Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI) 15 *ISSN (Printed) : 2579-7271*
Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim Riau *ISSN (Online) : 2579-5406*
Pekanbaru, 24 Oktober 2023

Chatbot Berbasis Perintah untuk Membangkitkan Use Case Diagram

Muhammad Fikry*[1], Yusra[2]
[1,2] Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau
Email: [1]muhammad.fikry@uin-suska.ac.id, [2]yusra@uin-suska.ac.id

Abstrak

Chatbot dapat membantu pemodelan perangkat lunak sebelum perangkat lunak tersebut diimplementasikan. Untuk membangun model, Unified Modeling Language (UML) digunakan untuk mengkomunikasikan berbagai pandangan berbeda dari perangkat lunak yang dikembangkan. Dari 14 jenis diagram UML, penelitian-penelitian terkait hanya membangkitkan *class diagram*, *state chart diagram*, dan *activity diagram*. Pada penelitian ini, dirancang bangun *chatbot* untuk membangkitkan *use case diagram* secara otomatis. Hal ini dikarenakan *use case diagram* dibuat terlebih dahulu dalam fase analisis sebelum adanya diagram-diagram lain, sehingga berfungsi sebagai titik awal yang memberikan gambaran tingkat tinggi tentang fungsionalitas perangkat lunak. Untuk saat ini, *chatbot* hanya menerima masukan perintah, belum teks berbahasa alami. Analisa dilakukan untuk menspesifikasikan perintah yang dimasukkan oleh pengguna, dialog yang berlangsung antara pengguna dan *chatbot*, dan *action* yang dilakukan oleh *chatbot*. Setelah tahapan perancangan dan implementasi, dilakukan pengujian *blackbox*. Berdasarkan hasil pengujian *blackbox* terhadap seluruh fungsionalitas *chatbot*, diketahui bahwa *chatbot* telah sesuai dengan hasil analisa dan perancangan.

Kata kunci: *berbasis perintah, chatbot, UML, use case diagram.*

Abstract

Chatbots can help with software modeling. Unified Modeling Language (UML) is used for communicating different views of the software being developed. Of the 14 types of UML diagrams, related studies only generate class diagrams, state chart diagrams, and activity diagrams. In this study, a chatbot is designed to automatically generate a use case diagram because it is created first in the analysis phase before other diagrams exist, so it serves as a starting point that provides a high-level picture of software functionality. The chatbot only accepts command input, not natural language text. The analysis is carried out to specify the commands entered by the user, the dialogue that takes place between the user and the chatbot, and the actions performed by the chatbot. After the design and implementation stages, black box testing is carried out. Based on the test result of all chatbot functionalities, the chatbot is functioning correctly.

***Keywords**:* chatbot, *command-based*, UML, *use case diagram*.

1. Pendahuluan

Chatbot adalah agen percakapan yang menyediakan layanan dan akses informasi melalui interaksi dengan penggunanya. Ada berbagai istilah yang digunakan untuk merujuk pada sistem yang menyediakan interaksi percakapan dengan pengguna, yaitu *chatbot*, *conversational bot*, *conversational agent*, *virtual assistant*, dan *digital assistant*. Antarmuka percakapan (*conversational user interface*) yang dimiliki *chatbot* memungkinkan pengguna berinteraksi dengan menggunakan teks, ujaran, sentuhan, dan berbagai mode input/output lainnya [1].

Chatbot dapat diaplikasikan di berbagai bidang, misalnya layanan publik [2]–[4], institusi pendidikan [5]–[7], kesehatan [8]–[10], kesehatan mental [11][12], e-commerce [13][14], pariwisata [15], dan restoran [16]. Pada bidang rekayasa perangkat lunak, *chatbot* dapat digunakan untuk membantu pemodelan (*assisted modeling*) perangkat lunak, sebelum perangkat lunak tersebut diimplementasikan. Untuk membangun model, bahasa pemodelan grafis seperti Unified Modeling Language (UML) digunakan untuk mengembangkan, memahami, dan mengkomunikasikan berbagai pandangan berbeda dari perangkat lunak. Perez-Soler [17] mengusulkan penggunaan *chatbot* untuk membuat salah satu jenis diagram UML, yaitu *class diagram*, secara kolaboratif melibatkan banyak pengguna dalam satu grup Telegram. *Chatbot* tersebut mengatur kolaborasi antar pengguna dan menginterpretasikan input pengguna (dalam bahasa alami) untuk membangkitkan *class diagram* secara bertahap.

