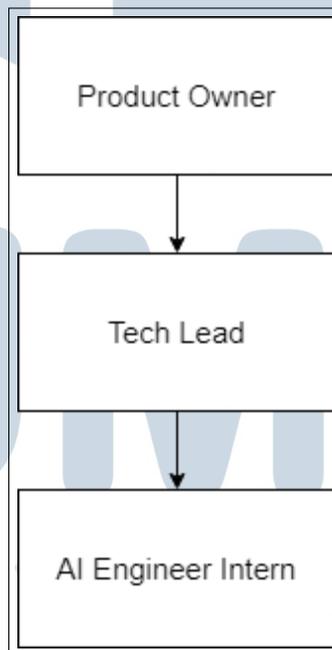


BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Pada proses kerja magang di PT. Siloam International Hospitals, penulis diberi tanggung jawab sebagai *AI Developer*. Selama proses kerja magang, Rio Gunawan selaku *Tech Lead Developer* dari *project* memberikan bimbingan dan pengawasan secara langsung.

Sebagai sarana komunikasi, para *developer* menggunakan aplikasi Whatsapp dan Teams. Tim mengadakan pertemuan setiap hari yang membahas tugas serta *insight* untuk menyelesaikan tugas tersebut dari *lead*. Jika proses pengerjaan sudah selesai, *developer* akan melakukan *request merge* untuk dicek kode serta hasilnya. Selain itu, *developer* juga menggunakan Postman untuk membagi API yang telah dibuat kepada tim *website* dan *mobile*.



Gambar 3.1. Kedudukan dan Organisasi Pelaksanaan Magang

3.2 Tugas yang Dilakukan

Tugas yang dilakukan sebagai *AI Engineer* pada proses magang di PT. Siloam International Hospitals adalah sebagai berikut:

1. Membuat API untuk *chatbot* dengan implementasi AI menggunakan *vector database Qdrant*
2. Melakukan *research* untuk kepentingan *chatbot* seperti *flow history* dan *embedding*
3. Membuat API CRUD untuk Doctor, QnA, dan Admin

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1, 3.2, dan 3.3.

Tabel 3.1. Pekerjaan yang dilakukan minggu ke 1 - 6 selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Melakukan setup repository dan mempelajari tentang Prompt AI Engineering. Kemudian membuat sistem chat room dari mengambil history yang dibagi - bagi per chatroom menggunakan <i>database SQLite</i>
2	Melakukan enhance pada tampilan <i>chatroom</i> dan mempelajari <i>vector store, embedding</i> dan <i>vector database</i> serta prompting dalam membuat <i>chatbot</i>
3	Mengimplementasi PostgreSQL pada <i>chatroom</i> , Memperbaiki <i>prompting</i> pada <i>chatbot</i> , dan melakukan <i>research</i> terhadap <i>workflow agent supervision</i> untuk <i>chatbot</i>
4	Melakukan <i>research database</i> yang bagus antara PostgreSQL dan mongodb. Setelah itu, melakukan <i>research</i> dalam menyimpan chat secara efisien dalam kasus ini menggunakan <i>custom checkpoint</i> pada MongoDB
5	Melakukan implementasi <i>custom checkpoint</i> menggunakan <i>database MongoDB</i> untuk <i>memory</i> dan <i>history</i> pada <i>chatbot internal</i>
6	Melakukan <i>improvement prompting</i> untuk <i>agent supervision</i> dan melakukan <i>improvement</i> pada <i>history</i> dan <i>memory chatbot</i>

Lanjut pada halaman berikutnya

Tabel 3.2. Pekerjaan yang dilakukan minggu ke 7 - 14 selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
7	Melakukan <i>improvement</i> pada tampilan Streamlit dan implementasi <i>temporary memory</i>
8	Membuat <i>temporary memory</i> untuk menyimpan key variable yang dikirim pengguna. Membuat pertanyaan dan <i>tool retriever</i> FAQ untuk diimplementasikan di workflow. Melakukan <i>research embedding model</i> yang akan digunakan dalam menyimpan pertanyaan FAQ pada <i>vector database</i>
9	Mengganti <i>embedding model</i> untuk simpan FAQ. Memperbaiki <i>tool</i> untuk <i>save</i> FAQ. Testing <i>tool save</i> FAQ dan <i>update</i> simpan <i>metadata</i> di <i>vector database</i> .
10	<i>Improve tool retriever</i> FAQ. Pindah ke <i>project chatbot</i> opd. Implementasi <i>database</i> MongoDB dan validasi JWT Token pada <i>chatbot</i> dokter. Implementasi <i>history</i> MongoDB pada <i>agent assistant</i> di <i>chatbot</i> dokter
11	<i>Update session</i> yang menandakan <i>chatroom</i> . Membuat beberapa API seperti <i>update token</i> dan <i>clear history</i> pada MongoDB dan Redis suatu <i>session</i> . Membuat <i>summary agent</i> untuk melakukan <i>summary</i> jika <i>chat history</i> melebihi batas. Melakukan <i>research</i> untuk <i>chatbot streaming</i>
12	Melakukan <i>improvement</i> dalam melakukan <i>summary</i> pada <i>memory agent</i> . Merancang ulang <i>flow chatbot session</i> dan membuat API untuk melakukan <i>get session id</i> menggunakan <i>id user</i> . Membuat API CRUD untuk FAQ
13	Melakukan revisi API CRUD FAQ. Membuat API CRUD untuk dokter dan admin. Implementasi Logging, CORS, dan Ratelimiter pada proyek <i>chatbot</i> opd. Melakukan <i>refactor code</i> pada bagian <i>controller</i>
14	Membuat Dockerfile untuk <i>chatbot</i> dokter dan melakukan <i>research</i> untuk membuat <i>chatbot</i> menghasilkan output <i>streaming</i>

Lanjut pada halaman berikutnya

Tabel 3.3. Pekerjaan yang dilakukan minggu ke 15 - 18 selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
15	Melakukan <i>hotfix</i> pada <i>flow session</i> pada <i>chatroom</i> . Membuat <i>collection</i> API untuk tim <i>website</i> dan <i>mobile</i> pada postman. <i>Refactor route</i> API dan penggunaan <i>cammalCase</i> ke <i>snack case</i>
16	Revisi <i>database chatroom</i> untuk menyimpan chat <i>history</i> dan <i>flow session</i> serta <i>handle expire session</i> . Menambahkan <i>pagination</i> pada API <i>get history</i> . Melengkapi dan revisi <i>collection</i> pada Postman
17	Melakukan <i>research</i> pada <i>custom checkpointer</i> . Mengganti <i>flow summary</i> pada <i>memory agent</i> . Memperbarui <i>collection</i> pada Postman
18	Melakukan <i>refactor code</i> dan <i>testing flow booking, reschedule dan cancel</i> janji temu dengan dokter. Membuat API <i>change profile</i> dan melakukan <i>improve</i> saat <i>handle payer</i> . Merapikan <i>route api</i> pada <i>chatbot</i> dokter

3.4 Requirements

Berikut adalah tugas - tugas yang telah dikerjakan beserta penjelasannya.

3.4.1 Get Chatbot Session

Pada bagian ini, pasien meminta id session untuk membuat *chatroom* dengan *chatbot*. Hal ini dilakukan agar dapat membedakan antara *chatroom* satu dengan yang lain. Ini menjadi bagian yang penting agar AI dapat mengerti yang sedang dibahas pada satu *chatroom* tanpa melibatkan *chatroom* yang lain.

3.4.2 Send Message Chatbot

Pada bagian ini, pasien mengirim pesan yang ditanyakan kepada AI. Pasien harus melakukan login dahulu agar kita dapat mengetahui siapa yang membuat janji temu. Namun, pasien juga dapat mengubah *profile* untuk membuat janji temu dengan dokter sebagai *profile* yang lain.

3.4.3 CRUD Doctor

Bagian ini, pengguna dapat mengelola data-data dokter yang nantinya digunakan untuk kebutuhan *chatbot*. Fitur ini memberikan pengguna akses untuk melihat, menambahkan, mengubah, dan menghapus data mengenai dokter. Selain itu, pengguna dapat dengan mudah memelihara.

Data pada dokter berisikan id, nama, *unique key*, rumah sakit tempat dokter bekerja, spesialis dan subspecialis. Dengan begini, pengguna dapat mencari dokter berdasarkan nama, rumah sakit, spesialis atau subspecialis. Dengan begitu, fitur ini meningkatkan efisiensi dalam mengelola data dokter dan mengurangi kemungkinan untuk terjadi kesalahan.

3.4.4 CRUD QnA

Bagian ini, pengguna dapat mengelola data-data QnA yang nantinya digunakan untuk kebutuhan *chatbot*. Fitur ini memberikan pengguna akses untuk melihat, menambahkan, mengubah, dan menghapus data mengenai QnA. Selain itu, pengguna dapat dengan mudah memelihara. Data pada QnA berisikan pertanyaan dan jawaban. Fitur ini membantu pengguna untuk mengelola dan menggunakan data QnA dengan efisien.

3.5 Perancangan

Pada saat melakukan diskusi antara tim *product* dan *developer*, tim *developer* diminta untuk membuat *chatbot* yang memungkinkan pasien untuk membuat janji temu dengan dokter sesuai jadwal yang diinginkan. *Chatbot* dapat merekomendasikan spesialis yang dituju berdasarkan keluhan yang diberikan pasien. Selain membuat janji temu, pasien juga dapat meminta ubah jadwal dan membatalkan janji temu yang sudah dibuat. Untuk menyamakan dengan *customer service*, *chatbot* dilengkapi dengan kemampuan untuk menjawab pertanyaan seputar rumah sakit. Tim *product* memberikan *requirements* yang dibutuhkan dalam *chatbot* dokter. Setiap minggu pada hari Selasa dan Kamis, tim *product* dan tim *developer* akan melakukan pertemuan untuk membahas *update requirements* dan *progress* pengerjaan.

Chatbot dapat memiliki *agent* untuk membantunya dalam menjalankan tugasnya [3]. *Chatbot* dokter memiliki 2 *agent* yaitu QnA Assistant dan Doctor Assistant. Tiap *agent* dilengkapi *tools* untuk membantu *agent* dalam menjalankan

tugasnya masing-masing. *Tools* adalah sebuah *function* yang digunakan oleh LLM yang dapat dilengkapi dengan nama dan deskripsi agar LLM dapat mengetahui kapan digunakan. *Chatbot* ini dibuat dengan framework *Retrieval-Augmented Generation* (RAG). RAG adalah teknik menggabungkan pengambilan informasi dan pembuatan teks agar respon yang diberikan lebih objektif [4]. AI dibuat dengan bantuan *retrieval tools* yaitu untuk mengambil data-data seperti nama dokter dan rumah sakit. Pengambilan data ini disimpan pada *vector database*. Objek berupa nama dokter dan rumah sakit disimpan dalam bentuk vektor. Dengan memanfaatkan data yang disimpan sebagai vektor, penggunaan pencarian kemiripan sangat efektif dalam operasi pengambilan data berbasis vektor [5]. Selain itu, *chatbot* dibantu oleh database MongoDB untuk menyimpan *history chat* dengan pasien agar *chatbot* mengerti konteks *chat* dengan pasien. *History* sudah diimplementasi *summary* jika pasien telah melakukan *chat* lebih dari 5 kali agar *history* yang diolah *chatbot* tidak menumpuk.

