

Implementasi *Chatbot* Sebagai Asisten *Virtual* Untuk Membantu Mahasiswa Dalam Mendapatkan Informasi Akademik

Abdurahman Amin¹⁾, Tommy Bustomi²⁾, dan Kusno Harianto³⁾

^{1,2,3}Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer Widya Cipta Dharma
^{1,2,3}Samarinda, 75123

E-mail: 2043032@wicida.ac.id¹⁾, tbustomi@gmail.ac.id²⁾, kusno@wicida.ac.id³⁾

ABSTRAK

Penelitian ini bertujuan untuk mengembangkan chatbot sebagai asisten virtual guna membantu mahasiswa STMIK Widya Cipta Dharma dalam memperoleh informasi akademik. Penelitian dilakukan di Bagian Administrasi Akademik dan Kemahasiswaan dengan metode pengumpulan data melalui wawancara serta studi pustaka yang berkaitan dengan chatbot.

Metode pengembangan yang digunakan adalah linear sequential, dengan perangkat lunak pendukung Visual Studio Code dan Android Studio. Hasil dari penelitian ini menunjukkan bahwa implementasi chatbot dapat memberikan kemudahan bagi mahasiswa dalam mengakses informasi akademik. Mahasiswa cukup mengajukan pertanyaan, dan chatbot akan memberikan jawaban sesuai dengan data yang tersedia.

Kata Kunci: Chatbot, Asisten Virtual, Informasi Akademik, Linear Sequential, Android Studio, Visual Studio Code

Implementation of Chatbot as a Virtual Assistant to Help Students Obtain Academic Information

ABSTRACT

This study aims to develop a chatbot as a virtual assistant to help students at STMIK Widya Cipta Dharma obtain academic information. The research was conducted in the Academic and Student Affairs Administration Division using data collection methods such as interviews and literature studies related to chatbots.

The development method used is linear sequential, with supporting software including Visual Studio Code and Android Studio. The results of this study show that the implementation of a chatbot can facilitate students in accessing academic information. Students can simply ask questions, and the chatbot will provide answers based on available data.

Keywords: Chatbot, Virtual Assistant, Academic Information, Linear Sequential, Android Studio, Visual Studio Code.

1. PENDAHULUAN

Informasi akademik mencakup berbagai data dan petunjuk yang diperlukan oleh mahasiswa untuk menjalani aktivitas perkuliahan dengan lancar. Hal ini termasuk jadwal kelas, pengumuman penting, prosedur administrasi, dan pedoman akademik. Akses yang cepat dan tepat terhadap informasi ini sangat penting untuk mendukung kelancaran proses belajar mengajar serta menghindari kesalahan dan kebingungan yang dapat mengganggu kinerja akademik mahasiswa.

Saat ini, informasi akademik umumnya tersedia melalui berbagai platform digital seperti website, aplikasi mobile, dan sistem informasi akademik yang disediakan oleh institusi pendidikan. Di Sekolah Tinggi Manajemen Informatika dan Komputer Widya Cipta Dharma (STMIK WICIDA), Buku Pedoman Akademik 2023 menjadi salah satu sumber utama bagi mahasiswa untuk mencari informasi terkait perkuliahan. Namun, meskipun akses digital ini mempermudah mahasiswa dalam

memperoleh informasi, sering kali masih ada kendala dalam hal pencarian informasi secara cepat dan efisien.

Beberapa masalah yang sering dihadapi oleh mahasiswa dalam mengakses informasi akademik antara lain adalah kompleksitas antarmuka sistem informasi akademik, kurangnya panduan yang interaktif, serta keterbatasan dalam hal interaksi langsung. Mahasiswa sering kali mengalami kesulitan menemukan informasi spesifik seperti jadwal kelas atau prosedur administrasi, yang dapat mengakibatkan penundaan atau ketidakakuratan dalam pengambilan keputusan. Hal ini menunjukkan perlunya solusi yang lebih interaktif dan responsif untuk membantu mahasiswa.

Solusi yang dapat diimplementasikan untuk mengatasi masalah ini adalah dengan mengembangkan chatbot sebagai asisten virtual. Chatbot ini akan dirancang menggunakan Visual Code Studio dengan bahasa pemrograman Python. Dengan menggunakan algoritma Dialog Management, chatbot ini diharapkan

dapat memberikan bantuan yang cepat dan akurat kepada mahasiswa dalam mencari informasi akademik. Perancangan alur percakapan yang efektif akan dilakukan untuk memastikan chatbot dapat menjawab pertanyaan mahasiswa dengan tepat. Pengujian Whitebox dan Blackbox testing akan digunakan untuk mengevaluasi kinerja dan kehandalan chatbot.

Diharapkan dengan implementasi chatbot sebagai asisten virtual, mahasiswa STMIK WICIDA dapat mengakses informasi akademik dengan lebih mudah, cepat, dan akurat. Solusi ini diharapkan dapat meningkatkan efisiensi dan pengalaman pengguna dalam mengakses informasi akademik, serta mengurangi beban administrasi yang harus ditangani secara manual oleh staf akademik. Dengan demikian, chatbot ini tidak hanya akan mendukung kebutuhan informasi mahasiswa, tetapi juga berkontribusi pada peningkatan kualitas layanan akademik di STMIK WICIDA.

2. RUANG LINGKUP

2.1 Rumusan Masalah

Berdasarkan dari latar belakang tersebut didapatkan rumusan masalah yang akan dibahas adalah “implementasi chatbot sebagai asisten virtual untuk membantu mahasiswa STMIK WICIDA dalam mendapatkan informasi akademik”.

2.2 Batasan Masalah

- 1) Fokus pada pengembangan chatbot sebagai asisten virtual untuk mahasiswa dalam lingkungan STMIK WICIDA.
- 2) Menggunakan Buku Pedoman Akademik STMIK WICIDA 2023 sebagai salah satu sumber informasi.
- 3) Menggunakan Visual Code Studio dengan bahasa pemrograman python dan Android Studio dalam membuat aplikasi chatbot.
- 4) Analisis perancangan alur percakapan yang efektif untuk memastikan chatbot memberikan bantuan yang tepat kepada mahasiswa STMIK WICIDA.
- 5) Aplikasi Chatbot menggunakan Algoritma Dialog Management.
- 6) Menggunakan pengujian Whitebox dan Blackbox testing.
- 7) Batasan ruang lingkup meliputi hanya pengembangan dan evaluasi chatbot untuk informasi akademik STMIK WICIDA, seperti kalender akademik dan prosedur administrasi.
- 8) Penelitian tidak akan membahas aspek non-teknis seperti dampak psikologis atau sosial dari penggunaan chatbot dalam konteks pendidikan tinggi.

2.3 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mengembangkan dan mengimplementasikan sebuah chatbot sebagai asisten virtual yang dapat membantu mahasiswa STMIK WICIDA dalam mengakses informasi akademik dengan lebih efisien.

2.4 Manfaat Penelitian

Manfaat Penelitian ini akan dirasakan oleh pihak-pihak berikut:

- 1) Mahasiswa STMIK WICIDA
Tersedianya aplikasi ini, maka Mahasiswa akan lebih nyaman dalam mengetahui informasi-informasi yang ingin diketahui seputar akademik.
- 2) Staff STMIK WICIDA
Tersedianya aplikasi ini, maka staff akan lebih mudah untuk mengolah informasi yang hanya fokus ke satu sumber dan tidak mengganggu aktivitas lainnya.

3. BAHAN DAN METODE

3.1 Algoritma Manajemen Dialog – Switch Statement

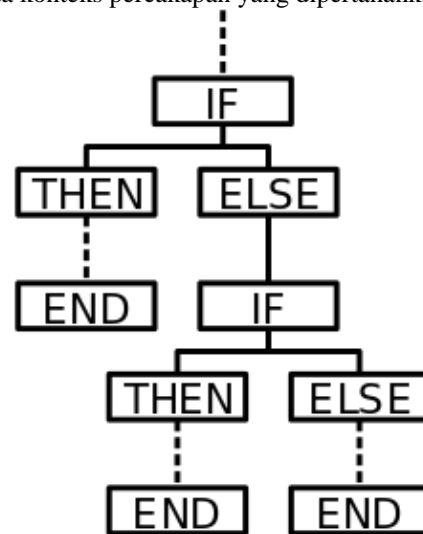
Gaya *Switch Statement* dalam membangun chatbot dengan algoritma dialogue management adalah sederhananya aturan *if-then* sederhana. Pengguna selalu memiliki inisiatif (Cañas, dkk, 2021).

