

Perancangan Aplikasi Chatbot FAQ Berbasiskan Aplikasi Android

Alheta Wahyu Matalarens*¹, Dwi Setyowati²

^{1,2,3}Jurusan Rekaayasa Sistem Komputer, Institut Sains dan Teknologi AKPRIND, Yogyakarta
e-mail: *¹alheta.matalarens@gmail.com, ²dwisetyowati@akprind.ac.id

Abstrak

Banyak pengguna di Indonesia mengalami kesulitan mendapatkan informasi terkait aplikasi yang mereka gunakan, contohnya pelanggan Traveloka kesulitan berinteraksi dengan customer support melalui E-mail dan telepon. Chatbot muncul sebagai solusi teknologi yang menggantikan interaksi manusia dalam mencari informasi, seperti yang diterapkan di platform lain seperti Shopee. Pengembangan aplikasi dilakukan menggunakan Android Studio dengan Kotlin, serta *library* seperti Retrofit untuk pengambilan data yang berbentuk JSON melalui REST API, untuk antarmuka dialog serta Text to Speech. Machine Learning dengan model tanya jawab BERT dan dataset Stanford Question Answering Dataset (SQuAD) versi 2.0 yang diterjemahkan ke bahasa Indonesia. Backend menggunakan bahasa Python dan RESTful API untuk menerima HTTP Request seperti *Post* dan *Get*, didukung oleh Terraform untuk deploy otomatis di Google Cloud Platform. Hasilnya adalah aplikasi Android bernama "Loka," memungkinkan pengguna mudah menanyakan pertanyaan melalui chatbot terkait FAQ Traveloka, memberikan solusi bagi yang enggan membaca FAQ langsung. Diharapkan solusi ini dapat mengatasi masalah komunikasi informasi aplikasi di Traveloka secara efektif melalui aplikasi yang dirancang.

Kata kunci—Chatbot, Pengembangan Aplikasi, Android

Abstract

Many users in Indonesia experience difficulties in obtaining information related to the applications they use; for instance, Traveloka customers face challenges in interacting with customer support through email and telephone. Chatbots emerge as a technological solution that replaces human interaction in seeking information, as implemented on other platforms like Shopee. The application development was carried out using Android Studio with Kotlin, along with libraries such as Retrofit for fetching data in JSON format through REST API, for the dialogue interface as well as Text to Speech functionality. Machine Learning was employed using a question-answering model like BERT, and the Stanford Question Answering Dataset (SQuAD) version 2.0 translated into Indonesian language was utilized. The backend is built using Python and RESTful API to handle HTTP Requests such as Post and Get, supported by Terraform for automatic deployment on the Google Cloud Platform. The result is an Android application named 'Loka,' enabling users to easily inquire through a chatbot regarding Traveloka's FAQs, providing a solution for those who prefer not to read the FAQs directly. This solution is expected to effectively address communication issues regarding Traveloka application information through the designed application.

Keywords—Chatbot, Application Development, Android

1. PENDAHULUAN

Pengguna aplikasi yang ada di Indonesia kebanyakan sedang mengalami masalah untuk mencari informasi mengenai

aplikasi yang mereka gunakan. Beberapa kasus pelanggan khususnya pada Traveloka, mereka kesusahan untuk mencari informasi lewat email dan telepon dari customer support.

Chatting Robot atau juga bisa disebut chatbot adalah salah satu kecanggihan teknologi chatting yang nantinya menggantikan suatu pekerjaan manusia [1]. Dengan sistem kerja yang dilakukan oleh chatbot, nantinya manusia atau user dapat dengan mudah mencari sebuah informasi yang diinginkan. Chatbot sekarang ini sudah mulai diterapkan di dalam sebuah aplikasi salah satu contohnya chat bot pada Shopee. Chatbot pertama ditulis oleh Joseph Weizenbaum, profesor MIT pada tahun 1966. Pada waktu itu tentu saja chatbot dibuat masih amat sangat sederhana [2].

