

Pengembangan Layanan Informasi FAQ Berbasis *Chatbot* di Badan Pendapatan Daerah Kota Surabaya Menggunakan *Framework* RASA

Aderio Rendy Harenza¹, Aries Dwi Indriyanti²

^{1,2} Sistem Informasi, Universitas Negeri Surabaya

¹aderiorendy.21006@mhs.unesa.ac.id

²ariesdwi@unesa.ac.id

Abstrak— Badan Pendapatan Daerah (Bapenda) Kota Surabaya sebagai lembaga publik yang berperan penting dalam pengelolaan pajak dan pendapatan daerah harus terus meningkatkan kualitas pelayanannya kepada masyarakat khususnya bagi para wajib pajak. Salah satu bentuk layanan informasi yang tersedia di Bapenda Kota Surabaya adalah fitur *Frequently Asked Question* (FAQ) yang dapat diakses pada *website* resminya. Namun layanan tersebut terbatas dan kurang efektif dalam menjawab berbagai pertanyaan dari masyarakat. Untuk mengatasi permasalahan tersebut, penelitian ini mengembangkan sistem *chatbot* berbasis *framework* RASA yang terintegrasi dengan sistem monitoring berbasis web. Penelitian ini menerapkan dua metode, yaitu metode CRISP-DM untuk pengembangan model *chatbot* dan metode *Rapid Application Development* (RAD) untuk pengembangan sistem monitoring. Hasil pengujian menunjukkan bahwa *chatbot* memiliki akurasi dengan nilai *F1-score* sebesar 0.946, nilai *precision* 0.954, dan tingkat keberhasilan *story* sebesar 84,6%. Sistem monitoring juga berjalan baik berdasarkan pengujian *black box testing* yang telah dilakukan. Dengan demikian, solusi ini cukup efektif dalam meningkatkan efisiensi layanan informasi dan meringankan beban petugas dalam memberikan informasi kepada masyarakat khususnya para wajib pajak.

Kata Kunci— *Frequently Ask Question*, *Chatbot*, *Framework* RASA, *Natural Language Understanding*, Badan Pendapatan Daerah.

I. PENDAHULUAN

Pada era globalisasi yang semakin berkembang ini, kemajuan teknologi informasi telah mengubah pola dan struktur hubungan antara masyarakat dan pemerintah. Saat ini, infrastruktur teknologi di Indonesia masih sedikit menggunakan sumber daya manusia dan sumber daya pendukungnya, hal tersebut menyebabkan masalah dalam menyampaikan informasi dan layanan publik kepada masyarakat [1]. Setiap bentuk usaha, baik yang dikelola oleh pemerintah maupun pihak swasta, terus berupaya meningkatkan mutu pelayanannya guna menghadapi persaingan yang semakin sengit antar organisasi [2]. Kemajuan teknologi informasi membuat lembaga-lembaga sudah mulai memperbaiki sistem kerja mereka untuk memberikan layanan yang lebih baik kepada masyarakat [3].

Salah satu lembaga atau instansi pemerintah yang memiliki peran sangat penting dalam melayani masyarakat adalah Badan Pendapatan Daerah (Bapenda). Bapenda dipimpin oleh Kepala Badan, yang bertanggung jawab kepada Walikota melalui

Sekretaris Daerah, dan melaksanakan fungsi penunjang dan pembantuan di bidang keuangan [4]. Tugas utama dan fungsi Bapenda adalah melaksanakan tugas pemerintah daerah dalam hal pajak dan retribusi daerah, serta pendapatan lainnya [5]. Tugas ini sangat krusial karena pendapatan daerah menjadi pondasi bagi pembiayaan berbagai program pembangunan dan pelayanan publik yang dilakukan oleh pemerintah daerah.

Sebagai lembaga publik yang memberikan layanan kepada masyarakat, Bapenda harus meningkatkan kualitas layanan agar meningkatkan kepuasan masyarakat [5]. Oleh sebab itu penyediaan layanan informasi terhadap masyarakat khususnya para wajib pajak terkait berbagai kewajiban dan hak mereka akan menjadi tantangan besar bagi Bapenda. Seperti contoh pada Bapenda Kota Surabaya yang menyediakan layanan informasi di *website* resminya mengenai pertanyaan yang sering ditanyakan atau biasa disebut *Frequently Ask Question* (FAQ). Namun informasi pada *website* tersebut masih bersifat terbatas. Hal tersebut kurang efisien karena daftar pertanyaan yang tersedia hanya sedikit atau terbatas karena masih menerapkan sistem lama berupa daftar pertanyaan dan jawaban saja. Hal tersebut juga kurang efektif untuk membantu para petugas untuk menjawab kebutuhan pertanyaan dari masyarakat khususnya para wajib pajak. Oleh karena itu, layanan informasi FAQ pada Bapenda Kota Surabaya perlu ditingkatkan untuk membantu petugas menjawab pertanyaan dari para wajib pajak.

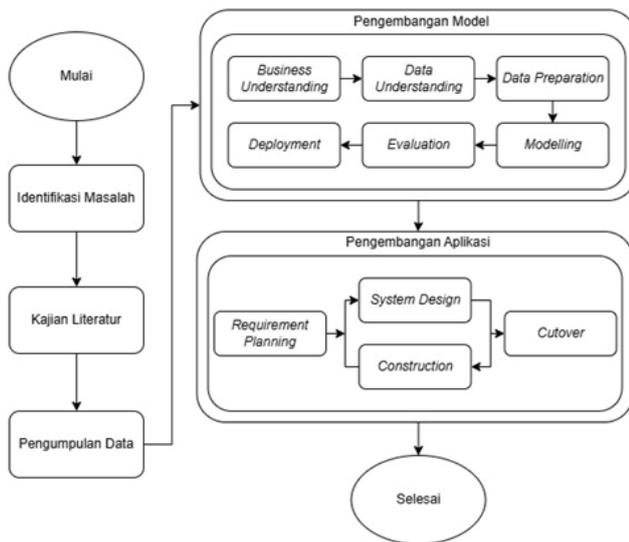
Salah satu solusi potensial untuk mengatasi tantangan ini adalah dengan menerapkan teknologi *chatbot*. *Chatbot* memungkinkan pengguna mendapatkan bantuan dengan cepat dan akurat tanpa keterlibatan manusia secara langsung, membuat interaksi antara manusia dan teknologi lebih sederhana dan efisien [6]. *Chatbot* adalah program atau layanan yang memungkinkan pengguna dapat berinteraksi melalui percakapan teks [7]. Implementasi teknologi *chatbot* ini tidak hanya akan meningkatkan efisiensi layanan informasi yang disediakan oleh Bapenda Kota Surabaya, tetapi juga akan mengurangi beban kerja petugas. Selain itu, *chatbot* dapat beroperasi selama 24 jam, memberikan fleksibilitas yang lebih besar kepada para wajib pajak untuk mengakses informasi sesuai kebutuhan mereka [8].

Ada banyak kerangka kerja atau *framework* yang dapat digunakan untuk membangun sebuah *chatbot*, salah satunya adalah *framework* RASA [9]. RASA adalah *framework* yang bersifat *open source* untuk mengembangkan sebuah *chatbot* [10]. Kerangka kerja ini menawarkan fitur yang

memungkinkan pengembang melatih *chatbot* agar dapat memahami dan menanggapi pertanyaan dari pengguna [11]. RASA memiliki perbedaan dibandingkan dengan interaksi FAQ sebelumnya karena didasarkan pada percakapan yang lebih alami, seperti bagaimana manusia berinteraksi satu sama lain dengan mempertimbangkan konteks dan tindakan apa yang harus diambil [12]. Secara keseluruhan, penggunaan teknologi *chatbot* dengan *framework* RASA membawa banyak manfaat seperti peningkatan efisiensi layanan informasi hingga mengurangi beban kerja petugas.

II. METODOLOGI PENELITIAN

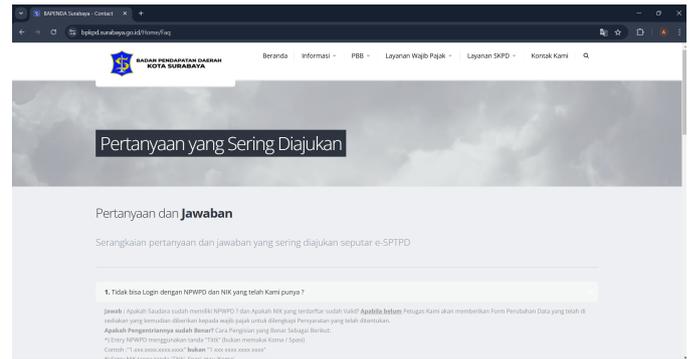
A. Kerangka Penelitian



Gbr. 1 Kerangka Penelitian.