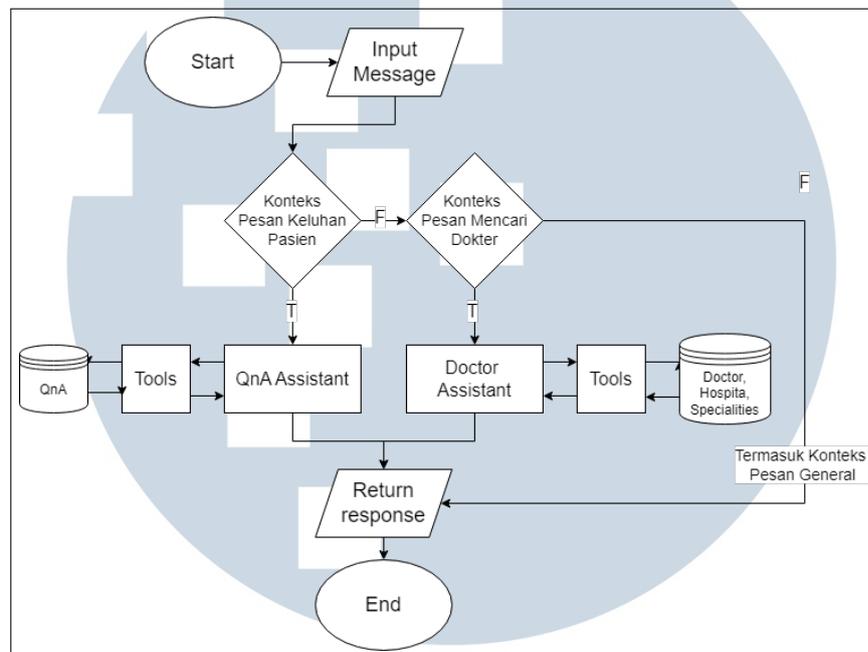
Terdapat 4 *collections* data yang disimpan pada Qdrant atau *vector database*. Data ini dilakukan embed ke dalam *vector database* menggunakan CRON setiap jam 12 malam agar data yang diperbarui. Data yang disimpan pada *vector database* digunakan oleh *chatbot* menggunakan *query* dengan filter kata atau *semantic search* atau filter dari variabel setiap data.

Colletions ini terdiri dari dokter, rumah sakit, spesialis, dan QnA. Data dokter terdapat 2856 dengan masing-masing 12 variabel. Data rumah sakit terdapat 41 rumah sakit dengan masing-masing 8 variabel. Data spesialis terdapat 33 dengan masing-masing 3 variabel. Terakhir data QnA terdapat 348 dengan masing-masing 2 variabel untuk menjawab pertanyaan seputar Siloam.

Dalam pembuatan *chatbot* dokter, terdapat 3 langkah untuk membuat janji temu dengan dokter. Pertama, *chatbot* dapat memberikan rekomendasi dokter dari keluhan pasien. Pasien juga dapat memberikan nama rumah sakit yang dia mau atau memberikan akses lokasi untuk menemukan rumah sakit Siloam terdekat. Kedua, *chatbot* memberikan rekomendasi *payer type* dari janji temu dokter yang pernah dibuat atau pasien dapat memilih ulang setelah memilih dokter. Ketiga, sesudah pasien memilih *payer type*, *chatbot* dapat membuat janji temu dengan dokter dan *chatbot* dapat memberikan kesimpulan tentang janji temu yang dibuat ke pasien. Diluar dari 3 langkah tersebut, *chatbot* dapat mengganti jadwal dan membatalkan janji temu yang sudah dibuat. Selain itu, *chatbot* dapat menggantikan *profile* sesuai keinginan pasien sebelum membuat janji temu dengan dokter. *Chatbot* ini hanya ada tampilan untuk pasien dan tidak ada tampilan admin atau dokter. *Chatbot* ini

juga tidak mempunyai *alert/warning*.

3.5.1 Flowchart *Workflow Chatbot Dokter*



Gambar 3.2. Flowchart *Workflow Chatbot Dokter*

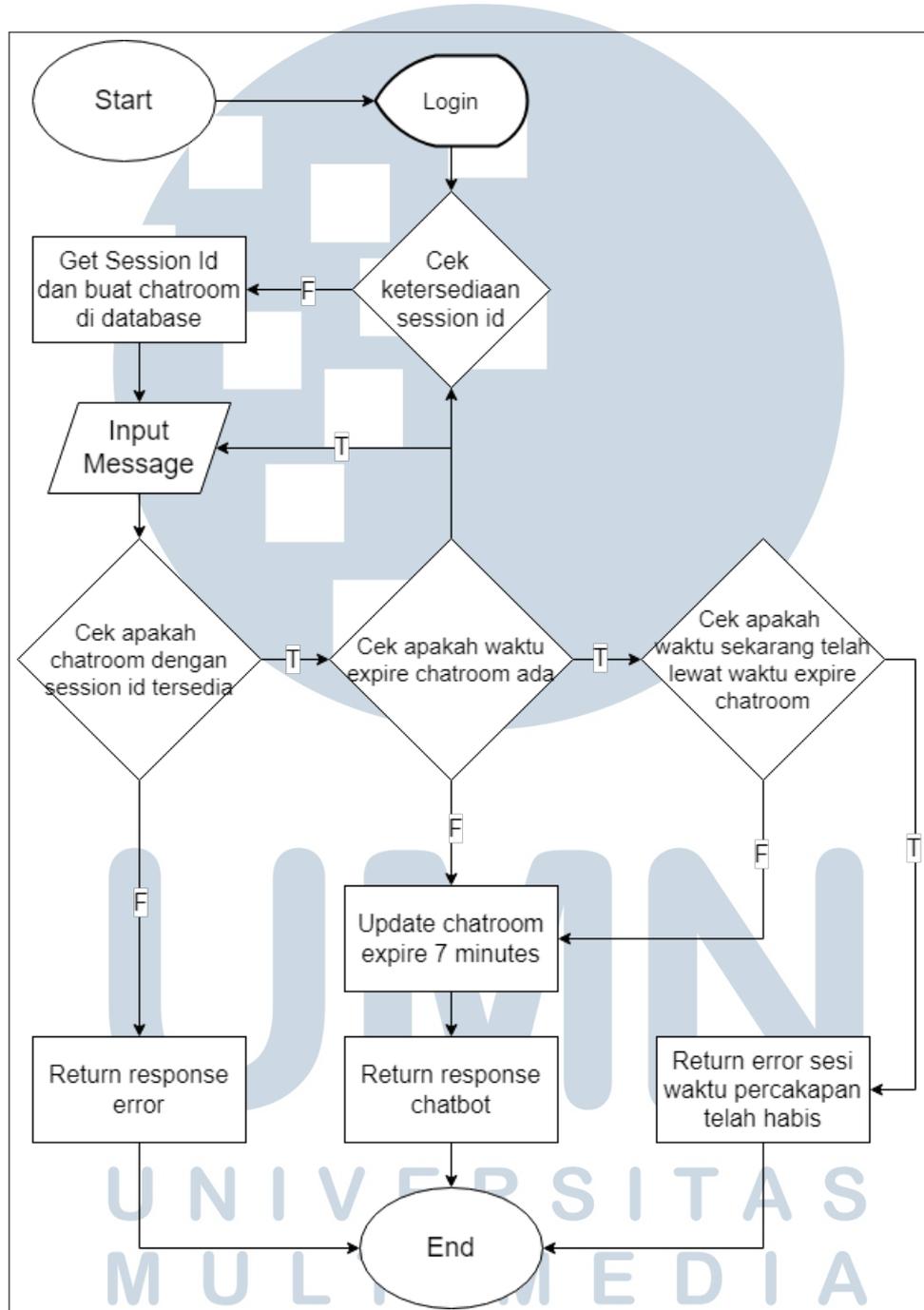
Flowchart pada gambar 3.2 berikut adalah alur *chatbot* dokter dalam menerima pesan dari pasien. Pada flowchart ini, pertama kali pesan dari pasien akan dianalisa dahulu. Jika pesan yang dikirim tidak berhubungan dengan keluhan atau membuat janji temu dengan dokter, maka *chatbot* akan menolak menjawab pertanyaan pasien. Jika pertanyaan tersebut berupa keluhan atau seputar rumah sakit, *chatbot* akan mengarahkan ke *agent* QnA Assistant. QnA Assistant hanya dilengkapi tools untuk mencari data QnA atau data seputar rumah sakit Siloam. QnA Assistant menggunakan *tools* untuk mencari data QnA pada *vector database* sesuai dengan pesan yang pasien kirim. *Tools* akan mengembalikan data QnA ke QnA Assistant dan menggunakan data dari *tools* untuk merespon kembali ke pasien.

Doctor Assistant ini dilengkapi dengan *tools* 9 *tools*. *Tools* yang dibuat penulis adalah *upcoming appointment tools* untuk melihat jadwal janji temu yang akan datang dan *reschedule appointment tools* untuk menjadwalkan janji temu yang akan datang. *Tools* ini menggunakan data dari *vector database* Qdrant. Data yang digunakan diperbarui setiap jam 12 malam dengan data dari tim *website* dan *mobile*.

Jika pasien memberikan pesan terkait membuat janji temu dengan nama dokter atau spesialis, maka *chatbot* akan mengarahkan ke *Doctor Assistant*. *Doctor Assistant* akan memberikan rekomendasi nama dokter tergantung detail yang diberikan oleh pasien. *Doctor Assistant* dapat memberikan rekomendasi dokter berdasarkan rumah sakit terdekat jika pasien membagikan lokasinya. Setelah pasien memberikan nama dokter, *Doctor Assistant* akan memberikan jadwal yang tersedia kepada pasien. Pasien akan memberikan jadwal yang diinginkan dan *Doctor Assistant* akan meminta konfirmasi kepada pasien. Pasien dapat melakukan cek ulang dokter, rumah sakit dan jadwal yang akan dibuat dan mengkonfirmasi kepada *chatbot*. *Chatbot* akan meminta *Doctor Assistant* untuk membuat janji temu dan memberikan ringkasan janji temu yang dibuat kepada pasien.



3.5.2 Flowchart Session

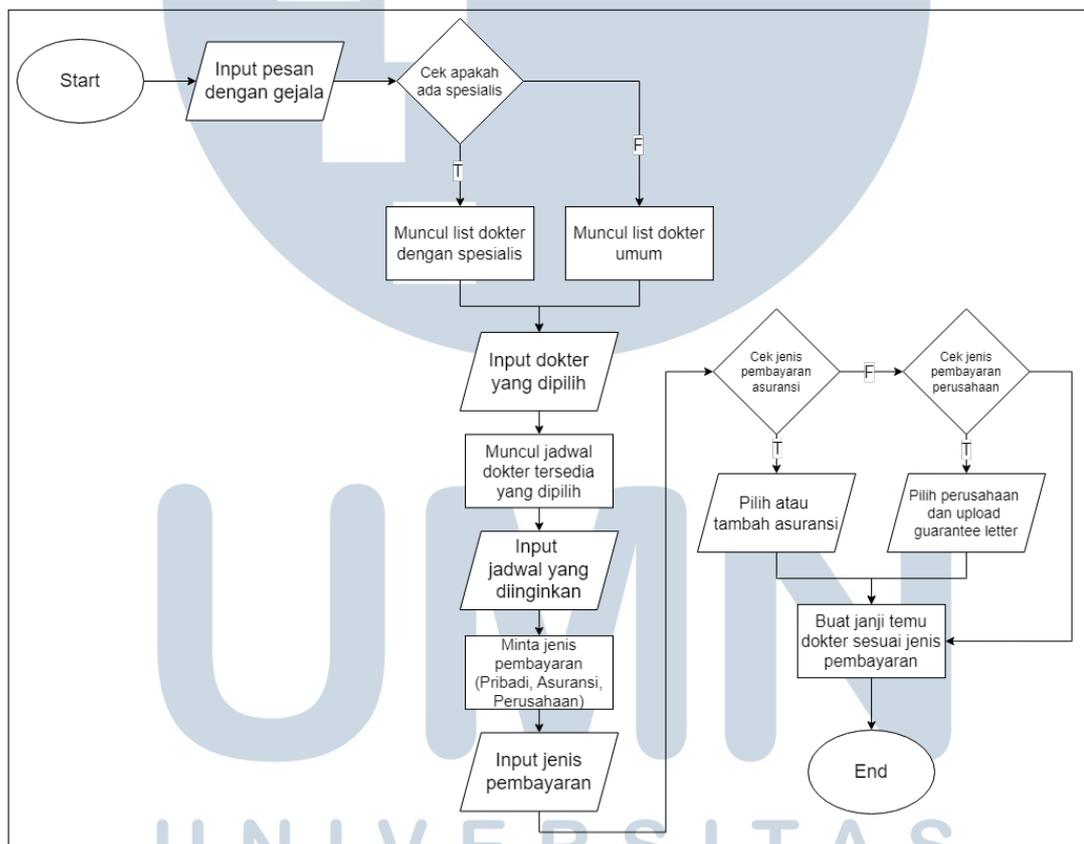


Gambar 3.3. Flowchart Session

Flowchart pada gambar 3.3 berikut adalah alur session untuk mengakses *chatbot* dokter. Pertama, pasien harus sudah melakukan login dahulu untuk mendapatkan informasi mengenai pasien yang menggunakan *chatbot* dokter. Jika