Kelebihan yang didapatkan menggunakan gaya *switch statement*:

Mudah diimplementasikan dan dikelola.

Adapun, Kekurangan yang didapatkan menggunakan gaya *switch statement*:

Tidak terlalu menarik bagi pengguna. Tanpa status: tidak ada konteks percakapan yang dipertahankan.



Gambar 1 Contoh Switch Statement

Pada Gambar 1 adalah contoh *switch statement*. dapat dilihat jika “*if-then*” atau dalam bahasa manusia “jika benar”, maka *end* atau selesai. Jika “*if-else*” atau “jika tidak”, maka bisa dilakukan perulangan ke “*if-then*” lagi atau bisa juga langsung *end* atau selesai jika sudah tidak dapat lagi melakukan perulangan.

3.2 System Usability Scale (SUS)

System Usability Scale adalah alat pengukuran yang dapat digunakan untuk mengukur tingkat *usability* sebuah sistem (Paramitha, dkk, 2023).

Beberapa keunggulan menggunakan *system usability scale* antara lain:

- 1) Mudah digunakan dan diterima oleh responden

- 2) Dapat digunakan pada sample penelitian yang kecil dengan hasil yang akurat
- 3) Terbukti valid dalam menentukan apakah sistem sudah dapat digunakan dengan baik

System Usability Scale menggunakan skala Likert satu hingga lima yaitu 1 sangat tidak setuju, 2 tidak setuju, 3 netral, 4 setuju, dan 5 sangat setuju. Pertanyaan kuesioner system usability scale pun perlu disusun secara berurutan yaitu:

- 1) Saya berpikir akan menggunakan sistem ini lagi
- 2) Saya merasa sistem ini rumit untuk digunakan
- 3) Saya merasa sistem ini mudah digunakan
- 4) Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini
- 5) Saya merasa fitur-fitur sistem ini berjalan dengan semestinya
- 6) Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini)
- 7) Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat
- 8) Saya merasa sistem ini membingungkan
- 9) Saya merasa tidak ada hambatan dalam menggunakan sistem ini
- 10) Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini

Adapun cara menghitung hasil pengukuran system usability scale yaitu:

- 1) Untuk setiap pertanyaan pada urutan ganjil kurangi dengan nilai satu. Contoh pertanyaan 1 memiliki skor 4. Maka kurangi 4 dengan 1 sehingga skor pertanyaan 1 adalah 3.
- 2) Untuk setiap pertanyaan pada urutan genap kurangi nilainya dari lima. Contoh pertanyaan 2 memiliki skor 1. Maka kurangi 5 dengan 1 sehingga skor pertanyaan 2 adalah 4.
- 3) Tambahkan nilai-nilai dari pernyataan bernomor ganjil dan genap. Kemudian hasil penjumlahan tersebut dikalikan dengan 2,5.

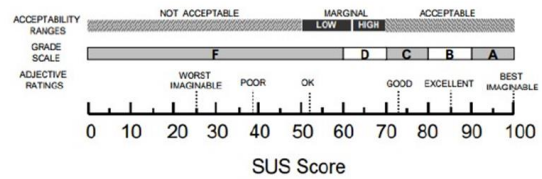
Aturan perhitungan skor untuk berlaku pada 1 responden. Untuk perhitungan selanjutnya, skor SUS dari masing-masing responden dicari skor rata-ratanya dengan menjumlahkan semua skor dan dibagi dengan jumlah responden. Berikut rumus menghitung skor SUS:

$$\bar{x} = \frac{\sum x}{n} \quad (1)$$

Diketahui \bar{x} (1) adalah skor rata-rata, $\sum x$ adalah jumlah skor SUS, dan n adalah jumlah responden.

Walau tidak dapat membantu dalam menentukan faktor atau fitur yang masih bermasalah pada sistem, system usability scale dapat membantu dalam menentukan apakah sistem sudah dapat digunakan dengan baik (*usability*).

Dapat dilihat pada gambar 2 menunjukkan contoh data skor SUS.



Gambar 2. Contoh data skor

3.3 Whitebox Testing

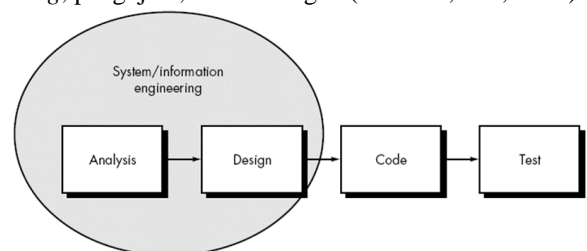
White box testing atau yang dapat diartikan menjadi “pengujian kotak putih” adalah pengujian yang dilakukan untuk menguji perangkat lunak dengan cara menganalisa dan meneliti struktur *internal* dan kode dari perangkat lunak. Untuk melakukan pengujian ini, penguji/tester perlu memiliki kemampuan dalam memahami kode dari suatu program sehingga pengujian ini tidak bisa dilakukan oleh sembarang orang (Pei, dkk, 2019).

3.4 Blackbox Testing

Black box testing atau dapat disebut juga *Behavioral Testing* adalah pengujian yang dilakukan untuk mengamati hasil *input* dan *output* dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik. Untuk melakukan pengujian, penguji tidak harus memiliki kemampuan menulis kode program. Pengujian ini dapat dilakukan oleh siapa saja (Sutiah & Supriyono, 2021).

3.5 Metode Linear Sequential

Model *Linear Sequential* terkadang disebut siklus hidup klasik atau model air terjun, model sekuensial linier menunjukkan sistematis, pendekatan sekuensial untuk pengembangan perangkat lunak yang dimulai pada tingkat sistem dan kemajuan melalui analisis, desain, *coding*, pengujian, dan dukungan (Solikhin, dkk, 2018).



Gambar 3. Model Linear Sequential

Pada gambar 3 adalah Model Linear Sequential. Gambar tersebut menggambarkan model sekuensial linier untuk rekayasa perangkat lunak. Model setelah siklus rekayasa konvensional, model sekuensial linier meliputi kegiatan sebagai berikut:

1) Analisis.

Proses pengumpulan persyaratan adalah diintensifkan dan difokuskan secara khusus pada perangkat lunak. Untuk memahami sifat program yang akan dibangun,

software engineer (“*analisis*”) harus memahami *domain* informasi untuk perangkat lunak, serta diperlukan fungsi, perilaku, kinerja, dan *interface*. Persyaratan untuk kedua sistem dan perangkat lunak didokumentasikan dan ditinjau dengan pelanggan.

2) *Design*.

Desain perangkat lunak sebenarnya adalah proses dengan banyak langkah yang berfokus pada empat atribut yang berbeda dari program: struktur data, arsitektur perangkat lunak, representasi *interface*, dan prosedural (algoritmik) secara rinci. Proses desain menerjemahkan kebutuhan ke dalam sebuah representasi perangkat lunak yang dapat dinilai kualitasnya sebelum *coding* dimulai. Seperti Kebutuhan, desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.

3) *Code*.

Langkah pembuatan kode program melakukan tugas ini. Desain harus diterjemahkan ke dalam bentuk yang dapat dibaca oleh mesin. Jika desain dilakukan dengan cara yang rinci, pembuatan kode dapat diselesaikan secara mekanis.

4) *Test*.

Setelah kode telah dihasilkan, pengujian program dimulai. Proses pengujian berfokus pada *internal* logis dari perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada *eksternal* fungsional; yaitu, melakukan pengujian untuk mengungkap kesalahan dan memastikan bahwa pendefinisian masukan akan menghasilkan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

4. PEMBAHASAN

Perlu dilakukan beberapa tahapan seperti analisis, desain, coding, lalu pengujian. Tahapan-tahapan tersebut akan dijelaskan sebagai berikut.