Kerangka penelitian ini menjelaskan langkah-langkah sistematis yang diambil untuk mencapai tujuan penelitian. Sesuai dengan gambar 1, penelitian ini akan membuat dua sistem yaitu sebuah model *chatbot* beserta sistem monitoringnya sehingga akan menggunakan dua metode yaitu metode *Cross-Industry Standard Process for Data Mining* (CRISP-DM) dan metode *Rapid Application Development* (RAD). Metode CRISP-DM digunakan untuk mengembangkan model *chatbot*, sedangkan metode RAD digunakan untuk pengembangan sistem monitoring *chatbot* berbasis *website*.

B. Identifikasi Masalah



Gbr. 2 FAQ Website Bapenda Kota Surabaya.

Badan Pendapatan Daerah Kota Surabaya menghadapi tantangan dalam menyediakan informasi FAQ yang efektif dan berguna bagi masyarakat. *Frequently Ask Question* atau disingkat FAQ adalah sistem yang membantu pengguna dalam menemukan jawaban atas pertanyaan yang sering diajukan dalam penyelenggaraan layanan tertentu [13]. Banyak pertanyaan bersifat repetitif atau berulang yang ditanyakan oleh masyarakat kepada petugas yang sebenarnya sudah disediakan pada FAQ di *website* resmi Bapenda Kota Surabaya. Masalah-masalah ini menunjukkan kebutuhan mendesak untuk mengotomatisasi layanan FAQ melalui *chatbot*, yang diharapkan dapat menjawab pertanyaan umum secara cepat dan akurat tanpa keterlibatan manusia.

Layanan informasi FAQ yang berjalan saat ini merupakan sistem statis berupa daftar pertanyaan dan jawaban yang tersedia di situs web Bapenda Kota Surabaya seperti yang ditunjukkan gambar 2. FAQ ini berfungsi sebagai sumber informasi mandiri yang berisi jawaban atas pertanyaan yang sering diajukan terkait layanan dan prosedur tertentu. Pengguna dapat mengakses daftar ini untuk mencari informasi dasar tentang layanan yang ditawarkan. Informasi di situs web ini seringkali tidak mencakup seluruh pertanyaan yang mungkin dicari oleh pengguna, sehingga masyarakat tetap membutuhkan bantuan langsung dari petugas. Jumlah staf terbatas, akibatnya waktu tanggap terhadap pertanyaan masyarakat menjadi tidak konsisten dan sering terjadi keterlambatan dalam memberikan jawaban.

Untuk mengatasi kelemahan-kelemahan tersebut, dikembangkan sebuah sistem *chatbot* berbasis FAQ. *Chatbot* adalah sistem percakapan cerdas yang mampu memproses bahasa manusia secara alami [14]. *Chatbot* akan dibangun menggunakan *framework* RASA yang mampu menjawab pertanyaan masyarakat secara otomatis, tanpa keterlibatan petugas. Tidak seperti sistem *chatbot* lainnya, karena bersifat *open source*, RASA sangat fleksibel dan memungkinkan kostumisasi yang luas [15]. *Framework* RASA juga dilengkapi dengan kemampuan pemahaman bahasa alami atau NLU (*Natural Language Understanding*), sehingga dapat memahami dan merespons berbagai pertanyaan dengan cepat dan akurat. Dengan *chatbot* ini, masyarakat dapat mengajukan pertanyaan kapan saja, termasuk di luar jam kerja.

C. Kajian Literatur

Kajian literatur merupakan tahapan dalam penelitian yang melibatkan penelaahan dan analisis literatur terkait dengan topik yang diteliti. Tujuan dari kajian ini adalah untuk memahami fokus serta temuan dari penelitian sebelumnya, yang dapat membantu menghindari pengulangan studi yang sama dan mengidentifikasi topik penelitian baru berdasarkan pola-pola yang telah ada. Dalam penelitian ini, kajian literatur difokuskan pada studi sebelumnya mengenai *chatbot*, khususnya penggunaan *framework* pengembangan *chatbot* berbasis NLU. *Natural Language Understanding* (NLU) adalah salah satu cabang dari *Natural Language Processing* (NLP) yang berkaitan dengan kemampuan program untuk menangkap dan memahami maksud atau pesan (*intent*) yang ingin disampaikan oleh manusia [16].

D. Pengumpulan Data

Badan Pendapatan Daerah (Bapenda) adalah organisasi di bawah pemerintah daerah yang bertanggung jawab atas penerimaan pendapatan daerah dengan mengoordinasikan dan mengumpulkan pajak, retribusi, bagi hasil pajak, dana perimbangan, serta sumber pendapatan lainnya [17]. Data primer diperoleh melalui wawancara dengan *stakeholder* atau pemangku kepentingan seperti admin atau staff IT Badan Pendapatan Daerah Kota Surabaya untuk mengidentifikasi pertanyaan yang paling sering diajukan oleh masyarakat sebanyak 25 pertanyaan umum. Sedangkan data sekunder meliputi dokumentasi yang sudah ada yaitu daftar FAQ di situs *website* resmi Badan Pendapatan Daerah Kota Surabaya. Untuk setiap pertanyaan dari data primer dan sekunder nantinya akan dibuat variasi menjadi 5 pertanyaan yang akan digunakan untuk melatih model *chatbot* dan dioptimalkan agar sesuai dengan kebutuhan yang diharapkan.

E. Pengembangan Model

Penelitian ini akan menggunakan metode CRISP-DM untuk pengembangan model *chatbot*. *Cross-Industry Standard Process for Data Mining* atau biasa disingkat CRISP-DM adalah metode pengembangan sistem yang fleksibel, yang dapat diterapkan pada berbagai jenis proyek data mining yang memungkinkan para pengembang untuk menyesuaikan setiap tahapannya sesuai dengan kebutuhan dan tujuan proyek [18]. Model *chatbot* akan dibuat terlebih dahulu sebelum masuk ke tahap pengembangan sistem monitoring.

1) Business Understanding

Tahap pertama yang perlu dilakukan pada proses pengembangan model ialah *Business Understanding* atau pemahaman bisnis. Pada tahap ini ada beberapa hal yang perlu dilakukan seperti menentukan tujuan dan persyaratan yang jelas terkait *chatbot* yang akan dikembangkan. Pada tahap ini, dilakukan identifikasi tujuan utama dari pengembangan *chatbot* yaitu untuk memberikan layanan informasi FAQ yang lebih efisien di Badan Pendapatan Daerah Kota Surabaya. *Chatbot* ini dirancang untuk membantu masyarakat khususnya para wajib pajak mendapatkan informasi terkait layanan pajak

yang ada di web resmi secara cepat dan akurat, sehingga mengurangi beban staf dan meningkatkan kepuasan pengguna.

2) Data Understanding

Selanjutnya masuk ke tahap *data understanding* atau pemahaman data. Penelitian ini akan menggunakan data yang diperoleh dari proses wawancara dengan pemangku kepentingan dari Bapenda Kota Surabaya terkait pertanyaan yang sering diajukan masyarakat. Selain itu, peneliti juga akan menggunakan data FAQ yang sudah ada di *website* resmi Bapenda Kota Surabaya. Setelah data dikumpulkan, selanjutnya akan dilakukan pemberian *intent* untuk setiap pertanyaan. Setiap *intent* nantinya akan memiliki 5 variasi pertanyaan. Pertanyaan dengan *intent* yang sama akan dikelompokkan menjadi satu. Untuk *intent* yang hanya memiliki satu pertanyaan nantinya akan dibuat variasi pertanyaan yang mirip atau serupa.

3) Data Preparation

Framework RASA memiliki komponen dataset sendiri yang terdiri dari 6 bagian yang akan dikonfigurasi untuk melatih model *chatbot*. Komponen-komponen tersebut berupa format YML/YAML sesuai dengan standar RASA. Komponen-komponen tersebut berisi *NLU*, *Rules*, *Story*, *Config*, dan *Credentials*. Data NLU pada *RASA Framework* terdiri dari contoh pesan pengguna yang diberi label atau *intent* sesuai dengan maksud di setiap kategori. *Rules* atau aturan berisi kumpulan contoh yang menggambarkan tahapan serta alur dialog *chatbot*. Setiap urutan tersusun rapi dalam bentuk cerita (*story*) yang memuat informasi tentang *intent* dan aksi (*action*) yang terkait. Data *domain* dalam *RASA Framework* mencakup elemen-elemen percakapan seperti tindakan, maksud, entitas, dan slot. *Config* atau konfigurasi berisi pengaturan penting yang mencakup *language*, *pipeline*, dan *policy* yang digunakan untuk melatih model *chatbot*. Sedangkan *credential* berisi token aktivasi yang memungkinkan model *chatbot* yang telah dilatih dapat digunakan dan terhubung dengan berbagai platform komunikasi.