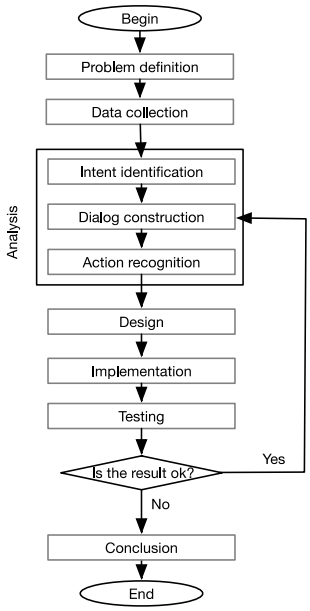
Pemodelan merupakan bagian penting dalam siklus hidup pengembangan perangkat lunak (*software development life cycle*, disingkat SDLC). Ada empat fase dasar di dalam SDLC, yaitu perencanaan, analisis, desain, dan implementasi [18]. Pada awal fase analisis, ditentukan kebutuhan perangkat lunak. Hasilnya menjadi masukan bagi langkah-langkah berikutnya, baik dalam fase analisis maupun desain, untuk membuat model (fungsional, struktural, dan *behavioral*) perangkat lunak. Dokumen kebutuhan perangkat lunak dan deskripsi domain yang dihasilkan pada awal fase analisis biasanya tersedia dalam bentuk teks berbahasa alami. Berbagai *task* pemrosesan bahasa alami dapat digunakan untuk membangkitkan diagram UML [19] dan model domain [20] secara otomatis dari teks tersebut. Hal ini dapat mempercepat proses pengembangan perangkat lunak, dan mempermudah sinkronisasi kebutuhan, model, dan perangkat lunak yang dihasilkan.

Menurut spesifikasi UML 2.5, ada 14 jenis diagram UML. Landhäußer [19] membangkitkan *class diagram*, *state chart diagram*, dan *activity diagram*, sementara Perez-Soler [17] hanya membangkitkan *class diagram* yang memperlihatkan pandangan (*view*) statis dari sistem. Selain *class diagram*, ada dua diagram yang sering digunakan untuk mendapatkan pandangan (*view*) lain dari sistem. Kedua diagram tersebut yaitu *use case diagram* yang menampilkan berbagai kemungkinan interaksi pengguna dan sistem, serta *sequence diagram* yang memperlihatkan pandangan (view) dinamis dari sistem. Baik Landhäußer [19] maupun Perez-Soler [17] belum menganalisis pembangkitan *use case diagram* dan *sequence diagram*.

Berdasarkan latar belakang masalah, pokok permasalahan dalam penelitian ini adalah bagaimana merancang bangun *chatbot* untuk membantu pemodelan (*assisted modeling*) perangkat lunak melalui pembangkitan diagram UML. Diagram UML yang dibangkitkan adalah *use case diagram*. Hal ini dikarenakan *use case diagram* dibuat terlebih dahulu dalam fase analisis sebelum adanya diagram-diagram lain. *Chatbot* menerima masukan berupa perintah diawali karakter slash (/), bukan teks berbahasa alami, dikenal sebagai *chatbot* berbasis perintah (disebut juga berbasis aturan, *keyword*, atau *transactional*). Hal ini dikarenakan penelitian ini merupakan langkah awal menuju *chatbot* yang dapat menerima bahasa alami, dengan terlebih dahulu mengidentifikasi *intent*, *entity*, dialog, dan *action* yang diperlukan. Tujuan penelitian ini yaitu merancang bangun *chatbot* berbasis *web* untuk membangkitkan *use case diagram*, serta melakukan pengujian terhadap *chatbot* yang telah dirancang bangun. Detail penelitian dijabarkan dalam bab Metode Penelitian yang memuat langkah-langkah penelitian, dan bab Hasil dan Pembahasan yang memuat analisa yang telah dilakukan dan hasil yang diperoleh. Akhirnya, bab Kesimpulan berisikan kesimpulan beserta saran untuk penelitian selanjutnya.

2. Metode Penelitian

Permasalahan dalam penelitian ini adalah bagaimana merancang bangun *chatbot* untuk membantu pemodelan (*assisted modeling*) perangkat lunak melalui pembangkitan diagram UML, yaitu *use case diagram*. Ada sejumlah tahapan sebagaimana diperlihatkan pada Gambar 1.



Gambar 1. Tahapan penelitian

Pada tahapan pengumpulan data, dikumpulkan kalimat-kalimat instruksi yang digunakan ketika membuat *use case diagram*. Kalimat-kalimat tersebut diperoleh dari responden meliputi mahasiswa jurusan Teknik Informatika semester 4, dan mahasiswa semester 5 ke atas. Mahasiswa semester 4 belum mendapatkan mata kuliah Rekayasa Perangkat Lunak Berorientasi Obyek (RPLBO) yang disajikan di semester 5, sehingga tidak memiliki pengetahuan mengenai istilah-istilah UML terkait pembuatan *use case diagram*. Mereka dilibatkan untuk mendapatkan kalimat-kalimat yang lebih umum ketika membuat diagram. Responden diberikan suatu gambar *use case diagram* yang diambil dari [18] dengan modifikasi, kemudian diminta menuliskan kalimat-kalimat instruksi yang digunakan ketika membuat *use case diagram*.