Berdasarkan uraian diatas, diperlukan dikembangkan suatu aplikasi android untuk mempermudah pengguna dalam bertanya tentang FAQ atau pertanyaan yang sering ditanyakan. Pada latar belakang di atas, pembuatan laporan ini penulis memberi judul "Perancangan Aplikasi Chatbot Faq Berbasiskan Aplikasi Android pada Studi Independen Bangkit Academy 2022". Dengan ini diharapkan pihak pemberi masalah pada Company-Based Project yaitu Traveloka dapat menyelesaikan masalah yang diberikan tersebut.

2. METODE PENELITIAN

2.1 Analisis Desain Sistem

Pada beberapa kasus yang ada di Traveloka sebagai pengguna adalah mereka kesusahan untuk menanyakan setiap informasi via telepon dan email. Mengutip dari Kompas.com (2019) dengan artikel yang berjudul "Semua Ada di FAQ BKN, Pelamar CPNS 2019 Jangan Malas Baca, Ini Link-nya" dapat dinyatakan halaman FAQ pada sebuah website sebenarnya adalah metode yang kurang baik untuk bisnis dan juga masalah kita Ketika menggunakan aplikasi Traveloka sebagai customer ketika mencari informasi.

Ide dari company capstone ini penting untuk memberi solusi atau alternatif yang dapat menyelesaikan masalah dari malas membaca FAQ.

2.2 Perancangan Desain Sistem

Pada perancangan desain sistem ini terdiri dari 3 bagian kelompok yang bertugas untuk merancang sebuah aplikasi yaitu.

a. Machine Learning

Model yang dirancang adalah model tanya jawab yang berbasiskan dari model yang dari model yang dirancang oleh Cahya Irawan bernama Indonesian BERT base model (uncased), Dataset yang digunakan adalah Stanford Question Answering Dataset atau SQuAD versi 2.0 [3]. Dataset ini adalah dataset tanya jawab yang cukup populer, dataset ini berisi konteks yang digunakan model machine learning untuk memahami maksud dari pertanyaan yang ditanyakan, pertanyaan itu sendiri, dan juga jawaban [4]. Sebelum dataset tersebut digunakan, isi dari dataset tersebut perlu diterjemahkan ke bahasa indonesia dikarenakan model yang akan kami rancang menggunakan bahasa indonesia dan dataset yang digunakan wajib berbahasa Indonesia. Proses menerjemahkan dataset menggunakan library deep_translator yang digunakan untuk menghubungkan program dengan layanan terjemahan seperti Google Translate, file dataset yang berbentuk JSON akan disortir isinya dan nantinya akan diterjemahkan ke bahasa indonesia, lalu dataset yang sudah diterjemahkan akan disusun menjadi format yang digunakan oleh huggingface, yang merupakan website yang berisi banyak sekali model pre-trained dan juga dataset. Setelah dataset telah diterjemahkan dan diformat ulang maka proses transfer learning dapat dimulai [5].

b. Mobile Development

Menggunakan beberapa komponen yaitu Figma yang digunakan untuk mendesain aplikasi sebelum mendesainnya di Android Studio, kemudian ada Android Studio sebagai IDE untuk pengembangan aplikasi, menggunakan Bahasa Kotlin sebagai bahasa pengembangan dari aplikasi, menggunakan perpustakaan retrofit yang dapat digunakan untuk mengambil dan mengolah data JSON berbasiskan REST API, perpustakaan FORIS untuk menampilkan dialog peringatan dan mengimplementasikan metode *Text to Speech* berefrensikan dari Dery Sudrajat [6].

c. Cloud Computing

Backend menggunakan Bahasa Python sebagai bahasa utama dan menggunakan RESTful API untuk menerima HTTP Request seperti POST dan GET. Alasan menggunakan bahasa Python adalah penggunaannya yang mirip dengan model Machine Learning yang

sama menggunakan Python juga. Kita menggunakan Terraform untuk otomatisasi deploy infrastruktur di Google Cloud Platform agar mempunyai workflow yang konsisten dan sumber daya yang efisien untuk infrastruktur. Kita menggunakan 3 jenis Terraform yaitu Google Cloud Storage bucket creation, mendeploy aplikasi backend di Cloud Run dengan Http Load Balancer sebagai proxy dan mendeploy backend di Kubernetes Engine sebagai rencana improvisasi kedepannya [7,8]

3. HASIL DAN PEMBAHASAN

Aplikasi chatbot ini dirancang dan dibuat berdasarkan bahasa pemrograman Kotlin sebagai bahasa utama dalam pengembangan aplikasi Android. Pada pengembangan ini penulis menggunakan arsitektur Model View ViewModel (MVVM) sebagai praktek clean code [9].