4) Modelling

Proses pemodelan dalam RASA secara umum memiliki tiga tahapan sesuai dengan tiga komponen utama pada file *config* pada *framework* RASA yaitu *language*, *pipeline*, dan *policies*. *Language* atau bahasa merupakan komponen untuk menentukan bahasa apa yang akan digunakan dalam *chatbot* nantinya. *Pipeline* adalah inti dari pemrosesan teks di RASA yang mencakup rangkaian komponen untuk melakukan preprocessing dan modeling. *Policies* adalah komponen yang menentukan logika percakapan di *RASA Core*, termasuk bagaimana *chatbot* merespons pengguna berdasarkan konteks percakapan.

5) Evaluation

Proses evaluasi *chatbot* berbasis *RASA Framework* dilakukan dengan memanfaatkan fitur pengujian otomatis yang telah disediakan oleh RASA. Pengujian tersebut terbagi menjadi dua sesuai dengan komponen yang ada pada RASA

yaitu NLU test dan Core (Dialogue Management) test. Penjelasan lebih lengkap dari dua pengujian tersebut adalah sebagai berikut.

- *NLU Test*

Pada saat perintah rasa test nlu dijalankan, RASA akan mengambil data dari file seperti data/nlu.yml dan secara default atau otomatis membagi data menjadi 80% data latih dan 20% data uji. Selanjutnya, model akan memprediksi *intent* dan *entity* pada data uji, dan hasil prediksi tersebut akan menghasilkan *confusion matrix*. *Confusion Matrix* merupakan tabel yang dapat memvisualisasikan kinerja *classifier* dengan membandingkan hasil prediksi kelas dengan nilai sebenarnya dari data [19].

- *Core Test*

Untuk pengujian Core (Dialog Management) test, RASA menguji kemampuan model dalam mengelola dialog dan menentukan aksi (*action*) yang tepat berdasarkan konteks percakapan. Pengujian ini dilakukan menggunakan file uji percakapan (tests/test_stories.yml). Ketika perintah rasa test core dijalankan, RASA akan melakukan simulasi percakapan berdasarkan skenario-skenario yang telah ditentukan. Model akan memprediksi urutan aksi berdasarkan intent yang dikenali, state percakapan sebelumnya, dan kebijakan (policies) yang dikonfigurasi, seperti *MemoizationPolicy* untuk mencocokkan story secara persis, *RulePolicy* untuk mengikuti aturan eksplisit, *TEDPolicy* untuk memprediksi aksi berdasar konteks dengan RNN, dan *UnexpectEDIntentPolicy* untuk menangani *intent* tidak terduga secara eksplisit.

6) Deployment

Setelah menyelesaikan proses *modeling* dan pelatihan dataset, model *chatbot* yang dikembangkan dan memenuhi kriteria evaluasi yang memadai, akan dibuat bagian anatar muka atau *frontend* agar bisa digunakan oleh pengguna. Untuk tampilan antarmuka digunakan teknologi berbasis web seperti HTML, CSS, dan *Javascript*. Komunikasi antara *frontend* dan *chatbot* dilakukan melalui *webhook* API RASA.

F. Pengembangan Aplikasi

Setelah model *chatbot* berhasil dibuat, tahap selanjutnya yaitu pengembangan aplikasi berbasis *website* berupa sistem monitoring *chatbot*. Penelitian ini akan menggunakan metode *Rapid Application Development* (RAD) untuk mengembangkan sistem monitoring tersebut. Metode RAD dirancang untuk mempercepat proses pengembangan aplikasi serta melakukan pembaruan secara berkelanjutan dengan fokus pada masukan dari pengguna [20].

1) Requirement Planning

Requirement Planning merupakan langkah awal yang penting untuk pemetaan kebutuhan dari sebuah aplikasi. Pada tahap ini peneliti akan melibatkan *stakeholder* seperti Kepala Koordinator IT yang ada di Bapenda Kota Surabaya. Peneliti akan melakukan diskusi dan wawancara terkait rancangan

aplikasi yang telah dibuat, sehingga *stakeholder* dapat melakukan validasi dan memberikan masukan tentang rancangan aplikasi yang akan dibangun. Berdasarkan permasalahan yang telah dijelaskan sebelumnya, peneliti telah merumuskan karakteristik pengguna pada Tabel 3.1 berikut.

TABEL I
 KARAKTERISTIK PENGGUNA

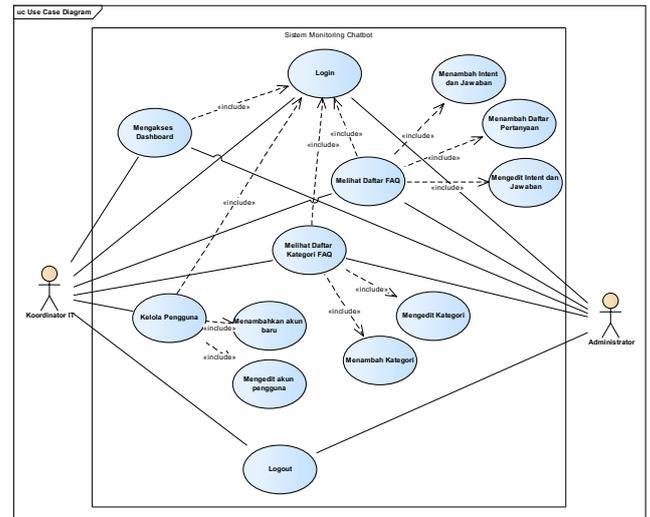
No.	Pengguna	Karakteristik
1	Koordinator IT (Super Admin)	Koordinator IT dapat mengakses seluruh menu dan fitur yang ada dalam sistem seperti mengakses dashboard, mengelola kategori FAQ, mengelola pertanyaan dan jawaban FAQ, serta dapat mengelola akun petugas yang dapat mengakses sistem.
2	Administrator (Admin)	Admin hanya dapat mengakses dashboard, mengelola kategori FAQ, dan mengelola pertanyaan dan jawaban FAQ.

Setelah karakteristik untuk masing-masing pengguna teridentifikasi pada tabel I, maka selanjutnya peneliti melakukan analisis untuk kebutuhan fungsional dan non fungsional sistem. Kebutuhan fungsional disusun berdasarkan fitur-fitur utama yang dibutuhkan oleh pengguna. Fitur-fitur ini dirancang agar dapat mendukung aktivitas pengguna sesuai dengan perannya dalam sistem.

TABEL III
 KEBUTUHAN FUNGSIONAL SISTEM

No.	Pengguna	Kebutuhan Fungsional
1	Koordinator IT	Dapat melakukan <i>login</i> .
		Dapat melihat ringkasan atau dashboard sistem monitoring <i>chatbot</i> .
		Dapat melihat daftar kategori FAQ.
		Dapat menambahkan kategori FAQ baru.
		Dapat mengedit kategori FAQ yang tersedia.
		Dapat melihat daftar FAQ.
		Dapat menambahkan FAQ baru berupa intent dan jawaban.
		Dapat menambahkan atau mengedit pertanyaan pada FAQ.
		Dapat melihat dan mengedit <i>intent</i> dan jawaban pada FAQ.
		Dapat melihat daftar petugas yang memiliki hak akses.
		Dapat menambahkan akun petugas baru.
		Dapat mengedit akun petugas yang tersedia.

2	Administrator	Dapat melakukan <i>logout</i> .
		Dapat melakukan <i>login</i> .
		Dapat melihat ringkasan atau dashboard sistem monitoring <i>chatbot</i> .
		Dapat melihat daftar kategori FAQ.
		Dapat menambahkan kategori FAQ baru.
		Dapat mengedit kategori FAQ yang tersedia.
		Dapat melihat daftar FAQ.
		Dapat menambahkan FAQ baru berupa <i>intent</i> dan jawaban.
		Dapat menambahkan atau mengedit pertanyaan pada FAQ.
		Dapat melihat dan mengedit <i>intent</i> dan jawaban pada FAQ.



Gbr. 3 Use Case Diagram Sistem Monitoring Chatbot.

Sementara itu, kebutuhan non fungsional merujuk pada serangkaian batasan, karakteristik, dan atribut yang harus dimiliki oleh sistem, yang tidak berkaitan langsung dengan fungsi atau perilaku spesifik sistem, tetapi lebih menekankan pada kualitas sistem secara keseluruhan.

