

RANCANG BANGUN LINGKUNGAN PEMROGRAMAN PYTHON DENGAN METODE *CHATBOT* PADA *PLATFORM* WHATSAPP

Syariz Nadhifa Aruriansyah^{1*}, Anis Cherid², Hadi Santoso³, Dwi Anindyani Rochmah⁴

^{1,2,3,4}Fakultas Ilmu Komputer, Universitas Mercu Buana
Email: ¹*41519010029@student.mercubuana.ac.id, ²anis.cherid@mercubuana.ac.id,
³hadi.santoso@mercubuana.ac.id, ⁴dwi.anindya@mercubuana.ac.id

(*: *Corresponding Author*)

(Naskah masuk: 14 Agustus 2023, diterima untuk diterbitkan: 18 Agustus 2023)

Abstrak

Kemajuan *Chatbot* telah banyak diadopsi oleh berbagai perusahaan sebagai dukungan bagi layanan pelanggan. *Chatbot* sendiri merupakan sebuah teknologi yang mensimulasikan atau menirukan percakapan layaknya seperti manusia melalui pesan teks. Pada dasarnya, pengembangan sistem *Chatbot* pada penelitian ini bertujuan untuk memudahkan pengguna jika ingin belajar pemrograman dalam bahasa pemrograman Python. Meskipun sudah tersedia aplikasi berbasis web untuk tujuan ini, yaitu Google Colab, antarmuka aplikasi web ini sulit untuk digunakan jika melakukan pemrograman pada layar yang berukuran kecil yang terdapat pada perangkat *mobile*. Dengan latar belakang masalah tersebut, penulis membangun sistem *Chatbot* yang dapat membantu pengguna dalam melakukan pemrograman Python melalui Whatsapp. Eksekusi terhadap kode program yang dikirimkan melalui Whatsapp diimplementasikan dengan cara memanfaatkan fitur bahasa pemrograman Python untuk mengeksekusi kode program dalam bentuk *string*. Selain itu, penulis menyediakan fitur manajemen kode program untuk banyak pengguna, sehingga sistem *Chatbot* ini dapat dipergunakan oleh banyak pengguna sekaligus.

Kata kunci: *Chatbot, Python, Whatsapp, Teknologi, Software*

DESIGN AND BUILD A PYTHON PROGRAMMING ENVIRONMENT USING THE *CHATBOT* METHOD ON THE WHATSAPP PLATFORM

Abstract

Chatbot advances have been widely adopted by various companies as support for customer service. Chatbot itself is a technology that simulates or imitates human-like conversations via text messages. Basically, the development of the Chatbot system in this research aims to make it easier for users if they want to learn programming in the Python programming language. Although a web-based application is available for this purpose, namely Google Colab, this web application interface is difficult to use when programming on the small screen found on a mobile device. With this problem as a background, the author built a chatbot system that can help users carry out Python programming via WhatsApp. Execution of program code sent via WhatsApp is implemented by utilizing the Python programming language feature to execute program code in string form. Apart from that, the author provides a program code management feature for many users, so that this Chatbot system can be used by many users at once.

Keywords: *Chatbot, Python, Whatsapp, Technology, Software*

1. PENDAHULUAN

Perkembangan teknologi mengalami pertumbuhan yang disebabkan banyaknya pengguna komputer dan *smartphone* yang semakin meluas, digunakan dalam berbagai keperluan, seperti kepentingan perusahaan, pendidikan, dan kesehatan. kemajuan *Chatbot* telah banyak diadopsi oleh berbagai perusahaan sebagai dukungan bagi layanan pelanggan. *Chatbot* merupakan aplikasi yang diciptakan untuk memfasilitasi komunikasi antara manusia dan mesin [1].

Chatbot atau *chatting robot* adalah sebuah aplikasi yang mampu mensimulasikan percakapan dengan pengguna. Hal ini dilakukan dengan memberikan tanggapan yang sesuai dengan pola pertanyaan atau pesan yang telah dituliskan oleh pengguna. Pembuatan *Chatbot* melibatkan penyusunan kumpulan data dasar yang terdiri dari berbagai pola pertanyaan dan jawaban. Data ini akan digunakan oleh *Chatbot* untuk merespon pertanyaan-pertanyaan yang diajukan oleh pengguna dengan jawaban yang sesuai [2].

Pada dasarnya, pengembangan sistem *Chatbot* dibuat karena untuk memudahkan pengguna jika ingin melakukan pemrograman. Bahasa pemrograman python telah menyediakan sebuah aplikasi berbasis web, yaitu Google Colab. Google Colab dapat diakses melalui perangkat apapun seperti *desktop* dan *mobile*. Namun, untuk tampilan Interface pada Google Colab jika diakses menggunakan *mobile* tampilannya sangat kecil sehingga cukup sulit untuk menggunakan Google Colab di perangkat *mobile*. Maka dari itu, untuk memudahkan pengguna dalam melakukan pemrograman, peneliti membuat sistem *Chatbot* yang dapat digunakan untuk melakukan pemrograman bahasa Python. Kemudian untuk menggunakan sistem *Chatbot* ini dapat diakses melalui media *chatting* yaitu Whatsapp.

Dengan permasalahan tersebut, penulis dapat membuat rancangan sistem *Chatbot*. Dimana, sistem ini dapat membantu pengguna dalam melakukan pemrograman sehingga *User* tidak perlu menggunakan software tambahan karena *Chatbot* ini dapat diakses hanya melalui Whatsapp. Whatsapp merupakan sebuah aplikasi perpesanan yang tidak hanya sebagai media komunikasi [3]. Namun Whatsapp dapat dikembangkan sebagai media pembelajaran khususnya pemrograman interaktif. Perbedaan aplikasi pemrograman interaktif yang dibuat oleh penulis dengan aplikasi pemrograman lainnya yaitu mereka berbasis web yang dimana, aplikasi tersebut bisa diakses melalui *device* apapun seperti *desktop*, *mobile phone*, dan lain-lain. Jika aplikasi yang berbasis web diakses melalui *mobile phone*, Maka *User Interface* yang ditampilkan sangat kecil sehingga pengguna sulit untuk menggunakannya. Maka dari itu, penulis membuat aplikasi ini untuk memudahkan pengguna saat menggunakannya.

Selain itu penulis akan berusaha mengembangkan fitur pemrograman interaktif yaitu fitur yang menerima *Input User* ketika aplikasi sedang berjalan. Masalah ini sudah berhasil dipecahkan oleh berbagai *compiler* yang berbasis web. Dalam penelitian ini penulis akan berusaha membangun aplikasi *Chatbot* yang memiliki fitur manajemen kode program untuk banyak pengguna, sehingga sistem *Chatbot* ini dapat dipergunakan oleh banyak pengguna sekaligus.

2. METODE

2.1. Metode Pengembangan Sistem

Pada penelitian ini menggunakan metode pengembangan perangkat lunak berbasis model prototipe. Model prototipe ini dirancang secara terstruktur dan melibatkan beberapa tahap yang dilalui dalam proses pembuatannya. Apabila pada tahap akhir ditemukan bahwa sistem yang dibangun belum sempurna, maka dilakukan evaluasi ulang.