pasien belum pernah melakukan *chat*, maka sistem akan meminta id session untuk membuat *chatroom*. Pasien yang telah mendapatkan id session dapat mengirim pesan ke *chatbot*. Saat pasien mengirim pesan ke *chatbot*, sistem akan melakukan cek apakah session tersebut ada dan masa berlaku sessionnya. Jika session ada dan masa berlaku masih aktif, maka *chatbot* akan memberikan respon dari AI sebaliknya jika session tidak ada atau masa berlaku tidak aktif, maka *chatbot* akan memberikan respon bahwa sesi percakapan telah habis.

3.5.3 Flowchart *Booking* Dokter untuk Penyakit yang Diketahui

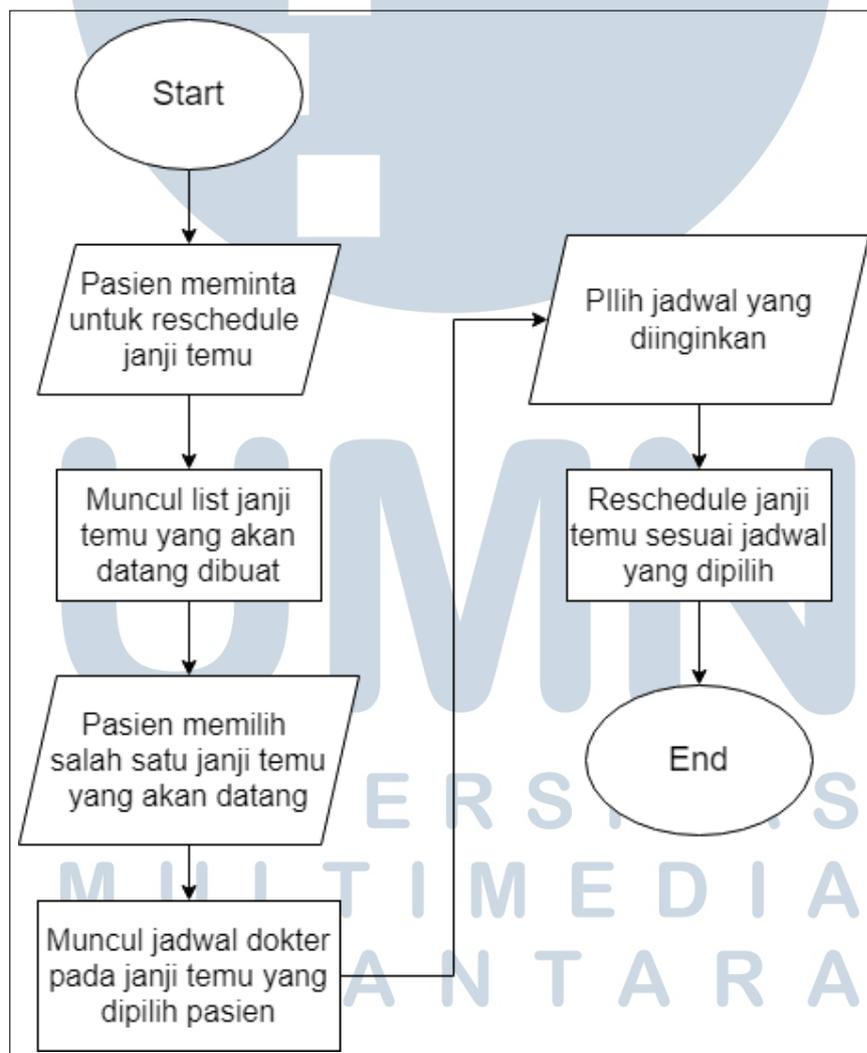


Gambar 3.4. Flowchart *Booking* Dokter untuk Penyakit yang Diketahui dengan *Chatbot*

Flowchart pada gambar 3.4 berikut adalah alur pasien dalam membuat janji temu dengan dokter melalui *chatbot*. Pertama pasien mau membuat janji temu dengan dokter dengan memberikan nama dokter, spesialis atau gejala penyakit. Jika pasien memberikan spesialis, *chatbot* menampilkan *list* dokter dengan spesialis yang diminta. Jika pasien tidak menyebutkan spesialis atau memberikan gejala,

Chatbot menampilkan *list* dokter umum. Hal ini karena terdapat limitasi pada *prompt* untuk mencegah meningkatnya token (*cost*) dan waktu yang dibutuhkan AI untuk mengolah informasi *prompt*. Kemudian, pasien memilih dokter dari *list* yang diberikan. *Chatbot* memberikan jadwal yang tersedia oleh dokter tersebut dan pasien memilih jadwal yang diinginkan. Terakhir, *chatbot* meminta jenis pembayaran yang pasien mau. Jika pasien sudah memberikan jenis pembayaran, *chatbot* akan membuat janji temu dan memberikan detail janji temu yang sudah dibuat kepada pasien.

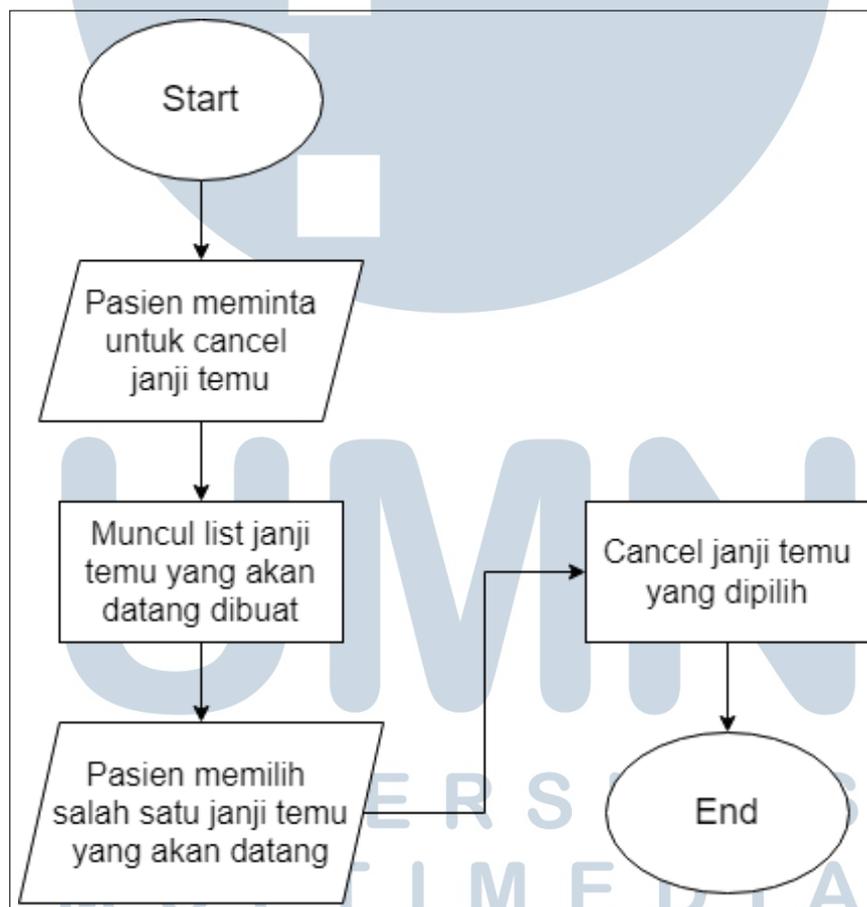
3.5.4 Flowchart *Reschedule* Janji Temu



Gambar 3.5. Flowchart *Reschedule* Janji Temu

Flowchart pada gambar 3.5 berikut adalah alur pasien dalam menjadwalkan ulang janji temu yang akan datang. Pertama pasien meminta *chatbot* untuk *reschedule* janji temu agar *chatbot* memberikan jadwal janji temu yang akan datang. Kemudian pasien memilih salah satu janji temu yang akan datang melalui nomor atau kode janji temu. *Chatbot* akan memberikan jadwal dokter pada janji temu yang dipilih oleh pasien. Pasien kemudian memilih jadwal yang diinginkan. Terakhir, *chatbot* menjadwalkan ulang janji temu dan memberikan detail janji temunya ke pasien.

3.5.5 Flowchart Cancel Janji Temu



Gambar 3.6. Flowchart Cancel Janji Temu

Flowchart pada gambar 3.6 berikut adalah alur pasien dalam membatalkan janji temu yang akan datang. Pertama, pasien meminta *chatbot* untuk membatalkan janji temu agar *chatbot* memberikan jadwal janji temu yang akan datang.

Kemudian, pasien memilih salah satu janji temu yang akan datang melalui nomor atau kode janji temu. Terakhir, *chatbot* membatalkan janji temu dan memberikan info detail janji temu yang dibatalkan.