TABEL III
KEBUTUHAN NON FUNGSIONAL SISTEM

No.	Pengguna	Kebutuhan non Fungsional
1	Koordinator IT	Memiliki akses untuk melihat dan mengelola akun petugas yang ada.
		Dapat mengubah peran (role) pengguna.
		Dapat mengubah status akun (aktif/nonaktif).
		Dapat membuat kata sandi baru untuk petugas yang lupa password.
2	Administrator	Hanya memiliki akses untuk melakukan manajemen data kategori, FAQ, dan melihat dashboard, tanpa bisa mengelola akun pengguna lain.

2) System Design

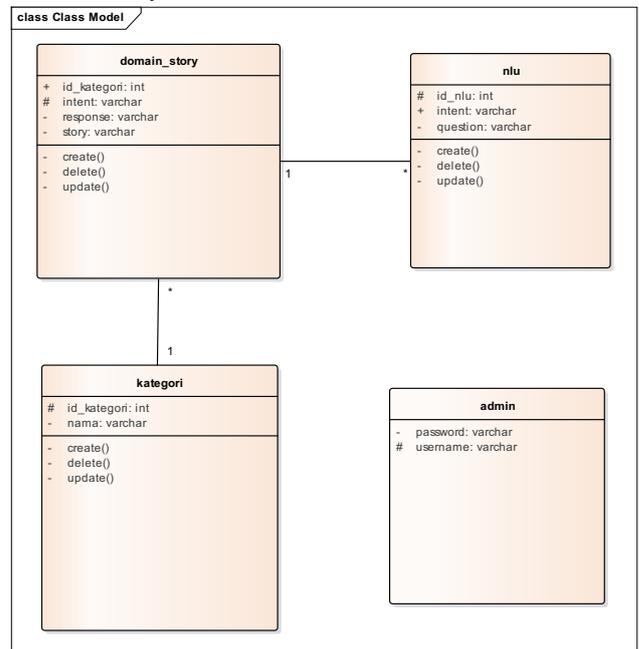
Langkah berikutnya adalah merancang desain sistem monitoring *chatbot* yang akan dibangun seperti diagram UML dan kerangka visual sebagai acuan dari sistem yang akan dibangun.

- **Use Case Diagram**

Use Case diagram adalah jenis diagram UML yang menggambarkan interaksi antara pengguna (aktor) dan fungsi-fungsi utama dalam sebuah sistem. Diagram ini menunjukkan skenario penggunaan yang mencakup peran pengguna serta hubungan mereka dengan proses atau fitur yang ada di dalam sistem.

- **Class Diagram**

Class diagram membantu pengembang untuk memahami bagaimana data diatur dalam sistem dan bagaimana kelas-kelas berinteraksi satu sama lain untuk membuat struktur *database* nantinya. Dengan menggunakan *class diagram*, pengembang dapat memvisualisasikan entitas utama, atribut, dan hubungan antar kelas sebelum membangun struktur *database* yang sebenarnya.

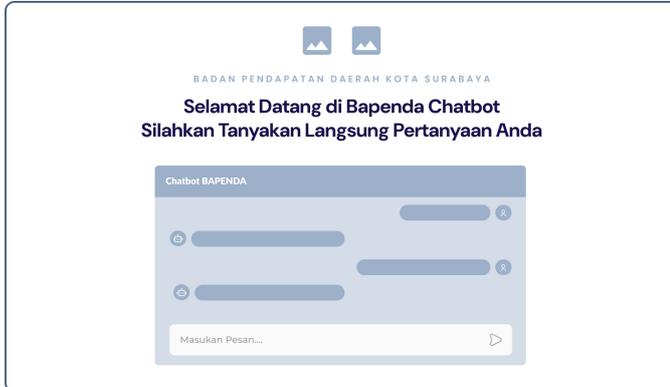


Gbr. 4 Class Diagram Sistem Monitoring Chatbot.

- **Wireframe**

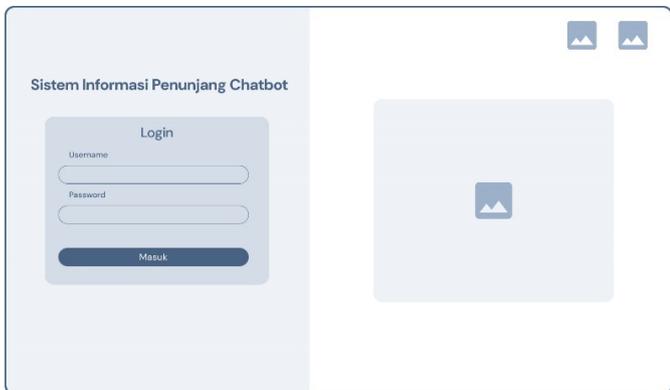
Setelah mengidentifikasi diagram UML, selanjutnya yaitu membuat kerangka visual atau *wireframe* dari

sistem monitoring *chatbot* yang akan dibuat. Dengan *wireframe*, pengembang dan pemangku kepentingan dapat memahami alur interaksi pengguna dengan sistem dan memastikan bahwa elemen-elemen penting ditempatkan dengan strategis untuk memudahkan penggunaan.



Gbr. 5 Wireframe Halaman Chatbot.

Pada halaman menu *chatbot* terdapat sebuah jendela percakapan interaktif seperti pada gambar 5. Desain antarmuka dibuat sederhana dan minimalis guna meningkatkan efisiensi serta kenyamanan pengguna. Pendekatan ini memungkinkan pengguna untuk langsung fokus pada fungsi utama yakni mengajukan pertanyaan, tanpa harus terganggu oleh elemen visual yang tidak relevan.



Gbr. 6 Wireframe Halaman Login.

Pada halaman login berisi username dan password sebagai akses masuk ke sistem informasi yang ditunjukkan pada gambar 6. Setelah pengguna memasukkan kredensial yang valid, sistem akan melakukan proses autentikasi dan mengarahkan pengguna ke halaman *dashboard* utama sesuai dengan hak akses yang dimiliki. Jika terjadi kesalahan pada input, seperti *username* atau *password* yang salah, maka sistem akan menampilkan notifikasi kesalahan untuk memberi

tahu pengguna agar memeriksa kembali data yang dimasukkan.



Gbr. 7 Wireframe Sistem Monitoring Chatbot.

Pada Halaman menu *dashboard* berisi ringkasan informasi seperti yang ditunjukkan oleh gambar 7. Di sebelah kanan terdapat sebuah navbar yang berisi menu-menu yang ada di sistem monitoring *chatbot*. Menu-menu tersebut berguna untuk mengelola daftar kategori FAQ, daftar FAQ, dan daftar pengguna sistem.

3) Construction

Tahapan *construction* mengimplementasikan dengan menulis kode program berdasarkan rancangan desain sistem yang telah dibuat. Pembuatan sistem akan menggunakan bahasa pemrograman C# dan menggunakan *framework* ASP.NET Core. *Database* yang digunakan disesuaikan dengan *database* pada *chatbot* agar memungkinkan integrasi data antara *chatbot* dan sistem monitoring yang dikembangkan. Dengan demikian, setiap perubahan atau penambahan data pada sistem monitoring *chatbot* akan secara otomatis tercermin dalam *chatbot*, sehingga memudahkan proses pemeliharaan data secara *real-time*.

4) Cutover

Cutover merupakan tahap akhir yang berisi pengujian akhir yang akan dilakukan oleh pemangku kepentingan atau *stakeholder*. Langkah pengujian sangat penting untuk memastikan program bekerja dengan baik dan memenuhi persyaratan. Untuk penelitian ini penulis menggunakan metode *Black Box Testing* sebagai metode pengujianya.

III. HASIL DAN PEMBAHASAN

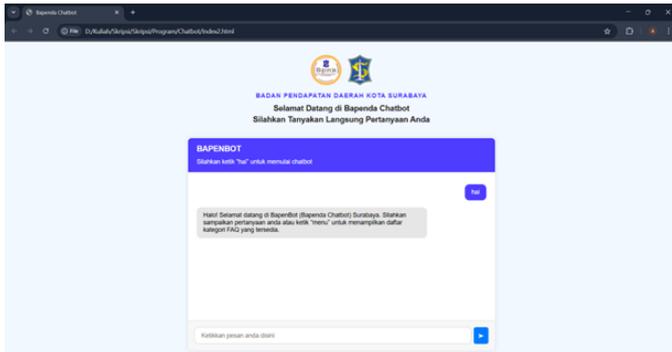
A. Hasil

Hasil pengembangan mencakup dua komponen. Pertama model *chatbot* yang berfungsi untuk berinteraksi langsung dengan pengguna atau masyarakat, khususnya para wajib pajak. Kedua sebuah sistem monitoring yang dirancang untuk memudahkan *stakeholder* khususnya para koordinator IT dan administrator Bapenda Kota Surabaya dalam mengelola dan memelihara data *chatbot*. Kedua komponen ini saling

terintegrasi, sehingga menciptakan sebuah ekosistem yang lengkap antara pengguna dan pengelola.

