

BAB II

TINJAUAN PUSTAKA

2.1. Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (instruction) atau pernyataan (statement) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output (Jogiyanto, 2008). Gabungan dari beberapa aplikasi biasa disebut dengan paket atau application suite. Sebagai contoh Microsoft Office, dimana terdapat aplikasi pengolah kata, lembar kerja, pengolah angka, dan beberapa aplikasi lain dalam satu paket.

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut, dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut. Aplikasi mempunyai arti yaitu, pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi, yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan, maupun pemrosesan data yang diharapkan. Pengertian aplikasi secara umum adalah, alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya. Aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user (Abdurahman & Riswaya, 2014).

2.2. Android

Android adalah sebuah kumpulan perangkat lunak untuk perangkat mobile yang mencakup sistem operasi, middleware dan aplikasi utama mobile. (Safaat, 2012). Menurut Meier (2012).

Android adalah sebuah software open source yang termasuk sistem operasi, middleware, dan kunci aplikasi mobile beserta sekumpulan library API untuk menulis sebuah aplikasi yang dapat membentuk tampilan, rasa, dan fungsi pada perangkat yang menjalankannya.

Seiring dengan perkembangan jaman, Android bertransformasi menjadi platform yang senantiasa bergerak cepat dalam hal inovasi. Tidak dapat dipungkiri lagi bahwa pembuat dari Android adalah Google, sehingga hal ini dapat terjadi. Dengan melakukan akuisisi Android, Google kemudian mengembangkan

sebuah platform tersebut. Didalam Android sendiri terdapat OS berbasis linux, Graphic User Interface (GUI), serta web browser dan aplikasi end user yang bisa diunduh, juga bisa terdapat aplikasi open source yang bisa dimodifikasi oleh berbagai user.

2.3. Android Studio

Android Studio merupakan Integrated Development Environment (IDE) yang berguna untuk pengembangan aplikasi Android, diluncurkan tanggal 16 Mei 2013 pada Google I/O conference oleh Android.



Gambar 2.1 Logo Android Studio

Android Studio sendiri menggunakan bahasa Java dan berbasis IntelliJ IDEA dari JetBrains. Android Studio memiliki banyak fitur yang berguna untuk meningkatkan produktivitas dalam membangun aplikasi Android, serta memiliki code editor dan tools pengembangan IntelliJ yang kuat. Beberapa fitur tersebut, diantaranya:

1. Build system dengan basis Gradle yang dinamis sesuai dengan kebutuhan.
2. *Emulator* yang dapat berjalan dengan cepat dan memiliki fitur lengkap.
3. Kompatibel untuk semua perangkat *Android*.
4. Alat pengujian serta *framework* yang ekstensif.
5. *Instant Run* yang memudahkan *developer* untuk menjalankan aplikasi secara instan tanpa membangun APK baru.
6. Lint yang dapat meningkatkan kinerja, fungsi, kompatibilitas versi dan masalah yang lain.
7. Code Template dan integrasi GitHub yang bisa memudahkan developer dalam mengembangkan fitur aplikasi secara umum dan melakukan

import, contoh: code.

8. Adanya fitur C++ dan NDK.
9. Built-in support untuk Google Cloud Platform.

(Android, 2020)

2.4. Firebase

Firebase berfokus untuk menyediakan basis data secara realtime dan backend, sebagai layanan (Backend as a Service). Terdapat API yang berfungsi untuk melakukan sinkronisasi antara aplikasi data klien, untuk kemudian disimpan pada cloud Firebase. Tidak hanya itu, Firebase mempunyai sebuah library untuk berbagai macam platform klien, yang berfungsi untuk mengintegrasikan Android, iOS, JavaScript, Java, Objective-C dan Node Js, serta dapat biasa disebut dengan suatu layanan DbaaS (Database as a Service) dengan konsep realtime.



Gambar 2.2 Logo Firebase

Semua data Firebase Realtime Database disimpan sebagai objek JSON. Bisa dianggap basis data sebagai JSON tree yang di-host di cloud. Tidak seperti basis data SQL, firebase tidak ada tabel atau rekaman. Ketika ditambahkan ke JSON tree, data akan menjadi simpul dalam struktur JSON yang ada. Meskipun basis data menggunakan JSON tree, data yang tersimpan dalam basis data bisa diwakili sebagai tipe bawaan tertentu yang sesuai dengan tipe JSON yang tersedia, untuk membantu menulis lebih banyak kode yang bisa dipertahankan (Firebase Realtime Database, 2020).

2.5. NLP

Natural Language Processing (NLP) merupakan salah satu cabang ilmu AI yang berfokus pada pengolahan bahasa natural. Bahasa natural adalah bahasa yang secara umum digunakan oleh manusia dalam berkomunikasi satu sama lain.

Penelitian menggunakan pendekatan leksikon dan NLP sebelumnya telah digunakan untuk penyaringan gejala depresi pada pengguna Twitter, hasilnya diperoleh bahwa pendekatan leksikon dan NLP memiliki akurasi lebih baik dari pada menggunakan pendekatan machine.

