```
f create_chart(parameter, data, normal_limit, filename):
    data_column_index = 0
    values = [float(row[data_column_index]) for row in data]
    labels = [parameter, 'Others']
    # Check if there's only one value, then adjust labels accordingly
    if len(values) == 1:
        labels = [parameter]

    percentages = [(value / sum(values)) * 100 for value in values]
    # Create the pie chart
    fig, ax = plt.subplots()
    wedges, texts, autotexts = ax.pie(
        percentages,
        labels=labels,
        autopct='%1.1f%%',
        startangle=90,
        colors=['skyblue', 'lightgrey'],
        wedgeprops=dict(width=0.4),
    )
    # Add text to each sector
    for autotext in autotexts:
        autotext.set_color('white')
        autotext.set_size(8) # Set font size to fit
    # Add normal limit line
    ax.axhline(y=normal_limit, color='r', linestyle='--', label=f'Normal Limit for {parameter}')
    ax.set_title(f'{parameter} Overview')
    ax.axis('equal')
    ax.legend()
    # Save the chart to a file
    plt.savefig(filename)
    plt.close()
```

Gambar 4.2-10 Visualisasi Grafik

Untuk opsi Insight, bot menghasilkan visualisasi grafik berdasarkan data medis yang disimpan dalam database. Dengan menggunakan matplotlib, bot membuat grafik yang menampilkan informasi seperti gula darah sewaktu, gula darah puasa, tekanan darah sistolik, dan tekanan darah diastolik dari rekam medis pengguna.

Dengan menggabungkan teknologi-teknologi ini dengan pemrograman yang efisien, bot Telegram ini mampu memberikan pengalaman pengguna yang baik dalam mengelola profil dan rekam medis mereka secara digital. Dengan kesederhanaan dan keterampilan teknis, bot ini menjadi alat yang bermanfaat dalam pemantauan kesehatan dan pengelolaan data medis pengguna.

4.3 Hasil Implementasi

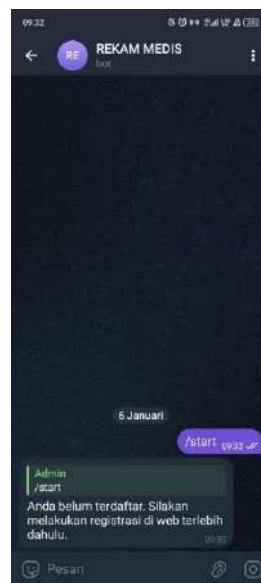
Hasil implementasi dari bot Telegram ini adalah adanya *platform* yang memudahkan pengguna dalam mengelola profil kesehatan dan pencatatan rekam medis mereka.

4.3.1 Interface

Antarmuka pengguna yang baik akan memudahkan pengguna untuk berinteraksi dengan bot, menyediakan pengalaman pengguna yang intuitif, dan memastikan informasi disampaikan dengan jelas dan efektif.

1. Tampilan Awal

Setelah pengguna memulai bot dengan perintah /start, mereka disambut dengan pesan selamat datang. Jika pengguna belum terdaftar, mereka diberitahu untuk melakukan registrasi terlebih dahulu melalui web.



Gambar 4.3-1 Tampilan Awal

Pesan selamat datang ini memberikan pengguna pemahaman awal tentang apa yang dapat mereka lakukan dengan bot.

3. Menu Utama

Setelah menerima pesan selamat datang, pengguna diberikan menu utama yang berisi opsi seperti Profil, Rekam Medis, dan Insight.

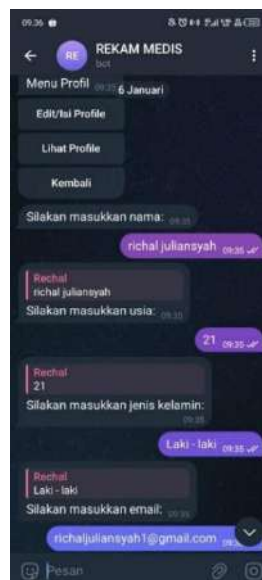


Gambar 4.3-2 Menu Utama

Menu ini disajikan dalam bentuk keyboard inline untuk kenyamanan pengguna. Pengguna dapat dengan mudah memilih opsi yang mereka inginkan dengan mengetuk tombol yang sesuai.

3. Profil

Di bawah opsi Profil, pengguna dapat memilih untuk mengedit profil mereka atau melihat profil saat ini. Jika pengguna memilih untuk mengedit profil.



Gambar 4.3-3 Profil

mereka akan diminta untuk memasukkan informasi profil seperti nama, usia, jenis kelamin, dan alamat. Proses pengisian formulir ini terstruktur dengan baik dan disertai dengan panduan yang jelas untuk memastikan pengguna memberikan informasi yang benar.

4. Rekam Medis

Di bawah opsi Rekam Medis, pengguna memiliki opsi untuk menambahkan rekam medis baru atau melihat rekam medis yang sudah ada. Saat menambahkan rekam medis baru,

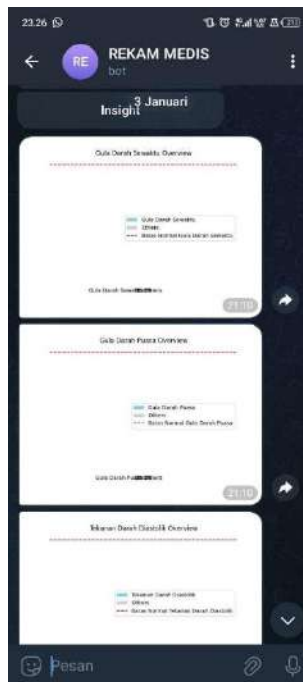


Gambar 4.3-4 Rekam Medis

pengguna diminta untuk memasukkan informasi seperti gula darah sewaktu, gula darah puasa, tekanan darah sistolik, tekanan darah diastolik, dan catatan. Proses ini diarahkan dengan jelas untuk memastikan pengguna memasukkan data yang relevan.

5. Insight

Opsi Insight menyajikan pengguna dengan visualisasi dari data medis yang mereka masukkan sebelumnya.



Bot menghasilkan grafik berdasarkan data gula darah dan tekanan darah dari rekam medis pengguna. Grafik ini disampaikan kepada pengguna secara bergantian, dengan judul yang jelas dan pemformatan yang baik untuk memastikan pemahaman yang optimal.

Implementasi dari sisi antarmuka pengguna ini memastikan pengalaman pengguna yang baik dengan bot Telegram. Dengan menyediakan antarmuka yang intuitif, informatif, dan responsif, bot ini membantu pengguna dengan efisien dalam mengelola profil dan rekam medis mereka.