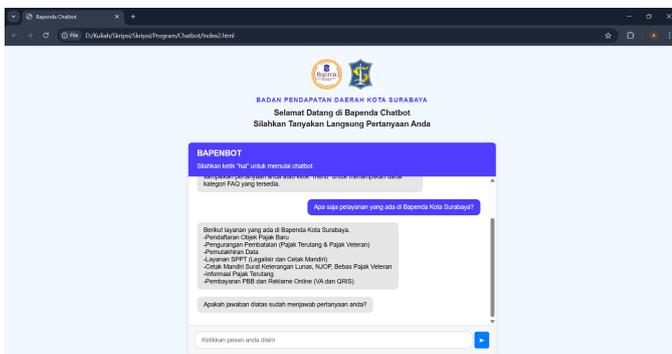
1) Model Chatbot

Model *chatbot* yang dikembangkan diberi nama “BapenBot” yang merupakan singkatan dari Bapenda *Chatbot*. BapenBot dirancang untuk membantu pengguna umum, khususnya masyarakat dan para wajib pajak. Pengembangan model *chatbot* bertujuan untuk menggantikan FAQ sebelumnya yang masih statis agar dapat meningkatkan efisiensi dalam penyampaian informasi kepada masyarakat.



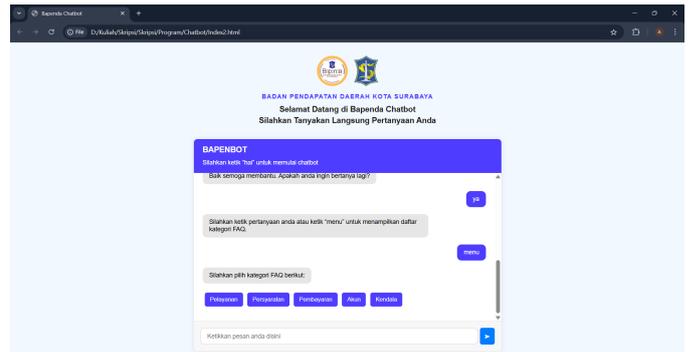
Gbr. 8 Halaman Jendela *Chatbot*.

Untuk mengaktifkan *chatbot*, pengguna harus menginputkan kata “hai” seperti yang diinstruksikan pada bagian atas jendela *chatbot*. Ketika pengguna menginputkan atau mengirimkan kata “hai” *chatbot* akan aktif dan menampilkan pesan seperti pada gambar 8. Pengguna dapat menuliskan pertanyaannya secara langsung atau menginputkan kata “menu” untuk menampilkan daftar kategori FAQ yang tersedia.



Gbr. 9 Pengguna Bertanya Pada *Chatbot*.

Ketika pengguna menanyakan sebuah pertanyaan langsung kepada *chatbot* seperti pada gambar 9, *chatbot* langsung memberikan jawaban atas pertanyaan tersebut. Kemudian *chatbot* akan mengirimkan pesan verifikasi apakah jawaban yang diberikan sudah menjawab atau belum terkait pertanyaan yang telah diajukan. Pengguna dapat mengirimkan kata “sudah” untuk menandai bahwa jawaban yang diberikan sudah benar.

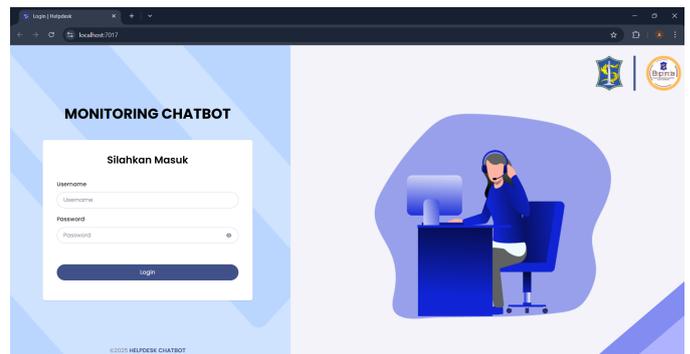


Gbr. 10 Pengguna Memilih Menu Pada *Chatbot*.

Apabila pengguna menginputkan kata menu, *chatbot* akan menampilkan daftar kategori FAQ yang tersedia seperti yang ditunjukkan pada gambar 10. Pengguna dapat memilih kategori FAQ yang diinginkan dengan menekan tombol kategori yang muncul.

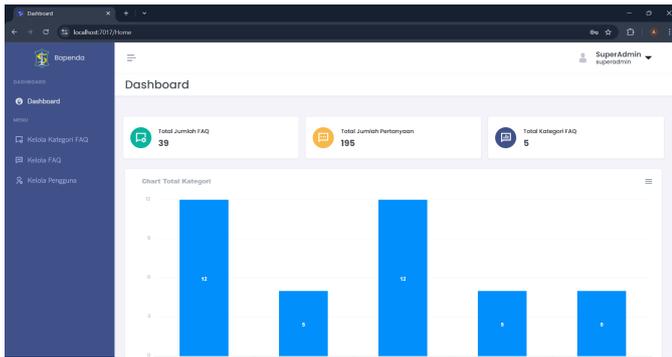
2) Sistem Monitoring *Chatbot*

Sistem monitoring *chatbot* merupakan sebuah platform berbasis web yang dikembangkan untuk memudahkan pengelolaan *chatbot* yang telah dibuat. Sistem ini dirancang agar administrator atau petugas yang bertanggung jawab dapat mengelola data *chatbot* dengan lebih efisien. Dengan adanya sistem ini, *chatbot* dapat terus diperbarui dan diperbaiki sesuai dengan kebutuhan Bapenda Kota Surabaya.



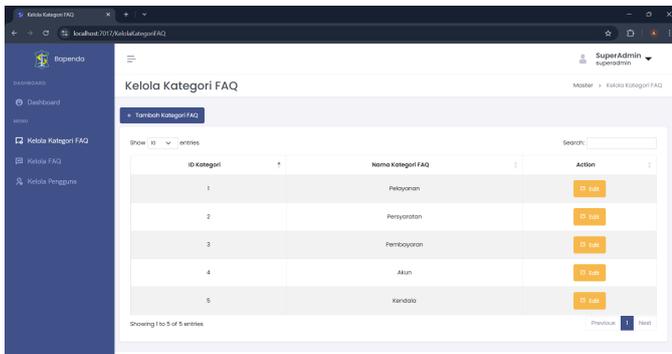
Gbr. 11 Halaman Login Sistem Monitoring *Chatbot*.

Halaman *login* merupakan gerbang utama bagi pengguna sebelum dapat mengakses sistem monitoring *chatbot*. Pengguna perlu memasukkan *username* dan *password* yang telah terdaftar untuk masuk ke sistem. Jika informasi yang dimasukkan benar, pengguna akan diarahkan ke *dashboard*.



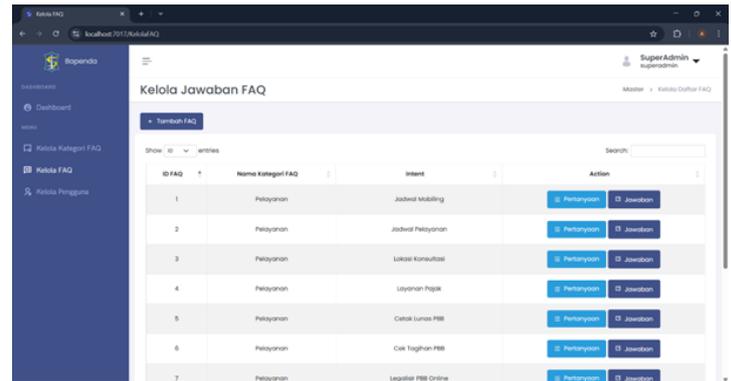
Gbr. 12 Halaman Dashboard Sistem Monitoring Chatbot.