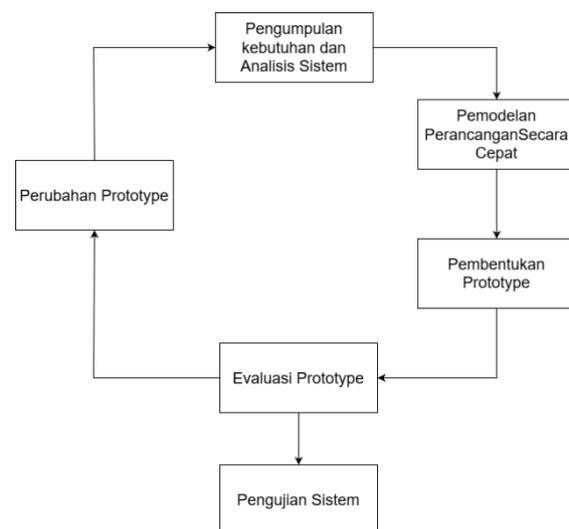
Tahapan dalam penelitian ini disesuaikan dengan model yang diimplementasikan dan meliputi langkah-langkah berikut:

1. Pengumpulan Kebutuhan dan Analisis Sistem

Tahap awal yang digunakan dalam penelitian ini adalah pengumpulan kebutuhan dan analisis sistem. Pada tahap ini, dilakukan pengenalan terhadap kebutuhan sistem. Dalam mengembangkan sistem *Chatbot* ini, diperlukan beberapa pendukungnya, seperti Python, Javascript, dan pendukung lainnya.

2. Pemodelan perancangan secara cepat

Tahap berikutnya adalah Pemodelan perancangan secara cepat, yang berfungsi sebagai patokan dalam pembuatan prototipe. Dengan tujuan untuk membuat prototipe yang cukup untuk memperlihatkan potensi sistem kepada pengguna dan mendapatkan umpan balik awal.



Gambar 1. Tahapan Penelitian

3. Pembentukan prototipe

Pada langkah ini, prototipe dibentuk berdasarkan rancangan model yang sebelumnya telah dibuat. Prototipe ini mungkin tidak memiliki semua fitur yang lengkap, tetapi harus mampu mencerminkan fungsionalitas inti yang diinginkan.

4. Evaluasi prototipe

Dalam tahap ini, penilaian dilakukan terhadap prototipe yang telah disesuaikan dengan persyaratan. Jika masih belum sesuai dengan persyaratan, maka langkah selanjutnya dapat dilakukan, yaitu melakukan perubahan prototipe tersebut.

5. Perubahan prototipe

Tahapan ini melibatkan pengguna dalam pengujian prototipe untuk menyempurnakan prototipe yang dibangun. Pengguna akan memberikan umpan balik tentang kegunaan, kekurangan, dan perbaikan yang dapat dilakukan agar menghasilkan *prototipe* yang sesuai dengan kebutuhan.

6. Pengujian Sistem

Tahap terakhir dari metode penelitian ini adalah pengujian sistem. Pengujian sistem dilakukan dengan menggunakan skenario *blackbox testing* dengan melihat input dan output pada sistem

Chatbot. Pada tahap ini pengujian sistem dilakukan untuk melihat akurasi sistem sistem yang telah dibuat sebelumnya guna memastikan sistem tersebut berfungsi dengan baik.

2.2. Library dan Komponen Perangkat yang Digunakan

Python merupakan salah satu bahasa pemrograman tingkat tinggi yang bersifat *interpreter*, *interactive*, *object oriented*, dan dapat beroperasi hampir di semua *platform* [4]. Kelebihan dari python ialah pengguna dapat memahami dan mempelajari bahasa pemrograman python tanpa perlu menyimpan *commands* yang diberikan kedalam sebuah berkas. Sehingga Python dapat memanfaatkan *interpreter* sebagai penerjemah agar pengguna dapat memberikan *command* secara interaktif.

Flask adalah *framework web* yang ditulis dengan python. Yang dimana dapat diklasifikasikan ke dalam *micro-framework* karena tidak memerlukan alat atau pustaka tertentu dan memiliki basis data bawaan [5]. Flask juga menyediakan *library* dan kumpulan kode program yang dapat digunakan untuk membangun sebuah aplikasi, tanpa harus mel akukannya dari awal [6]. Namun dukungan *extension flask* yang dapat menambah fitur aplikasi. Ekstensi ada untuk pemetaan relasional objek, validasi formulir, penanganan unggahan, berbagai teknologi autentikasi terbuka, dan beberapa alat terkait kerangka kerja umum. Ekstensi diperbarui jauh lebih sering daripada program inti Flask [7].

Waitress merupakan implementasi dari protokol WSGI (Web Server Gateway Interface) dan digunakan untuk menjalankan aplikasi web Python secara lokal atau untuk menyediakan aplikasi web yang dapat diakses melalui jaringan. Waitress dirancang untuk memberikan performa yang baik dan dapat menangani beban yang tinggi.

JSON (*JavaScript Object Notation*) adalah sebuah format yang digunakan untuk pertukaran dan penyimpanan data. JSON memiliki struktur yang sederhana dan terbaca oleh manusia serta mudah dipahami oleh mesin. JSON menggunakan sintaksis mirip dengan bahasa pemrograman JavaScript, yang terdiri dari pasangan nama-nilai (*key-value pairs*). Data dalam format JSON direpresentasikan dalam bentuk objek atau array.

Aplikasi *Chatbot* ialah sebuah perangkat lunak kecerdasan buatan yang diciptakan untuk berinteraksi dengan pengguna melalui teks. Terdapat berbagai macam bot yang dapat mengartikan dan merespon pesan manusia, walaupun pada dasarnya, bot hanya mampu mengidentifikasi kata kunci yang dimasukkan dan memberikan balasan berdasarkan kata kunci yang paling sesuai atau menggunakan pola kata yang serupa dari kumpulan data yang telah disimpan di *server* [8].

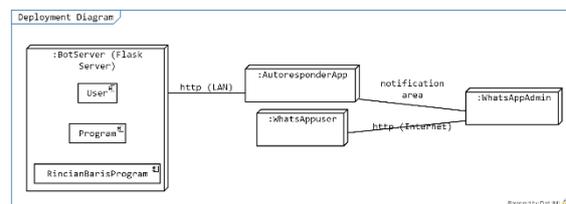
Whatsapp adalah layanan perpesanan populer untuk *smartphone* yang dimana pengguna dapat

berbagi pesan teks, suara, dan video antara orang atau dalam obrolan grup dalam format sinkron atau asinkron [9]. Namun tidak hanya sebagai media komunikasi, Whatsapp dapat dikembangkan dan dimodifikasi menjadi media pembelajaran yang efektif [3].