4.4 Pengujian

Dalam pengembangan bot Telegram, penting untuk melakukan beberapa jenis pengujian, termasuk pengujian unit untuk memastikan fungsi-fungsi kode berjalan dengan benar, UAT untuk menilai kepuasan pengguna akhir, dan pengujian dengan berbagai ID pengguna untuk memastikan bot dapat menangani berbagai skenario penggunaan. Dengan melakukan pengujian yang komprehensif ini, pengembang dapat memastikan bahwa bot bekerja sesuai yang diharapkan, memenuhi kebutuhan pengguna, dan memberikan pengalaman pengguna yang positif. Setelah

mengimplementasikan berbagai pendekatan pengujian untuk fitur-fitur Telegram dalam pengembangan *chatbot*, hasil pengujian dapat dirangkum sebagai berikut:

4.4.1 Pengujian Unit

Rangkaian tes yang ditulis menggunakan kerangka kerja *pytest* untuk menguji fungsi-fungsi terkait fungsionalitas bot, khususnya yang melibatkan interaksi dengan database. Kasus pengujian ini terutama berfokus pada:

1. Menguji apakah suatu ID pengguna tertentu ada dalam database atau tidak .
2. Mengambil data profil pengguna.
3. Mengambil data rekam medis.

Berikut ini adalah kode untuk pengujian unit pada bot rekam medis diabetes :

```
import pytest
from unittest.mock import MagicMock
from bot import (
    check_id,
    get_profile_data,
    get_rekam_medis_data,
)

@pytest.fixture
def mock_database_cursor():
    # Create a mock database cursor
    mock_cursor = MagicMock()
    return mock_cursor

def test_check_id_user_exists(mock_database_cursor):
    # Test when user ID exists in the database
    user_id = 1116739768
    mock_database_cursor.fetchone.return_value = (123,) # Simulate user ID exists in the database
    result = check_id(user_id)
    assert result == True # Expecting True since user ID exists

def test_check_id_user_not_exists(mock_database_cursor):
    # Test when user ID doesn't exist in the database
    user_id = 1233523544655
    mock_database_cursor.fetchone.return_value = None # Simulate user ID doesn't exist in the database
    result = check_id(user_id)
    assert result == False # Expecting False since user ID doesn't exist

def test_get_profile_data(mock_database_cursor):
    # Test getting user profile data
    user_id = 1116739768
    get_profile_data(user_id, mock_database_cursor)
    mock_database_cursor.execute.assert_called_once_with("SELECT * FROM pasien WHERE user_id = %s",
    (user_id,))

def test_get_rekam_medis_data(mock_database_cursor):
    # Test getting rekam medis data
    user_id = 1116739768
    get_rekam_medis_data(user_id, mock_database_cursor)
    mock_database_cursor.execute.assert_called_once_with("SELECT * FROM report WHERE user_id = %s",
    (user_id,))
```

Gambar 4.4-1 Kode Pengujian

Hasil dari pengujian unit menggunakan *pytest* :

```
PS D:\magang\iset\test> pytest
===== test session starts =====
platform win32 -- Python 3.11.2, pytest-8.0.2, pluggy-1.4.0
rootdir: D:\magang\iset\test
plugins: anyio-3.6.2, asyncio-0.23.5.post1
asyncio: mode=Mode.STRICT
collected 4 items

test_bot.py .... [100%]

===== 4 passed in 5.42s =====
```

Gambar 4.4-2 Hasil Uji unit

Berikut adalah Tabel hasil pengujian unit:

Tabel 4.4-1 Hasil Pengujian Unit

No.	Kasus Uji	Status
1	Menguji keberadaan ID pengguna dalam database	Berhasil
2	Mengambil data profil pengguna	Berhasil
3	Mengambil data rekam medis	Berhasil

Pengujian unit telah berhasil mengevaluasi setiap unit fungsionalitas dalam fitur Telegram secara terpisah. Dengan menggunakan alat Pytest, pengujian ini memastikan bahwa setiap fungsi terhubung dengan fitur Telegram berjalan dengan baik dan sesuai dengan spesifikasi yang diharapkan.

4.4.2 Pengujian Penerimaan Pengguna (UAT) untuk Fitur Telegram

Pengujian melibatkan pengguna akhir atau pemangku kepentingan dalam proses pengujian untuk memastikan bahwa fitur atau fungsionalitas yang dikembangkan sesuai dengan kebutuhan dan harapan mereka.

1. Persiapkan Lingkungan Uji: Pastikan bahwa lingkungan uji mencerminkan lingkungan produksi sebanyak mungkin. Ini termasuk menginstal aplikasi Telegram pada perangkat yang akan digunakan oleh pengguna uji, dan memastikan ketersediaan koneksi internet yang stabil.

```
Feature: Testing Telegram Bot
  Scenario: Testing profile editing
    Given a user wants to edit their profile
    When the user enters their name as "John Doe"
    And the user enters their age as "30"
    And the user enters their gender as "Male"
    And the user enters their email as "john.doe@example.com"
    And the user enters their phone number as "123456789"
    And the user enters their weight as "70 kg"
    And the user enters their height as "180 cm"
    And the user enters their address as "123 Main St, City"
    Then the user profile should be updated successfully

  Scenario: Testing record medical data
    Given a user wants to record their medical data
    When the user enters their blood sugar fasting as "120 mg/dL"
    And the user enters their blood sugar random as "140 mg/dL"
    And the user enters their systolic blood pressure as "120 mmHg"
    And the user enters their diastolic blood pressure as "80 mmHg"
    And the user enters notes as "No specific notes"
    Then the medical record should be saved successfully
```

Gambar 4.4-3 Skenerio Behave

2. Jalankan Skenario Uji: Persiapkan serangkaian skenario uji yang mencakup berbagai kasus penggunaan yang mungkin terjadi. Skenario ini harus mencakup fungsionalitas utama dari fitur Telegram yang akan di

```
from behave import given, when, then
from bot import update_profile_data, save_rekam_medis_data

@given('a user wants to edit their profile')
def step_user_wants_to_edit_profile(context):
    # Setup for the test, if any
    pass

@when('the user enters their name as "{name}"')
def step_user_enters_name(context, name):
    context.profile_data['name'] = name

# Implement other steps similarly...

@then('the user profile should be updated successfully')
def step_profile_should_be_updated_successfully(context):
    success = update_profile_data(context.profile_data)
    assert success

@when('the user enters their blood sugar fasting as "{fasting}"')
def step_user_enters_blood_sugar_fasting(context, fasting):
    context.medical_data['fasting'] = fasting

# Implement other steps similarly...

@then('the medical record should be saved successfully')
def step_medical_record_should_be_saved_successfully(context):
    success = save_rekam_medis_data(context.medical_data)
    assert success
```