4.1 Analisis

4.1.1 Analisis Data

Dalam pengembangan aplikasi *chatbot* ini, data yang digunakan berasal dari Buku Pedoman Akademik STMIK WICIDA Tahun 2023/2024. Buku pedoman ini menjadi sumber utama untuk membangun basis pengetahuan *chatbot*, yang meliputi aturan akademik, proses administrasi, dan informasi lain yang relevan. Data dari buku pedoman ini diolah menjadi format yang dapat dipahami oleh *chatbot*, sehingga mampu memberikan jawaban yang akurat dan sesuai dengan konteks kebutuhan pengguna.

4.1.2 Analisis User

Setelah memberikan kuesioner kepada mahasiswa, diketahui bahwa banyak yang memerlukan *chatbot* untuk membantu mendapatkan informasi akademik. Selain itu juga mahasiswa ingin respon yang cepat dan penggunaan yang mudah. Tidak lupa juga bahwa mahasiswa meyakini bahwa *chatbot* dapat membantu mengurangi beban administrasi kampus.

4.1.3 Analisis Kebutuhan Fungsional

Dari banyaknya fitur-fitur yang ada pada *chatbot* pada umumnya, pada penelitian ini fitur yang digunakan hanyalah fitur teks. Untuk informasi yang bersumber dari buku pedoman berisi informasi-informasi seperti sejarah, peraturan umum akademik, sistem administrasi, program studi, kemahasiswaan, informasi umum, dan lain-lain. Sedangkan informasi yang tidak didapatkan dari buku pedoman berupa kalender akademik.

4.1.4 Analisis Kebutuhan Non Fungsional

Adapun perangkat yang akan digunakan selama penelitian. Untuk perangkat keras (*Hardware*) :

- 1) Intel core-i5 gen 7
- 2) RAM 20gb
- 3) *Harddisk* 1tb

Untuk perangkat lunak (*Software*) :

- 1) Visual Studio Code
- 2) Android Studio

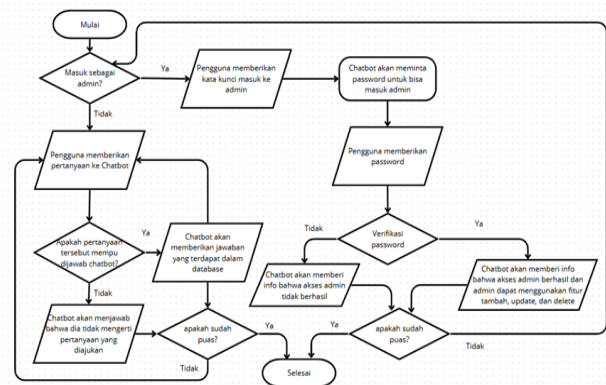
4.1.5 Analisis Sistem

Algoritma yang akan digunakan adalah *dialog management* dengan gaya *switch statement*. Yaitu pengguna akan melakukan percakapan dengan sistem dan sistem akan mencari kondisi yang tepat untuk menjawab percakapan pengguna.

4.2 Desain

4.2.1 Flowchart

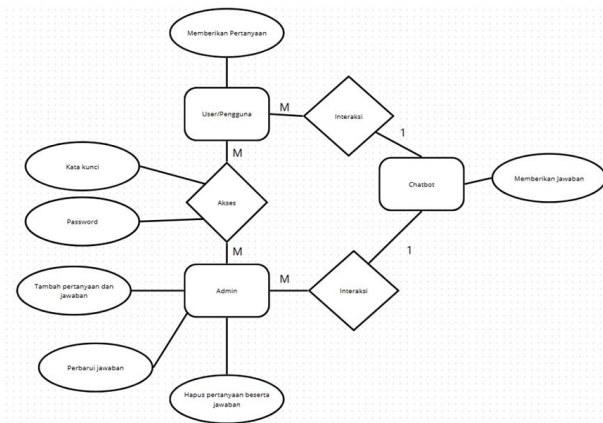
Berdasarkan hasil analisis, maka akan dirancang *flowchart* desain sistem sebagai berikut :



Gambar 4 Flowchart desain sistem

Pada Gambar, *Flowchart* ini menggambarkan alur kerja sistem *chatbot* dari awal hingga akhir. *Flowchart* dimulai dengan pengguna memberikan *input* berupa teks atau pertanyaan. *Input* tersebut diterima oleh sistem yang kemudian diproses oleh algoritma *dialog management* berbasis *switch statement*. Sistem memeriksa *database* untuk menemukan jawaban yang sesuai dengan *input* pengguna. Jika jawaban ditemukan, maka sistem akan memberikan respon kepada pengguna. *Flowchart* ini membantu menggambarkan alur dari proses kerja *chatbot* sehingga mempermudah pengembangan dan pengujian sistem.

4.2.2 ERD

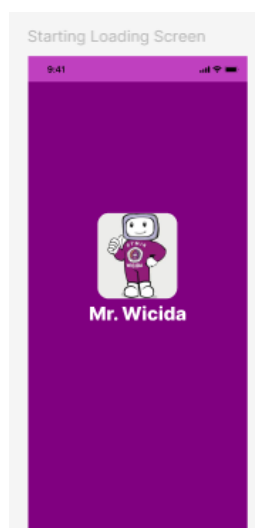


Gambar 5 ERD database

Pada gambar, Gambar ini menunjukkan *Entity Relationship Diagram* (ERD) dari database yang digunakan dalam aplikasi. ERD menjelaskan struktur tabel dalam *database*, relasi antar tabel, dan atribut yang digunakan. Tetapi, jika struktur *database* yang dimiliki hanya terdiri dari satu tabel, yaitu tabel utama yang menyimpan seluruh data yang dibutuhkan oleh *chatbot*. Maka tabel ini berfungsi sebagai pusat data untuk mengelola *input* pengguna, respon *chatbot*, dan atribut tambahan yang mendukung fungsi *chatbot*.

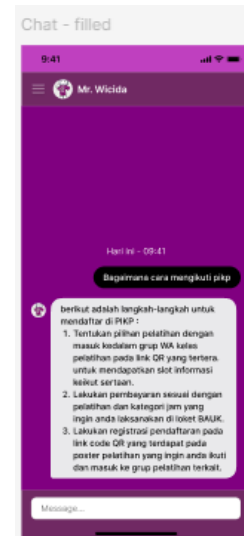
4.2.3 User Interface

Pada desain *User Interface*, alat yang digunakan adalah *Figma*.



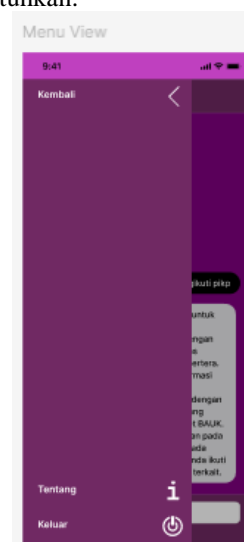
Gambar 6 Tampilan Splash

Pada gambar, Tampilan *Splash* ini merupakan tampilan awal yang muncul ketika aplikasi pertama kali dijalankan. Fungsi utama dari Tampilan *Splash* adalah memberikan waktu bagi sistem untuk mempersiapkan elemen-elemen penting sebelum masuk ke antarmuka utama. Pada tampilan ini biasanya ditampilkan logo aplikasi, teks motivasi, atau animasi sederhana untuk agar tidak membosankan. Tampilan *splash* juga penting untuk memberikan kesan pertama yang positif kepada pengguna.



Gambar 7 Tampilan halaman chat

Pada gambar, Halaman ini merupakan inti dari aplikasi, di mana pengguna berinteraksi langsung dengan *chatbot*. *Interface* dirancang agar sederhana dan mudah digunakan, dengan kolom *input* untuk mengetik pertanyaan dan area *chat* untuk menampilkan percakapan antara pengguna dan *chatbot*. *Chatbot* akan menjawab pertanyaan pengguna berdasarkan data yang ada di *database*. Desain ini dirancang untuk memberikan pengalaman pengguna yang intuitif, memastikan bahwa pengguna dapat dengan cepat menemukan informasi yang mereka butuhkan.