Pada pengembangan aplikasi ini bagian Machine Learning menggunakan Bahasa pemrograman Python untuk melatih data yang nantinya digunakan untuk sebagai model.

Bagian Cloud Computing bertugas mendeploy hasil dari bagian Machine Learning dan mengolahnya menjadi API yang nantinya akan digunakan oleh bagian dari Mobile Development untuk mengambil hasil data yang diminta oleh aplikasi

3.1 Implementasi Figma sebagai Tampilan Aplikasi

Tahap awal pada pembangunan aplikasi ini yaitu perancangan wireframe dan mockup pada Figma yang nantinya akan di implementasikan di Android Studio sebagai layout dari setiap halaman dari aplikasi [10].

3.2 Implementasi Retrofit

Pada tahap ini kita menggunakan library tambahan pada Android Studio yang berguna untuk mengambil data pada API. Setelah data didapatkan, maka data dapat diolah dan ditampilkan sebagai View dari hasil tersebut.

3.3 Implementasi *speech to text*

Pada implementasi isi digunakan fitur *speech to text* sebagai mana dari fitur dari Accessibility Android. Ini digunakan agar pengguna dapat lebih mudah menggunakan

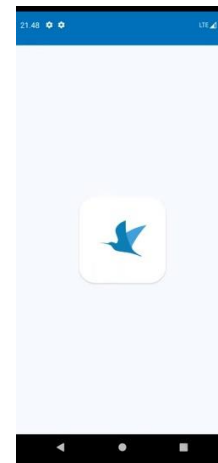
aplikasi yang hanya menggunakan suara saja dapat menjadi teks untuk chat aplikasi

3.4 Implementasi Sistem

Pada implementasi isi digunakan fitur *speech to text* sebagai mana dari fitur dari Accessibility Android. Ini digunakan agar pengguna dapat lebih mudah menggunakan aplikasi yang hanya menggunakan suara saja dapat menjadi teks untuk chat aplikasi

1) Halaman Splash Screen

Halaman ini adalah halaman yang muncul pertama kali Ketika membuka dari aplikasi ini. Pada halaman ini terdapat delay 1,5 detik guna sebagai persiapan pembukaan aplikasi ini



Gambar 1 Halaman *Splash Screen*

2) Halaman Onboarding

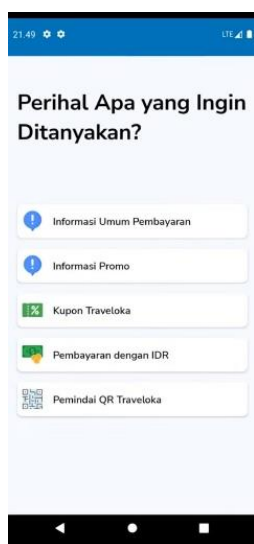
Halaman Onboarding muncul Ketika pengguna sama sekali belum membuka aplikasi ini. Untuk selanjutnya halaman Onboarding tidak akan muncul lagi karena sudah menekan tombol selanjutnya yang menyebabkan aplikasi menyimpan data.