Gambar 4.4-4 Kode Uji Behave Hasil Pengujian


```

behave
program system anda berhasil dijalankan
program system anda berhasil dijalankan
Feature: Testing Telegram Bot # features/user_profile.feature:1

Scenario: Testing profile editing # features/user_profile.feature:2
  Given a user wants to edit their profile # steps/user_profile_steps.py:4
  When the user enters their name as "John Doe" # steps/user_profile_steps.py:9
  And the user enters their age as "30" # steps/user_profile_steps.py:13
  And the user enters their gender as "Male" # steps/user_profile_steps.py:17
  And the user enters their email as "john.doe@example.com" # steps/user_profile_steps.py:21
  And the user enters their phone number as "123456789" # steps/user_profile_steps.py:25
  And the user enters their weight as "70 kg" # steps/user_profile_steps.py:29
  And the user enters their height as "180 cm" # steps/user_profile_steps.py:33
  And the user enters their address as "123 Main St, City" # steps/user_profile_steps.py:37

Scenario: Testing record medical data # features/user_profile.feature:13
  Given a user wants to record their medical data # steps/user_profile_steps.py:46
  When the user enters their blood sugar fasting as "120 mg/dL" # steps/user_profile_steps.py:51
  And the user enters their blood sugar random as "140 mg/dL" # steps/user_profile_steps.py:55
  And the user enters their systolic blood pressure as "120 mmHg" # steps/user_profile_steps.py:59
  And the user enters their diastolic blood pressure as "80 mmHg" # steps/user_profile_steps.py:63
  And the user enters notes as "No specific notes" # steps/user_profile_steps.py:67

1 feature passed, 0 failed, 0 skipped
2 scenarios passed, 0 failed, 0 skipped
15 steps passed, 0 failed, 0 skipped, 0 undefined

```

Gambar 4.4-5 Hasil Uji UAT Behave

3. Tabel hasil pengujian

Tabel 4.4-2 Hasil Pengujian UAT

No.	Fitur	Status
1	Profile Editing	Berhasil
2	Record Medical Data	Berhasil

Pengujian penerimaan pengguna melibatkan pengguna aktual atau representatif dalam skenario pengujian realistis. Dengan menggunakan alat behave dan , fitur Telegram telah divalidasi sesuai dengan harapan pengguna dan kasus penggunaan yang umum atau penting dalam Telegram.

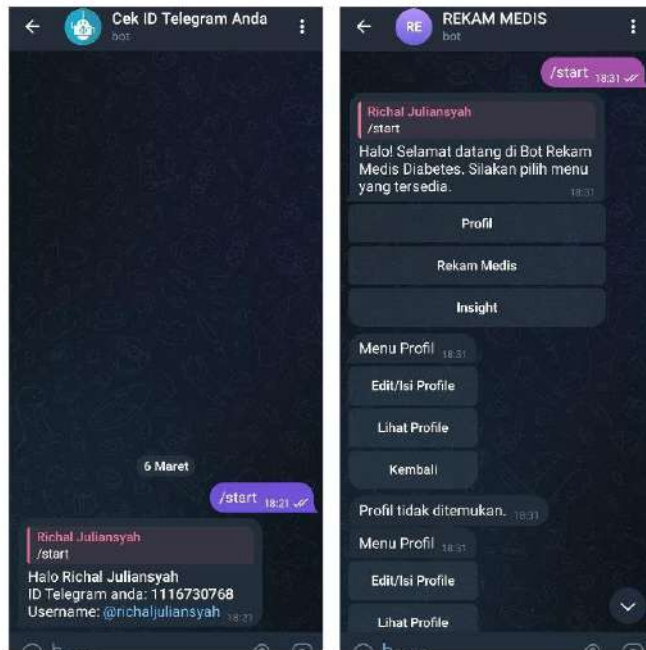
4.4.3 Pengujian dengan Berbagai ID Pengguna

Pengujian dengan berbagai ID pengguna menjadi krusial untuk memastikan bahwa bot Telegram dapat beroperasi dengan baik dan memberikan respons yang tepat kepada berbagai jenis pengguna. Dalam subbab ini, kita akan membahas bagaimana melakukan pengujian dengan ID pengguna yang berbeda

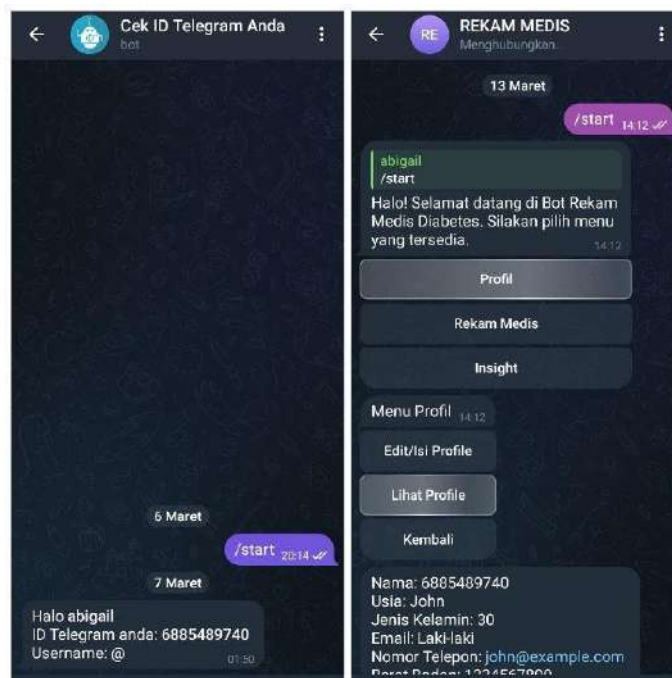
1. Pengguna terdaftar

Pertama, kita akan menguji bot dengan pengguna baru yang belum terdaftar dalam sistem. Pengguna baru diharapkan untuk menerima pesan selamat

datang dan diarahkan untuk melakukan registrasi melalui web. Pengujian ini memastikan bahwa bot memberikan respons yang tepat kepada pengguna yang belum terdaftar:



Gambar 4.4-6 user 1



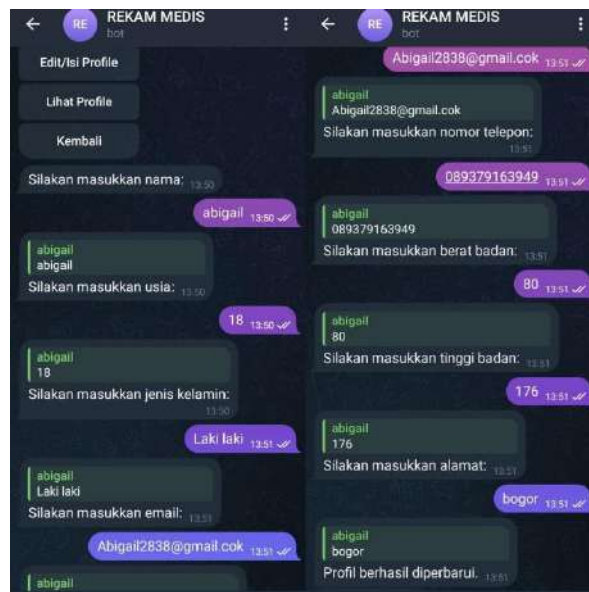
Gambar 4.4-7 User 2

2. Pengguna Mengisi Profil

Kemudian, kita akan menguji proses pengisian profil dengan pengguna yang telah terdaftar. Pengguna akan diminta untuk memasukkan informasi profil mereka seperti nama, usia, dan alamat. Pengujian ini memastikan bahwa bot dapat menerima input dari pengguna dengan benar dan menyimpannya ke dalam database :