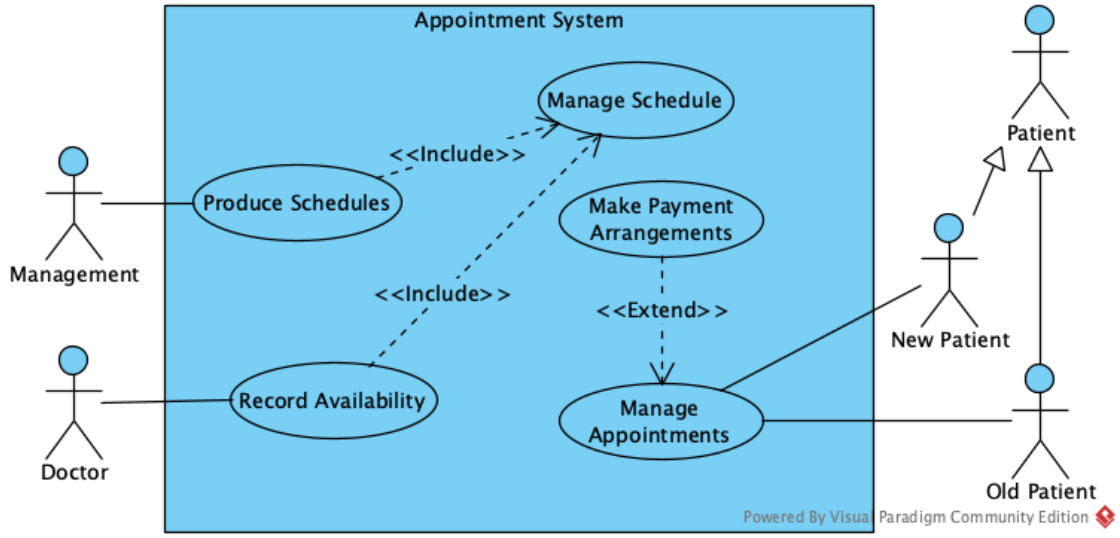
Analisa dilakukan untuk menspesifikasikan kebutuhan *chatbot* melalui *intent identification*, *dialog construction*, dan *action recognition*. Kalimat-kalimat instruksi yang diperoleh dari tahapan pengumpulan data digunakan untuk mendapatkan *intent* dan *entity*. Oleh karena penelitian ini dibatasi pada masukan berupa perintah yang diawali karakter slash (/) dan bukan bahasa alami, maka *intent* yang ditemukan menjadi dasar pembuatan perintah yang dimengerti oleh *chatbot*. Selanjutnya, didefinisikan struktur dialog yang diperlukan *chatbot* untuk memberikan respon. Respon yang dihasilkan oleh *chatbot* disebut *action*, dimana satu intent dapat menginisialisasi banyak *action*. Akhirnya, dilakukan perancangan struktur basis data, dan antar muka *chatbot*.

Chatbot diimplementasikan dengan menggunakan teknologi *web*. Setelah chatbot diimplementasikan, pengujian dilakukan menggunakan *blackbox testing* untuk mengetahui apakah *chatbot* sesuai dengan hasil analisa dan perancangan. Pada tahapan akhir dari penelitian ini dilakukan penarikan kesimpulan berdasarkan hasil tahapan-tahapan sebelumnya, serta penyampaian saran untuk pengembangan *chatbot* di masa datang.

3. Hasil dan Analisa

Data yang dikumpulkan dalam penelitian ini adalah kalimat-kalimat instruksi yang digunakan ketika membuat *use case diagram*. Responden diberikan suatu gambar *use case diagram* yang diambil dari [18] dengan modifikasi, kemudian diminta menuliskan kalimat-kalimat instruksi yang digunakan ketika membuat *use case diagram*. Gambar 2 *memperlihatkan use case diagram* yang diberikan kepada responden.

ISSN (Printed) : 2579-7271
ISSN (Online) : 2579-5406



Gambar 2. Use case diagram

Pada penelitian ini, diperoleh responden meliputi mahasiswa jurusan Teknik Informatika semester 4 sebanyak 31 orang, dan mahasiswa semester 5 ke atas sebanyak 103 orang. Total responden sebanyak 134 orang. Jawaban responden menjadi acuan untuk mendapatkan *intent* dan *entity* pada tahapan selanjutnya. Seperti teks berbahasa alami pada umumnya, jawaban responden bisa berantakan dan mengandung salah ketik.

Berdasarkan kalimat-kalimat instruksi yang diperoleh dari tahapan pengumpulan data, dilakukan identifikasi *intent* dan *entity*. Identifikasi *intent* dimaksudkan untuk mengkategorikan maksud (niat) dari pengguna dalam suatu percakapan. Sementara itu, identifikasi *entity* dimaksudkan untuk menentukan jenis informasi yang diekstrak dari masukan pengguna.

Oleh karena penelitian ini dibatasi pada masukan berupa perintah yang diawali karakter slash (/) dan bukan bahasa alami, maka *intent* dan *entity* yang ditemukan menjadi dasar pembuatan perintah yang dimengerti oleh *chatbot*. Kedepannya, kalimat-kalimat instruksi tersebut akan digunakan untuk melatih *chatbot* sehingga dapat memahami masukan teks berbahasa alami.

Spesifikasi perintah meliputi :

1. Suatu perintah (command) dapat dipisahkan menjadi satu atau lebih argumen (argument) dengan pemisah karakter spasi.
2. Argumen pertama adalah nama perintah yang diawali karakter slash (/).
3. Argumen-argumen lain yang mengikuti nama perintah berupa satu atau lebih pasangan opsi (option) dan nilai (value).
4. Opsi adalah argumen yang digunakan untuk memodifikasi perilaku dari suatu perintah, dan penulisannya diawali karakter hypen (-).
5. Nilai adalah argumen yang digunakan untuk menyatakan nilai dari suatu opsi, dan penulisannya diawali dan diakhir karakter tanda petik ganda ("), atau hanya karakter titik (.) yang menandakan *chatbot* akan mengembalikan pilihan nilai sesuai opsinya.