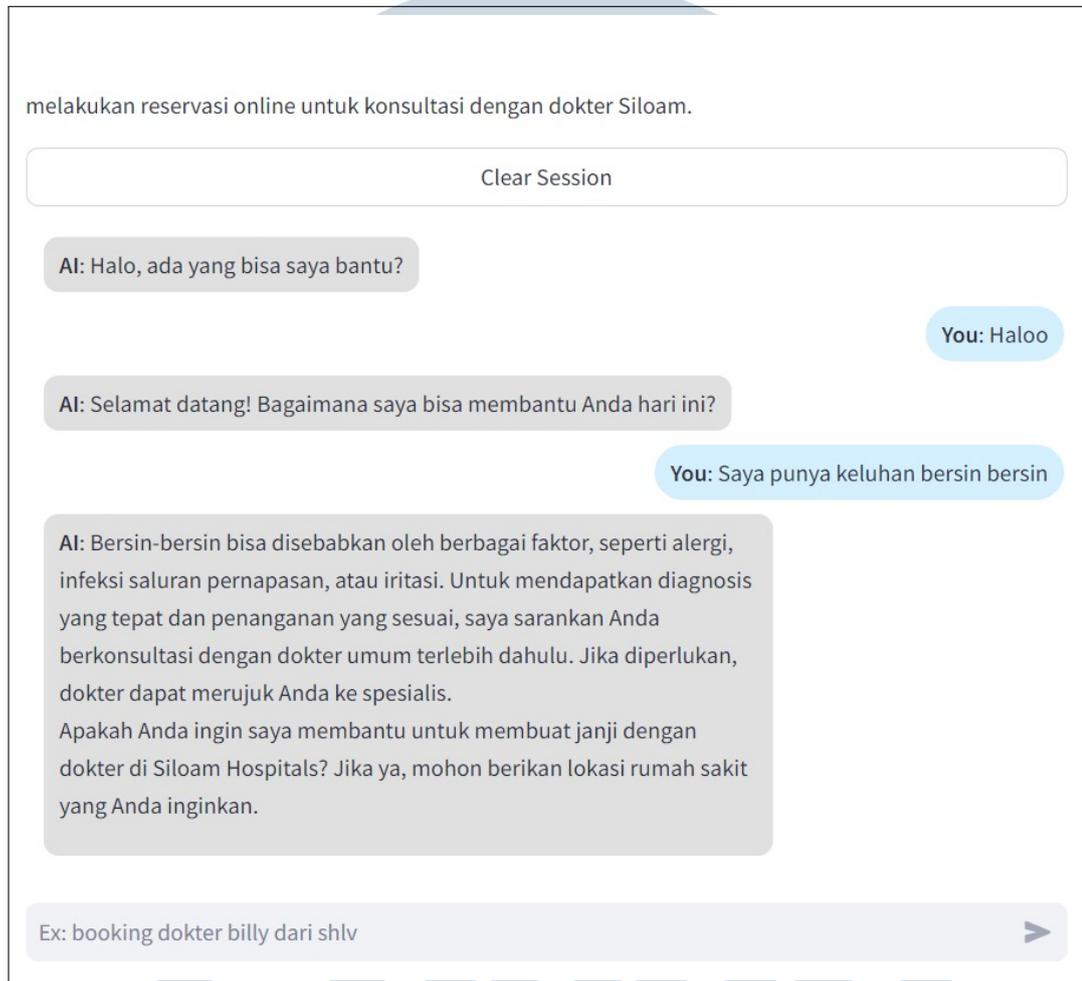
3.6 Implementasi

Dalam implementasi ini, *chatbot* yang telah dibuat akan diuji menggunakan aplikasi berbasis *web* yang dibuat menggunakan framework Streamlit. Aplikasi akan menguji *chatbot* dengan melakukan *requests* pada API yang telah dibuat. Penggunaan aplikasi bertujuan untuk mempermudah mengamati seluruh pembicaraan dengan *chatbot*.

Hasil implementasi ini menggunakan data *User Acceptance Testing* (UAT) untuk memastikan *chatbot* yang telah dibuat memenuhi kebutuhan *user*. Data UAT dianalisis untuk mengidentifikasi potensi *bug* serta untuk mengetahui apakah fitur yang dibuat telah berjalan sesuai ekspektasi. Hasil evaluasi yang diberikan akan digunakan untuk menyempurnakan *chatbot* sebelum diluncurkan ke *production*. Beberapa *bug* dan *fix* yang diuji pada UAT akan ditunjukkan pada lampiran.



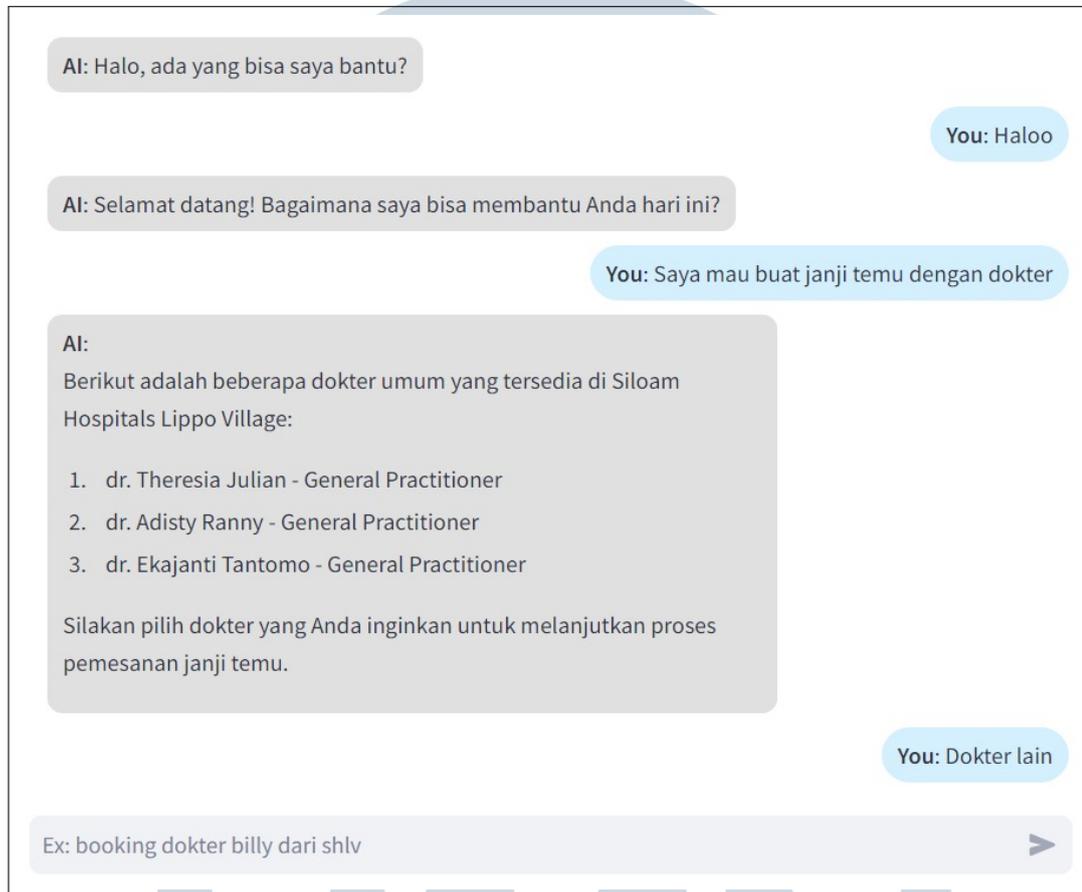
3.6.1 Contoh Konsultasi Keluhan



Gambar 3.7. Contoh Konsultasi Keluhan

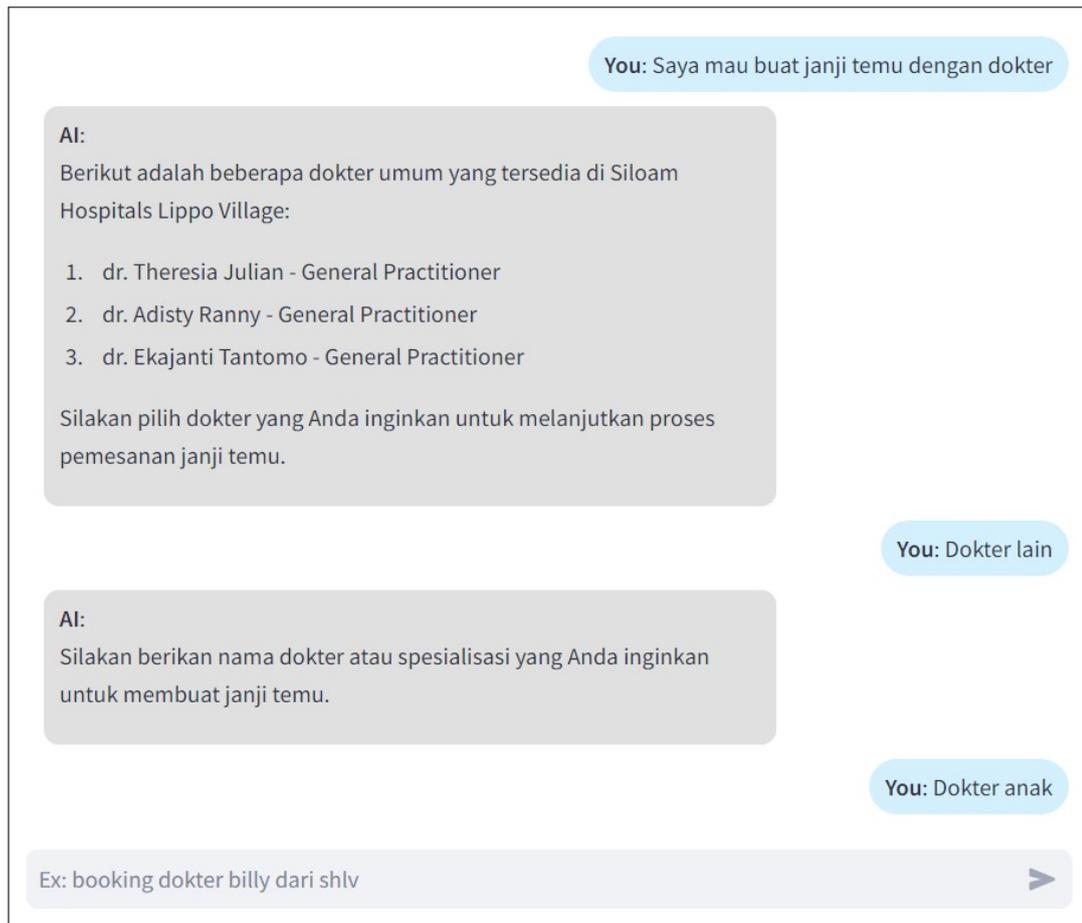
Pada gambar 3.7 adalah contoh konsultasi keluhan, pasien menyapa *chatbot* dahulu. Setelah itu, pasien memberikan keluhan tentang penyakitnya. *Chatbot* akan melakukan pencarian ke data QnA dan merespon balik ke pasien dengan data tersebut. Tidak hanya merespon dengan data, *chatbot* juga menawarkan untuk membuat janji temu dengan dokter di rumah sakit siloam.

3.6.2 Contoh Membuat Janji Temu Dokter



Gambar 3.8. Contoh Membuat Janji Temu Dokter 1

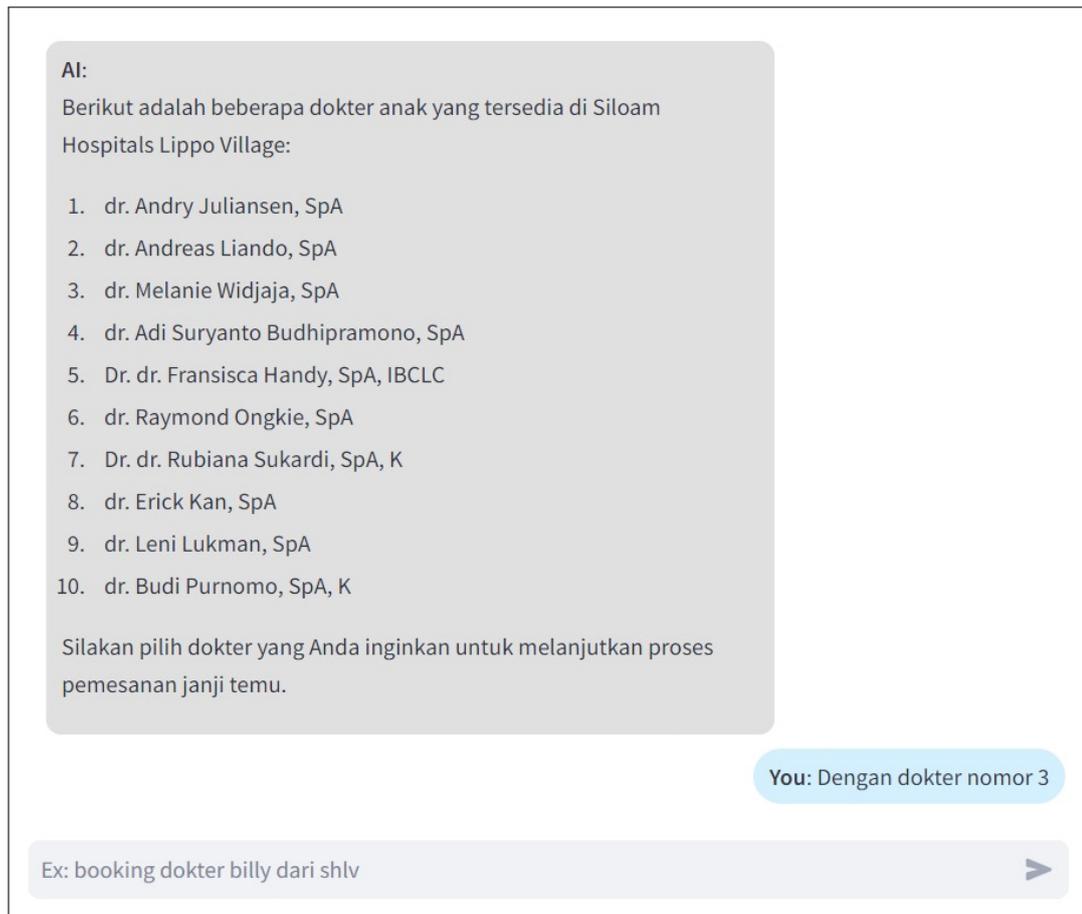
Pada contoh seperti di gambar 3.8 ini diawali dengan pasien menyapa *chatbot*. *Chatbot* membalas dengan sambutan dan pasien membalas dengan mau membuat janji temu dengan dokter. *Chatbot* memberikan rekomendasi dokter umum di rumah sakit Siloam Hospitals Lippo Village. Rekomendasi dokter yang *chatbot* berikan ada di Siloam Hospitals Lippo Village karena pasien menghidupkan lokasi dan rumah sakit Siloam terdekatnya adalah Siloam Hospitals Lippo Village. Namun, pada contoh ini, pasien memilih untuk membuat janji temu dengan dokter lain.