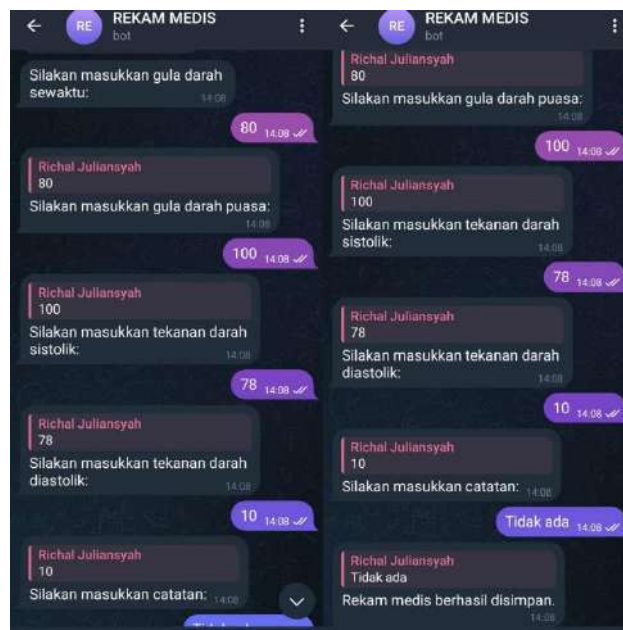
Gambar 4.4-8 Pengisian Profil User 1



Gambar 4.4-9 Pengisian Profil User 2

3. Pengguna Menambahkan Rekam Medis

Pengujian selanjutnya melibatkan pengguna yang ingin menambahkan rekam medis baru. Bot harus memandu pengguna melalui proses pengisian formulir dengan baik, memvalidasi input, dan menyimpan data rekam medis dengan benar. Pengguna juga harus menerima konfirmasi setelah rekam medis berhasil ditambahkan :



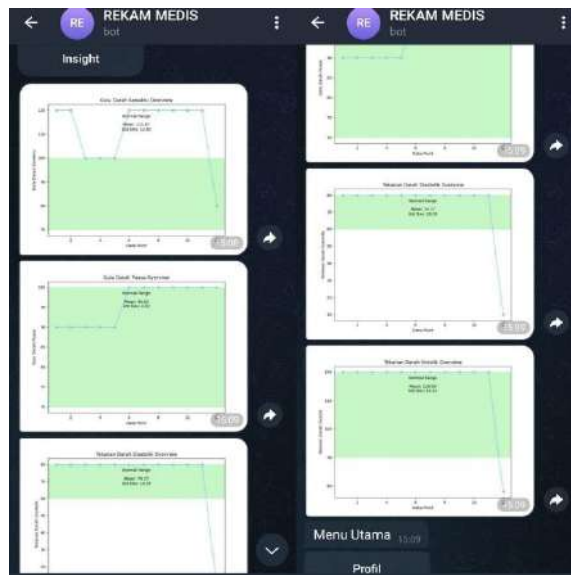
Gambar 4.4-10 Tambah Rekam medis user 1



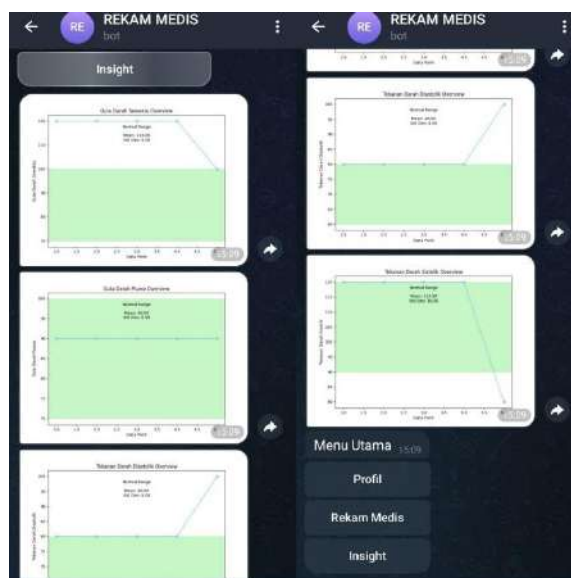
Gambar 4.4-11 Tambah Rekam medis user 2

4. Pengguna Melihat Insight

Terakhir, kita akan menguji pengguna yang ingin melihat insight dari data rekam medis mereka. Bot harus menghasilkan grafik yang akurat berdasarkan data yang tersedia dan menyampaikannya kepada pengguna dengan jelas. Pengguna harus dapat memahami informasi yang disajikan dalam grafik dengan mudah.



Gambar 4.4-12 Insight user 1



Gambar 4.4-13 Insight User 2

Setelah melakukan serangkaian pengujian dengan berbagai ID pengguna, kami dengan senang hati mengumumkan bahwa seluruh pengujian telah berhasil. Bot

Telegram telah menunjukkan kinerja yang baik dalam merespons berbagai jenis pengguna, mulai dari pengguna baru hingga pengguna yang telah terdaftar. Semua fitur, mulai dari registrasi hingga pengisian profil, penambahan rekam medis, dan pengambilan insight, telah berjalan dengan lancar dan sesuai dengan harapan. berikut adalah tabel hasil pengujian untuk masing-masing fitur:

1. Registrasi

Tabel 4.4-3 Pengujian Register

No.	ID Pengguna	Status Pengujian
1	User 1	Berhasil
2	User 2	Berhasil
3	User 3	Berhasil
4	User 4	Berhasil
5	User 5	Berhasil

2. Pengisian Profil

Tabel 4.4-4 Pengisian Profil

No.	ID Pengguna	Status Pengujian
1	User 1	Berhasil
2	User 2	Berhasil
3	User 3	Berhasil
4	User 4	Berhasil
5	User 5	Berhasil

3. Penambahan Rekam Medis

Tabel 4.4-5 Tambah Rekam medis

No.	ID Pengguna	Status Pengujian
1	User 1	Berhasil
2	User 2	Berhasil
3	User 3	Berhasil
4	User 4	Berhasil

5	User 5	Berhasil
---	--------	----------

4. Melihat Insight

Tabel 4.4-6 Melihat Insight

No.	ID Pengguna	Status Pengujian
1	User 1	Berhasil
2	User 2	Berhasil
3	User 3	Berhasil
4	User 4	Berhasil
5	User 5	Berhasil

Berdasarkan hasil pengujian yang luas dan menyeluruh dengan berbagai ID pengguna serta fitur yang beragam, Dapat ditarik bahwa bot Telegram telah terbukti efektif dalam operasinya. Bot ini telah menunjukkan respons yang tepat terhadap pengguna baru maupun yang telah terdaftar, menjalankan setiap fitur dengan lancar tanpa masalah yang signifikan, dan memberikan interaksi yang mudah dipahami melalui pesan yang jelas dan grafik yang informatif. Dengan kinerja yang konsisten dan kemampuan untuk berinteraksi dengan baik dengan berbagai jenis pengguna, bot Telegram ini dapat diandalkan sebagai alat yang efektif dalam memberikan layanan yang dibutuhkan, seperti dalam konteks pelayanan kesehatan atau manajemen informasi medis.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dalam implementasi *chatbot* berbasis Telegram untuk mengelola rekam medis penderita diabetes, proses analisis dan perancangan sistem yang matang menjadi dasar yang penting. Tahap ini melibatkan identifikasi kebutuhan perangkat keras dan perangkat lunak, serta merancang arsitektur sistem yang efisien.