Gambar 2 Halaman Onboarding

3) Halaman Kategori Pertanyaan

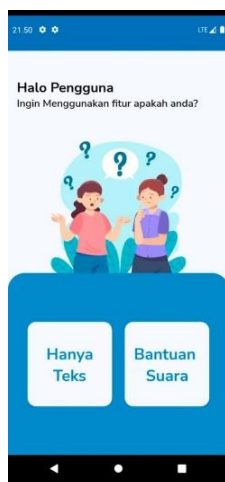
Pada halaman ini pengguna dapat kategori pertanyaan yang akan dipilih mengenai dari FAQ tersebut. Dalam mekanisme tersebut, Intent dari tiap tombol memiliki id extra tersendiri yang nantinya akan diterima pada Activity tujuan yang berfungsi sesuai dari kategori pertanyaan yang dipilih



Gambar 3 Halaman pemilihan kategori pertanyaan

4) Halaman Pemilihan Fitur Accessibility

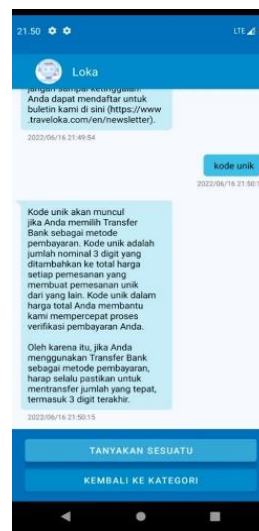
Pada halaman ini pengguna harus memilih menggunakan fitur Accessibility speech to text atau tidak. Mekanisme ini dapat terbuat berkat fitur Accessibility yang ada pada sistem operasi Android. Per tombol ini juga memiliki id unique yang nanti akan diberikan kepada tujuan Activity agar bisa terapkan.



Gambar 4 Halaman pemilihan Accessibility

5) Halaman Main Chat

Halaman ini merupakan tampilan utama dari aplikasi ini. Disini pengguna dapat mulai mengetikkan pertanyaan atau dapat menggunakan fitur speech to text yang berguna Ketika pengguna membutuhkannya



Gambar 5 Halaman Chat

3.4 Pengujian dan Hasil Analisis

Pada pengujian sistem aplikasi Chatbot yang dikembangkan ini dilakukan menggunakan metode UI Test. Metode ini yang merupakan dari library Espresso dapat digunakan untuk menguji apakah UI yang ditampilkan tersebut sesuai apa atau tidak yang diharapkan. Framework pengujian Espresso, yang disediakan oleh AndroidX Test, menyediakan API untuk menulis pengujian UI guna mensimulasikan interaksi pengguna dalam satu aplikasi target. Pengujian Espresso dapat dijalankan di perangkat yang menjalankan Android 2.3.3 (API level 10) dan yang lebih tinggi. Manfaat utama menggunakan Espresso adalah framework pengujian ini menyediakan sinkronisasi tindakan pengujian secara otomatis dengan UI aplikasi yang sedang Anda uji. Espresso mendeteksi kapan thread utama tidak digunakan, sehingga dapat menjalankan perintah pengujian Anda pada waktu yang tepat, yang akan meningkatkan keandalan pengujian (Developer Android, 2022)

Hasil dari tabel pengujian berikut ini :

Tabel 1 Uji UI Test

No	Pengujian	Test Case	Ekspetasi	Hasil
1	Pemilihan kategori pertanyaan	Android mensimulasikan sebagai pengguna menklik salah satu dari tombol semua kategori pertanyaan	Masuk ke Halaman Chat	Berhasil
2	Pemilihan accessibility	Android mensimulasikan sebagai pengguna menklik salah satu dari tombol pilihan accessibility	Masuk ke Halaman Kategori	Berhasil
3	Onboarding	Android mensimulasikan sebagai pengguna menklik tombol terakhir pada halaman Onboarding	Masuk ke Halaman Pemilihan Accessibility	Berhasil

4. KESIMPULAN

Berdasarkan perancangan, pengembangan dan pengujian aplikasi chatbot yang telah dilakukan, diperoleh kesimpulan sebagai berikut :

- 1) Aplikasi chatbot ini menggunakan Bahasa Kotlin sebagai Bahasa utama dalam pengembangannya yang direkomendasi oleh Google menjadikannya pengembangan ini lebih efisien
- 2) Aplikasi chatbot ini dapat menggunakan fitur accessibility berupa speech to text yang memudahkan pengguna dalam menggunakan aplikasi ini
- 3) Pengguna dapat menggunakan aplikasi ini Ketika ingin mencari suatu informasi