Gambar 3.9. Contoh Membuat Janji Temu Dokter 2

Sebelumnya pasien meminta untuk membuat janji temu dengan dokter lain. Pada gambar 3.9, *Chatbot* meminta pasien untuk memberikan spesialisasi atau nama dokter yang pasien inginkan. Pasien kemudian memberikan spesialisasi dokter anak.

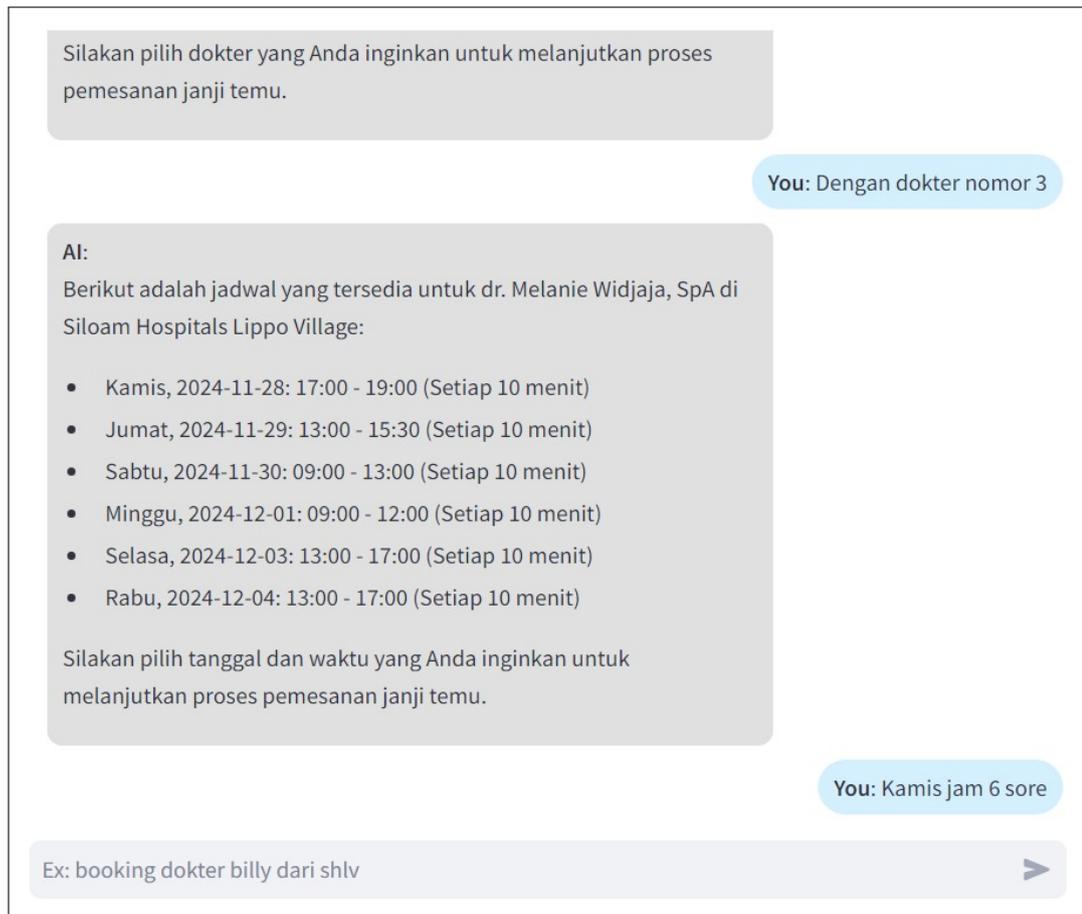
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.10. Contoh Membuat Janji Temu Dokter 3

Pada gambar 3.10 yang dimana sebelumnya pasien telah memberikan spesialisasi dokter yang dia inginkan yaitu dokter anak. *Chatbot* memberikan 10 pilihan dokter dengan spesialisasi anak di rumah sakit yang sama yaitu Siloam Hospitals Lippo Village. Pasien memilih dokter nomor 3 yaitu dr. Melanie Widjaja, SpA.

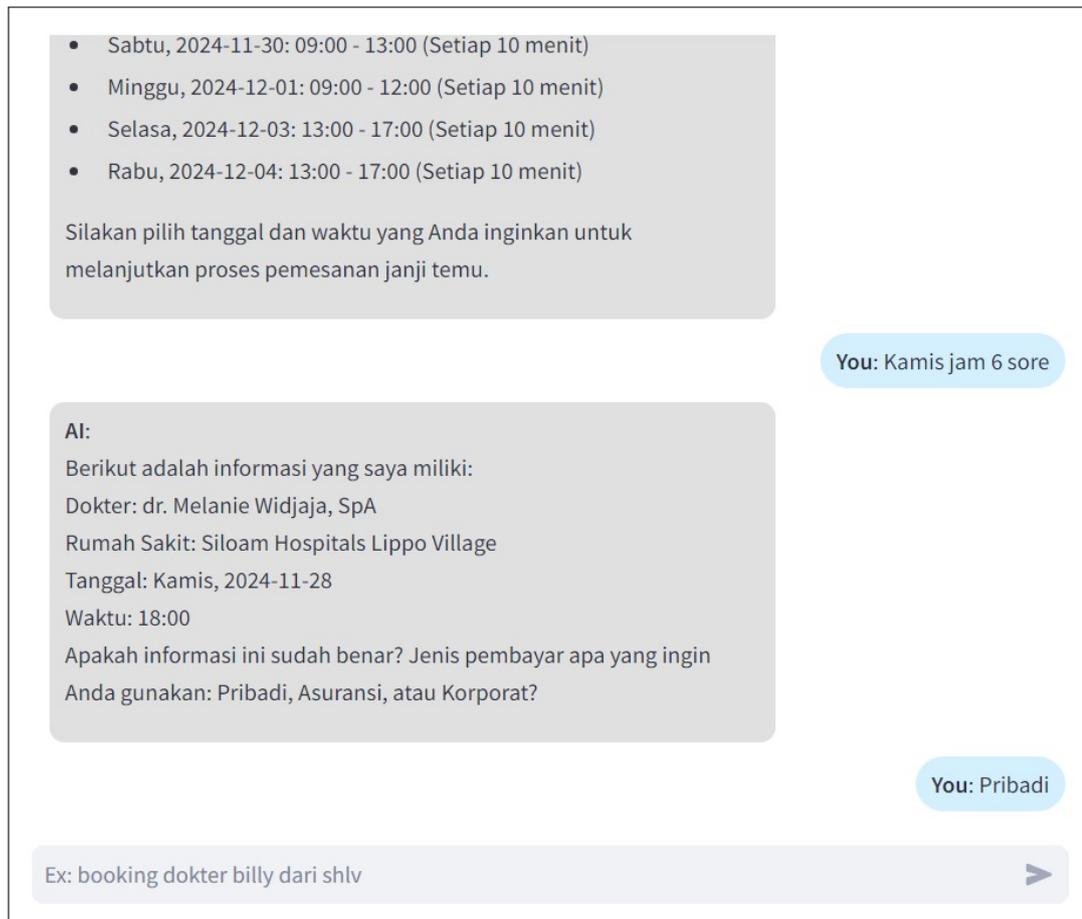
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.11. Contoh Membuat Janji Temu Dokter 4

Pada gambar 3.11, Pasien sebelumnya meminta untuk membuat janji temu dengan dr. Melanie Widjaja, SpA. *Chatbot* memberikan jadwal yang tersedia selama 1 minggu kedepan. Pasien meminta pada hari kamis tanggal 28 November 2024 pada jam 6 sore.

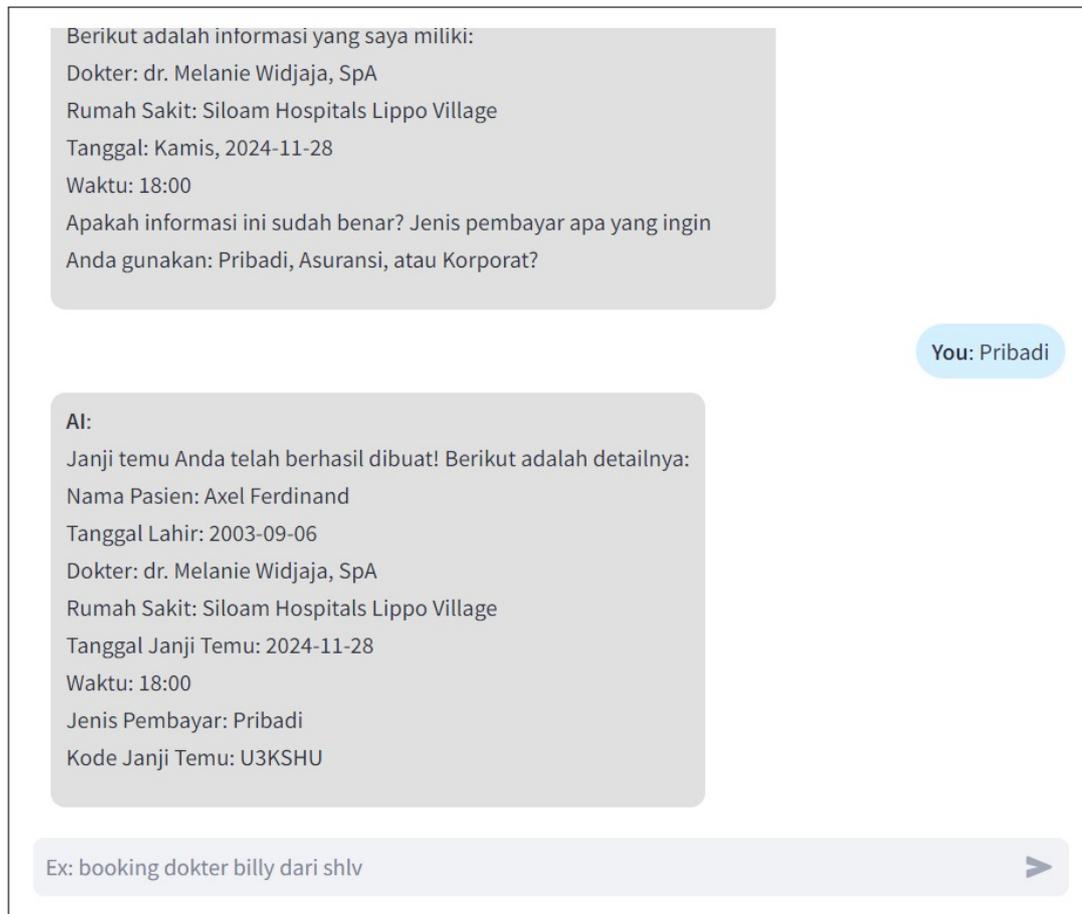
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.12. Contoh Membuat Janji Temu Dokter 5