Autoresponder for Whatsapp merupakan sebagai API serta *server* dan juga wadah *database* yang akan memberikan perintah *machine learning* dengan konsep sistem kecerdasan buatan untuk membuat aplikasi berbasis *machine learning* secara otomatis melalui fasilitas *Chatbot*. *Autoresponder* akan berjalan pada model *Conversation Platform* yang dimana, mengirimkan informasi kepada pelanggan dalam bentuk pesan teks yang telah disimpan dalam *database* melalui nomor telepon yang terhubung ke internet dan juga pusat layanan Whatsapp [10].

3. HASIL DAN PEMBAHASAN

Metode yang digunakan pada sistem *Chatbot* dapat dijelaskan pada gambar berikut :



Gambar 2. Deployment Diagram Pada Sistem Chatbot

Pada *Deployment Diagram* berikut menjelaskan bagaimana sistem *Chatbot* bekerja. Pengguna yang sebagai *WhatsappUser* dapat berinteraksi dengan *Chatbot* dengan instruksi yang sudah diinput ke sistem. Apabila pengguna mengirimkan pesan, maka pesan tersebut akan diteruskan ke *WhatsappAdmin* melalui internet. Ketika *WhatsappAdmin* menerima pesan, kemudian *Autoresponder* akan menarik pesan tersebut melalui *notification area* pada perangkat *WhatsappAdmin* yang dimana *Autoresponder* merupakan sebagai Whatsapp API pada sistem *Chatbot* ini. Setelah *Autoresponder* menarik pesan tersebut, lalu diteruskan ke *server* sistem *Chatbot* melalui *http LAN* (Local Area Network). *Server* pada sistem *Chatbot* ini menggunakan *Flask Server* yang digunakan untuk menghubungkan ke *database*.

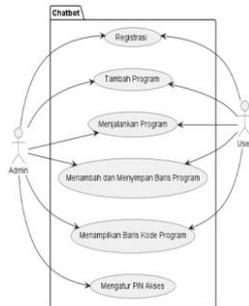
3.1. Usecase Aplikasi Chatbot

3.1.1. Usecase Diagram

Usecase Diagram merupakan bentuk diagram yang menggambarkan interaksi antara sistem informasi yang akan dihasilkan dengan satu atau beberapa pelaku [8]. *Usecase Diagram* memberikan gambaran *visual* tentang fungsionalitas utama serta interaksi antara pengguna dan sistem tersebut dapat membantu *Software Developer* untuk memahami kebutuhan sistem dan mengkomunikasikan fungsionalitas yang diharapkan dari sistem tersebut.

Pada gambar 3 *Usecase Diagram* ini menjelaskan fitur-fitur yang ada pada sistem *Chatbot*. Sistem

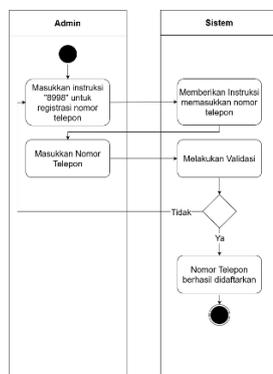
tersebut terdapat dua *actor*, yaitu Admin dan *User*. Pada dasarnya, Admin dan *user* dapat menggunakan fitur nya secara keseluruhan. Namun, berbeda halnya pada *user* yang dimana *user* tidak dapat menggunakan fitur konfigurasi PIN Akses. Karena, PIN Akses digunakan untuk mengelola *user* pada saat melakukan registrasi pada *user*.



Gambar 3. Usecase Diagram Aplikasi Chatbot

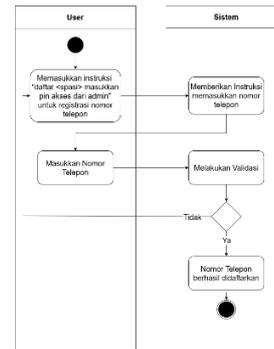
3.1.2. Activity Diagram

Activity Diagram merupakan jenis diagram yang digunakan dalam pemodelan sistem untuk membentuk *visual* dari alur kerja atau aktivitas serta menggambarkan urutan proses yang terjadi pada sebuah sistem. *Activity Diagram* berguna dalam pemodelan sistem karena dapat membantu pengembang sistem untuk memahami alur kerja sistem secara *visual* dan memudahkan dalam mengidentifikasi masalah atau kekurangan dalam sistem [11].



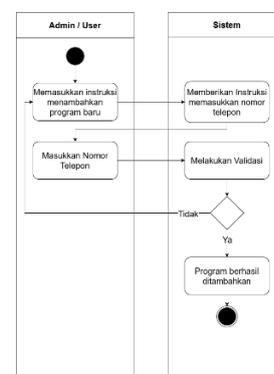
Gambar 4. Activity Diagram Registrasi Admin

Pada *Activity* ini, admin dapat melakukan registrasi pada nomor telepon admin ke sistem *Chatbot* agar admin dapat menggunakan sistem *Chatbot* ini.



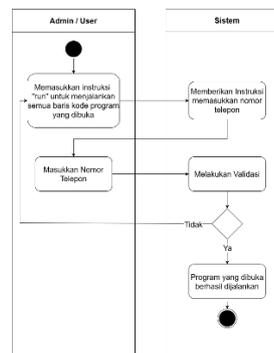
Gambar 5. Activity Diagram Registrasi User

Pada *Activity* ini, *User* dapat melakukan registrasi pada nomor telepon *User* ke sistem *Chatbot* agar pengguna dapat menggunakan sistem *Chatbot* ini.



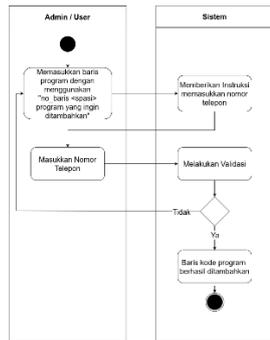
Gambar 6. Activity Diagram Tambah Program

Kemudian pada *Activity* ini pengguna dapat menambahkan program ke sistem *Chatbot*.



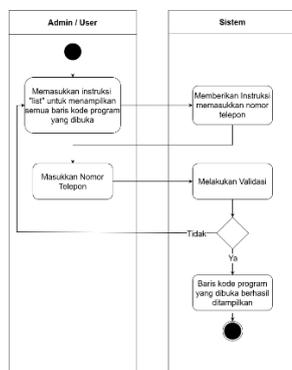
Gambar 7. Activity Diagram Menjalankan Program

Pada *Activity* ini pengguna dapat menjalankan program yang sudah dibuat yang dimana, pengguna dapat mengetahui hasil dari program tersebut.



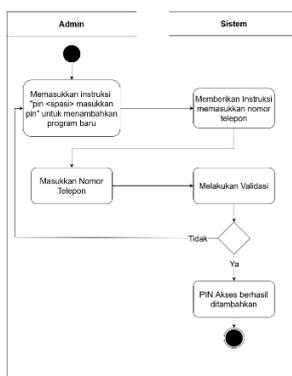
Gambar 8. Activity Diagram Menambah dan Menyimpan Baris Program

Pada *Activity* ini, pengguna dapat menambahkan baris program. Setelah ditambahkan, baris program tersebut disimpan secara otomatis pada sistem *Chatbot*.