1. Dari hasil perancangan dan implementasi chatbot berbasis Telegram untuk mengelola rekam medis penderita diabetes, terbukti bahwa chatbot berfungsi dengan baik. Chatbot telah dirancang dengan beberapa komponen utama, termasuk antarmuka pengguna menggunakan platform Telegram untuk interaksi, modul pengolahan bahasa alami (Natural Language Processing, NLP) untuk memahami dan memproses bahasa alami pengguna, serta penggunaan database untuk menyimpan informasi profil dan rekam medis dengan efisiensi. Selain itu, sistem pengingat dan notifikasi diterapkan untuk mengingatkan pengguna tentang jadwal pengukuran kadar gula darah dan aktivitas kesehatan lainnya. Fitur-fitur yang diimplementasikan dalam chatbot termasuk pengisian profil, penambahan rekam medis, dan opsi insight kesehatan yang memberikan pengguna kemudahan dalam mencatat dan mengelola informasi kesehatan mereka secara digital.
2. Dari segi efektivitas chatbot ini terbukti efektif dalam operasinya. Respons yang tepat, antarmuka pengguna yang intuitif, dan visualisasi data yang informatif menjadikan chatbot ini alat yang dapat diandalkan dalam manajemen rekam medis penderita diabetes. Pengujian unit, pengujian penerimaan pengguna (UAT) dan Pengujian dengan Berbagai ID Pengguna menunjukkan bahwa chatbot mampu menjalankan fitur-fiturnya dengan baik dan positif.

5.2 Saran

1. Pengembangan Fitur: Perlu dirancang rencana pengembangan untuk menambahkan fitur-fitur baru yang dapat meningkatkan fungsionalitas bot, seperti integrasi notifikasi, rekomendasi berdasarkan data kesehatan, atau dukungan untuk *multiple users*.
2. Penelitian Lebih Lanjut: Melakukan penelitian lebih mendalam mengenai efektivitas penggunaan bot Telegram dalam manajemen kesehatan, termasuk analisis dampaknya terhadap kesadaran kesehatan dan kualitas hidup penderita diabetes.
3. Personalisasi Lebih Lanjut: Evaluasi dan implementasikan lebih banyak opsi personalisasi pada profil pengguna, memungkinkan pengguna untuk menyesuaikan antarmuka sesuai preferensi mereka.
4. Peningkatan Keamanan: Terus melakukan evaluasi dan perbaikan pada aspek keamanan sistem, mempertimbangkan teknologi terbaru untuk melindungi data pengguna.
5. Pemantauan dan Pemeliharaan: Implementasikan mekanisme pemantauan yang lebih canggih dan otomatis untuk memudahkan pemeliharaan sistem dan mengurangi waktu downtime.

Saran-saran ini diharapkan dapat memberikan kontribusi positif bagi penelitian selanjutnya dalam meningkatkan kualitas dan keberlanjutan sistem yang telah dikembangkan tanpa mengabaikan prinsip-prinsip etika penelitian dan integritas akademis.

DAFTAR REFERENSI

- [1] p. j. tengah, “edukasi pencegahan diabetes pada lansia di rw 13 jebres,” *pengabd. komunitas*, vol. 02, no. 01, pp. 64–70, 2023.
- [2] k. p. dharmawan, “rancangan dan implementasi telegram *chatbot* dengan dialogflow (studi kasus: pariwisata bali),” *researchgate.net*, no. may, 2021.
- [3] h. priambodo and a. muhajirin, “perancangan *chatbot* pendaftaran siswa dengan telegram bot design a *chatbot* for student registration using telegram bot,” *j. inform. inf. secur.*, vol. 3, no. 1, pp. 73–88, 2022, doi: 10.31599/jiforty.v3i1.1332.
- [4] a. a. akhsan and f. faizah, “analisis dan perancangan interaksi *chatbot* reminder dengan user-centered design,” *j. sist. inf.*, vol. 13, no. 2, p. 78, 2017, doi: 10.21609/jsi.v13i2.555.
- [5] o. n. putra, h. hardiyono, and e. d. p. pitaloka, “evaluasi konversi sputum dan faktor korelasinya pada penderita tuberkulosis paru kategori i dengan diabetes melitus,” *j. farm. dan ilmu kefarmasian indones.*, vol. 8, no. 1, p. 38, 2021, doi: 10.20473/jfiki.v8i12021.38-45.
- [6] n. k. n. antari, *diabetes melitus tipe 2*, vol. 4, no. 13. 2017. [online]. available: https://simdos.unud.ac.id/uploads/file_penelitian_1_dir/653f627b3ce1272d209353541c305cee.pdf
- [7] w. maryati, a. o. wannay, and d. p. suci, “hubungan kelengkapan informasi medis dan keakuratan kode diagnosis diabetes mellitus,” *j. rekam medis dan inf. kesehat.*, vol. 1, no. 2, p. 96, 2018, doi: 10.31983/jrmik.v1i2.3852.
- [8] m. f. salim and s. sugeng, “analisis rekam medis penderita diabetes mellitus melalui implementasi teknik data mining di rsup dr. sardjito yogyakarta,” *j. kesehat. vokasional*, vol. 2, no. 2, p. 167, 2018, doi: 10.22146/jkesvo.30331.
- [9] s. sugiono, “pemanfaatan *chatbot* pada masa pandemi covid-19 : kajian fenomena society 5 . 0 *chatbot* utilization during the covid-19 pandemic:revisiting the concept of society 5 . 0,” *j. pikom (penelitian komun. dan pembangunan)*, vol. 22, no. 2, pp. 133–148, 2021.
- [10] d. ismawati and i. prasetyo, “the development of telegram bot through short story,” vol. 456, no. bicomst, pp. 209–212, 2020, doi: 10.2991/assehr.k.201021.049.
- [11] d. n. bestari and a. wibowo, “an iot-based real-time weather monitoring system using telegram bot and thingsboard *platform*,” *int. j. interact. mob. technol.*, vol. 17, no. 6, pp. 4–19, 2023, doi: 10.3991/ijim.v17i06.34129.
- [12] a. h. hasan and m. f. hilmi, “input process output (ipo) ai *chatbot* as personal learning assistant for programming coursework,” *malaysian j. distance educ.*, vol. 22, no. 1, pp. 16–26, 2022, doi: 10.36777/mjde2022.22.1.2.
- [13] s. m. arif and h. purwoko, “pengayaan kemampuan olah basis data *mysql* pada ukm lcc politeknik lp3i jakarta,” *j. pkm pengabd. kpd. masy.*, vol. 4, no. 4, p. 394, 2021, doi: 10.30998/jurnalpkm.v4i4.6403.
- [14] a. niamilah, a. a. alfin, and i. kurniasari, “siklus hidup pengembangan sistem basis data pada sistem informasi buku tamu di badan pusat statistik kabupaten kediri menggunakan *mysql*,” *j. nas. komputasi dan teknol. inf.*, vol. 6, no. 1, pp. 115–121, 2023, doi: 10.32672/jnkti.v6i1.5830.