Pertanyaan yang Sering Ditanyakan yang lebih cepat, efektif dan Efisien daripada mencari secara manual pada website

- 4) Pengujian UI Test menghasilkan sistem yang sesuai dengan UI yang diharapkan Ketika pengguna menggunakan aplikasi ini

5. SARAN

Aplikasi ini masih terdapat UI yang warnanya belum sesuai Ketika mode gelap dari android diaktifkan yang membuat UI ini kurang menarik

UCAPAN TERIMA KASIH

Terimakasih kepada Bangkit Academy yang telah memberikan capstone akhir yang berupa studi kasus masalah pelanggan pada company Traveloka

DAFTAR PUSTAKA

- [1] Adamopoulou, Eleni. "An Overview of Chatbot Technology" AIAI 2020: Artificial Intelligence Applications and Innovations, (2020): 373.
- [2] Nama belakang penulis, nama depan penulis. "Chatbot for E-Learning: A Case of Study" ternational Journal of Mechanical Engineering and Robotics Research Vol. 7, No 2 (2018): 528.
- [3] Rondeau, Marc. "Systematic Error Analysis of the Stanford Question Answering Dataset" Proceedings of the Workshop on Machine Reading for Question Answering, W18-2602 (2018): 12
- [4] Mahesh, Batta. "Machine Learning Algorithms – A Review." International Journal of Science and Research (IJSR) (2018): 381.
- [5] Bourhis, Pierre. "JSON: Data model and query languages" Information Systems volume 89 (2020)
- [6] Oliveira, Victor. "On the Adoption of Kotlin on Android Development: A Triangulation Study" 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER) (2020)
- [7] Rashid, Aaqib. "Cloud Computing Characteristics and Services: A Brief Review" International Journal of Computer Sciences and Engineering Volume 7, Issue-2 (2019)

- [8] Srivastava, Pristavu. "A Review Paper on Cloud Computing" International Journals of Advanced Research in Computer Science and Software Engineering Volume 8, Issue-6 (2018)
- [9] Arif, Muhammad. "Implementasi Pola Arsitektur Model-View-ViewModel (MVVM) pada Sistem Informasi Akademik Universitas Darussalam Gontor Berbasis Mobile" (2019)
- [10] Muhammad, Muhyidin. "Perancangan Ui/Ux Aplikasi My Cic Layanan Informasi Akademik Mahasiswa Menggunakan Aplikasi Figma" Vol 10, No 2 (2020):210
- [11] Wikipedia, 2022. Android (sistem operasi) – Wikipedia Bahasa Indonesia, ensiklopedia bebas. [online] Wikipedia. Available at:
<[https://id.wikipedia.org/wiki/Android_\(sistem_operasi\)](https://id.wikipedia.org/wiki/Android_(sistem_operasi))> [Accessed 31 July 2022].
- [12] Wikipedia, 2022. Kotlin (Bahasa Pemrograman) – Wikipedia Bahasa Indonesia, ensiklopedia bebas. [online] Wikipedia. Available at:
<[https://id.wikipedia.org/wiki/Kotlin_\(bahasa_pemrograman\)](https://id.wikipedia.org/wiki/Kotlin_(bahasa_pemrograman))> [Accessed 31 July 2022].
- [13] Wikipedia, 2022. Github – Wikipedia Bahasa Indonesia, ensiklopedia bebas. [online] Wikipedia. Available at:
<<https://id.wikipedia.org/wiki/GitHub/>> [Accessed 31 July 2022].
- [14] Wikipedia, 2022. Android Studio - Wikipedia. [online] Wikipedia. Available at:
<https://en.wikipedia.org/wiki/Android_Studio/> [Accessed 31 July 2022].
- [15] Android. D., 2022. Menguji UI untuk satu aplikasi | Developer Android | Android Developers. developers. Available at:
<<https://developer.android.com/training/testing/ui-testing/espresso-testing?hl=id/>> [Accessed 31 July 2022].
-