Gambar 9. Activity Diagram Menampilkan Baris Kode Program

Pada *Activity* ini pengguna dapat menampilkan baris kode program yang sudah disimpan pada sistem *Chatbot* dimana, pengguna dapat melihat pada baris kode program tersebut apabila terdapat kesalahan dalam memasukkan kode program.



Gambar 10. Activity Diagram Konfigurasi PIN Akses

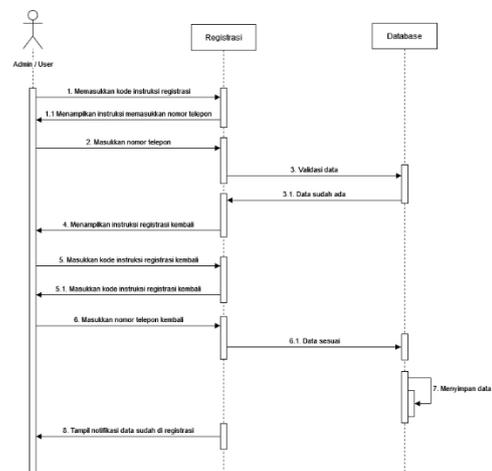
Pada *Activity* ini, Admin dapat membuat pin untuk memberikan akses pendaftaran pada *User*. Dengan adanya PIN Akses ini tidak semua *User* dapat melakukan pendaftaran.

3.1.3. Sequence Diagram

Sequence Diagram adalah sebuah gambaran yang digunakan untuk mengilustrasikan komunikasi antara

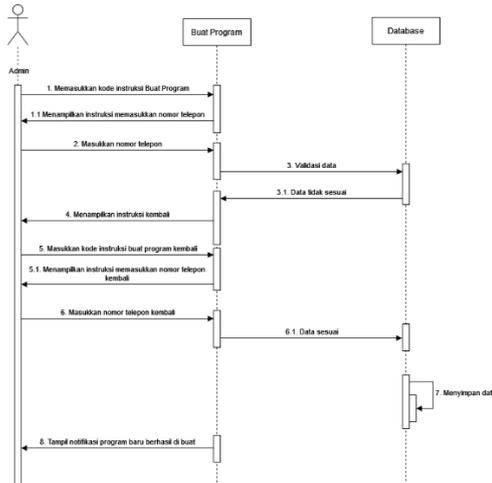
objek-objek di dalam suatu sistem [12]. Objek-objek tersebut digambarkan sebagai kotak vertikal yang diberi label dengan nama objek atau kelas yang bersangkutan. Pesan-pesan yang dikirim antara objek-objek tersebut digambarkan sebagai panah dengan label yang menunjukkan nama pesan beserta parameter-parameter yang dikirim. Panah tersebut menggambarkan urutan pesan yang dikirim dari satu objek ke objek lain.

Penggunaan *Sequence Diagram* sangat bermanfaat dalam analisis dan perancangan sistem yang kompleks, karena memberikan visualisasi yang jelas mengenai interaksi antara objek-objek. Diagram ini membantu dalam memahami alur pesan dan kerja sama antara komponen-komponen sistem, serta mengidentifikasi ketergantungan dan hubungan antara objek-objek tersebut.



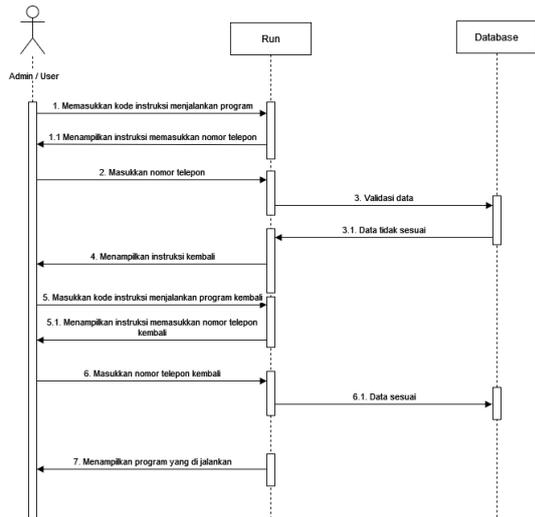
Gambar 11. Sequence Diagram Registrasi

Pada *Sequence* ini menjelaskan bagaimana alur sistem *Chatbot* dapat melakukan proses registrasi agar pengguna dapat menggunakan sistem *Chatbot* ini.



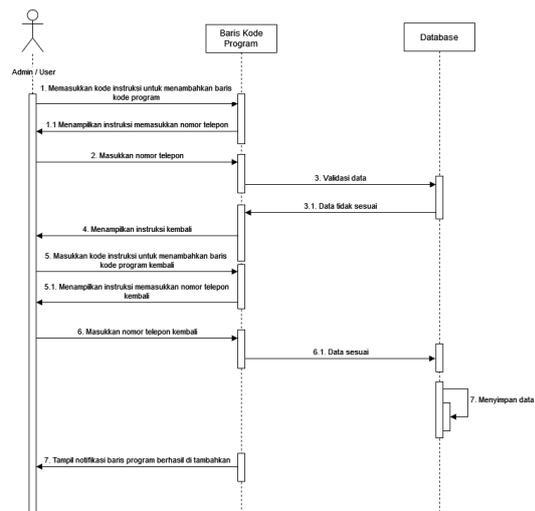
Gambar 12. Sequence Diagram Tambah Program

Pada *Sequence Diagram* ini menjelaskan bagaimana sistem *Chatbot* menambahkan program agar pengguna dapat menambahkan program tersebut.



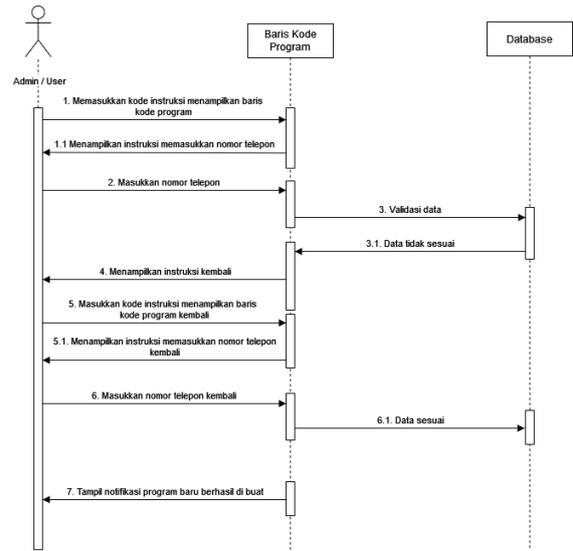
Gambar 13. Sequence Diagram Menjalankan Program

Pada *Sequence Diagram* ini menjelaskan bagaimana sistem *Chatbot* agar pengguna dapat menjalankan program yang sudah dibuat sehingga, pengguna dapat mengetahui hasil dari program tersebut.