- [15] m. a. s. panatagama, “migrasi basis data non-relasional mongoddb ke *mysql* menggunakan pentaho data integration,” *automata*, no. query date: 2023- 01-09 02:10:29, 2020.
- [16] e. tri utami and j. hamongan siregar, “perancangan alur kerja pemasaran pada media komunikasi digital menggunakan basis data *mysql*,” *prosiding*, vol. 3, pp. 225–233, 2023, doi: 10.59134/prosidng.v3i.363.
- [17] o. g. khoirunnisa and d. djuniadi, “implementasi algoritma aes untuk keamanan data rekam medis,” *petir*, vol. 15, no. 1, pp. 21–27, 2021, doi: 10.33322/petir.v15i1.1333.
- [18] d. kastowo and s. raharjo, “analisis perbandingan penyimpanan data rekam medis elektronik berstandar fhir pada sistem basis data: bigchaindb, *mysql* dan mongoddb,” *jnanaloka*, pp. 37–4x, 2023, doi: 10.36802/jnanaloka.2023.v4-no01-37-4x.

LAMPIRAN

1. Kode Pemograman *Chatbot* Telegram

```
from io import BytesIO
import os
import mysql.connector
import logging
import matplotlib.pyplot as plt
from aiogram import Bot, Dispatcher, executor, types
from aiogram.contrib.fsm_storage.memory import MemoryStorage
from aiogram.dispatcher import FSMContext
from aiogram.dispatcher.filters import Command
from aiogram.dispatcher.filters.state import State, StatesGroup

# Konfigurasi logging
logging.basicConfig(level=logging.INFO)

# Koneksi ke MySQL
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="mydatabase3"
)

# Inisialisasi bots
bot = Bot(token="6760519530:AAHJPOBpwKzLidnVnehnrUSM3QFNAYzik4")
dp = Dispatcher(bot, storage=MemoryStorage())

# State untuk pengisian/edit profil
class ProfileStates(StatesGroup):
    NAMA = State()
    USIA = State()
    JENIS_KELAMIN = State()
    EMAIL = State()
    NOMOR_TELEPON = State()
    BERAT_BADAN = State()
    TINGGI_BADAN = State()
    ALAMAT = State()

# State untuk pengisian rekam medis
class RekamMedisStates(StatesGroup):
    GULA_DARAH_SEWAKTU = State()
    GULA_DARAH_PUASA = State()
    TEKANAN_DARAH_SISTOLIK = State()
```

```

TEKANAN_DARAH_DIASTOLIK = State()
DELETE_ID = State()

@dp.message_handler(Command("start"))
async def start(message: types.Message):
    user_id = message.from_user.id

    # Periksa apakah pengguna telah terdaftar
    id = check_id(user_id)

    if id:
        await message.reply("Halo! Selamat datang di Bot Rekam Medis Diabetes. Silakan pilih
menu yang tersedia.",
                             reply_markup=menu_keyboard())
    else:
        # Jika pengguna belum terdaftar, catat ID pengguna ke dalam log atau tempat
penyimpanan yang sesuai
        log_unregistered_user(user_id)

        await message.reply("Anda belum terdaftar. Silakan melakukan registrasi di web
terlebih dahulu.")

def log_unregistered_user(user_id):
    print(f"ID Pengguna Tidak Terdaftar: {user_id}")

def check_id(user_id):
    mycursor = mydb.cursor()
    sql = "SELECT id FROM users WHERE user_id = %s"
    val = (user_id,)
    mycursor.execute(sql, val)
    result = mycursor.fetchone()
    mycursor.close()

    # Periksa apakah ada baris yang sesuai dengan user_id
    if result:
        return True
    else:
        return False

# Corrected register_user function
def register_user(user_id, db_cursor):
    sql = "INSERT INTO users (user_id) VALUES (%s)"
    val = (user_id,)
    db_cursor.execute(sql, val)

```

```

mydb.commit()

# Keyboard untuk menu
def menu_keyboard():
    keyboard = types.InlineKeyboardMarkup()
    keyboard.add(types.InlineKeyboardButton(text="Profil", callback_data="profile_menu"))
    keyboard.add(types.InlineKeyboardButton(text="Rekam Medis",
callback_data="rekam_medis_menu"))
    keyboard.add(types.InlineKeyboardButton(text="Insight", callback_data="insight"))
    return keyboard

# Command /profile
@dp.callback_query_handler(text='profile_menu')
async def profile_menu(callback_query: types.CallbackQuery):
    await bot.send_message(callback_query.from_user.id, "Menu Profil",
reply_markup=profile_keyboard())

# Keyboard untuk menu profil
def profile_keyboard():
    keyboard = types.InlineKeyboardMarkup()
    keyboard.add(types.InlineKeyboardButton(text="Edit/Isi Profile",
callback_data="edit_profile"))
    keyboard.add(types.InlineKeyboardButton(text="Lihat Profile",
callback_data="view_profile"))
    keyboard.add(types.InlineKeyboardButton(text="Kembali", callback_data="back_to_menu"))
    return keyboard

# Command /backtomenu
@dp.callback_query_handler(text='back_to_menu')
async def back_to_menu(callback_query: types.CallbackQuery):
    await bot.send_message(callback_query.from_user.id, "Kembali ke menu utama",
reply_markup=menu_keyboard())

# Command /editprofile
@dp.callback_query_handler(text='edit_profile')
async def edit_profile(callback_query: types.CallbackQuery):
    await bot.send_message(callback_query.from_user.id, "Silakan masukkan nama:")
    await ProfileStates.NAMA.set()

# Menangani input nama saat edit profil
@dp.message_handler(state=ProfileStates.NAMA)
async def process_edit_nama(message: types.Message, state: FSMContext):
    async with state.proxy() as data:

```

```

        data['nama'] = message.text

    await message.reply("Silakan masukkan usia:")
    await ProfileStates.USIA.set()

# Menangani input usia saat edit profil
@dp.message_handler(state=ProfileStates.USIA)
async def process_edit_usia(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['usia'] = message.text

    await message.reply("Silakan masukkan jenis kelamin:")
    await ProfileStates.JENIS_KELAMIN.set()

# Menangani input jenis kelamin saat edit profil
@dp.message_handler(state=ProfileStates.JENIS_KELAMIN)
async def process_edit_jk(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['jenis_kelamin'] = message.text

    await message.reply("Silakan masukkan email:")
    await ProfileStates.EMAIL.set()

# Menangani input email saat edit profil
@dp.message_handler(state=ProfileStates.EMAIL)
async def process_edit_email(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['email'] = message.text

    await message.reply("Silakan masukkan nomor telepon:")
    await ProfileStates.NOMOR_TELEPON.set()

# Menangani input nomor telepon saat edit profil
@dp.message_handler(state=ProfileStates.NOMOR_TELEPON)
async def process_edit_nomor_telepon(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['nomor_telepon'] = message.text

    await message.reply("Silakan masukkan berat badan:")
    await ProfileStates.BERAT_BADAN.set()

# Menangani input berat badan saat edit profil
@dp.message_handler(state=ProfileStates.BERAT_BADAN)
async def process_edit_berat_badan(message: types.Message, state: FSMContext):
    async with state.proxy() as data:

```

```

        data['berat_badan'] = message.text

    await message.reply("Silakan masukkan tinggi badan:")
    await ProfileStates.TINGGI_BADAN.set()

# Menangani input tinggi badan saat edit profil
@dp.message_handler(state=ProfileStates.TINGGI_BADAN)
async def process_edit_tinggi_badan(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['tinggi_badan'] = message.text

    await message.reply("Silakan masukkan alamat:")
    await ProfileStates.ALAMAT.set()

# Menangani input alamat saat edit profil
@dp.message_handler(state=ProfileStates.ALAMAT)
async def process_edit_alamat(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['alamat'] = message.text

    # Update data profil ke database
    await update_profile_data(message.from_user.id, data)

    await message.reply("Profil berhasil diperbarui.")

# Reset state
await state.reset_state()

# Kembali ke menu profil
await bot.send_message(message.chat.id, "Menu Profil", reply_markup=profile_keyboard())

# Fungsi untuk menyimpan atau memperbarui data profil ke database
def update_profile_data(user_id, data):
    mycursor = mydb.cursor()
    sql = "UPDATE penderita SET nama=%s, usia=%s, jenis_kelamin=%s, email=%s,
    nomor_telepon=%s, berat_badan=%s, tinggi_badan=%s, alamat=%s WHERE user_id=%s"
    val = (data['nama'], data['usia'], data['jenis_kelamin'], data['email'],
    data['nomor_telepon'], data['berat_badan'], data['tinggi_badan'], data['alamat'], user_id)
    mycursor.execute(sql, val)
    mydb.commit()

# Function to handle updating profile
async def handle_update_profile(message: types.Message, data):
    update_profile_data(message.from_user.id, data)
    await message.reply("Profil berhasil diperbarui.")

```



```

# Send inline keyboard menu
await message.answer("Menu Profil", reply_markup=profile_keyboard())

# Command /viewprofile
@dp.callback_query_handler(text='view_profile')
async def view_profile(callback_query: types.CallbackQuery):
    profile_data = get_profile_data(callback_query.from_user.id)
    if profile_data:
        profile_text = f>Nama: {profile_data['nama']}\n"
        profile_text += f"Usia: {profile_data['usia']}\n"
        profile_text += f"Jenis Kelamin: {profile_data['jenis_kelamin']}\n"
        profile_text += f>Email: {profile_data['email']}\n"
        profile_text += f"Nomor Telepon: {profile_data['nomor_telepon']}\n"
        profile_text += f"Berat Badan: {profile_data['berat_badan']}\n"
        profile_text += f"Tinggi Badan: {profile_data['tinggi_badan']}\n"
        profile_text += f"Alamat: {profile_data['alamat']}"

        await bot.send_message(callback_query.from_user.id, profile_text)
    else:
        await bot.send_message(callback_query.from_user.id, "Profil tidak ditemukan.")

    # Kembali ke menu profil
    await bot.send_message(callback_query.from_user.id, "Menu Profil",
reply_markup=profile_keyboard())

# Fungsi untuk mendapatkan data profil dari database
def get_profile_data(user_id, db_cursor):
    sql = "SELECT * FROM penderita WHERE user_id = %s"
    val = (user_id,)
    db_cursor.execute(sql, val)
    result = db_cursor.fetchone()
    if result:
        profile_data = {
            'nama': result[1],
            'usia': result[2],
            'jenis_kelamin': result[3],
            'email': result[4],
            'nomor_telepon': result[5],
            'berat_badan': result[6],
            'tinggi_badan': result[7],
            'alamat': result[8]
        }
        return profile_data
    else:
        return None

```

```

# States untuk rekam medis baru
class RekamMedisStates(StatesGroup):
    GULA_DARAH_SEWAKTU = State()
    GULA_DARAH_PUASA = State()
    TEKANAN_DARAH_SISTOLIK = State()
    TEKANAN_DARAH_DIASTOLIK = State()
    CATATAN = State()

# Command /rekammedis
@dp.callback_query_handler(text='rekam_medis_menu')
async def rekam_medis_menu(callback_query: types.CallbackQuery):
    await bot.send_message(callback_query.from_user.id, "Menu Rekam Medis",
reply_markup=rekam_medis_keyboard())

# Keyboard untuk menu rekam medis
def rekam_medis_keyboard():
    keyboard = types.InlineKeyboardMarkup()
    keyboard.add(types.InlineKeyboardButton(text="Rekam Medis Baru",
callback_data="new_rekam_medis"))
    keyboard.add(types.InlineKeyboardButton(text="Lihat Rekam Medis",
callback_data="view_rekam_medis"))
    keyboard.add(types.InlineKeyboardButton(text="Hapus Rekam Medis",
callback_data="delete_rekam_medis"))
    keyboard.add(types.InlineKeyboardButton(text="Kembali", callback_data="back_to_menu"))
    return keyboard

# Command /newrekammedis
@dp.callback_query_handler(text='new_rekam_medis')
async def new_rekam_medis(callback_query: types.CallbackQuery):
    await bot.send_message(callback_query.from_user.id, "Silakan masukkan gula darah
sewaktu:")
    await RekamMedisStates.GULA_DARAH_SEWAKTU.set()

# Menangani input gula darah sewaktu saat rekam medis baru
@dp.message_handler(state=RekamMedisStates.GULA_DARAH_SEWAKTU)
async def process_new_gula_darah_sewaktu(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['gula_darah_sewaktu'] = message.text

    await message.reply("Silakan masukkan gula darah puasa:")

```

```

    await RekamMedisStates.GULA_DARAH_PUASA.set()

# Menangani input gula darah puasa saat rekam medis baru
@dp.message_handler(state=RekamMedisStates.GULA_DARAH_PUASA)
async def process_new_gula_darah_puasa(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['gula_darah_puasa'] = message.text

    await message.reply("Silakan masukkan tekanan darah sistolik:")
    await RekamMedisStates.TEKANAN_DARAH_SISTOLIK.set()

# Menangani input tekanan darah sistolik saat rekam medis baru
@dp.message_handler(state=RekamMedisStates.TEKANAN_DARAH_SISTOLIK)
async def process_new_tekanan_darah_sistolik(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['tekanan_darah_sistolik'] = message.text

    await message.reply("Silakan masukkan tekanan darah diastolik:")
    await RekamMedisStates.TEKANAN_DARAH_DIASTOLIK.set()

# Menangani input tekanan darah diastolik saat rekam medis baru
@dp.message_handler(state=RekamMedisStates.TEKANAN_DARAH_DIASTOLIK)
async def process_new_tekanan_darah_diastolik(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['tekanan_darah_diastolik'] = message.text

    await message.reply("Silakan masukkan catatan:")
    await RekamMedisStates.CATATAN.set()

# Menangani input catatan saat rekam medis baru
@dp.message_handler(state=RekamMedisStates.CATATAN)
async def process_new_catatan(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['catatan'] = message.text

    # Simpan data rekam medis ke database
    save_rekam_medis_data(message.from_user.id, data)

    await message.reply("Rekam medis berhasil disimpan.")

# Reset state
await state.reset_state()