Setelah berhasil login, pengguna akan masuk ke halaman dashboard yang berfungsi sebagai pusat informasi mengenai chatbot. Di halaman ini, administrator dapat melihat total jumlah FAQ, jumlah kategori FAQ, serta total pertanyaan yang telah ditambahkan. Selain itu terdapat chart atau grafik yang menampilkan jumlah FAQ berdasarkan kategori, sehingga administrator dapat dengan mudah menganalisis distribusi data dalam chatbot.



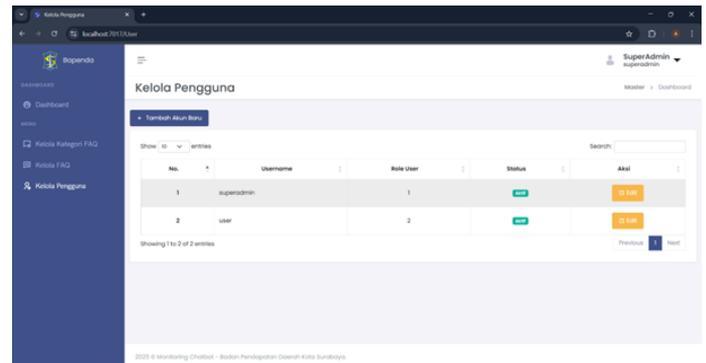
Gbr. 13 Halaman Kategori FAQ Sistem Monitoring Chatbot.

Pada halaman kelola kategori FAQ, administrator dapat mengelola kategori pertanyaan yang ada di dalam chatbot. Data kategori ditampilkan dalam bentuk datatable yang berisi ID kategori, nama kategori, serta opsi untuk mengedit kategori. Dengan adanya fitur pengelolaan kategori ini, administrator dapat secara fleksibel menyesuaikan struktur topik yang ada dalam chatbot agar tetap relevan dengan kebutuhan.



Gbr. 14 Halaman Kelola FAQ Sistem Monitoring Chatbot.

Halaman kelola FAQ merupakan fitur yang digunakan untuk mempermudah administrator dalam mengelola daftar pertanyaan dan jawaban yang akan digunakan oleh chatbot. Halaman ini berfungsi sebagai pusat pengelolaan daftar FAQ yang digunakan dalam chatbot. Tombol-tombol aksi tersebut memungkinkan administrator untuk mengedit atau memperbaiki pertanyaan maupun jawaban yang ada. Selain itu, tersedia juga tombol Tambah FAQ yang dapat digunakan untuk menambahkan pertanyaan baru ke dalam sistem.



Gbr. 15 Halaman Kelola Pengguna Sistem Monitoring Chatbot.

Halaman kelola pengguna berfungsi sebagai pusat manajemen akun bagi setiap petugas yang memiliki hak akses terhadap sistem monitoring chatbot. Halaman ini memungkinkan administrator untuk memantau dan mengatur seluruh akun yang terdaftar. Data pengguna ditampilkan dalam bentuk datatable yang mencakup username, peran pengguna (role), status akun, serta tombol edit. Selain itu, terdapat tombol tambah pengguna yang digunakan untuk menambahkan akun baru ke dalam sistem. Dengan fitur ini, pengelolaan hak akses menjadi lebih terorganisir dan aman, karena hanya pihak yang benar-benar berwenang yang dapat melakukan perubahan pada sistem. Fitur ini juga sangat penting untuk memastikan bahwa hanya pengguna yang memiliki otorisasi yang dapat mengelola chatbot.

B. Pembahasan

Bagian ini membahas mengenai proses pengembangan layanan informasi FAQ berbasis chatbot yang telah dirancang

dan dibangun untuk Bapenda Kota Surabaya. Pembahasan difokuskan pada dua komponen utama yaitu model *chatbot* yang menjadi inti dari sistem tanya jawab otomatis, serta sistem monitoring *chatbot* yang digunakan untuk mengelola dan memantau konten FAQ dan pengguna dalam sistem tersebut.

1) Model Chatbot

Chatbot yang dikembangkan dalam penelitian ini diberi nama “Bapenbot” yang berarti Bapenda *Chatbot*. Bapenbot dirancang menggunakan *framework* RASA yang merupakan salah satu *platform open-source* untuk membangun *chatbot* berbasis *machine learning*. Dalam proses pengembangannya, *chatbot* ini menggunakan *database* MySQL untuk menyimpan data FAQ serta algoritma *fuzzy matching* untuk meningkatkan kecocokan input pengguna dengan *intent* yang relevan. Selain itu, untuk tampilan antarmuka (*frontend*) digunakan HTML, CSS, dan *Javascript* agar *chatbot* dapat dengan mudah diintegrasikan ke dalam situs resmi maupun sistem informasi milik Bapenda Kota Surabaya.

Dataset untuk *chatbot* ini dikumpulkan melalui proses wawancara langsung dengan pihak Bapenda Kota Surabaya. Wawancara dilakukan bersama dengan Bapak Edi Gunawan selaku kepala koordinator IT dan Ibu Ulifatul Ro'idah selaku tenaga administrasi. Berdasarkan hasil wawancara tersebut diperoleh 25 daftar pertanyaan dan jawaban yang akan digunakan sebagai bahan untuk pembuatan *chatbot*.

Selain berasal dari hasil wawancara yang telah dilakukan, FAQ yang sebelumnya sudah tersedia di situs resmi Bapenda Kota Surabaya juga tetap akan dimanfaatkan sebagai bagian dari penyusunan data *chatbot*. Daftar FAQ yang ada di *website* resmi Bapenda Kota Surabaya berjumlah 14 pertanyaan dan jawaban. Sehingga total keseluruhan data FAQ berjumlah 39 pertanyaan beserta jawaban. Setiap data yang telah dikumpulkan kemudian akan diidentifikasi *intent* untuk tiap pertanyaan. Untuk tiap *intent* yang memiliki tema atau topik yang serupa akan dikelompokkan menjadi satu kategori. Berdasarkan hasil pengelompokan terdapat 5 kategori FAQ.

TABEL IV
 DAFTAR KATEGORI FAQ

No	Nama Kategori	Jumlah FAQ
1	Pelayanan	12
2	Persyaratan	5
3	Pembayaran	12
4	Akun	5
5	Kendala	5

Masing-masing pertanyaan yang telah dikategorikan akan dikembangkan menjadi lima variasi kalimat yang berbeda namun tetap memiliki makna yang sama. Proses variasi ini dilakukan untuk memperkaya *dataset* dan meningkatkan kemampuan *chatbot* dalam memahami berbagai bentuk pertanyaan yang diajukan oleh pengguna dengan gaya bahasa yang berbeda-beda. Dengan menyediakan lima variasi untuk setiap pertanyaan, model *chatbot* akan memiliki referensi yang lebih luas dalam mengenali *intent* dari input pengguna yang

beragam, baik dari segi susunan kata, penggunaan sinonim, hingga gaya bertanya.

Setelah pertanyaan dikumpulkan dan dikategorikan, seluruh varian pertanyaan serta jawaban akan dimasukkan ke dalam *database*. *Database* yang dibuat disesuaikan dengan desain *class diagram* yang telah dibuat pada bab sebelumnya. Rancangan *database* ini mengikuti struktur yang menggambarkan hubungan antara data kategori FAQ, *intent*, pertanyaan dan jawaban. Setelah itu masuk ke tahap selanjutnya yaitu mengisi komponen-komponen pada dataset RASA.

Komponen pertama yaitu *nlu* yang berisi daftar *intent* yang akan digunakan pada percakapan *chatbot*. *Intent* ini akan memicu respon atau jawaban sesuai dengan inputan yang diberikan oleh pengguna. Setiap *intent* dalam file *nlu* dilengkapi dengan beberapa contoh kalimat yang merepresentasikan berbagai cara pengguna dalam mengajukan pertanyaan atau menyampaikan pernyataan terkait topik tertentu.

Komponen kedua yaitu *rules* yang merupakan aturan atau pola interaksi antara pengguna dan *chatbot*. File ini digunakan untuk mengarahkan alur percakapan berdasarkan kondisi tertentu yang telah ditetapkan. Bapenbot dibangun menggunakan total 12 *rules* berdasarkan *flow* atau alur percakapan yang telah dibuat sebelumnya.

Komponen ketiga dalam pengembangan *chatbot* adalah *story* atau skenario percakapan yang berisi alur interaksi antara pengguna dan *chatbot* berdasarkan *intent* yang telah didefinisikan sebelumnya. Di dalam file *story*, dijelaskan langkah demi langkah bagaimana *chatbot* harus bertindak setelah menerima input tertentu, termasuk pengiriman balasan, pertanyaan lanjutan, atau pilihan menu tambahan yang bisa dipilih pengguna. Berdasarkan alur cerita yang telah disusun, Bapenbot memiliki sekitar 12 kemungkinan skenario percakapan yang dapat terjadi selama interaksi dengan pengguna.