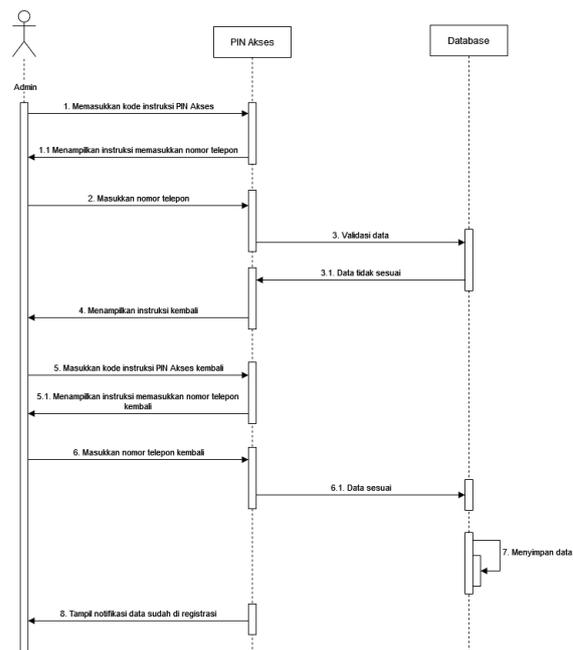
Gambar 14. Sequence Diagram Menambahkan dan Menyimpan Baris Program

Pada *Sequence Diagram* ini menjelaskan bagaimana sistem *Chatbot* dapat menambahkan baris program. Setelah ditambahkan, baris program tersebut disimpan secara otomatis pada sistem *Chatbot*.



Gambar 15. Sequence Diagram Menampilkan Baris Kode Program

Pada *Sequence Diagram* ini menjelaskan bagaimana sistem *Chatbot* dapat menampilkan baris kode program yang sudah disimpan pada sistem *Chatbot* dimana, pengguna dapat melihat pada baris kode program tersebut apabila terdapat kesalahan dalam memasukkan kode program.



Gambar 16. Sequence Diagram Konfigurasi PIN Akses

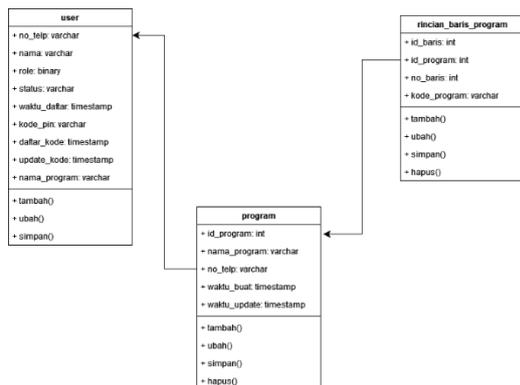
Pada *Sequence Diagram* ini, admin dapat membuat pin untuk memberikan akses pendaftaran pada *User*. Dengan adanya PIN Akses ini tidak semua *User* dapat melakukan pendaftaran.

3.1.4. Class Diagram

Class Diagram adalah jenis diagram yang ditujukan dalam pemodelan sistem untuk menggambarkan struktur sistem atau perangkat lunak [11]. Diagram ini memvisualisasikan struktur tetap

dari suatu sistem berbasis objek dengan menggambarkan kelas-kelas yang ada dalam sistem tersebut, atribut-atribut yang dimiliki oleh kelas-kelas tersebut, dan juga hubungan serta interaksi antara kelas-kelas tersebut.

Class Diagram dapat digunakan untuk memodelkan struktur tetap dalam pemrograman berbasis objek, yang pada akhirnya untuk mempermudah pemahaman tentang hubungan antar kelas dan struktur keseluruhan dari sistem. Diagram ini umumnya diterapkan pada tahap perancangan perangkat lunak sebelum dilakukannya implementasi.



Gambar 17. Class Diagram Aplikasi Chatbot

3.2. Implementasi Algoritma

Chatbot adalah program perangkat lunak otomatis yang berinteraksi dengan manusia. *Chatbot* dapat dikategorikan sebagai *Natural Language Processing*, yang dimana merupakan suatu cabang dalam bidang kecerdasan buatan yang berfokus pada pengolahan bahasa alami [13][14].

Cara kerja dari sistem *Chatbot* ini hanya memerlukan kata kunci teks yang telah disediakan, yang dimana akan memproses kata kunci tersebut untuk memberikan jawaban yang sesuai. Kemudian lingkup pertanyaan yang dapat diajukan kepada *Chatbot* sudah ditentukan sebelumnya agar tidak melampaui batas tersebut [15].

Adapun implementasi *Chatbot* dengan menggunakan whatsapp, sebagai berikut :

Pada gambar di bawah ini menjelaskan bagaimana sistem menghitung jumlah kata dalam suatu teks. Pada sintaks ini deklarasi fungsi diberi nama *count_words*. Kemudian pada *count_words* memiliki satu parameter yaitu *sentence* yang berfungsi untuk menghitung jumlah kata. Jadi, pada saat memanggil fungsi *count_words(sentence)* dan memberikan kalimat sebagai argumen, fungsi ini akan menghitung dan mengembalikan jumlah kata dalam kalimat tersebut.

```
def count_words(sentence):
    pattern = r"\b\w+\b" # Pola untuk mencocokkan kata-kata
    matches = re.findall(pattern, sentence)
    return len(matches)
```

Gambar 18. Fungsi Count Word

Kemudian untuk memproses jumlah kata pada suatu instruksi diperlukan nya nilai *variable* nomor

telepon dan nilai pada *variable* kode guna menjalankan instruksi tersebut. Jika memanggil *count_word(kode)*, maka jumlah kata pada kalimat pada *variable* kode akan dihitung. Apabila terdapat hanya satu kata saja maka akan memanggil fungsi proses satu kata. Namun, apabila terdapat dua kata maka akan memanggil fungsi proses dua kata.

```
def proses_request():
    json_data = request.get_json()
    content = json_data['query']
    no_telp = content['sender']
    kode = content['message']
    word_count = count_words(kode)
    print(f"Kode : {kode}\nNomor Telepon : {no_telp}", file=sys.stderr)

    if word_count == 1:
        reply_text = proses_satu_kata(kode, no_telp)
    elif word_count >= 2:
        reply_text = proses_dua_kata(kode, no_telp)

    data = {"replies": [
        {"message": reply_text}
    ]}
    json_string = json.dumps(data)
    return json_string
```

Gambar 19. Fungsi Proses Request

Setelah menghitung jumlah kata pada suatu kalimat, maka akan memproses dari tiap kata tersebut. Pada kasus ini peneliti mengimplementasikan proses satu kata yang dimana apabila terdapat kata *help* pada *variable* kode maka akan menampilkan isi dari *variable response*.

```
if kode == "help":
    if checkRole(no_telp) == True:
        response = "Berikut ini adalah instruksi yang dikawal oleh chatbot:\n\n" +
            "buatprogram instruksi untuk mulai membuat program baru\n" +
            "tampilprogram instruksi untuk menampilkan daftar program yang pernah dibuat\n" +
            "hapusprogramopsionalnomor_urut: instruksi untuk menghapus program dengan nomor urut pada tampil program\n" +
            "revisiopsionalnomor_urut: instruksi untuk mengedit program dengan nomor urut pada tampil program\n" +
            "list: instruksi untuk menampilkan baris program yang sudah dibuat\n" +
            "run: instruksi untuk menjalankan program yang sedang dibuat"
    else:
        return response
return "Maaf, Nomor anda belum mendapatkan layanan chatbot"
```