Pada gambar 3.12, pasien sebelumnya telah memberi jadwal untuk janji temu yang dia inginkan. *Chatbot* kemudian memberikan rangkuman janji temu yang akan dibuat dari nama dokter, rumah sakit, tanggal, dan waktu janji temu. Selain itu, *chatbot* meminta jenis pembayaran untuk janji temu ini yaitu diantara pribadi, asuransi, atau korporat.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

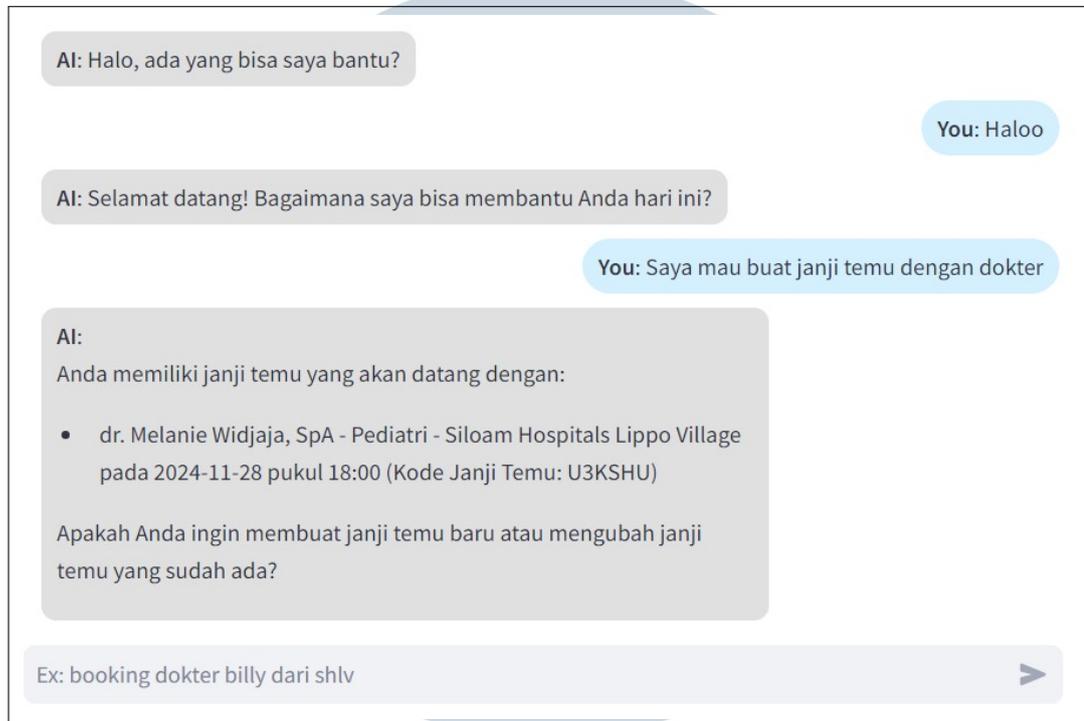


Gambar 3.13. Contoh Membuat Janji Temu Dokter 6

Pada gambar 3.13, pasien sebelumnya memberikan tipe pembayaran yang dia mau yaitu pribadi. *Chatbot* langsung melakukan cek dan pembuatan janji temu yang sudah diminta pasien. Terakhir, *chatbot* memberikan info terakhir bahwa janji temu yang dibuat telah berhasil dibuat. *Chatbot* juga memberikan nama pasien dan detail tentang janji temu yang telah dibuat beserta kode janji temu.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

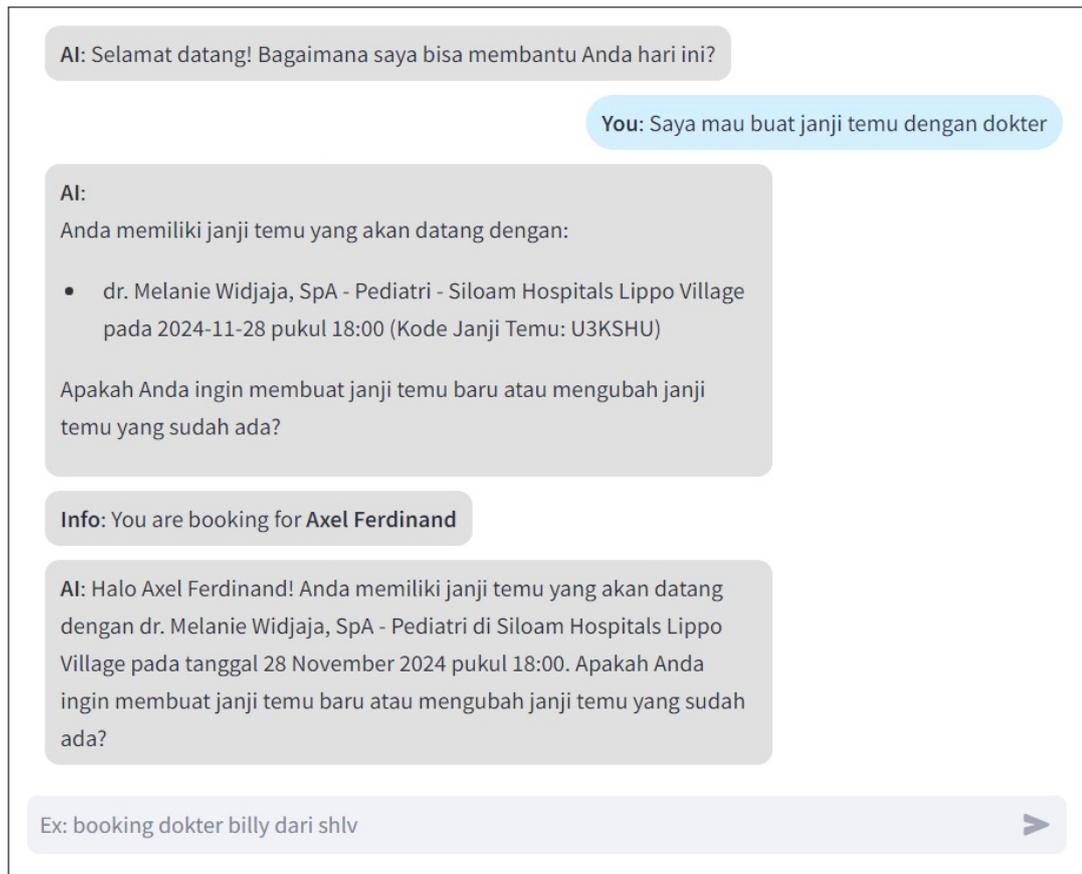
3.6.3 Contoh Ganti Profile Janji Temu Dokter



Gambar 3.14. Contoh Ganti Profile Janji Temu Dokter 1

Pada gambar 3.14, pasien mengawali percakapan dengan menyapa *chatbot*. Kemudian, pasien mau membuat janji temu dengan dokter. *Chatbot* memberikan janji temu sebelumnya karena pasien cenderung berobat dengan dokter yang ia temui sebelumnya untuk melakukan pemeriksaan kembali.

U I M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

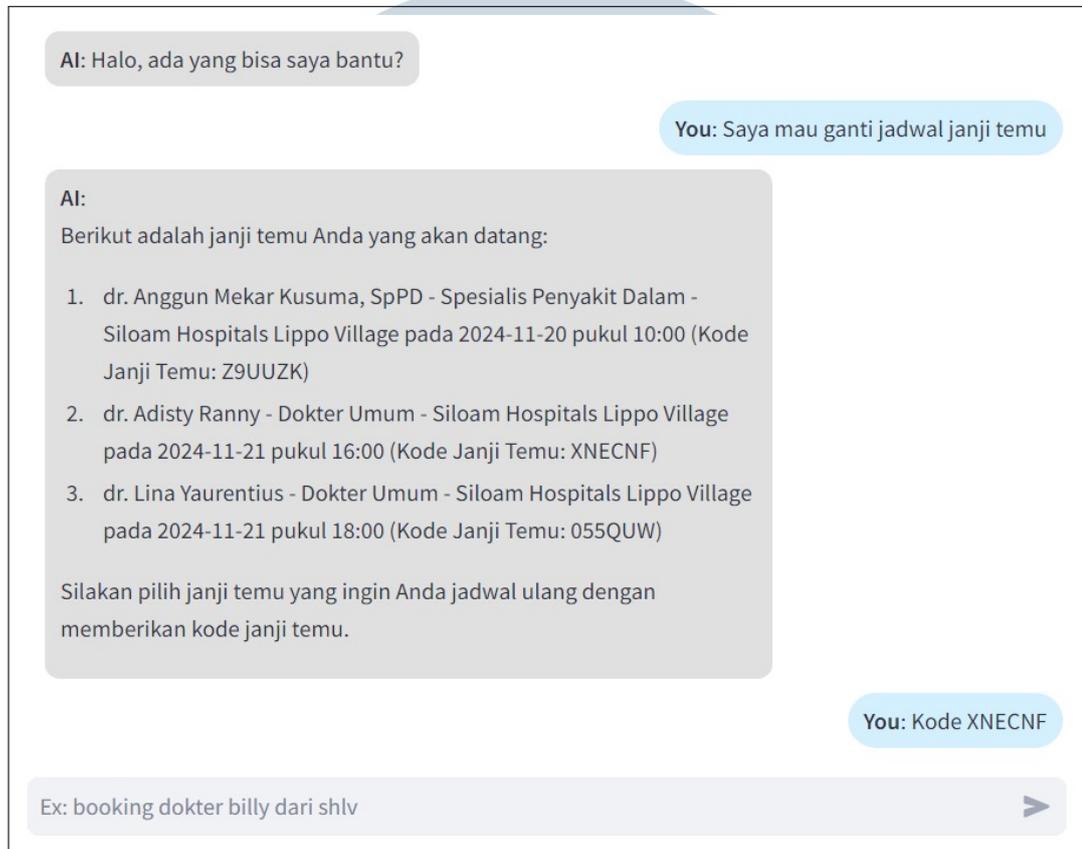


Gambar 3.15. Contoh Ganti Profile Janji Temu Dokter 2

Pada gambar 3.15, *chatbot* sebelumnya memberikan daftar janji temu yang akan datang. Kemudian pasien memilih *change profile* ke Axel Ferdinand. Untuk bagian memilih *change profile* akan tersedia saat diimplementasikan pada *website* dan *mobile*. *Change profile* akan menimbulkan sebuah info ke profile apa yang kita pilih. Setelah itu, *chatbot* akan merespon dengan memberikan merangkum percakapan sebelumnya.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