Gambar 8 Tampilan Halaman menu

Pada gambar, Tampilan ini menunjukkan halaman *menu* utama aplikasi, yang berfungsi sebagai navigasi ke fitur-fitur lain dalam aplikasi. Di halaman ini, pengguna dapat memilih untuk mengakses berbagai fitur, seperti halaman *chat*, informasi tentang pengembang, atau tombol untuk keluar dari aplikasi. Halaman menu dirancang untuk membantu pengguna menavigasi aplikasi dengan mudah, memastikan bahwa setiap fungsi dapat diakses dengan cepat dan tanpa kebingungan.



Gambar 9 Tampilan halaman tentang

Pada gambar, Halaman ini bertujuan untuk memberikan informasi tambahan kepada pengguna pengembang aplikasi. Informasi yang biasanya ditampilkan meliputi informasi terkait dengan pengembang. Halaman ini tidak hanya meningkatkan transparansi tetapi juga membangun hubungan yang lebih baik antara pengembang dan pengguna.

4.3 Coding

Dengan desain sistem yang sudah ada maka akan dilanjutkan dengan membangun program dan struktur database.

```

1 • create database chatbot_db;
2
3 CREATE TABLE conversation_flow (
4   id INT AUTO_INCREMENT PRIMARY KEY,
5   user_input VARCHAR(255),
6   bot_response VARCHAR(10000),
7   next_state VARCHAR(50),
8   created_by_admin BOOLEAN DEFAULT 0 -- kolom untuk menandai jika data ditambahkan oleh admin
9 );

```

Gambar 10 Kode mysql

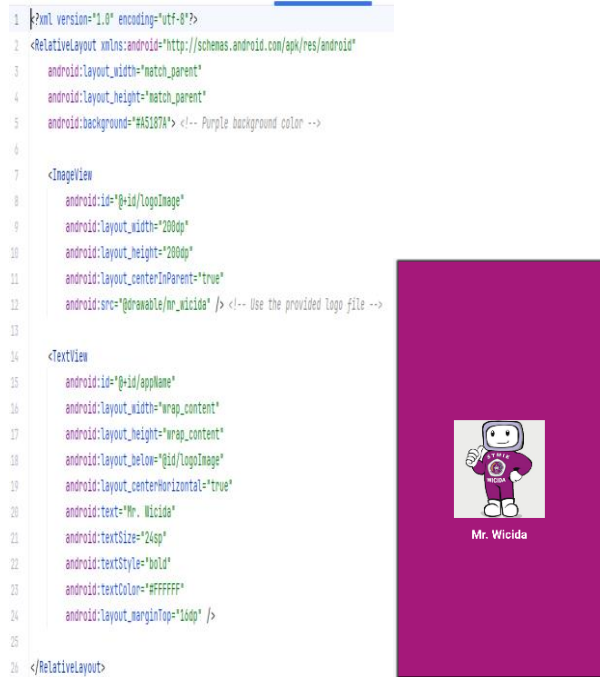
Pada gambar 10, adalah kode untuk membuat database. Nama database ini adalah *chatbot_db*. Lalu dilanjutkan dengan membuat tabel yang bernama *conversation_flow*. Isi tabel *conversation_flow* meliputi *id*, *user_input*, *bot_response*, *next_state*, dan *created_by_admin*.

Struktur database mysql tersebut akan digunakan pada API yaitu mencakup satu tabel dan beberapa kolom. Pada tabel 1 menjelaskan isi dari tabel *conversation_flow*.

Tabel 1. data tabel conversation_flow

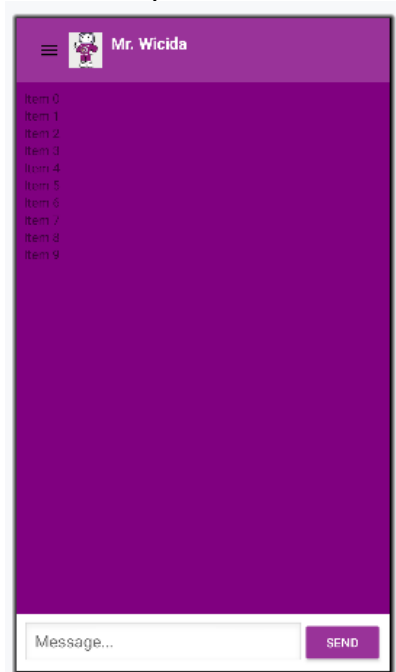
Kolom	Tipe Data	Keterangan
id	int	auto_increment, primary key
user_input	varchar(255)	Berisi kalimat pertanyaan
bot_respon se	varchar(10000)	Berisi kalimat jawaban
Next_state	varchar(50)	Null
Created_b y_admin	tinyint(1)	Jika baris ditambahkan oleh admin maka berisi 1, jika tidak maka 0

Dengan adanya database tersebut, dapat membantu menjalankan fungsi program. Tetapi sebelum itu dibuatlah tampilan chatbot menggunakan Android Studio.



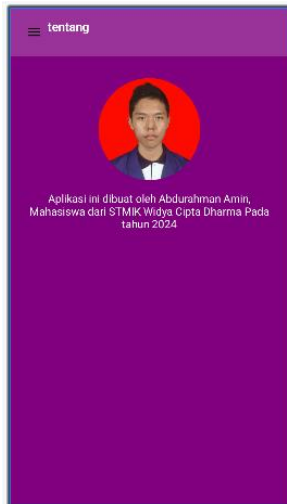
Gambar 11 Kode dan tampilan splash

Pada gambar 11, adalah tampilan splash yang berfungsi sebagai pembuka sebelum menampilkan halaman awal. *ImageView* untuk menampilkan gambar. *TextView* untuk menampilkan teks.



Gambar 12 Halaman awal (percakapan)

Pada gambar 12, adalah tampilan halaman awal yang dimana halaman ini digunakan untuk pengguna bisa berinteraksi dengan chatbot.



Gambar 13 Halaman tentang

Pada gambar 13, adalah halaman tentang. Halaman ini bertujuan agar pengguna dapat mengetahui informasi pengembang.

```
1 from flask import Flask, request, jsonify
2 import mysql.connector
3 from difflib import get_close_matches # Untuk mencari string mirip
```

Gambar 14 library API

Pada gambar 14, kode tersebut bertujuan untuk dapat digunakannya fungsi-fungsi yang akan dikerjakan.

```
4 # Initialize Flask app
5 app = Flask(__name__)
6
```

Gambar 15 Inisial aplikasi flask

Pada gambar 15, kode tersebut bertujuan untuk mengubah inisial app. Disini hanya menggunakan kode default.

```
7 # Koneksi ke database MySQL
8 db = mysql.connector.connect(
9     host="localhost",
10    user="root",
11    password="test01",
12    database="chatbot_db"
13 )
14
15 cursor = db.cursor()
16
```

Gambar 16 Koneksi ke database mysql

Pada gambar 16, kode tersebut bertujuan untuk menghubungkan ke database mysql.

```
17
18 # Fungsi untuk mendapatkan respons dari database
19 def get_response(user_input):
20     cursor.execute("SELECT user_input, bot_response, next_state FROM conversation_flow")
21     all_questions = cursor.fetchall()
22
23     # Cek apakah pertanyaan persis ada di database
24     exact_match = next((row for row in all_questions if row[0].lower() == user_input.lower()), None)
25     if exact_match:
26         return exact_match[1], None # Kembalikan jawaban langsung
27
28     # Cari pertanyaan serupa jika tidak ada hasil
29     all_user_inputs = [row[0] for row in all_questions]
30     suggestions = get_close_matches(user_input, all_user_inputs, n=1, cutoff=0.6) # Ambil satu saran dengan kemiripan minimal 60%
31
32     if suggestions:
33         suggested_question = suggestions[0]
34         suggested_response = next(row[1] for row in all_questions if row[0] == suggested_question)
35         return f"Apakah maksud Anda: '{suggested_question}'? Jika ya, jawab 'iya'.", suggested_response
36     else:
37         return "Maaf, saya tidak mengerti.", None
38
```

Gambar 17 Respon dari database

Pada gambar 17, kode tersebut bertujuan untuk mendapatkan respon dari database.

```
39 # Fungsi untuk mengecek apakah user adalah admin
40 def verify_admin(password):
41     if password == "Wicida":
42         return True
43     return False
44
```