Gambar 20. Proses Satu Kata

Namun berbeda halnya dengan proses dua kata. Pada fungsi proses dua kata akan mencocokkan pola dua kata dalam instruksi dengan menggunakan *regular expression*.

```
def proses_dua_kata(kode, no_telp):
    pattern = r"\b(\w+)\b\s+(\w+)\b"
    match = re.search(pattern, kode)
    if match:
        word1 = match.group(1)
        word2 = match.group(2)
    else:
        return "Maaf, ada kesalahan pada server"
```

Gambar 21. Proses Dua Kata

Regular expression terdapat pada *variable pattern* sehingga fungsi *re.search* akan mencari pola yang cocok dalam instruksi kode, maka akan disimpan pada *variable match*. Jika *variable match* terdapat kecocokan pada *word1* dan *word2* maka pencocokan berhasil. Jika tidak maka akan menampilkan pesan "maaf, ada kesalahan pada server".

Pada kasus ini peneliti mengimplementasikan proses dua kata yang dimana *variable word1* nya pin dan *variable word2* dimasukkan dengan angka, maka akan membuat pin baru pada *role admin*. Namun jika yang menggunakan instruksi pin *role* nya bukan *admin* maka akan menampilkan pesan "Maaf, hanya *admin* yang diizinkan mengubah pin".

```
if word1 == "pin":
    if checkRole(no_telp) == True:
        query = f"UPDATE user SET kode_pin = '{word2}', update_kode = '{datetime.datetime.now()}' WHERE no_telp = '{no_telp}'"
        cursor.execute(query)
        db.commit()
        return "Pin baru pendaftaran berhasil dibuat"
    else:
        return "Maaf, hanya admin yang diizinkan mengubah pin"
```

Gambar 22. Instruksi Proses Dua Kata

Kemudian algoritma yang dipergunakan untuk mengeksekusi kode program yang dikirimkan melalui Whatsapp adalah memanfaatkan fungsi bawaan dari bahasa pemrograman Python. Jika fungsi *exec()* dipanggil dengan menggunakan argumen berupa *string*, misalnya “print(“Hello”)”, maka instruksi tersebut akan dieksekusi sehingga pada *console* atau *standard output* akan ditampilkan tulisan “Hello”. Keluaran program pada *standard output* inilah yang dikirimkan sebagai respon dari *Chatbot* jika pengguna meminta untuk menjalankan (*run*) program.

```
codeOut = StringIO()
#json_obj = json.loads(json_str)
code = final
# capture output
sys.stdout = codeOut
s = ""
try:
    exec(code)
    if 'draw(' in code:
        print(link_image(no_telp))
        s = codeOut.getvalue()
except Exception as e:
    errorString = StringIO()
    traceback.print_exc(file=errorString)
    s = errorString.getvalue()
# restore stdout and stderr
sys.stdout = sys.__stdout__
sys.stderr = sys.__stderr__
codeOut.close()
return "*****" + s + "*****"
```

Gambar 23. Proses Menjalankan Program

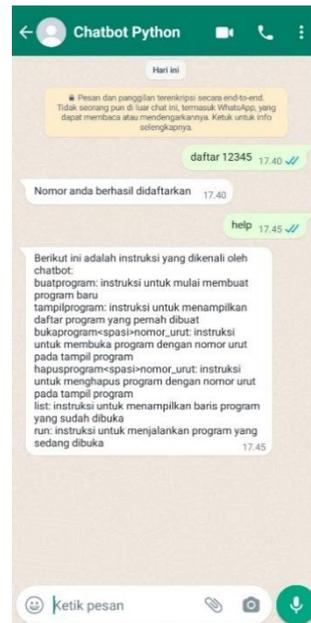
3.3. User Interface

Berikut ini merupakan tampilan *Interface* pada sistem *Chatbot* :



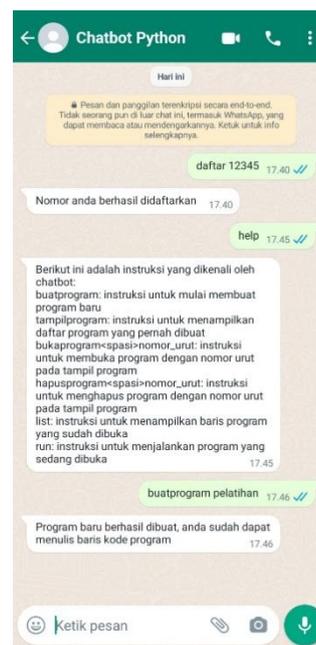
Gambar 24. Mendaftarkan pengguna ke sistem Chatbot

Pada gambar ini menjelaskan bagaimana cara melakukan proses registrasi agar pengguna dapat menggunakan sistem *Chatbot* ini.



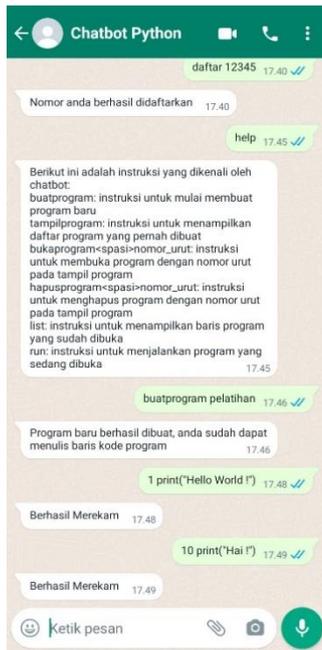
Gambar 25. Menampilkan instruksi dengan bantuan

Pada gambar ini menjelaskan bagaimana cara menggunakan instruksi pada sistem *Chatbot* agar pengguna dapat menggunakannya.



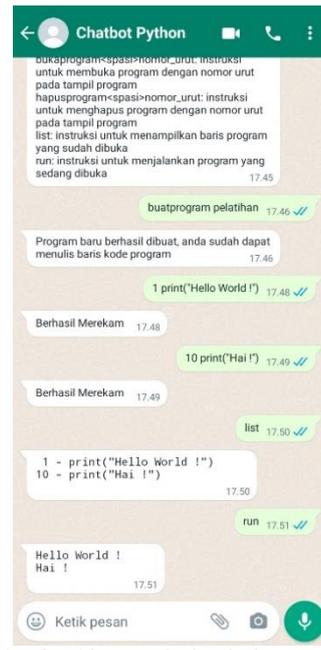
Gambar 26. Membuat program baru

Pada gambar ini menjelaskan bagaimana pengguna dapat menambahkan program tersebut ke sistem *Chatbot*.