```

```

# Kembali ke menu rekam medis
await bot.send_message(message.chat.id, "Menu Rekam Medis",
reply_markup=rekam_medis_keyboard())

# Keyboard untuk menu rekam medis
def rekam_medis_keyboard():
    keyboard = types.InlineKeyboardMarkup()
    keyboard.add(types.InlineKeyboardButton(text="Rekam Medis Baru",
callback_data="new_rekam_medis"))
    keyboard.add(types.InlineKeyboardButton(text="Lihat Rekam Medis",
callback_data="view_rekam_medis"))
    keyboard.add(types.InlineKeyboardButton(text="Hapus Rekam Medis",
callback_data="delete_rekam_medis"))
    keyboard.add(types.InlineKeyboardButton(text="Kembali", callback_data="back_to_menu"))
    return keyboard

# Fungsi untuk menyimpan data rekam medis ke database
def save_rekam_medis_data(user_id, data):
    mycursor = mydb.cursor()
    sql = "INSERT INTO report (user_id, gula_darah_sewaktu, gula_darah_puasa,
tekanan_darah_sistolik, tekanan_darah_diastolik, catatan) VALUES (%s, %s, %s, %s, %s, %s)"
    val = (user_id, data['gula_darah_sewaktu'], data['gula_darah_puasa'],
data['tekanan_darah_sistolik'], data['tekanan_darah_diastolik'], data['catatan'])
    mycursor.execute(sql, val)
    mydb.commit()

# Command /viewrekammedis
@dp.callback_query_handler(text='view_rekam_medis')
async def view_rekam_medis(callback_query: types.CallbackQuery):
    rekam_medis_data = get_rekam_medis_data(callback_query.from_user.id)
    if rekam_medis_data:
        rekam_medis_text = "Rekam Medis:\n\n"
        for data in rekam_medis_data:
            rekam_medis_text += f"Minggu ke -:\n"
            rekam_medis_text += f"Gula Darah Sewaktu: {data[2]}\n"
            rekam_medis_text += f"Gula Darah Puasa: {data[3]}\n"
            rekam_medis_text += f"Tekanan Darah Sistolik: {data[4]}\n"
            rekam_medis_text += f"Tekanan Darah Diastolik: {data[5]}\n"
            rekam_medis_text += f"Catatan: {data[6]}\n\n"

        await bot.send_message(callback_query.from_user.id, rekam_medis_text)
    else:
        await bot.send_message(callback_query.from_user.id, "Tidak ada data rekam medis.")

```

```

# Fungsi untuk mengambil data rekam medis dari database
def get_rekam_medis_data(user_id, db_cursor):
    sql = "SELECT * FROM report WHERE user_id = %s"
    val = (user_id,)
    db_cursor.execute(sql, val)
    result = db_cursor.fetchall()
    return result

# Fungsi untuk menghapus data rekam medis dari database
def delete_rekam_medis_data(user_id, delete_id):
    mycursor = mydb.cursor()
    sql = "DELETE FROM report WHERE user_id = %s AND id = %s"
    val = (user_id, delete_id)
    mycursor.execute(sql, val)
    mydb.commit()
    return mycursor.rowcount > 0

# Command /insight
@dp.callback_query_handler(text='insight')
async def insight(callback_query: types.CallbackQuery):
    user_id = callback_query.from_user.id

    mycursor = mydb.cursor()
    sql = "SELECT gula_darah_sewaktu, gula_darah_puasa, tekanan_darah_diastolik,
tekanan_darah_sistolik FROM report WHERE user_id = %s"
    val = (user_id,)
    mycursor.execute(sql, val)
    results = mycursor.fetchall()
    mycursor.close()

    if results:
        # Memproses data dan membuat grafik
        create_chart('Gula Darah Sewaktu', results, 140, 'gula_darah_sewaktu_chart.png')
        create_chart('Gula Darah Puasa', results, 140, 'gula_darah_puasa_chart.png')
        create_chart('Tekanan Darah Diastolik', results, 100,
'tekanan_darah_diastolik_chart.png')
        create_chart('Tekanan Darah Sistolik', results, 140,
'tekanan_darah_sistolik_chart.png')

        # Mengirim grafik ke pengguna
        with open('gula_darah_sewaktu_chart.png', 'rb') as photo:
            await bot.send_photo(callback_query.from_user.id, photo)

```

```

with open('gula_darah_puasa_chart.png', 'rb') as photo:
    await bot.send_photo(callback_query.from_user.id, photo)
with open('tekanan_darah_diastolik_chart.png', 'rb') as photo:
    await bot.send_photo(callback_query.from_user.id, photo)
with open('tekanan_darah_sistolik_chart.png', 'rb') as photo:
    await bot.send_photo(callback_query.from_user.id, photo)

# Menghapus file gambar setelah dikirim
os.remove('gula_darah_sewaktu_chart.png')
os.remove('gula_darah_puasa_chart.png')
os.remove('tekanan_darah_diastolik_chart.png')
os.remove('tekanan_darah_sistolik_chart.png')

await bot.send_message(callback_query.from_user.id, "Menu Utama",
reply_markup=menu_keyboard())
else:
    await bot.send_message(callback_query.from_user.id, "Data rekam medis tidak
ditemukan!")

# Fungsi untuk membuat grafik
def create_chart(parameter, data, normal_limit, filename):
    data_column_index = 0 # Change to the appropriate column index for the parameter
    values = [float(row[data_column_index]) for row in data]
    labels = [parameter, 'Others'] # Ensure labels list has the correct length

    # Check if there's only one value, then adjust labels accordingly
    if len(values) == 1:
        labels = [parameter]

    percentages = [(value / sum(values)) * 100 for value in values]

    # Create the pie chart
    fig, ax = plt.subplots()
    wedges, texts, autotexts = ax.pie(
        percentages,
        labels=labels,
        autopct='%1.1f%%',
        startangle=90,
        colors=['skyblue', 'lightgrey'],
        wedgeprops=dict(width=0.4),
    )

    # Add text to each sector
    for autotext in autotexts:
        autotext.set_color('white')

```

```

        autotext.set_size(8) # Set font size to fit

# Add normal limit line
ax.axhline(y=normal_limit, color='r', linestyle='--', label=f'Normal Limit for
{parameter}')

ax.set_title(f'{parameter} Overview')
ax.axis('equal')
ax.legend()

# Save the chart to a file
plt.savefig(filename)
plt.close()

# Command /backtomenu
@dp.callback_query_handler(text='back_to_menu')
async def back_to_menu(callback_query: types.CallbackQuery):
    await bot.send_message(callback_query.from_user.id, "Kembali ke menu utama",
reply_markup=menu_keyboard)

# Memulai bot
print("program system anda berhasil dijalankan")
if __name__ == '__main__':
    executor.start_polling(dp, skip_updates=True)

```