Gambar 18 Fungsi verifikasi admin

Pada gambar 18, kode tersebut bertujuan sebagai fungsi untuk mengecek apakah pengguna adalah admin.

```
45 # Variable to track admin status
46 is_admin = False
47 is_requesting_admin_password = False
48 current_admin_action = None # None, "tambah", "update", "delete"
49 pending_data = {} # Menyimpan data sementara untuk proses admin
50 pending_suggestion_response = None # Menyimpan jawaban untuk saran
51
```

Gambar 19 Variabel admin

Pada gambar 19, kode 46 hingga 47 bertujuan sebagai variabel untuk memantau status admin. Kode 48 adalah aksi admin. Kode 49 adalah data yang disimpan sementara. Kode 50 adalah jawaban yang disarankan.

```
52 # Route to interact with the chatbot
53 @app.route("/chatbot", methods=['POST'])
54 def chatbot_api():
55     global is_admin, is_requesting_admin_password, current_admin_action, pending_data, pending_suggestion_response
56     data = request.get_json()
57     user_input = data.get('message', '')
58
```

Gambar 20 app.route dan API

Pada gambar 20, kode 42 adalah rute yang akan digunakan pada aplikasi android agar bisa terhubung ke API. Sedangkan kode 43 adalah inti kode yang menghubungkan variabel-variabel yang ada.

```
60 # Jika pengguna telah ditawarkan saran dan menjawab 'iya'
61 if pending_suggestion_response and user_input.lower() == "iya":
62     response = pending_suggestion_response
63     pending_suggestion_response = None
64     return jsonify({"response": response})
65
66 # Jika pengguna telah ditawarkan saran dan menjawab lainnya
67 if pending_suggestion_response:
68     pending_suggestion_response = None
69     return jsonify({"response": "Maaf, saya tidak mengerti."})
70
```

Gambar 21 Respon chatbot setelah memberi saran

Pada Gambar 21, kode tersebut bertujuan untuk memberikan jawaban ke pengguna setelah memberikan saran pertanyaan yang dimaksud oleh pengguna.

```
71 # If user claims to be admin
72 if user_input.lower() == "saya admin" and not is_requesting_admin_password:
73     is_requesting_admin_password = True
74     return jsonify({"response": "Berikan password untuk mengakses admin."})
75
```

Gambar 22 Kode masuk admin

Pada Gambar 22, kode 50 jika pengguna memasukkan "saya admin" maka chatbot akan meminta pengguna untuk memasukkan password.

```
76 # Jika chatbot sedang menunggu password admin
77 if is_requesting_admin_password:
78     if verify_admin(user_input):
79         is_admin = True
80         is_requesting_admin_password = False
81         return jsonify({"response": "Anda sekarang dalam mode admin."})
82     else:
83         is_requesting_admin_password = False
84         return jsonify({"response": "Password salah. Akses admin ditolak."})
85
```

Gambar 23 Verifikasi password

Pada Gambar 23, kode tersebut bertujuan untuk memastikan apakah password yang dimasukkan pengguna benar atau tidak. Jika benar maka pengguna akan mendapatkan pesan bahwa pengguna memasuki

mode admin, jika salah maka pengguna akan mendapatkan pesan *password* salah.

```

86 # Fungsi admin: tambah, ubah, hapus (berbasis percakapan)
87 if is_admin:
88     if current_admin_action is None:
89         if user_input.lower() == "tambah":
90             current_admin_action = "tambah"
91             pending_data.clear()
92             return jsonify({"response": "Berikan pertanyaan baru."})
93
94         elif user_input.lower() == "update":
95             current_admin_action = "update"
96             pending_data.clear()
97             return jsonify({"response": "Berikan pertanyaan yang ingin diupdate."})
98
99         elif user_input.lower() == "delete":
100            current_admin_action = "delete"
101            pending_data.clear()
102            return jsonify({"response": "Berikan pertanyaan yang ingin dihapus."})
103

```

Gambar 24 Fungsi admin

Pada Gambar 24, kode tersebut bertujuan jika admin memberikan perintah “tambah”, “*update*”, dan “*delete*” maka *chatbot* akan minta perintah selanjutnya.

```

104 # Proses tambah
105 if current_admin_action == "tambah":
106     if "question" not in pending_data:
107         pending_data["question"] = user_input
108         return jsonify({"response": "Berikan jawaban untuk pertanyaan ini."})
109     elif "answer" not in pending_data:
110         pending_data["answer"] = user_input
111         cursor.execute(
112             "INSERT INTO conversation_flow (user_input, bot_response, created_by_admin) VALUES (%s, %s, 1)",
113             (pending_data["question"], pending_data["answer"])
114         )
115         db.commit()
116         current_admin_action = None
117         pending_data.clear()
118         return jsonify({"response": "Pertanyaan dan jawaban baru berhasil ditambahkan."})
119

```

Gambar 25 Fungsi admin tambah

Pada Gambar 25, kode tersebut bertujuan jika admin memberikan perintah “tambah” maka *chatbot* akan meminta pertanyaan dan jawaban yang ingin ditambahkan oleh admin.

```

120 # Proses update
121 if current_admin_action == "update":
122     if "question" not in pending_data:
123         pending_data["question"] = user_input
124         return jsonify({"response": "Berikan jawaban baru untuk pertanyaan ini."})
125     elif "answer" not in pending_data:
126         pending_data["answer"] = user_input
127         cursor.execute(
128             "UPDATE conversation_flow SET bot_response = %s WHERE user_input = %s",
129             (pending_data["answer"], pending_data["question"])
130         )
131         db.commit()
132         current_admin_action = None
133         pending_data.clear()
134         return jsonify({"response": "Jawaban berhasil diubah."})
135

```

Gambar 26 Fungsi admin update

Pada Gambar 26, kode tersebut bertujuan jika admin memberikan perintah “*update*” maka *chatbot* akan meminta pertanyaan yang jawabannya ingin dirubah.

```

136 # Proses delete
137 if current_admin_action == "delete":
138     if "question" not in pending_data:
139         pending_data["question"] = user_input
140         cursor.execute("DELETE FROM conversation_flow WHERE user_input = %s", (pending_data["question"],))
141         db.commit()
142         current_admin_action = None
143         pending_data.clear()
144         return jsonify({"response": "Pertanyaan dan jawaban berhasil dihapus."})
145

```

Gambar 27 Fungsi admin delete

Pada Gambar 27, kode tersebut bertujuan jika admin memberikan perintah “*delete*” maka *chatbot* akan meminta pertanyaan yang ingin dihapus beserta jawabannya.

```

146 # General user input response
147 bot_response, next_state_or_suggestion = get_response(user_input)
148 if next_state_or_suggestion:
149     pending_suggestion_response = next_state_or_suggestion
150     return jsonify({"response": bot_response})
151

```

Gambar 28 Respon bot

Pada Gambar 28, kode tersebut bertujuan agar *chatbot* memeriksa database terkait pertanyaan pengguna.

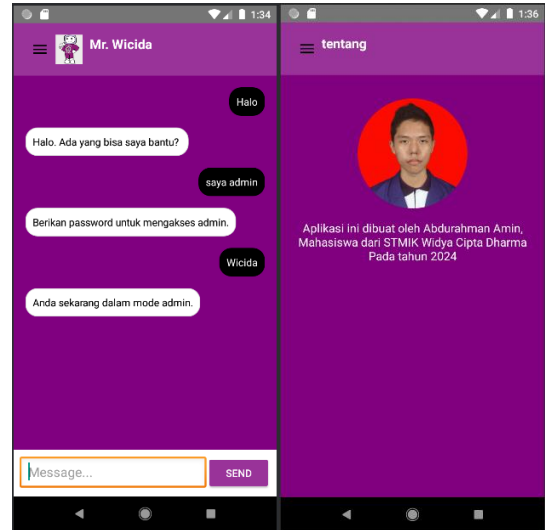
```

152 # Run the Flask app
153 if __name__ == '__main__':
154     app.run(host='0.0.0.0', port=5000, debug=True) #ngrok http 5000

```

Gambar 29 Menjalankan API

Pada Gambar 29, kode tersebut bertujuan sebagai API, yang dimana akan menampilkan *link* yang diperlukan oleh aplikasi android agar dapat terhubung ke API tersebut.