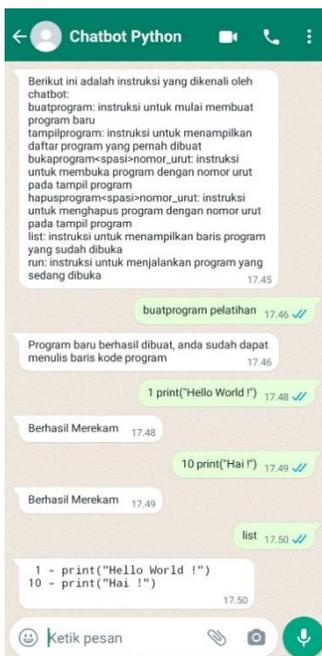
Gambar 27. Menambahkan baris program baru

Pada gambar berikut menjelaskan cara menambahkan baris kode program ke sistem *Chatbot*.



Gambar 29. Menjalankan kode program

Pada gambar berikut menjelaskan bagaimana cara menjalankan kode program sistem *Chatbot*.



Gambar 28. Menampilkan list baris kode program

Pada gambar berikut menampilkan semua baris kode program yang sudah ditambahkan ke sistem *Chatbot*.



Gambar 30. Menampilkan program yang sudah dibuat

Pada gambar berikut ini menjelaskan bagaimana sistem *Chatbot* dapat menampilkan program yang sebelumnya sudah disimpan pada sistem *Chatbot*.



Gambar 31. Membuka program

Pada gambar berikut menjelaskan bagaimana cara membuka suatu program yang sebelumnya sudah ditambahkan ke sistem *Chatbot*.



Gambar 32. Menghapus program

Pada gambar berikut menjelaskan bagaimana cara menghapus suatu program yang ada di sistem *Chatbot*.



Gambar 33. Menampilkan jika belum ada program yang dibuat

Pada gambar berikut menjelaskan apabila sistem *Chatbot* belum ada program yang dibuat oleh pengguna.

3.4. Pengujian Aplikasi

Berdasarkan hasil pengujian pada dua orang pengguna, di bawah ini disajikan respon yang diberikan atas pertanyaan terbuka pada kuesioner.

1. Pertanyaan 1: Apakah sistem *Chatbot* pembelajaran pemrograman Python memberikan manfaat untuk anda?

Jawaban responden 1: Ya, karena saya bisa membuat kode program python tanpa harus menghidupkan komputer *desktop* atau pun laptop.
 Jawaban responden 2: bermanfaat untuk pembelajaran pemrograman seandainya juga disertakan contoh-contoh program.

2. Pertanyaan 2: Apakah sistem *Chatbot* mudah digunakan untuk membuat kode program Python?
 Jawaban responden 1: Ya, karena proses menambahkan baris program dilakukan pada antarmuka Whatsapp yang mudah untuk diubah dan dibaca.

Jawaban responden 2: Cukup mudah digunakan, karena tersedia untuk mengubah dan menghapus baris program.

3. Pertanyaan 3: Apakah sistem *Chatbot* dapat mengeksekusi program python yang mengandung algoritma yang cukup rumit, misalnya algoritma sorting?

Jawaban responden 1: Ya, karena program dapat mengeksekusi *loop* sehingga dapat mengeksekusi algoritma sorting.

Jawaban responden 2: (responden 2 tidak mencoba fitur ini)

4. Pertanyaan 4: Apakah yang anda sarankan untuk perbaikan sistem *Chatbot* untuk pembelajaran pemrograman ini?

Jawaban responden 1: (responden 1 tidak memberikan saran)

Jawaban responden 2: Sebaiknya dilengkapi contoh-contoh program sebagai bahan pembelajaran.

3.5. Pengujian *Blackbox Testing*

Pengujian *Blackbox* merupakan pendekatan dalam menguji perangkat lunak di mana proses pengujian dilakukan tanpa memiliki pengetahuan mengenai bagaimana kode atau komponen internal perangkat lunak beroperasi. Dalam metode ini, fokus pengujian tertuju pada *input* yang diberikan kepada perangkat lunak serta *output* yang dihasilkannya, tanpa memerhatikan struktur internal atau implementasi kode yang digunakan. Tujuan utama dari pengujian *Blackbox* adalah memastikan bahwa perangkat lunak beroperasi sesuai dengan spesifikasi yang telah ditentukan dan memenuhi persyaratan fungsional yang telah digariskan, tanpa memperdulikan metode yang digunakan oleh perangkat lunak untuk mencapai hasil tersebut.

Tabel 1. Hasil Pengujian *Blackbox* Pendaftaran sistem *Chatbot*

Skenario Pengujian	Hasil yang diharapkan	Status
Melakukan pendaftaran dengan memasukkan PIN yang salah	Sistem akan menolak dan akan menampilkan “Maaf, PIN anda salah”	Berhasil
Melakukan pendaftaran tanpa memasukkan PIN	Sistem tidak mengenali instruksi	Berhasil
Melakukan pendaftaran dengan memasukkan PIN dengan benar	Sistem menerima dan menampilkan “Nomor anda berhasil didaftarkan”	Berhasil

Tabel 2. Hasil Pengujian *Blackbox* Membuat Program

Skenario Pengujian	Hasil yang diharapkan	Status
Melakukan pembuatan program tanpa memasukkan nama program	Sistem tidak mengenali instruksi	Berhasil
Melakukan pembuatan program tanpa melakukan pendaftaran	Sistem akan menolak dan akan menampilkan “Maaf, PIN anda salah”	Berhasil
Melakukan pembuatan program	Sistem akan menolak dan akan	Berhasil

Skenario Pengujian	Hasil yang diharapkan	Status
dengan memasukkan nama program yang sama dengan program sebelumnya	menampilkan “Maaf, nama program tersebut sudah ada. Silakan gunakan nama yang lain”	Berhasil
Melakukan pembuatan program dengan memasukkan nama program	Sistem menerima dan menampilkan “Program baru berhasil dibuat, anda sudah dapat menulis baris kode program”	

Tabel 3. Hasil Pengujian *Blackbox* Menampilkan Program

Skenario Pengujian	Hasil yang diharapkan	Status
Melakukan menampilkan program	Sistem menerima dan menampilkan semua daftar program yang sudah dibuat.	Berhasil

Tabel 4. Hasil Pengujian *Blackbox* Membuka Program

Skenario Pengujian	Hasil yang diharapkan	Status
Melakukan membuka program tanpa memasukkan nama program	Sistem tidak mengenali instruksi	Berhasil
Melakukan membuka program dengan memasukkan nama program yang tidak tersedia di sistem	Sistem akan menolak dan akan menampilkan “Program tidak ditemukan”	Berhasil
Melakukan membuka program dengan memasukkan nama program yang tersedia di sistem	Sistem menerima dan menampilkan “Program berhasil dibuka”	Berhasil

Tabel 5. Hasil Pengujian *Blackbox* Hapus Program

Skenario Pengujian	Hasil yang diharapkan	Status
--------------------	-----------------------	--------