Berdasarkan langkah-langkah membuat *use case diagram* menurut Dennis et al. (2015), dimodifikasi menjadi langkah-langkah berikut :

1. Membuat *use case diagram*, sekaligus membuat *system boundary* yang berisikan teks nama perangkat lunak yang dirancang bangun.
2. Membuat *use case* dan aktor. Oleh karena *use case* dan aktor merupakan notasi-notasi yang ada di dalam *use case diagram*, maka tidak masalah yang mana dibuat terlebih dahulu.
3. Membuat *relationship* menghubungkan antara *use case* dan aktor, antara *use case* dan *use case*, serta antara aktor dan aktor.

Berdasarkan langkah-langkah tersebut di atas, perintah yang dimasukkan oleh pengguna, dialog yang berlangsung antara pengguna dan *chatbot*, dan *action* yang dilakukan oleh *chatbot* yaitu :

1. Perintah /list digunakan untuk memperlihatkan daftar sesuatu, dengan format sebagai berikut.
    a. Perintah /list -type "Use Case Diagram"
        i. *Chatbot* memperlihatkan daftar *use case diagram* yang telah dibuat.
    b. Perintah /list -type "Actor"
        i. *Chatbot* memperlihatkan daftar aktor yang telah dibuat.
    c. Perintah /list -type "Use Case"
        i. *Chatbot* memperlihatkan daftar *use case* yang telah dibuat.
2. Perintah /create digunakan untuk membuat sesuatu, dengan penggunaan format berikut.
    a. Perintah /create -type "Use Case Diagram" -name .
        i. *Chatbot* meminta dimasukkan nama *use case diagram*.
        ii. Pengguna memasukkan nama *use case diagram*.
        iii. *Chatbot* membuat *use case diagram*, kemudian menampilkannya.
    b. Perintah /create -type "Actor" -name .
        i. *Chatbot* meminta dimasukkan nama aktor.
        ii. Pengguna memasukkan nama aktor.
        iii. *Chatbot* membuat aktor baru, kemudian menampilkannya *pada use case diagram* yang sedang ditampilkan.
    c. Perintah /create -type "Use Case"
        i. *Chatbot* meminta dimasukkan nama *use case*.
        ii. Pengguna memasukkan nama *use case*.
        iii. *Chatbot* membuat *use case* baru, kemudian menampilkannya pada *use case diagram* yang sedang ditampilkan.
3. Perintah /associate digunakan untuk membuat *relationship* antara suatu notasi dengan notasi lain yang ada pada *use case diagram*. Formatnya sebagai berikut.
    a. Perintah /associate -fromId . -fromCardinality . -type . -toId . -toCardinality . -label .
        i. *Chatbot* meminta pengguna untuk memilih notasi (aktor atau *use case*) asal.
        ii. Pengguna memilih suatu notasi sebagai notasi asal.
        iii. *Chatbot* meminta pengguna untuk memilih jenis *relationship*: *line*, *arrow*, *include*, *extend*, atau *generalization*.
        iv. Pengguna memilih salah satu jenis *relationship*.
        v. *Chatbot* meminta pengguna untuk memilih notasi (aktor atau *use case*) tujuan.
        vi. Pengguna memilih suatu notasi sebagai notasi tujuan, dan berbeda dari notasi asal.
        vii. *Chatbot* meminta pengguna untuk mengetikkan label dari *relationship*, bila diperlukan.
        viii. Pengguna mengetikkan label dari *relationship*, bila diperlukan.
        ix. *Chatbot* membuat *relationship* dari notasi asal ke notasi tujuan dan menampilkannya pada *use case diagram* yang sedang ditampilkan.
4. Perintah /rename digunakan untuk mengubah nama *use case diagram*, *use case* atau aktor.
5. Perintah /delete digunakan untuk menghapus *use case diagram*, *use case* atau aktor.
6. Perintah /dissociate digunakan untuk memutuskan *relationship*.
7. Perintah /open digunakan untuk menampilkan suatu diagram.
8. Perintah /close digunakan untuk menutup suatu diagram sehingga tidak tampil di layar.

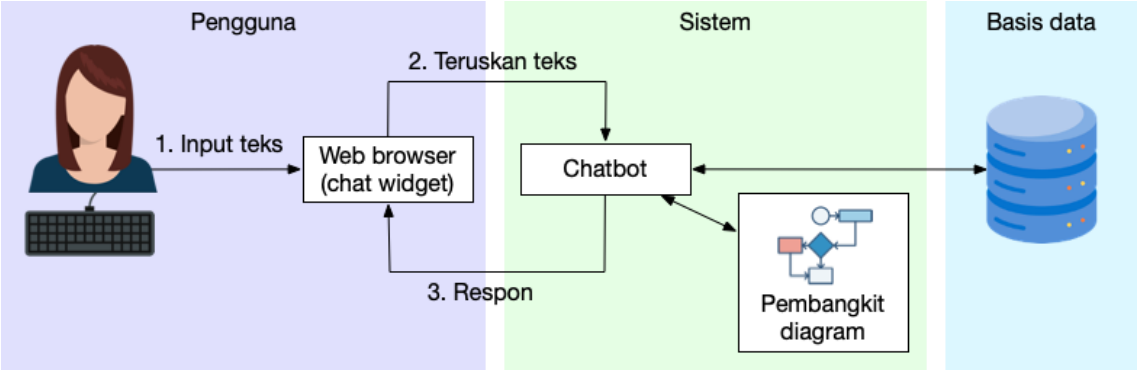
Dengan demikian, spesifikasi kebutuhan *chatbot* berbasis *web* untuk membangkitkan *use case diagram* yaitu :