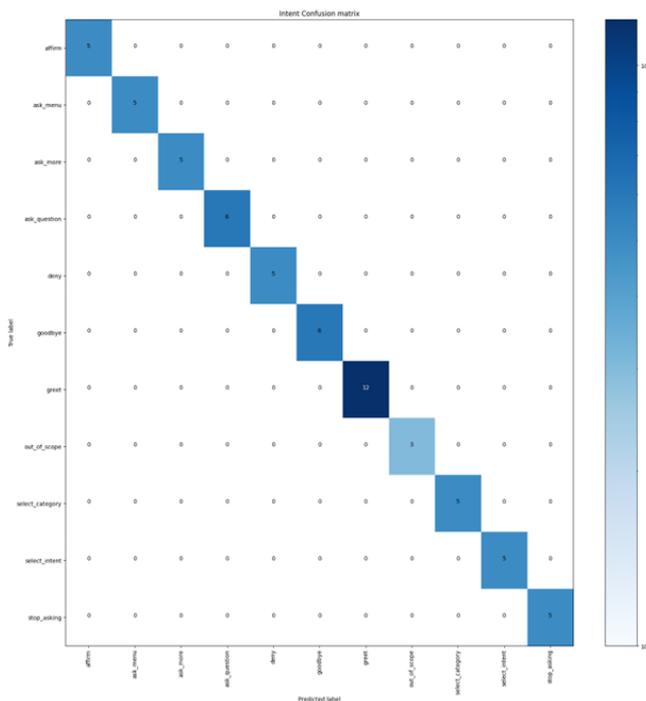
Komponen keempat yaitu *domain* yang berisi komponen-komponen utama yang digunakan oleh *chatbot* seperti daftar *intent*, entitas, aksi (*actions*). File ini berfungsi sebagai penghubung antara data pelatihan dan perilaku *chatbot*, sehingga sistem dapat memahami bagaimana merespons setiap *intent* yang diterima.

Setelah *dataset chatbot* selesai disusun, langkah selanjutnya ialah melakukan konfigurasi *chatbot* pada dua komponen sisanya yaitu pada komponen *config* dan *credentials*. Pembuatan *custom action* juga perlu dilakukan untuk menangani proses-proses dinamis tertentu seperti pengambilan data dari *database* dan pembuatan *payload button*. Pembuatan *custom actions* ini melibatkan file *actions* yang juga telah disediakan oleh *framework* RASA yang menggunakan bahasa pemrograman *python*.

Model *chatbot* yang telah selesai dibuat, selanjutnya akan dilakukan proses *train* atau pelatihan terlebih dahulu untuk membangun model *machine learning*. Pada tahap ini, file-file seperti *nlu*, *rules*, *stories*, *domain*, *config*, dan *actions* akan diproses untuk melatih *chatbot* agar dapat memahami *intent*

pengguna dan mengelola alur percakapan sesuai dengan skenario yang telah dirancang. Proses *training* ini bertujuan untuk menghasilkan model yang siap digunakan untuk pengujian dan implementasi pada *platform chatbot* yang akan dikembangkan.

Setelah proses pelatihan selesai dan model berhasil disimpan, langkah berikutnya adalah melakukan pengujian menggunakan metode pengujian bawaan dari framework RASA yaitu *NLU Test* dan *Core Test*. Pengujian NLU dilakukan untuk menguji kemampuan model dalam memahami *intent* dari input pengguna berdasarkan data yang sudah diberikan. Hasil pengujian ini akan menunjukkan seberapa akurat model dalam mengklasifikasikan *intent* dan *entity* dari sebuah kalimat.



Gbr. 16 Hasil Confusion Matrix Chatbot.

Hasil pengujian *Natural Language Understanding (NLU) test* pada *chatbot* ditunjukkan melalui *intent confusion matrix* yang dihasilkan pada gambar 16. Berdasarkan *matrix* tersebut, terlihat bahwa semua *intent* berhasil diklasifikasikan dengan benar tanpa adanya kesalahan klasifikasi antar *intent*. Hasil ini menunjukkan bahwa model NLU yang dikembangkan telah memiliki performa yang sangat baik dalam memahami dan mengklasifikasikan input pengguna. Tidak ditemukannya misklasifikasi antar *intent* yang membuktikan bahwa *pipeline NLU*, data *training*, dan *story* yang digunakan sudah cukup efektif.

```
[
2025-04-07 14:59:48 INFO rasa.core.test - Evaluation Results on ACTION level:
2025-04-07 14:59:48 INFO rasa.core.test - Correct: 105 / 111
2025-04-07 14:59:48 INFO rasa.core.test - F1-Score: 0.946
2025-04-07 14:59:48 INFO rasa.core.test - Precision: 0.954
2025-04-07 14:59:48 INFO rasa.core.test - Accuracy: 0.946
2025-04-07 14:59:48 INFO rasa.core.test - In-data fraction: 0
2025-04-07 14:59:48 INFO rasa.utils.plotting - Confusion matrix, without normalization:
[[39 0 0]]
```

Gbr. 17 Hasil Pengujian Core Test.

Berdasarkan gambar 17 diatas, hasil pengujian *Core Test* untuk *chatbot* menunjukkan performa yang cukup baik dalam menjalankan skenario percakapan (*stories*) yang telah disusun. Dari total 39 *story* yang diuji, sebanyak 33 *story* berhasil dijalankan dengan benar, menghasilkan tingkat akurasi sebesar 84,6%. Pada level aksi (*action*), *chatbot* mencatatkan performa tingkat koreksi sebesar 105 dari 111 aksi, atau setara dengan akurasi sebesar 94,6%. Selain itu, nilai *F1-Score* tercatat sebesar 0,946, dengan *precision* juga sebesar 0,954, yang menunjukkan bahwa *chatbot* mampu memilih aksi yang tepat dengan tingkat kesalahan yang rendah.

Karena hasil uji atau evaluasi model *chatbot* menunjukkan performa yang cukup baik, langkah selanjutnya yang dilakukan adalah membangun tampilan atau *frontend* pada *platform website*. Untuk membangun *frontend*, digunakan kombinasi teknologi berbasis *web* yaitu HTML, CSS, dan JavaScript. Komunikasi antara *frontend* dan *chatbot* dilakukan melalui *webhook API RASA*. *Webhook* ini memungkinkan *frontend* mengirimkan pesan pengguna ke server *chatbot* secara *real-time* menggunakan HTTP *request* dengan metode POST dan menerima balasan berupa prediksi *intent* serta respons teks yang telah dihasilkan oleh model *chatbot*.

2) Sistem Monitoring Chatbot

Sistem Monitoring *Chatbot* merupakan aplikasi berbasis web yang berfungsi untuk membantu koordinator IT maupun administrator dalam mengelola *chatbot*. Sistem ini dibangun menggunakan bahasa pemrograman C# dan menggunakan *framework ASP.Net Core*. *Framework* ini berbasis MVC (*Model-View-Controller*) yang membagi fungsionalitas sistem menjadi tiga bagian utama.

Setelah sistem monitoring selesai dikembangkan, langkah selanjutnya adalah melakukan pengujian untuk memastikan bahwa seluruh fitur berfungsi sesuai dengan kebutuhan dan spesifikasi yang telah dirancang. Pengujian dilakukan menggunakan metode *Black Box Testing* karena metode ini berfokus pada pengujian fungsionalitas sistem dari perspektif pengguna. Proses pengujian dilakukan secara langsung oleh Bapak Edi Gunawan selaku Kepala Koordinator IT di Bapenda Kota Surabaya, yang juga berperan sebagai stakeholder utama dalam proyek ini.