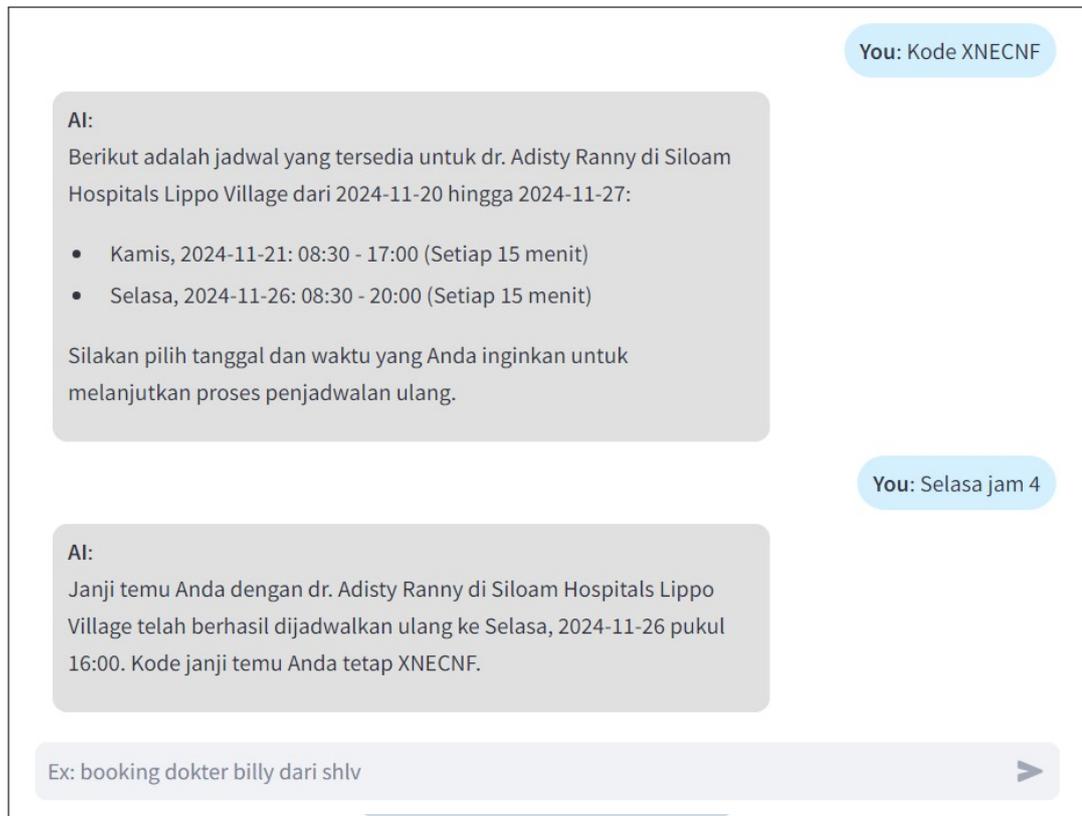
3.6.4 Contoh Ganti Jadwal Janji Temu Dokter



Gambar 3.16. Contoh Reschedule 1

Pada gambar 3.16 ini, pasien meminta *chatbot* untuk melakukan ganti pada jadwal janji temu yang akan datang. *Chatbot* memberikan list janji temu yang akan datang untuk pasien tersebut. Kemudian, pasien dapat memilih janji temu yang mau diganti jadwalnya dengan memberikan kode janji temu. Pasien memilih janji temu ke 2 dengan kode janji temu XNECNF.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

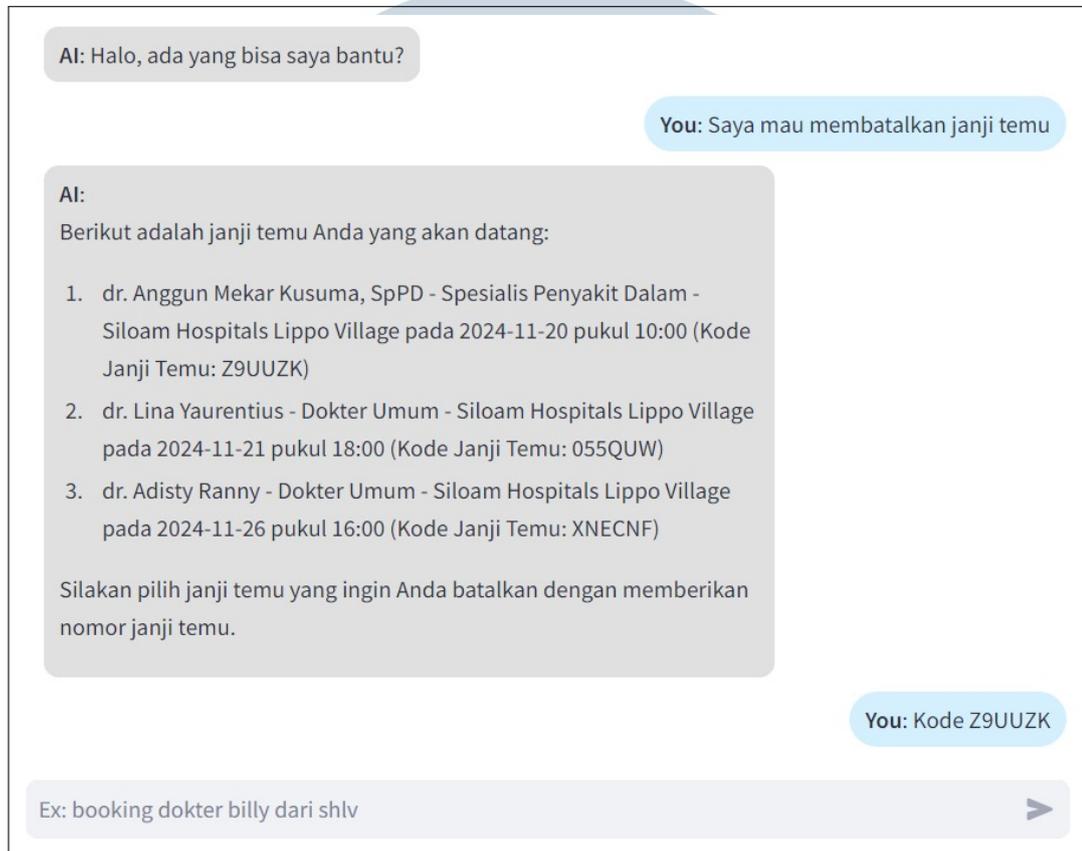


Gambar 3.17. Contoh Reschedule 2

Pasien telah memilih janji temu yang mau diganti jadwalnya. Pada gambar 3.17, *Chatbot* memberikan jadwal janji temu dokter tersebut untuk 1 minggu kedepan. Pasien memilih jadwal yang diinginkan yaitu hari Selasa tanggal 26 November 2024 jam 4 sore. *Chatbot* memberikan info bahwa janji temu berhasil diganti sesuai permintaan pasien tanpa mengubah kode janji temu. Jika jadwal janji temu tersebut sudah ada, *chatbot* akan memberikan info bahwa pergantian janji temu tidak dapat dilakukan.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

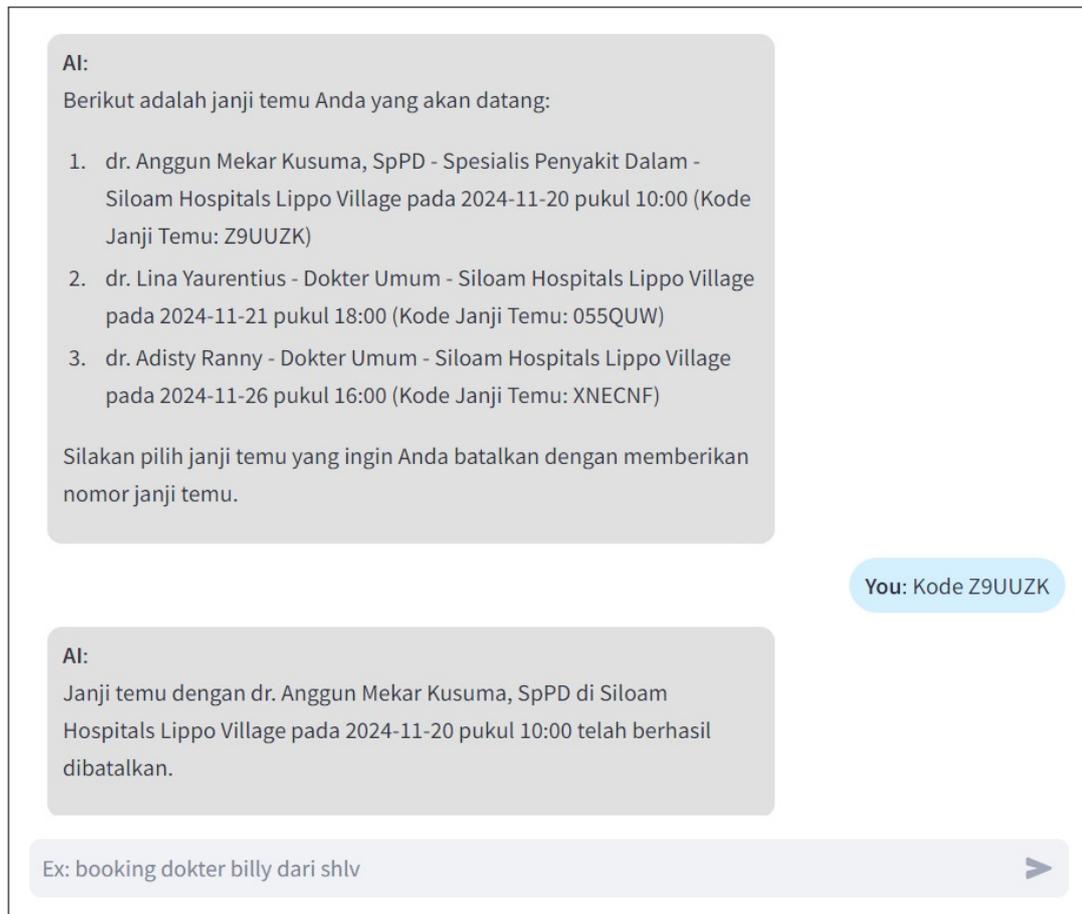
3.6.5 Contoh Batal Janji Temu Dokter



Gambar 3.18. Contoh Batal Janji Temu 1

Pada gambar 3.18 ini, pasien meminta *chatbot* untuk membatalkan janji temu yang akan datang. *Chatbot* memberikan list janji temu yang akan datang untuk pasien tersebut. Kemudian, pasien dapat memilih janji temu akan dibatalkan dengan memberikan kode janji temu. Pasien memilih janji temu ke 1 dengan kode janji temu Z9UUZK.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.19. Contoh Batal Janji Temu 2

Pasien telah memilih janji temu yang mau diganti jadwalnya. Pada gambar 3.19, *Chatbot* memberikan info bahwa janji temu yang dipilih oleh pasien telah dibatalkan. Info yang diberikan beserta nama dokter, rumah sakit, tanggal, dan waktu janji temu tersebut.