1. *Chatbot* memiliki antarmuka percakapan yang memungkinkan pengguna berinteraksi dengan menggunakan teks.
2. *Chatbot* dapat menerima perintah, kemudian melaksanakannya.

ISSN (Printed) : 2579-7271
ISSN (Online) : 2579-5406

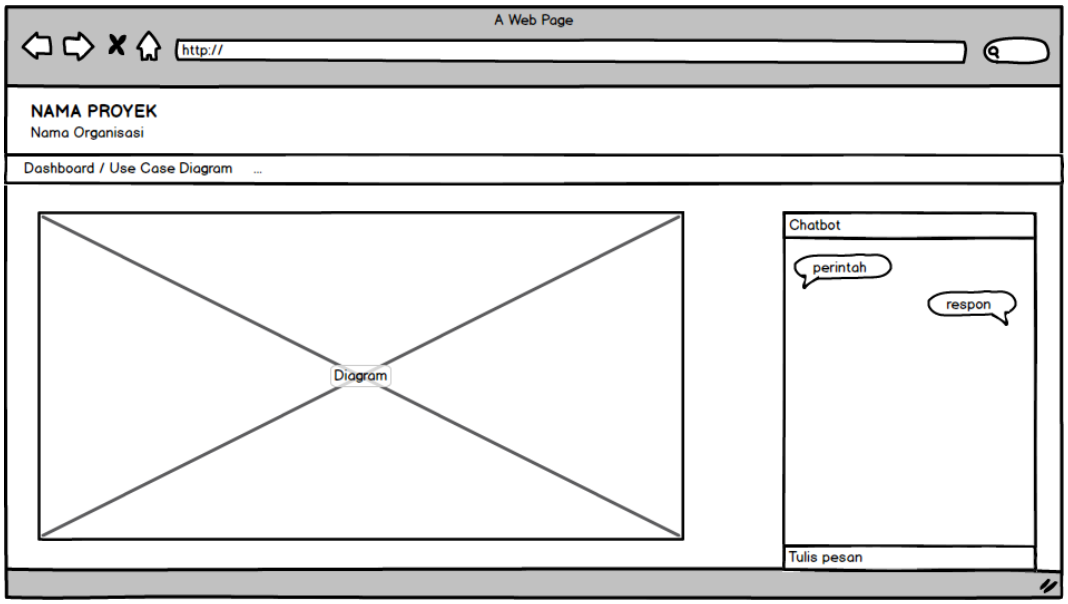
3. *Chatbot* memungkinkan pengguna mengelola *use case diagram*, meliputi membuat, menamai ulang, menghapus, membuka (menampilkan), dan menutup.
4. *Chatbot* memungkinkan pengguna mengelola *use case*, meliputi membuat, menamai ulang, dan menghapus.
5. *Chatbot* memungkinkan pengguna mengelola aktor, meliputi membuat, menamai ulang, dan menghapus.
6. *Chatbot* memungkinkan pengguna membuat relationship antar aktor, antara aktor dan *use case*, dan antar *use case*.
7. *Chatbot* memungkinkan pengguna menghapus *relationship* yang sudah dibuat..

Secara umum, alur penggunaan *chatbot* diperlihatkan pada Gambar 3, dengan penanganan sebagaimana aplikasi berbasis *web* pada umumnya.



Gambar 3. Alur penggunaan *chatbot*

Perancangan dilakukan untuk mendefinisikan basis data dan antarmuka. Basis data terdiri atas tabel *project*, *diagram*, *notation*, *diagram_notation*, dan *relation*. Rancangan antarmuka dibuat dengan menggunakan aplikasi Balsamiq Mockups. *Use case diagram* yang dihasilkan oleh *chatbot* ditampilkan pada halaman *web*, terpisah dari antarmuka *chatbot*. Hal ini dikarenakan ukuran gambar diagram terlalu besar jika ditampilkan di dalam antarmuka *chatbot*. Rancangan antarmuka diperlihatkan pada Gambar 4.