2.6. Chatbot

Chatbot adalah sebuah program komputer yang bertujuan untuk mensimulasikan sebuah kecerdasan buatan untuk dapat melakukan sebuah percakapan dengan manusia (Shawar & Atwell, 2002). Chatbot adalah sebuah implementasi dari bidang ilmu pengolahan bahasa alami, pembelajaran mesin, rekayasa perangkat lunak dan kecerdasan buatan. Sebuah chatbot akan dianggap menggunakan sebuah kecerdasan buatan, ketika masukan bahasa alami terdapat beberapa kata yang susunannya tidak sama pada sebuah kalimat, namun program mampu mempelajari bahwa masukan tersebut memiliki arti yang sama dengan masukan sebelumnya, sehingga program mampu merespon dengan balasan yang sesuai.

Chatbot dapat dikembangkan sesuai dengan kebutuhan yang ada, namun setiap chatbot yang dikembangkan memiliki tipe tersendiri dalam mengeksekusi masukan pengguna, berikut adalah tipe – tipe chatbot:

- a. **Button-Based Chatbot** Button-based chatbot adalah tipe chatbot yang masukan pengguna berdasarkan tombol perintah yang telah disediakan oleh chatbot tersebut. Tombol perintah yang disediakan oleh sebuah button-based chatbot adalah sebuah hirarki pohon keputusan. Pada button-based chatbot ini knowledge base yang disiapkan oleh pengembang tidak terlalu banyak, karena masukan pengguna hanya berdasarkan tombol yang disediakan oleh sistem. Kelemahan dari tipe chatbot ini adalah pengguna tidak dapat memberi masukan atau pertanyaan dengan leluasa, karena setiap tombol telah diatur dengan sebuah query untuk sebuah masukan.
- b. **Keyword Recognition-Based Chatbot** Keyword recognition-based chatbot adalah tipe sebuah chatbot yang merespon masukan pengguna dengan cara mengenali beberapa kata kunci yang ada pada masukan pengguna. Masukan yang diterima oleh chatbot jenis ini berupa kalimat atau susunan kata. Cara

keyword recognition-based chatbot menentukan sebuah respon atau balasan yang tepat kepada sebuah masukan yang telah diberikan adalah mengecek setiap kata sampai menemukan kata kunci yang cocok dengan knowledge base pada sistem. Tipe chatbot ini mempunyai kelemahan yaitu ketika merespon sebuah masukan yang mempunyai kata kunci yang sama.

- c. Contextual Chatbot Contextual chatbot adalah sebuah tipe chatbot yang memanfaatkan machine learning dan artificial intelligent untuk mengolah percakapan sistem dengan pengguna. Contextual chatbot dalam menerima sebuah masukan sama dengan keyword recognition-based chatbot yaitu menerima sebuah masukan berupa susunan kata atau kalimat. Perbedaan contextual chatbot dengan tipe chatbot yang lain yaitu setiap masukan pengguna akan diingat dan dipelajari lagi sehingga ketika ada sebuah masukan dengan susunan kata berbeda namun memiliki arti yang sama, sistem mampu memberi respon dengan tepat. Contextual chatbot memerlukan pengembangan knowledge base yang harus dikembangkan secara berkala, agar dapat mempelajari setiap masukan pengguna.

2.7. Dialogflow

Dialogflow (sebelumnya Api.ai, Speaktait) adalah pengembang teknologi interaksi manusia-komputer milik Google yang didasarkan pada percakapan bahasa alami. Perusahaan ini terkenal karena menciptakan Asisten (oleh Speaktait), teman virtual untuk smartphone Android, iOS, dan Windows Phone yang melakukan tugas dan menjawab pertanyaan pengguna dalam bahasa alami (Wikipedia).

Intent adalah tindakan khusus yang dapat dilakukan pengguna dengan menggunakan salah satu istilah yang ditentukan di konsol Dialogflow. Misalnya, pengguna dapat bertanya "Apakah akan turun hujan hari ini?" Atau "Di mana restoran pizza terdekat" - jika ini adalah istilah yang didefinisikan dalam konsol, maka mereka akan terdeteksi oleh Dialogflow dan maksud yang ditentukan di bawah kehendak dipicu.

Entitas adalah mekanisme Dialogflow untuk mengidentifikasi dan mengekstraksi data yang berguna dari input bahasa alami. Meskipun maksud memungkinkan agen untuk memahami motivasi di balik input pengguna tertentu,

entitas digunakan untuk memilih informasi tertentu yang disebutkan pengguna - apa pun dari alamat jalan sampai nama produk atau jumlah dengan unit. Setiap data penting yang ingin didapatkan dari permintaan pengguna akan memiliki entitas yang sesuai.

Fulfillment adalah fitur dari dialogflow untuk menghubungkan layanan ke server sendiri. Menghubungkan layanan Anda memungkinkan Anda untuk 11 mengambil tindakan berdasarkan end-user expressions dan mengirim respons dinamis kembali ke end-user. Misalnya, jika pengguna ingin menjadwalkan potong rambut pada hari Jumat, layanan dapat memeriksa database dari server kita dan menanggapi pengguna dengan informasi ketersediaan untuk hari Jumat.