Melakukan menghapus program tanpa memasukkan nama program	Sistem tidak mengenali instruksi	Berhasil
Melakukan menghapus program dengan memasukkan nama program yang tidak tersedia di sistem	Sistem akan menolak dan akan menampilkan “Program tidak ditemukan”	Berhasil
Melakukan membuka program dengan memasukkan nama program yang tersedia di sistem	Sistem menerima dan menampilkan “Program berhasil dihapus”	Berhasil

Tabel 6. Hasil Pengujian *Blackbox* Menampilkan Baris Kode Program

Skenario Pengujian	Hasil yang diharapkan	Status
Melakukan menampilkan baris kode program tanpa memasukkan kode program sebelumnya	Sistem menerima dan menampilkan “Belum ada baris yang dibuat”	Berhasil
Melakukan menampilkan baris kode program dengan memasukkan kode program sebelumnya	Sistem menerima dan menampilkan semua baris kode program yang sudah dibuat.	Berhasil

Tabel 7. Hasil Pengujian *Blackbox* Menjalankan Program

Skenario Pengujian	Hasil yang diharapkan	Status
Melakukan menjalankan baris kode program tanpa memasukkan kode program sebelumnya	Sistem menerima dan menampilkan “Belum ada program yang dibuka”	Berhasil
Melakukan menjalankan baris kode program dengan memasukkan	Sistem menerima dan menjalankan semua baris kode program	Berhasil

Skenario Pengujian	Hasil yang diharapkan	Status
kode program sebelumnya	yang sudah dibuat.	

Tabel 8. Hasil Pengujian *Blackbox* Menghapus Program

Skenario Pengujian	Hasil yang diharapkan	Status
Melakukan menghapus baris kode program tanpa memasukkan kode program sebelumnya	Sistem menerima dan menampilkan “Baris tidak ada”	Berhasil
Melakukan menghapus baris kode program dengan memasukkan kode program sebelumnya	Sistem menerima dan menghapus baris kode program.	Berhasil

4. KESIMPULAN

Berdasarkan hasil pengujian, dapat disimpulkan bahwa sistem *Chatbot* untuk menyediakan lingkungan pemrograman berbahasa Python telah berfungsi dengan baik. Sistem *Chatbot* dapat dipergunakan untuk merekam baris-baris kode program dan dapat mengeksekusi kode program yang keluarannya ditampilkan sebagai respon sistem *Chatbot* kepada pengguna.

DAFTAR PUSTAKA

- [1] R. Z. Ramadhani, H. Rusdianto, and V. Yahya, “Rancang Bangun Aplikasi Pusat Informasi Sekolah Dengan Penerapan *Chatbot* Menggunakan Aimi Berbasis Android Pada Smk Otomotif Al Husna Tangerang,” *Jurnal Teknik Informatika (JIKA) Universitas Muhammadiyah Tangerang*, vol. 3, no. 2, pp. 27–33, Oct. 2019, [Online]. Available: www.ellislab.com
- [2] R. Priambodo and T. Maya Kadarina, “Seminar Nasional Pengaplikasian Telematika (SINAPTIKA) Gamifikasi untuk Belajar Pemrograman menggunakan *Chatbot* pada Platform Discord,” *Seminar Nasional Pengaplikasian Telematika (SINAPTIKA)*, vol. 1, no. 1, 2021.
- [3] A. Husein, A. Sabiq, and M. I. Fahmi, “Mediating Quizzes as Assessment Tool through WhatsApp Auto-response in ELT Online Class,” *LANGKAWI Journal*, vol. 6, no. 2, 2020, [Online]. Available: <http://dx.doi.org/10.31>
- [4] B. A. Putra, “Rancang Bangun Sistem Absensi Berbasis Face Id di Bank Mandiri Sungai Rumbai dengan Bahasa Pemrograman Python,” *JVEIT (Journal of Vocational Education and Information Technology)*, vol. 3, no. 2, pp. 65–70, 2022, [Online]. Available: <http://ejournal.undhari.ac.id/index.php/jveit>

- [5] I. Arwani and B. Rahayudi, "Pengembangan *Chatbot* Yanies Cookies Untuk Pemesanan Kue Kering Berbasis Dialogflow," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 7, pp. 2121–2129, 2020, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [6] D. F. Ningtyas and N. Setiyawati, "Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request," *Jurnal Janitra Informatika dan Sistem Informasi*, vol. 1, no. 1, pp. 19–34, Apr. 2021, doi: 10.25008/janitra.v1i1.120.
- [7] N. Chauhan, M. Singh, A. Verma, A. Parasher, and G. Budhiraja, "Implementation of database using python flask framework," *International Journal of Engineering and Computer Science*, vol. 8, no. 12, pp. 24894–24899, Dec. 2019, doi: 10.18535/ijecs/v8i12.4390.
- [8] A. Santhy Toamain, "Rancang Aplikasi *Chatbot* Sebagai Virtual Asisten Dalam Pelayanan Pengguna Data Di Badan Pusat Statistik Provinsi Maluku," *Jurnal Teknologi Informasi*, vol. 7, 2021, [Online]. Available: <http://ejournal.urindo.ac.id/index.php/TI>
- [9] J. Walwema, "The WHO Health Alert: Communicating a Global Pandemic with WhatsApp," *J Bus Tech Commun*, vol. 35, no. 1, pp. 35–40, Jan. 2021, doi: 10.1177/1050651920958507.
- [10] M. Mufadhol, A. Wibowo, J. T. Santoso, J. Teguh, and Santoso.--D. Marketing, "Digital Marketing Techniques For Business Intelligence Systems Use Automated *Chatbot* Machine Learning," *PJAE*, vol. 17, no. 7, 2020.
- [11] I. M. Riyadhi, Intan Purnamasari, and Kamal Prihandani, "Penerapan Pola Arsitektur Mvvm Pada Perancangan Aplikasi Pengaduan Masyarakat Berbasis Android," *INFOTECH journal*, vol. 9, no. 1, pp. 147–158, May 2023, doi: 10.31949/infotech.v9i1.5246.
- [12] G. E. M. Pongantung, A. L. Kalua, and D. T. Salaki, "Sistem Informasi Pariwisata Daerah Kabupaten Minahasa Selatan Berbasis Web GIS," *Jurnal Ilmiah Sistem Informasi Akuntansi (JIMASIA)*, vol. 3, no. 1, pp. 24–32, Jun. 2023.
- [13] A. Kumar, P. Kumar Meena, and D. Panda, "CHATBOT IN PYTHON," *International Research Journal of Engineering and Technology*, 2019, [Online]. Available: www.irjet.net
- [14] D. Fajar Ramadhan, S. Noertjahjono, and J. Dedy Irawan, "Penerapan *Chatbot* Auto Reply Pada Whatsapp Sebagai Pusat Informasi Praktikum Menggunakan Artificial Intelligence Markup Language," 2020.
- [15] H. Eka Rosyadi, F. Amrullah, R. David Marcus, and R. Rahman Affandi, "Rancang Bangun *Chatbot* Informasi Lowongan Pekerjaan Berbasis Whatsapp dengan Metode NLP (*Natural Language Processing*)," *BRILIANT: Jurnal Riset dan Konseptual*, vol. 5, 2020, doi: 10.28926/briliant.