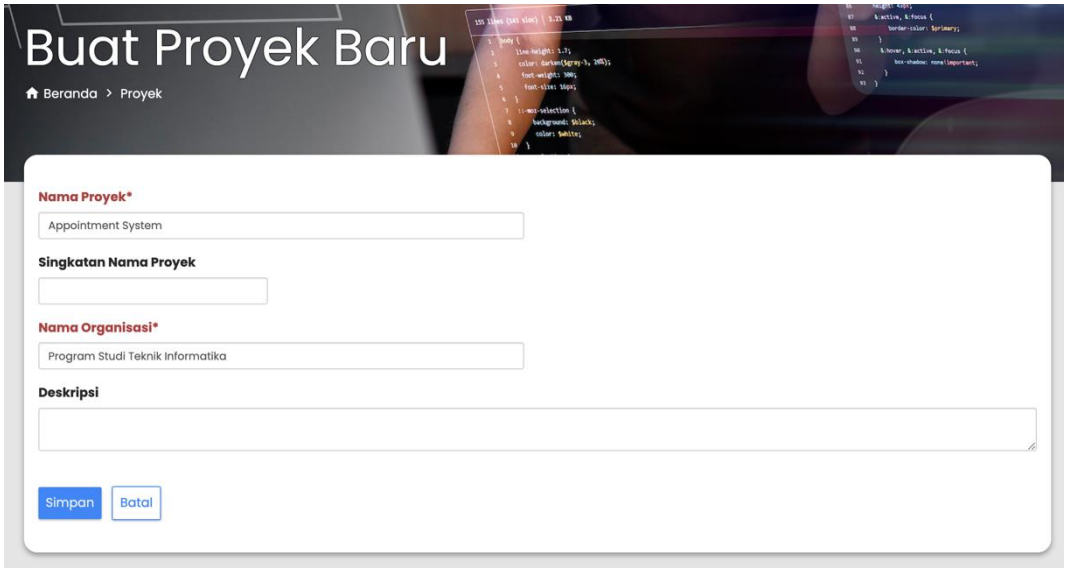


Gambar 4. Rancangan antar muka

ISSN (Printed) : 2579-7271
ISSN (Online) : 2579-5406

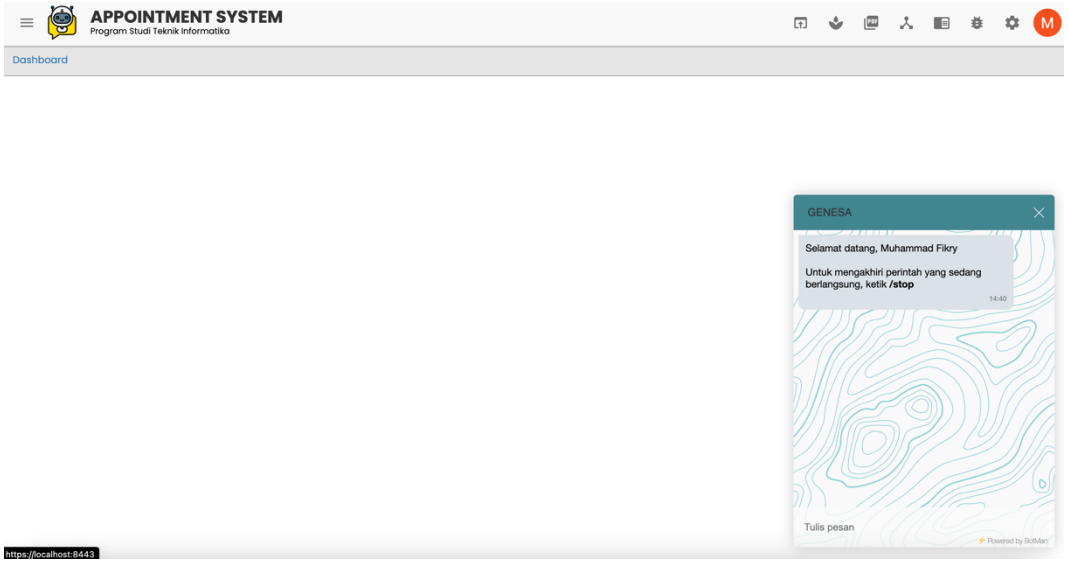
Berdasarkan hasil analisa dan perancangan, dilakukan implementasi *chatbot* untuk membangkitkan *use case diagram*. *Chatbot* ditempatkan di suatu halaman *web* dari aplikasi lain yang mengelola proyek, sehingga *chatbot* dapat digunakan untuk berinteraksi dengan proyek tertentu yang sedang dibuka oleh pengguna saja. Perangkat lunak yang digunakan untuk pengembangan aplikasi yaitu PHP, Botman, Python, dan PlantUML.

Hasil implementasi tampilan diperlihatkan pada gambar-gambar berikut. Pada tampilan aplikasi yang digunakan untuk mengelola proyek, ditambah proyek baru yaitu Appointment System. Tampilannya diperlihatkan pada Gambar 5.