2.8. UML

Menurut Zheng & Zhao (2014), UML adalah, suatu bahasa pemodelan dengan standar umum, yang berisikan suatu metode pemodelan statis dan dinamis. Metode pemodelan statis mendefinisikan atribut dan operasi objek-objek penting dalam sistem dan hubungan antara objek. Sedangkan, metode pemodelan dinamis mendefinisikan, karakteristik waktu objek dan mekanisme komunikasi untuk menyelesaikan tujuan dan tugas. Sedangkan menurut Lee (2012), Unified Modelling Language (UML) adalah suatu bahasa pemodelan visual yang berfungsi untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML adalah alat untuk menentukan dan memvisualisasikan sistem perangkat lunak. UML termasuk jenis diagram standar yang menggambarkan dan memetakan secara visual aplikasi komputer atau desain dan struktur sistem database. Terdapat beberapa diagram yang disediakan dalam UML, dibawah ini. Beberapa diagram yang disediakan dalam UML, antara lain:

1. Diagram use case bisnis,
2. Diagram use case,
3. Diagram aktivitas,
4. Diagram sekuensial,
5. Diagram kolaborasi,
6. Diagram kelas,
7. Diagram statechart,

8. Diagram komponen,
9. Diagram deployment.

(Sholiq & Robandi, 2010)

2.9. Diagram Use case

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case mempresentasikan sebuah interaksi antara actor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja dan sebagainya. Seorang actor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untk melakukan pekerjaan – pekerjaan tertentu (Suhendrik, 2011).

Menurut Sholiq & Robandi (2010), diagram use case menyajikan interaksi antara use case dan actor dalam sistem yang dikembangkan. Use case adalah fungsionalitas atau persyaratan sistem yang harus dipenuhi oleh sistem yang akan dikembangkan menurut pandangan pengguna sistem.

Use case adalah sebuah deskripsi dari kemungkinan urutan interaksi yang terjadi antara sistem dengan actor eksternal, terkait dengan tujuan tertentu yang ingin dicapai. Dengan menggunakan Use Case, akan lebih mudah untuk memahami interaksi antara actor dengan sistem yang akan dibuat (Cockburn, 2000).

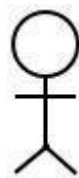
Di dalam diagram use case terdapat beberapa simbol-simbol seperti di bawah ini:

Use Case, pekerjaan tertentu di dalam sistem.



Gambar 2.3 Simbol use case

Actor, entitas manusia atau mesin yang berinteraksi dengan sistem.



Gambar 2.4 Simbol Actor

2.10. Diagram Aktivitas

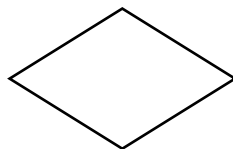
Menurut Suhendrik (2011), diagram aktivitas menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, proses yang mungkin terjadi dan bagaimana mereka berakhir. Sedangkan menurut Havaluddin (2011), activity diagram menggambarkan aktifitas-aktifitas, objek, state, transisi state dan event. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas. Di dalam diagram aktivitas terdapat beberapa simbol-simbol seperti di bawah ini:

1. Activity, menggambarkan interaksi antarmuka.



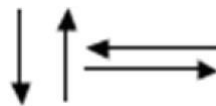
Gambar 2.5 Simbol Activity

2. Decision, suatu tindakan yang harus diambil pada kondisi tertentu.



Gambar 2.6 Simbol Decision

3. Line Connector, penghubung suatu simbol dengan simbol lainnya.



Gambar 2.7 Simbol Line Connector

4. Initial State, awalan aliran aktivitas



Gambar 2.8 Simbol Initial State

5. Final State, akhir aliran aktivitas



Gambar 2.9 Simbol Final State

2.11. Diagram Sekunsial

Diagram sekuensial menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case, interaksi yang terlibat, urutan antar operasi dan informasi yang diperlukan oleh masing-masing operasi (Suhendrik, 2011). Menurut Koutbi & Jakimi (2009), diagram sekuensial menjelaskan interaksi obyek yang disusun dalam suatu urutan waktu. Urutan waktu tersebut yaitu urutan kejadian yang dilakukan oleh seorang aktor yang menjalankan sistem. Diagram sekuensial memiliki kesederhanaan dan penggunaan yang luas dalam domain yang berbeda.

Di dalam diagram sekuensial terdapat beberapa simbol-simbol seperti di bawah ini:

1. Actor, menggambarkan orang yang sedang berinteraksi dengan sistem.
2. Entity Class, menggambarkan hubungan kegiatan yang dilakukan.
3. Boundary Class, memodelkan interaksi antara satu atau lebih aktor dengan sistem
4. Control Class, menggambarkan penghubung antara boundary dengan tebal.
5. Life Line, menggambarkan tempat mulai dan berakhirnya sebuah message.
Message, menggambarkan pengiriman pesan.

2.12. Diagram Kelas

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. Class diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, class

diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat (Haviluddin, 2011).