Gambar 30 Tampilan Aplikasi

Pada Gambar 30 adalah tampilan aplikasi *chatbot* yang sudah jadi.

4.4 Pengujian

Pada tahap ini, pengujian dilakukan agar dapat mengetahui apakah program yang dijalankan sesuai dengan yang diharapkan atau tidak.

4.4.1 Whitebox Testing

Pengujian ini dilakukan untuk memvalidasi logika program, memastikan alur eksekusi kode sesuai dengan desain, serta memeriksa adanya kesalahan pada struktur kode. Alur program yang akan diuji adalah masuk sebagai admin.

```

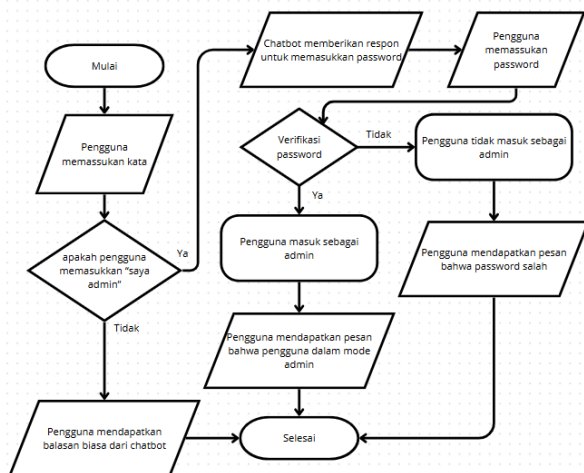
# If user claims to be admin
if user_input.lower() == "saya admin" and not is_requesting_admin_password:
    is_requesting_admin_password = True
    return jsonify({"response": "Berikan password untuk mengakses admin."})

# Jika chatbot sedang menunggu password admin
if is_requesting_admin_password:
    if verify_admin(user_input):
        is_admin = True
        is_requesting_admin_password = False
        return jsonify({"response": "Anda sekarang dalam mode admin."})
    else:
        is_requesting_admin_password = False
        return jsonify({"response": "Password salah. Akses admin ditolak."})

```

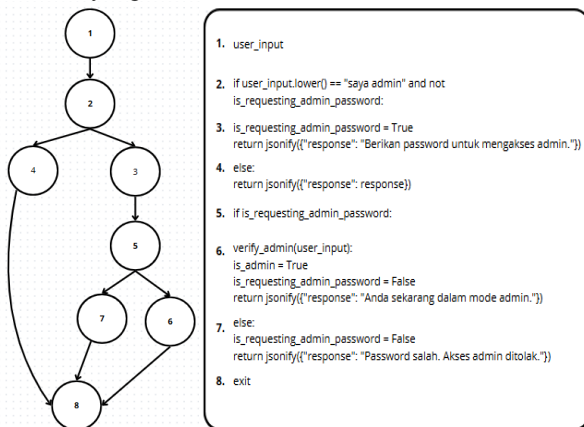
Gambar 31 Kode masuk sebagai admin

Pada gambar 31, jika pengguna memasukkan “saya admin” maka *chatbot* akan meminta *password* untuk mengakses admin. *Password* admin adalah “Wicida”. Jika pengguna memasukkan *password* yang benar maka akan mendapatkan respon “Anda sekarang dalam mode admin”, sedangkan jika pengguna memasukkan *password* yang salah maka akan mendapatkan respon “*Password* salah. Akses admin ditolak”.



Gambar 32 Alur masuk admin

Pada gambar 32, diketahui bahwa jika pengguna memasukkan kata kunci dengan benar maka *chatbot* akan meminta **password** yang akan mendapatkan dua jawaban berbeda tergantung dari benar atau tidak *password* yang diberikan.



Gambar 33 Basis path testing untuk masuk admin

Pada gambar 4.30 Diketahui terdapat dua jalur kondisi yaitu :

Jalur 1 : 1, 2, 4, 8.

Jalur 2 : 1, 2, 3, 5, 6, 8.

Jalur 3 : 1, 2, 3, 5, 7, 8.

Maka jika di Cyclomatic Complexity,
 Diketahui :

$$E = 9$$

$$N = 8$$

Maka :

$$CC = E - N + 2 \quad (2)$$

$$CC = 9 - 8 + 2$$

$$CC = 9 - 8 + 2$$

$$CC = 3$$

Diketahui CC (2) adalah *Cyclomatic Complexity*, E adalah jumlah garis (*edges*) dalam grafik alur kontrol, dan N adalah jumlah simpul (*nodes*) dalam grafik alur kontrol. Setelah dilakukan perhitungan tersebut,

Banyaknya *Cyclomatic Complexity* pada codingan adalah 3.

Tabel 2 Whitebox testing

Jalur	Input	Output yang diharapkan	Hasil
Jalur 1	Pengguna memasukkan teks bukan "saya admin"	Chatbot hanya merespon seperti inputan biasa	Chatbot menganggap sebagai pertanyaan biasa
Jalur 2	Pengguna memasukkan teks saya admin dan password benar	Chatbot memberikan akses admin ke pengguna	Chatbot memberikan pesan bahwa sudah masuk sebagai admin
Jalur 3	Pengguna memasukkan teks saya admin dan password salah	Chatbot menolak akses admin ke pengguna	Chatbot memberikan pesan bahwa password salah

4.4.2 Blackbox Testing

Pengujian ini dilakukan untuk memastikan sistem bekerja sesuai kebutuhan pengguna berdasarkan input dan output.

Tabel 3 Blackbox testing

No	Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Membuka aplikasi	Membuka aplikasi lalu menampilkan loading setelah membuka aplikasi	Menampilkan loading setelah membuka aplikasi	Tertampil loading setelah membuka aplikasi	Lulus
2	Membuka halaman awal	Membuka tampilan halaman awal setelah loading	Menampilkan halaman awal setelah loading	Tertampil halaman awal setelah loading	Lulus
3	Melakukan percakapan	Mengirim teks pertanyaan yang tersedia di database dan akan dijawab oleh chatbot	Pertanyaan yang diajukan pengguna akan dijawab oleh chatbot	Chatbot menjawab pertanyaan pengguna	Lulus
4	Kode masuk admin	Mengirim kode masuk "saya admin" untuk akses admin	Chatbot akan meminta "saya admin" untuk bisa memberikan akses	Chatbot meminta password untuk dimasukkan	Lulus
5	Admin	Memasukkan password "Wicida" untuk verifikasi akses admin	Pengguna memberikan password yang diminta chatbot untuk bisa mengakses admin	Chatbot memberikan akses admin ke pengguna	Lulus
6	Tambah	Menambah pertanyaan dengan kode "tambah" melalui chatbot setelah masuk sebagai admin	Chatbot akan meminta pertanyaan dan jawaban yang ingin ditambahkan	Chatbot meminta pertanyaan dan jawaban yang ingin ditambahkan	Lulus
7	Update	Memperbarui jawaban dengan kode "update" melalui chatbot setelah masuk sebagai admin	Chatbot akan meminta pertanyaan yang akan diperbarui jawabannya	Chatbot meminta pertanyaan yang jawabannya akan diperbarui	Lulus
8	Tombol Menu	Pengguna menekan tombol menu	Saat pengguna menekan tombol menu maka akan muncul beberapa tombol menu	Saat menekan tombol menu muncul beberapa tombol menu	Lulus

9	Tombol Tentang	Pengguna menekan tombol tentang	Saat pengguna menekan tombol tentang maka akan dialihkan ke tampilan tentang	Saat menekan tombol tentang, tampilan dialihkan ke tampilan tentang	Lulus
10	Tombol Chatbot	Pengguna menekan tombol chatbot	Saat pengguna menekan tombol chatbot maka akan dialihkan ke halaman awal	Saat menekan tombol chatbot, tampilan dialihkan ke halaman awal	Lulus
11	Tombol Keluar	Pengguna menekan tombol keluar	Saat pengguna menekan tombol keluar maka aplikasi akan tertutup	Saat menekan tombol keluar, aplikasi tertutup	Lulus

Setelah pengujian blackbox selesai, dilanjutkan dengan pengujian ke mahasiswa terkait aplikasi dan mengisi kuesioner dengan System Usability Scale.