Gambar 5. Tampilan aplikasi

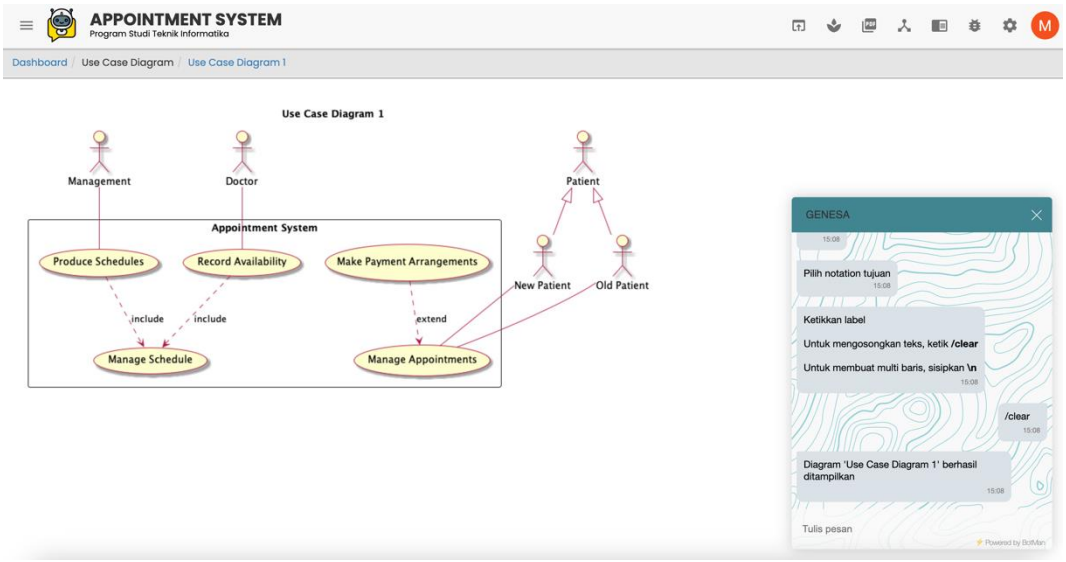
Pengguna dapat berinteraksi dengan *chatbot* pada halaman proyek Appointment System. Tampilannya diperlihatkan pada Gambar 6.



Gambar 6. Halaman proyek beserta tampilan *chatbot*

Hasil penggunaan *chatbot* untuk membuat *use case diagram* diperlihatkan pada Gambar 7.

ISSN (Printed) : 2579-7271
ISSN (Online) : 2579-5406



Gambar 7. Hasil pembuatan *use case diagram*

Setelah implementasi, pengujian dilakukan dengan menggunakan *blackbox testing* untuk mengetahui apakah *chatbot* sesuai dengan hasil analisa dan perancangan. Pengujian dilakukan dengan cara membuat *use case diagram* yang serupa dengan yang diberikan kepada responden. Hasil pengujian menggunakan *blackbox testing* diperlihatkan pada Tabel 1.

Tabel 1. Hasil pengujian *blackbox*

No	Deskripsi	Input	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
1	Membuat *use case diagram*	Nama *use case diagram*	*Use case diagram* dibuat dan ditampilkan	*Use case diagram* dibuat dan ditampilkan	Berhasil
2	Membuat aktor	Nama aktor : Management	Aktor dibuat dan ditampilkan	Aktor dibuat dan ditampilkan	Berhasil
3	Membuat *use case*	Nama *use case* : Produce Schedules	*Use case* dibuat dan ditampilkan	*Use case* dibuat dan ditampilkan	Berhasil
4	Membuat *relationship*	Jenis *relationship* : *line*	*Relationship* dibuat dan ditampilkan	*Relationship* dibuat dan ditampilkan	Berhasil
5	Mengubah nama *use case*	Nama *use case* : Record Availability	Nama *use case* berubah	Nama *use case* berubah	Berhasil
6	Menghapus *relationship*	Salah satu *relationship* yang ada di layar	*Relationship* dihapus	*Relationship* dihapus	Berhasil
7	Menghapus *use case*	Hapus *use case* Record Availability	*Use case* dihapus	*Use case* dihapus	Berhasil
8	Menutup use *case diagram* yang sedang ditampilkan	-	*Use case diagram* ditutup	*Use case diagram* ditutup	Berhasil
9	Membuka *use case diagram* yang ditutup sebelumnya	Pilih Use Case Diagram, kemudian pilih nama Use Case Diagram 1	*Use case diagram* ditampilkan	*Use case diagram* ditampilkan	Berhasil

Berdasarkan hasil pengujian *blackbox* terhadap seluruh fungsionalitas *chatbot*, diketahui bahwa *chatbot* telah sesuai dengan hasil analisa dan perancangan.

4. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini yaitu prototipe *chatbot* berbasis perintah untuk membangkitkan *use case diagram* telah berhasil dirancang bangun. *Chatbot* dapat menerima masukan berupa perintah diawali karakter *slash* (/). Berdasarkan hasil analisis, diperoleh 8 (delapan) perintah yang digunakan untuk membuat *use case diagram*. Berdasarkan hasil pengujian *blackbox*, fungsionalitas *chatbot* telah sesuai dengan hasil analisa dan perancangan.

Saran yang dapat diberikan untuk penelitian selanjutnya yaitu menambahkan dukungan masukan teks berbahasa alami dari pengguna dalam Bahasa Indonesia sebagai penganti perintah yang diawali karakter slash (/), misalnya dengan memanfaatkan Rasa Framework atau Dialogflow. Selain itu, perlu dilakukan User Acceptance Test (UAT) dengan meminta pengguna menggunakan *chatbot* untuk membuat *use case diagram* sesuai deskripsi yang disediakan. Chatbot dapat dikembangkan untuk mendukung diagram-diagram UML lainnya, seperti *activity diagram*, *sequence diagram*, *class diagram*, dan *deployment diagram*.

Referensi

[1] McTear, M. F., “The rise of the conversational interface: A new kid on the block?,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, pp. 38–49. doi: 10.1007/978-3-319-69365-1_3.