3.7 Testing

Dalam *testing* ini, *chatbot* yang telah dibuat akan dilakukan uji coba terhadap hasil dari *chatbot* dengan *score*. *Testing* ini lebih berfokus dalam *chatbot* menentukan spesialis atau gejala pasien. *Testing* dilakukan dengan melakukan input dan dinilai apakah hasilnya sesuai dengan ekspektasi. Setiap input akan dilakukan percobaan sebanyak 10 kali. Hasil testing diuraikan seperti pada 3.4, 3.5 dan 3.6

Tabel 3.4. *Accuracy Test* Berdasarkan Input Terhadap Spesialis dan Keluhan

Input	Expected Output	Score
<p>Saya punya keluhan demam muntah-muntah</p>	<p>Berdasarkan keluhan Anda, saya sarankan untuk konsultasi dengan dokter umum terlebih dahulu. Berikut adalah beberapa dokter umum yang tersedia di Siloam Hospitals Lippo Village: dr. Theresia Julian - General Practitioner, dr. Susanti Wijaya - General Practitioner, dr. Karina Helsa - General Practitioner, TEAM DOKTER MEMORY AGING CLINIC - General Practitioner, dr. Gloria Stefanie Ferdian - General Practitioner, dr. Lina Yaurentius - General Practitioner, dr. Willy Wanta - General Practitioner, dr. Palupi Nur Indayani - General Practitioner Silakan pilih dokter yang Anda inginkan untuk melanjutkan proses pemesanan.</p>	<p>90%</p>
<p>Saya mau bertemu dengan dokter mata</p>	<p>Berikut adalah beberapa dokter spesialis mata yang tersedia di Siloam Hospitals Lippo Village. Silakan pilih dokter yang Anda inginkan: dr. Widyastuti, SpM - Oftalmologi (Mata), dr. Maria Larasati Susyono, SpM - Oftalmologi (Mata), dr. Dion Oscar Iskandar, SpM - Oftalmologi (Mata), dr. Karliana Kartasa Taswir, SpM - Oftalmologi (Mata), dr. Endang M Johani, SpM - Oftalmologi (Mata), dr. Elvira, SpM - Oftalmologi (Mata), dr. Yoseph MPA Siahaan, SpM - Oftalmologi (Mata), dr. Heru Mahendrata Singgih, SpM - Oftalmologi (Mata), dr. Josiah Irma, SpM - Oftalmologi (Mata) Silakan pilih nomor dari 1 hingga 9 sesuai dengan dokter yang Anda inginkan.</p>	<p>95%</p>

Tabel 3.5. *Accuracy Test* Berdasarkan Input Terhadap Spesialis dan Keluhan

Input	Expected Output	Score
saya mau konsultasi ke dokter gigi	Berikut adalah beberapa dokter gigi yang tersedia di Siloam Hospitals Lippo Village. Silakan pilih dokter yang Anda inginkan: drg. Priscilla Lavine, SpKGA - Pediatric Dentist, drg. Monica Santosa, SpKGA - Pediatric Dentist, drg. Lily N Sitanggang - Dentist, Dr. drg. Andri Hardianto, SpBM(K), MD, Ph.D - Oral and Maxillofacial Surgeon, drg. Hartono Pudjowibowo - Dentist, drg. Jessica Santoso, SpKG - Dental Conservation - Endodontics Subspecialist, drg. Theresia Dhearine Pratiwi, SpKGA - Pediatric Dentist, Dr. drg. Andi S. Budihardja, SpBM, K - Oral and Maxillofacial Surgeon, drg. Shirley Clarissa - Dentist, drg. Suzanty Ariany, SpKGA - Pediatric Dentist Silakan pilih nomor dari 1 hingga 10 sesuai dengan dokter yang Anda inginkan.	95%
pinggang saya sakit, apa yang harus saya lakukan	Untuk nyeri pinggang, ada beberapa kemungkinan penyebabnya, seperti ketegangan otot, masalah pada tulang belakang, atau kondisi lainnya. Disarankan untuk berkonsultasi dengan dokter spesialis ortopedi atau spesialis rehabilitasi medis yang dapat melakukan pemeriksaan lebih mendalam dan memberikan perawatan yang sesuai. Namun, untuk evaluasi awal, Anda juga bisa berkonsultasi dengan dokter umum yang dapat melakukan pemeriksaan awal dan merujuk Anda ke spesialis jika diperlukan. Jika Anda ingin, saya bisa membantu untuk membuat janji dengan dokter di Siloam Hospitals. Apakah Anda memiliki lokasi rumah sakit yang diinginkan?	100%

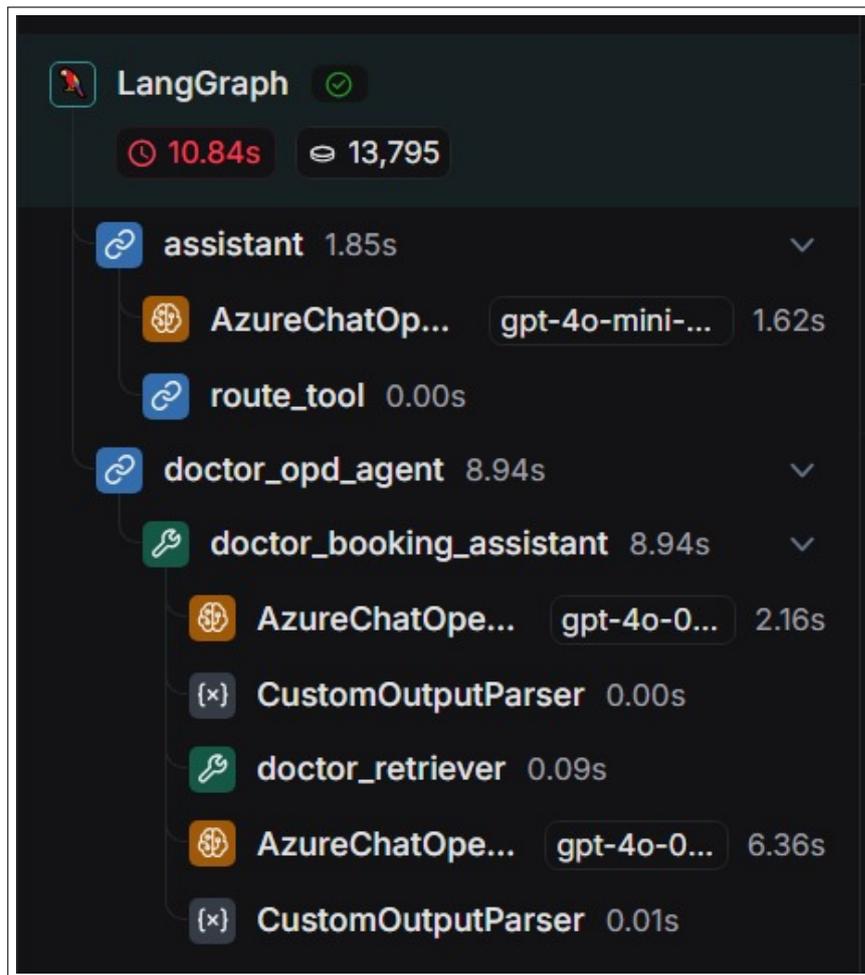
Tabel 3.6. *Accuracy Test* Berdasarkan Input Terhadap Spesialis dan Keluhan

Input	Expected Output	Score
saya mau konsultasi mengenai tulang	Berdasarkan informasi yang Anda berikan, saya sarankan untuk konsultasi dengan dokter umum terlebih dahulu. Berikut adalah beberapa dokter umum yang tersedia di Siloam Hospitals Lippo Village: dr. Theresia Julian - General Practitioner, dr. Susanti Wijaya - General Practitioner, dr. Karina Helsa - General Practitioner, TEAM DOKTER MEMORY AGING CLINIC - General Practitioner, dr. Gloria Stefanie Ferdian - General Practitioner, dr. Lina Yaurentius - General Practitioner, dr. Willy Wanta - General Practitioner, dr. Palupi Nur Indayani - General Practitioner Silakan pilih dokter yang Anda inginkan untuk melanjutkan proses pemesanan.	90%

Dari hasil *testing*, Ada beberapa yang mempunyai *score* 90% karena daftar dokter yang diberikan tidak selalu sama sehingga respon tidak 100% sama. Untuk input yang memiliki *score* 100% karena respon yang diberikan sama yaitu saran terhadap gejala pasien. Selain itu ada beberapa penyebab lain kenapa respon hanya 90%.

Pada input "Saya punya keluhan demam muntah-muntah", *chatbot* sekali memberikan respon saran kepada pasien dimana ekspektasinya *chatbot* langsung memberikan daftar dokter umum. Untuk input "saya mau konsultasi mengenai tulang", *chatbot* sekali memberikan respon daftar dokter tulang dimana ekspektasinya *chatbot* memberikan daftar dokter umum.

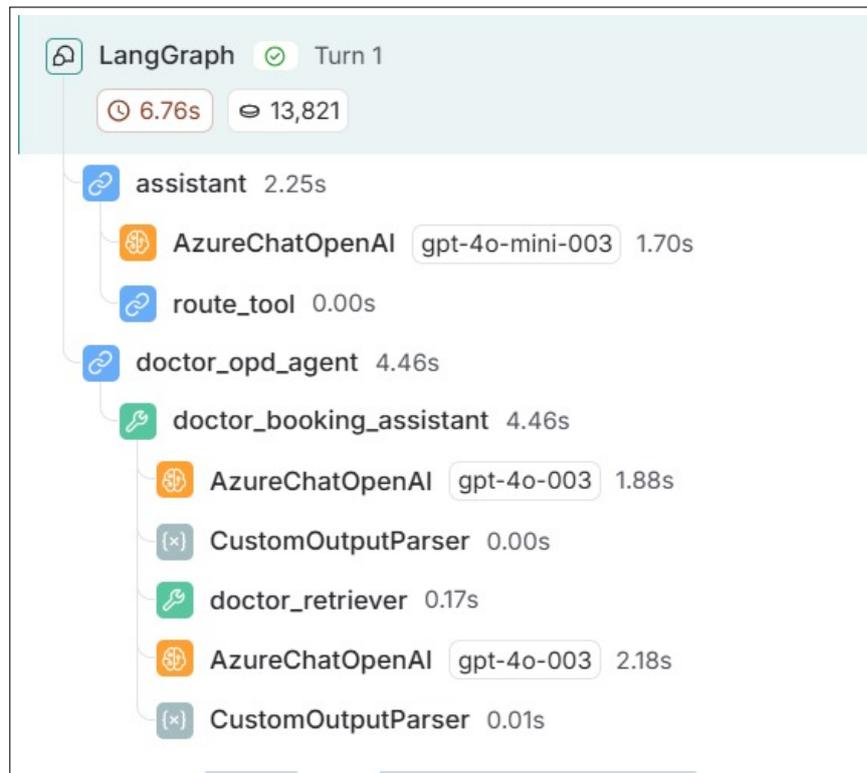
Hasil dari *testing* ini menunjukkan setiap input memiliki *score* diatas 90%. Berarti respon yang diberikan oleh *chatbot* yang dibuat sesuai dengan ekspektasi. Namun, terdapat kendala lain yaitu *chatbot* memiliki *time consuming* yang cukup lama.



Gambar 3.20. *Time Consuming Chatbot Local*

Pada Gambar 3.20, *Chatbot* ini memiliki *time consuming* yang lama karena LLM yang memakan waktu lama dalam mengarahkan pertanyaan ke *agent* dan membentuk respon. Hal ini juga karena *prompt* yang digunakan oleh *chatbot* panjang. Selain itu, adanya implementasi *history* agar *chatbot* dapat mengetahui konteks *chat* sebelumnya dengan pasien. Namun, hal ini dapat dipercepat jika *hardware* mendukung.

U
M
M
N
U
N
I
V
E
R
S
I
T
A
S
M
U
L
T
I
M
E
D
I
A
N
U
S
A
N
T
A
R
A



Gambar 3.21. *Time Consuming Chatbot UAT*

Pada Gambar 3.21 adalah *time consuming* pada *chatbot* yang sudah di run pada server UAT. Dapat dilihat *time consuming* yang dibutuhkan lebih cepat daripada local. Jadi solusi dari *time consuming* yang lama adalah *hardware* yang mumpuni.

3.8 Kendala dan Solusi yang Ditemukan

Kendala pada proses pelaksanaan magang adalah banyaknya pengerjaan fitur yang diganti atau tidak digunakan karena *requirements* yang berubah. Hal ini membuat pengerjaan *chatbot* dokter menjadi kurang efisien karena banyak waktu pengerjaan untuk fitur yang tidak digunakan atau diubah.

Solusi dari kendala ini adalah untuk melakukan perencanaan dahulu sebelum membuat fitur. Perencanaan dapat dibahas dengan tim *developer* agar lebih matang dan tidak ada perubahan selama atau sesudah pengerjaan fitur.