Berikut ini adalah daftar pertanyaan yang diajukan ke mahasiswa :

- 1) Saya merasa aplikasi ini mudah digunakan.
- 2) Saya merasa fungsi-fungsi dalam aplikasi ini terlalu rumit.
- 3) Saya merasa percaya diri saat menggunakan aplikasi ini.
- 4) Saya merasa bahwa saya membutuhkan bantuan teknis untuk dapat menggunakan aplikasi ini.
- 5) Fitur-fitur dalam aplikasi ini terintegrasi dengan baik.
- 6) Saya merasa ada terlalu banyak ketidakkonsistenan dalam aplikasi ini.
- 7) Saya merasa bahwa sebagian besar orang dapat dengan cepat belajar menggunakan aplikasi ini.
- 8) Saya merasa aplikasi ini sangat membingungkan untuk digunakan.
- 9) Saya merasa nyaman menggunakan aplikasi ini.
- 10) Saya merasa bahwa menggunakan aplikasi ini memerlukan banyak usaha.

Pertanyaan tersebut menggunakan skala 1 sampai 5 untuk menjawab, seperti :

- 1) Jika memilih angka 1 berarti Sangat Tidak Setuju.
- 2) Jika memilih angka 2 berarti Tidak Setuju.
- 3) Jika memilih angka 3 berarti Netral.
- 4) Jika memilih angka 4 berarti Setuju.
- 5) Jika memilih angka 5 berarti Sangat Setuju.

No	Nama Lengkap	NIM	Program Studi	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Charlos Herdiyanto	2043015	Teknik Informatika	4	3	3	4	4	3	4	4	4	4
2	Aff Qur Rahman	2043012	Teknik Informatika	4	2	4	2	4	2	4	2	2	2
3	Raihan Aliff Wahyudinur	2043051	Teknik Informatika	5	2	5	5	3	5	2	5	3	3
4	Ahmad Fadhil Aqram Pasennangi	2043022	Teknik Informatika	5	1	4	1	5	1	5	2	5	2
5	Didit Christian Nenosae	2043046	Teknik Informatika	4	3	5	5	4	5	4	3	4	3
6	Putri Nuraini Mawardah	2342005	Bisnis Digital	5	1	5	3	4	4	3	3	4	2
7	satrio ardy	2043073	Teknik Informatika	5	5	5	1	5	5	5	5	5	5
8	Muhammad Ivan Wardani	2043021	Teknik Informatika	5	3	3	2	4	3	3	2	4	3
9	Andre Irawan Ameng	2043043	Teknik Informatika	4	3	4	4	4	3	4	2	3	2
10	Bayu Pratama	2043044	Teknik Informatika	4	4	4	3	3	3	4	3	3	3
11	Ferbian Allo Rendi	2043036	Teknik Informatika	5	2	5	3	4	3	4	2	4	2

Gambar 34 Hasil responden

Pada gambar 34, adalah gambar hasil kuesioner yang telah diisi oleh responded. Hasil jawaban tersebut belum diubah sesuai prosedur SUS. Maka perlu diubah dahulu agar bisa melakukan perhitungan. Hal yang perlu diubah adalah :

- 1) Untuk setiap pertanyaan pada urutan ganjil kurangi dengan nilai satu. Contoh pertanyaan 1 memiliki skor 4. Maka kurangi 4 dengan 1 sehingga skor pertanyaan 1 adalah 3.
- 2) Untuk setiap pertanyaan pada urutan genap kurangi nilainya dari lima. Contoh pertanyaan 2 memiliki skor 1. Maka kurangi 5 dengan 1 sehingga skor pertanyaan 2 adalah 4.
- 3) Tambahkan nilai-nilai dari pernyataan bernomor genap dan ganjil. Kemudian hasil penjumlahan tersebut dikalikan dengan 2,5.

No	Nama Lengkap	NIM	Program Studi	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Jumlah	Jumlah x 2,5
1	Charlos Herdiyanto	2043015	Teknik Informatika	3	2	2	1	3	2	3	1	3	1	21	52,5
2	Aff Qur Rahman	2043012	Teknik Informatika	3	3	3	3	3	3	3	1	3	3	28	70
3	Raihan Aliff Wahyudinur	2043051	Teknik Informatika	4	3	4	0	4	2	4	3	4	2	30	75
4	Ahmad Fadhil Aqram Pasennangi	2043022	Teknik Informatika	4	4	3	4	4	4	4	3	4	3	37	92,5
5	Didit Christian Nenosae	2043046	Teknik Informatika	3	2	4	0	3	0	3	2	3	2	22	55
6	Putri Nuraini Mawardah	2342005	Bisnis Digital	4	4	4	2	3	1	2	2	3	3	28	70
7	satrio ardy	2043073	Teknik Informatika	4	0	4	0	0	0	4	0	4	0	16	40
8	Muhammad Ivan Wardani	2043021	Teknik Informatika	4	2	2	3	3	2	2	3	3	2	26	65
9	Andre Irawan Ameng	2043043	Teknik Informatika	3	2	3	1	3	2	3	2	3	2	25	62,5
10	Bayu Pratama	2043044	Teknik Informatika	3	1	3	2	2	2	3	2	2	2	22	55
11	Ferbian Allo Rendi	2043036	Teknik Informatika	4	3	4	2	3	2	3	3	3	3	30	75
Total :														712,5	

Gambar 35 Hasil responden yang telah di ubah

Pada Gambar 35, adalah gambar yang sudah diubah sesuai prosedur SUS. Dengan data yang sudah ada maka dapat dilakukan perhitungan.

Diketahui :

$$\sum x = 712,5$$

$$n = 11$$

$$\bar{x} = ?$$

Maka :

$$\bar{x} = \frac{\sum x}{n}$$

$$\bar{x} = \frac{712,5}{11}$$

$$\bar{x} = 64,7727 \quad (3)$$

Dari penjumlahan tersebut (3), ditemukan bahwa skor SUS adalah 64,7727. Jika dilihat melalui data skor di gambar 2, maka aplikasi ini memiliki *Acceptability Ranges* dengan *Marginal Low*. Untuk *Grade Scale* yaitu D. Lalu, *Adjective Ratings* diantara OK dan *GOOD*.

5. KESIMPULAN

Implementasi *chatbot* sebagai asisten *virtual* dapat membantu mahasiswa dalam mendapatkan informasi akademik tetapi ada beberapa kekurangan seperti terlalu spesifiknya format pertanyaan dan tampilan yang kurang ramah untuk orang awam sehingga setelah dilakukan SUS, nilai yang didapat adalah 64,7727. Menurut data skor SUS, maka aplikasi mendapatkan *Acceptability Ranges* dengan *Marginal Low*, *Grade Scale* D, dan *Adjective Ratings* diantara OK dan *GOOD*.

6. SARAN

Saran untuk Penelitian Selanjutnya:

- 1) Sebaiknya dikembangkan sistem yang tidak hanya menerima pertanyaan spesifik, tetapi juga mampu

mendeteksi kesesuaian teks melalui mekanisme pencocokan teks yang lebih fleksibel, seperti sistem pencarian berbasis similaritas. Selain itu, dapat ditambahkan fitur pintasan atau tombol yang menampilkan format pertanyaan.

- 2) Disarankan untuk menggunakan sumber data yang lebih mutakhir agar informasi yang diberikan tetap relevan dan akurat.
- 3) Perlu dilakukan pengembangan *User Interface* (UI) khusus untuk admin agar lebih intuitif dan memudahkan dalam pengelolaan pertanyaan serta jawaban.
- 4) Tampilan aplikasi perlu diperbarui secara berkala agar tetap sesuai dengan tren desain terkini dan meningkatkan pengalaman pengguna.
- 5) Sistem keamanan perlu ditingkatkan untuk memastikan perlindungan data pengguna dan mencegah potensi ancaman terhadap aplikasi.