TABEL V
PENGUJIAN BLACK BOX TESTING

No	Komponen yang Diuji	Hasil yang Diharapkan	Kesimpulan
1	Fitur Login	Menampilkan halaman dashboard sistem jika username dan password benar	Berhasil
2	Fitur Login	Menampilkan pesan kesalahan jika username atau password salah	Berhasil
3	Menu Kategori FAQ	Menampilkan halaman menu kelola kategori FAQ	Berhasil

4	Fitur Tambah Kategori FAQ	Menyimpan kategori FAQ baru dan menampilkan pada tabel	Berhasil
5	Fitur Edit Kategori FAQ	Menyimpan perubahan yang dilakukan pada kategori FAQ	Berhasil
6	Menu Kelola FAQ	Menampilkan halaman menu kelola FAQ	Berhasil
7	Fitur Tambah FAQ	Menyimpan intent dan jawaban baru dan menampilkan pada tabel	Berhasil
8	Fitur Pertanyaan	Menyimpan daftar pertanyaan yang diinputkan	Berhasil
9	Fitur Jawaban	Mampu menyimpan perubahan yang dilakukan pada fitur jawaban	Berhasil
10	Menu Kelola Pengguna	Menampilkan halaman menu kelola pengguna	Berhasil
11	Fitur Tambah Akun	Menyimpan akun pengguna baru dan menampilkan pada tabel	Berhasil
12	Fitur Edit Pengguna	Menyimpan perubahan pada akun pengguna yang dipilih	Berhasil
13	Fitur Logout	Mengeluarkan pengguna dari sistem monitoring	Berhasil

IV. KESIMPULAN

Berdasarkan hasil penelitian dan proses pengembangan layanan informasi FAQ berbasis *chatbot* di Badan Pendapatan Daerah Kota Surabaya menggunakan *framework* RASA telah melalui tahapan analisis, pengumpulan dan pengkategorian data FAQ, penyusunan dataset, konfigurasi model, serta integrasi dengan *frontend* melalui *webhook* API dan *database*. Selain itu, sistem monitoring berbasis web juga dikembangkan menggunakan arsitektur *Model-View-Controller* (MVC) dengan bahasa pemrograman *C#* dan *framework* *ASP.Net Core* untuk memudahkan pengelolaan data oleh administrator. Hasil pengujian menunjukkan bahwa *chatbot* memiliki akurasi dengan nilai *F1-score* sebesar 0.946, nilai *precision* 0.954, dan tingkat keberhasilan *story* sebesar 84,6%. Sistem monitoring juga berjalan baik berdasarkan pengujian *black box testing* yang telah dilakukan. Dengan demikian, solusi ini efektif dalam meningkatkan efisiensi layanan informasi dan meringankan beban petugas dalam memberikan informasi kepada masyarakat khususnya para wajib pajak.

UCAPAN TERIMA KASIH

Dengan penuh rasa syukur kepada Allah SWT atas limpahan rahmat dan hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan baik. Penulis menyampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan, bantuan, dan motivasi selama proses pelaksanaan penelitian. Semoga hasil dari penelitian ini dapat memberikan kontribusi yang bermanfaat, khususnya di bidang informatika, serta menjadi bagian dari kemajuan ilmu pengetahuan dan teknologi di masa mendatang.

REFERENSI

- [1] N. Rachmatullah dan F. Purwani, "Analisis Pentingnya Digitalisasi & Infrastruktur Teknologi Informasi Dalam Institusi Pemerintahan : E-Government," *JF*, vol. 12, no. 1, pp. 14–19, Mei. 2022, doi: 10.37859/jf.v12i1.3512.
- [2] F. Kusumanigrum dan A. D. Indriyanti, "Rancang Bangun Sistem Informasi Penilaian Kepuasan Pelanggan pada Kantor Pelayanan Pajak Pratama Mojokerto dengan Menggunakan Metode Pengembangan Sistem Spiral," *J. Emerg. Inf. Syst. Bus. Intell. (JEISBI)*, vol. 3, no. 1, pp. 45–49, Januari. 2022, doi: 10.26740/jeisbi.v3i1.44329.
- [3] Bambang Suprianto, "Literature Review: Penerapan Teknologi Informasi dalam Meningkatkan Kualitas Pelayanan Publik," *jpg*, vol. 8, no. 2, pp. 123–128, Juni. 2023, doi: 10.36982/jpg.v8i2.3015.
- [4] J. Fernos dan I. Wipi, "Pengaruh Kompensasi dan Motivasi Terhadap Kinerja Karyawan pada Badan Pendapatan Daerah (BAPENDA) Kota Padang," *Jurnal Valuasi: Jurnal Ilmiah Ilmu Manajemen dan Kewirausahaan*, vol. 3, no. 2, 2023.
- [5] A. R. Purba, "Pelaksanaan Tugas dan Fungsi Badan Pendapatan Daerah Dalam Pengelolaan Pajak dan Retribusi Daerah Provinsi Sumatera Utara," *Metadata*, vol. 5, no. 2, pp. 16–29, Mei. 2023, doi: 10.47652/metadata.v5i2.366.
- [6] A. Rachman, I. Mardhiyah, dan M. Jannah, "Implementasi Chatbot FAQ pada Aplikasi Monev Kinerja Direktorat Jenderal Anggaran Menggunakan Framework Rasa Open Source," *Media Online*, vol. 4, no. 1, pp. 62–72, 2023.
- [7] S. Rubaeah et al, "SISPAC: Chatbot untuk Diagnosis dan Penanganan Hipertensi," *Seminar Nasional Dinamika Informatika*, vol. 5, 2021.
- [8] M. R. Harisi dan E. M. Hiwono, "Pengaruh Chatbot Usage terhadap Customer Satisfaction," *Jurnal Ilmiah Manajemen Ekonomi dan Akuntansi*, vol. 1, ed. Februari, pp. 66–73, 2024.
- [9] D. Wulandari dan J. S. Wibowo, "Implementasi Chatbot Menggunakan Framework Rasa Untuk Layanan Informasi Wisata Di Kota Pati," *INTECOMS*, vol. 6, no. 2, pp. 794–801, September. 2023, doi: 10.31539/intecom.v6i2.7107.
- [10] J. P. Gujjar dan V. N. Kumar, "Open Source Chatbot Development Framework – RASA," *Asian Journal of Advances in Research*, vol. 5, no. 1, pp. 451–453, 2022.
- [11] A. Annas dan S. Wibisono, "Implementasi Chatbot Menggunakan Framework Rasa Untuk Layanan Informasi Objek Wisata Di Kabupaten Pemalang," *INTECOMS*, vol. 7, no. 1, pp. 123–129, Januari. 2024, doi: 10.31539/intecom.v7i1.8425.
- [12] R. K. Sharma dan M. Joshi, "An Analytical Study and Review of open source Chatbot framework, Rasa," *IJERT*, vol. V9, no. 06, p. IJERTV9IS060723, Juni. 2020, doi: 10.17577/IJERTV9IS060723.
- [13] I. K. Resika Arthana, G. R. Dantes, L. J. E. Dewi, K. Setemen, and N. W. Marti, "Pengembangan Prototype Frequently Asked Question (FAQ) UNDIKSHA Dengan Pendekatan User Centered Design," *j. pendidik. teknologi. kejuruan.*, vol. 18, no. 1, p. 77, Januari. 2021, doi: 10.23887/jptk-undiksha.v18i1.32141.
- [14] R. Kumar dan M. M. Ali, "A Review on Chatbot Design and Implementation Techniques," *International Research Journal of Engineering and Technology*, 2020.
- [15] A. Anindyati, "Analisis dan Perancangan Aplikasi Chatbot Menggunakan Framework Rasa dan Sistem Informasi Pemeliharaan Aplikasi (Studi Kasus: Chatbot Penerimaan Mahasiswa Baru Politeknik Astra)," *JTIK*, vol. 10, no. 2, pp. 291–300, April. 2023, doi: 10.25126/jtiik.20231026409.

- [16] D. Adi, L. W. Santoso, dan A. N. Tjondrowiguno, "Pengenalan Intent pada Natural Language Understanding Berbahasa Indonesia dengan Menggunakan Metode Convolutional Neural Network," *Jurnal Infra*, vol. 8, 2020.
- [17] M. Mail, N. Zainuddin, dan N. Ningsih, "Perancangan Enterprise Architecture Menggunakan TOGAF ADM (Architecture Development Method) pada BAPENDA Kab. Kolaka," *Seminar Nasional Pemanfaatan Sains dan Teknologi Informasi*, vol. 1, no. 1, pp. 1–5, 2023.
- [18] T. E. Putri dan G. Ramadhan, "Penerapan Chatbot sebagai Alat Pembelajaran untuk Pengembangan Pendidikan Karakter," *Indonesian Journal of Computer Science and Engineering (IJCSSE)*, vol. 1, 2024.
- [19] R. E. Widiyanti and H. Yulianton, "Implementasi Chatbot Rekomendasi Kuliner Kota Semarang Dengan Framework RASA," *INTECOMS*, vol. 7, no. 4, pp. 1333–1340, Juli. 2024, doi: 10.31539/intecom.v7i4.8819.
- [20] R. C. Kusuma dan A. D. Indriyanti, "Perancangan sistem informasi building construction menggunakan metode Rapid Application Development (RAD)," *Journal of Emerging Information System and Business Intelligence (JEISBI)*, vol. 3, no. 4, pp. 64–70, 2022.