[2] Moreira, F., J. Arango, G. M. Ramirez, and J. L. Sanchez, “Development of a Chatbot with NLP for the Prefecture of Los Rios in Ecuador,” in *17th Iberian Conference on Information Systems and Technologies (CISTI)*, Madrid, Spain: ITMA, 2022. doi: 10.23919/CISTI54924.2022.9820488.

[3] Khadija, M. A., W. Nurharjadmo, and Widyawan, “Deep Learning Generative Indonesian Response Model Chatbot for JKN-KIS,” in *2022 1st International Conference on Smart Technology, Applied Informatics, and Engineering (APICS)*, 2022, pp. 70–74. doi: 10.1109/APICS56469.2022.9918686.

[4] Hasan, I., S. Rizvi, S. Jain, and S. Huria, “The AI enabled Chatbot Framework for Intelligent Citizen-Government Interaction for Delivery of Services,” in *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2021, pp. 601–606.

[5] Supreetha, H. V and S. Sandhya, “Implementation of an Educational Chatbot using Rasa Framework,” vol. 3075, no. 9, pp. 29–35, 2022, doi: 10.35940/ijitee.G9189.0811922.

[6] Fauzia, L., “Implementation of Chatbot on University Website Using RASA Framework,” no. December 2021, 2022, doi: 10.1109/ISRITI54043.2021.9702821.

[7] Sebastian, D. and K. A. Nugraha, “Academic Customer Service Chatbot Development using TelegramBot API,” in *2021 2nd International Conference on Innovative and Creative Information Technology (ICITech)*, 2021, pp. 221–225. doi: 10.1109/ICITech50181.2021.9590140.

[8] Badlani, S., “Multilingual Healthcare Chatbot Using Machine Learning,” no. May 2021, 2022, doi: 10.1109/INCET51464.2021.9456304.

[9] Lescano, N. L., E. Yamao, E. X. V Sánchez, and M. A. P. E. Santillan, “Iterative design and implementation of a chatbot for sexual and reproductive health counseling in Peru,” in *2022 IEEE XXIX International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, 2022, pp. 1–4. doi: 10.1109/INTERCON55795.2022.9870050.

[10] Yimyam, W. and M. Ketcham, “Occupational Disease Risk Assessment System Using Artificial Intelligence System and Chatbot,” in *2022 International Conference on Cybernetics and Innovations (ICCI)*, 2022, pp. 1–5. doi: 10.1109/ICCI54995.2022.9744143.

[11] Omarov, B., S. Narynov, Z. Zhumanov, E. Alzhanova, and A. Gumar, “Artificial Intelligence Enabled Conversational Agent for Mental Healthcare,” vol. 6, no. 3, pp. 1544–1555, 2022.

[12] Liu, H., H. Peng, X. Song, C. Xu, and M. Zhang, “Using AI chatbots to provide self-help depression interventions for university students: A randomized trial of effectiveness,” *Internet Interv.*, vol. 27, p. 100495, 2022, doi: 10.1016/j.invent.2022.100495.

[13] Liew, T. W., G. Guan, G. Goh, and S. Tan, “The effects of designing conversational commerce chatbots with expertise cues,” no. January 2022, 2021, doi: 10.1109/HSI52170.2021.9538741.

[14] Rakhra, M., “E-Commerce Assistance with a Smart Chatbot using Artificial Intelligence,” pp. 6–10, 2021.

[15] AlHumoud, S. *et al.*, “Rahhal : A Tourist Arabic Chatbot,” in *2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, 2022. doi: 10.1109/SMARTTECH54121.2022.00028.

[16] Garg, R. *et al.*, “NLP Based Chatbot for Multiple Restaurants,” in *2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)*, 2021, pp. 439–443. doi: 10.1109/SMART52563.2021.9676218.

[17] Perez-Soler, S., E. Guerra, J. De Lara, and F. Jurado, “The rise of the (modelling) bots: Towards assisted modelling via social networks,” *ASE 2017 - Proc. 32nd IEEE/ACM Int. Conf. Autom. Softw. Eng.*, no. May 2018, pp. 723–728, 2017, doi: 10.1109/ASE.2017.8115683.

[18] Dennis, A., B. H. Wixom, and D. Tegarden, *Systems Analysis and Design : An Object-Oriented*

Approach with UML, 5th ed. Wiley, 2015. doi: 10.1016/0950-5849(89)90057-8.

[19] Landhäußer, M., S. J. Körner, and W. F. Tichy, "From requirements to UML models and back: How automatic processing of text can support requirements engineering," *Softw. Qual. J.*, vol. 22, no. 1, pp. 121–149, 2014, doi: 10.1007/s11219-013-9210-6.

[20] Arora, C., M. Sabetzadeh, L. Briand, and F. Zimmer, "Extracting domain models from natural-language requirements: Approach and industrial evaluation," *Proc. - 19th ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. Model. 2016*, pp. 250–260, 2016, doi: 10.1145/2976767.2976769.

[21] Franciscatto, M. H., M. Didonet, D. Fabro, L. Augusto, and O. Gonçalves, "Querying Multidimensional Big Data Through a Chatbot System," no. May, 2022, doi: 10.1145/3477314.3507692.