7. DAFTAR PUSTAKA

- Akhir, U. (2020). Sistem Informasi Notifikasi Absensi Kepada Orang Tua Dengan Algoritma Md5 Berbasis Client Server (Studi Kasus: Man 1 In Repository.Pancabudi.Ac.Id, diakses 21 Mei 2024.
- Akhyar Lubis dan Isnar Sumartono. 2023. Implementasi Layanan Akademik Berbasis Chatbot untuk Meningkatkan Interaksi Mahasiswa. *RESOLUSI : Rekayasa Teknik Informatika dan Informasi*. Vol 3 (5). 247 dan 251
- Akmal, N. K., & Dasaprawira, M. N. (2022). Rancang bangun Application Programming Interface (API) menggunakan gaya arsitektur GraphQL untuk pembuatan sistem informasi pendataan anggota Unit Kegiatan Mahasiswa (UKM) studi kasus UKM Starlabs. *Jurnal SITECH : Sistem Informasi Dan Teknologi*, 5(1). <https://doi.org/10.24176/sitech.v5i1.7937>, diakses 4 Maret 2024.
- Anggraeni, I. (2019). Pengertian Implementasi dan PendapaT Ahli. *Journal of Chemical Information and Modeling*, 53(9), 16–36, diakses 14 Mei 2024.
- Anggraini, Y., Pasha, D., Damayanti, D., & Setiawan, A. (2020). SISTEM INFORMASI PENJUALAN SEPEDA BERBASIS WEB MENGGUNAKAN FRAMEWORK CODEIGNITER. *Jurnal Teknologi Dan Sistem Informasi*, 1(2). <https://doi.org/10.33365/jtsi.v1i2.236>, diakses 21 Mei 2024.
- Asnal, H., Junadhi, Jamaris, M., Mardainis, & Irawan, Y. (2022). Workshop UI/UX Design dan Prototyping dengan Figma di SMK Taruna Masmur Pekanbaru. *J-PEMAS - Jurnal Pengabdian Masyarakat*, 3(1). <https://doi.org/10.33372/j-pemas.v3i1.800>
- Bariyah, S. H., & Imania, K. A. N. (2022). Pengembangan Virtual Assistant Chatbot Berbasis Whatsapp Pada Pusat Layanan Informasi Mahasiswa Institut Pendidikan Indonesia - Garut. *JURNAL PETIK*, 8(1). <https://doi.org/10.31980/jpetik.v8i1.1575> , diakses 14 Mei 2024.
- Bayu Puji Nursidik. 2023. Penggunaan Chatbot dalam Implementasi User Interface dan User Experience Toko Sembako dengan Pendekatan User Centered Design. Tugas Akhir. Fakultas Teknik, Tasikmalaya : Universitas Siliwangi.
- Cañas, P., Griol, D., & Callejas, Z. (2021). Towards versatile conversations with data-driven dialog management and its integration in commercial platforms. *Journal of Computational Science*, 55. <https://doi.org/10.1016/j.jocs.2021.101443> , diakses 20 Mei 2024.
- Eleni Adamopoulou dan Lefteris Moussiades. 2020. An Overview of Chatbot Technology. Dalam *Artificial Intelligence Applications and Innovations. AIAI 2020. IFIP Advances in Information and Communication Technology*, vol 584. (Ilias Maglogiannis, Lazaros Iliadis, dan Elias Pimenidis). Hal 373–383. Springer Cham : Neos Marmaras.
- Fitriyani Puspa Samodra. 2023. Akademik adalah Segala Hal yang Bersifat Ilmiah, Berikut Ulasannya, (Online), <https://www.liputan6.com/hot/read/5431761/akademik-adalah-segala-hal-yang-bersifat-ilmiah-berikut-ulasannya> , diakses 14 Mei 2024.
- Guntoro, Loneli Costaner, dan Lisnawita. 2020. Aplikasi Chatbot untuk Layanan Informasi dan Akademik Kampus Berbasis Artificial Intelligence Markup Language (AIML). *Digital Zone: Jurnal Teknologi Informasi & Komunikasi*. Vol 11 (2). 291 dan 293.
- Juniati, C. (2021). Aplikasi Penjualan Unit Sepeda Motor Berbasis Website Pada PT. Aneka Makmur Sejahtera Capem Kota Baru Pontianak. *Jurnal Sistem Informasi Akuntansi*, 2(2). <https://doi.org/10.31294/justian.v2i02.996>
- Khairina F. Hidayati. 2021. Mengenal Gantt Chart, Obat Antimolor untuk Proyek Kantormu, (Online), <https://glints.com/id/lowongan/gantt-chart-adalah/> , diakses 23 April 2024.
- Ni Nyoman Emang Smrti, I Putu Gd Sukenada, A., Ni Kadek, D. T. R., Adnan, A., & Pande Putu Ode, J. (2023). Flowgorithm Sebagai Penunjang



- Pembelajaran Algoritma dan Pemrograman. *Jurnal Bangkit Indonesia*, 12(1). <https://doi.org/10.52771/bangkitindonesia.v12i1.218>
- Nicolash Jeremy Onoma Lase, Rizal Setya Perdana, dan Dahnil Syauqy. 2023. Implementasi Chatbot Frequently Asked Question (FAQ) Universitas Brawijaya berbasis Voice Command pada Raspberry Pi 4 dengan Metode Naïve Bayes. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*. Vol : 7 (6). 2971 dan 2972, diakses 14 Mei 2024.
- Nurul Afifa, Randy Erfa Saputra, dan Ratna Astuti Nugrahaeni. 2023. Implementasi NLP Pada Chatbot Layanan Akademik Dengan Algoritma Bert. *e-Proceeding of Engineering*. Vol.10 (1). 384 dan 385.
- Paramitha, V. A. M., Martini, S., & Mufidah, I. (2023). PERANCANGAN MOBILE WEBSITE CLOUD KITCHEN LAA ZURU MENGGUNAKAN METODE DESIGN THINKING. *RADIAL : Jurnal Peradaban Sains, Rekayasa Dan Teknologi*, 11(2). <https://doi.org/10.37971/radial.v11i2.408>.
- Pei, K., Cao, Y., Yang, J., & Jana, S. (2019). Deepxplore: Automated whitebox testing of deep learning systems. *Communications of the ACM*, 62(11). <https://doi.org/10.1145/3361566> , diakses 23 April 2024.
- Sabbrina, A., Sufa, A. oktavia, Ritonga, D. putra, Siregar, E. R. sari, & . N. (2023). Pengenalan Konsep Dasar Dan Penggunaan Database Manajemen Sistem (DBMS). *Jurnal Sains Dan Teknologi (JSIT)*, 3(3). <https://doi.org/10.47233/jsit.v3i2.907> , diakses 4 Maret 2024.
- Singh, A., Sharma, S., & Singh, S. (2016). Android Application Development using Android Studio and PHP Framework. *International Journal of Computer Applications Recent Trends in Future Prospective in Engineering & Management Technology*, 975–8887 , diakses 20 Mei 2024.
- Solikhin, I., Sobri, M., & Saputra, R. (2018). Sistem Informasi Pendataan Pengunjung Perpustakaan (Studi kasus : SMKN 1 Palembang). *Jurnal Ilmiah Betrik*, 9(03). <https://doi.org/10.36050/betrik.v9i03.40>
- Sutiah, S., & Supriyono, S. (2021). Software testing on e-learning Madrasahs using Blackbox testing. *IOP Conference Series: Materials Science and Engineering*, 1073(1). <https://doi.org/10.1088/1757-899x/1073/1/012065> , diakses 23 April 2024.
- Syahrudin, A. N., & Kurniawan, T. (2018). Input dan Output pada Bahasa Pemrograman Python. *Jurnal Dasar Pemrograman Python STMIK*, June 2018, diakses 6 Mei 2024.
- Thomas Hamilton. 2024. Path Testing & Basis Path Testing in Software (Examples), (Online), <https://www.guru99.com/basis-path-testing.html> , diakses 7 Mei 2024.
- Zain Ahmad Taufik dan Supriyanto. 2023. Implementasi Chatbot untuk Layanan Frequently Asked Question Akademik dengan Penggunaan Dialogflow. *Jurnal Saintekom : Sains, Teknologi, Komputer dan Manajemen*. Vol. 13 (1). 3 dan 4.