

APLIKASI *CHATBOT* MENGGUNAKAN *DIALOGFLOW API* UNTUK INFORMASI JADWAL MISA DI GEREJA KATOLIK BERBASIS ANDROID

Eko Prasetyo Sutono¹, Alif Finandhita²

^{1,2}Program Studi Teknik Informatika, Universitas Komputer Indonesia
Jl. Dipatiukur Nomor 102-116 Bandung
Eko10117907@email.unikom.ac.id¹, alif.finandhita@email.unikom.ac.id²

ABSTRAK

Penelitian ini bertujuan untuk mengetahui bagaimana efektivitas dalam penyampaian informasi jadwal misa kepada umat Katolik. Metode penelitian ini menggunakan metode kualitatif dengan pendekatan deskriptif yaitu memaparkan pengolahan data jadwal misa, data gereja dan data pastor pada aplikasi yang akan dibangun. Teknik pengumpulan data yang digunakan ialah observasi, wawancara dan studi kepustakaan yang berkaitan dengan aplikasi yang dibangun, dan metode perancangan yang digunakan adalah metode *waterfall* dengan menggunakan diagram *UML (Unified Modelling Language)* sebagai metode *software* dan juga diimplementasikan dengan bahasa pemrograman *Java*. Permasalahan yang terjadi adalah penyampaian atau media yang digunakan dalam menyampaikan informasi jadwal misa kepada umat Katolik yaitu dengan memasang lembaran pengumuman yang dipajang di papan informasi yang masih kurang efektif. Hasil kuesioner yang disebarkan kepada beberapa responden dapat diambil solusi untuk menyelesaikan permasalahan tersebut yaitu membangun suatu aplikasi *mobile* berupa aplikasi *chatbot* dengan implementasi dari teknologi *Google* yaitu *Dialogflow API*, yang dapat berinteraksi dengan umat menggunakan bahasa sehari-hari. Aplikasi *chatbot* ini dapat memberikan suatu informasi kepada umat Katolik tentang jadwal misa yang berlokasi di gereja Keuskupan Bandung.

Kata Kunci : *Chatbot, Dialogflow API, Android, Jadwal Misa, Gereja Katolik*

1. PENDAHULUAN

Misa merupakan yang berasal dari bahasa latin yaitu *missa*. Kata misa menjadi sebutan yang sangat populer di telinga umat Katolik untuk perayaan Ekaristi pada gereja Katolik. Kata misa juga dapat

digunakan dalam perayaan liturgi lain, doa-doa, unsur-unsur perayaan sakramen tobat, bacaan-bacaan ataupun ibadat harian. Kemudian kata misa dapat diartikan dengan pembubaran jemaat atau umat yang telah selesai dalam suatu pertemuan atau perayaan liturgi atau ekaristi [1]. Di setiap gereja Katolik atau disebut dengan paroki bahwa setiap jadwal misa berbeda-beda, baik pada jadwal misa harian ataupun jadwal misa mingguan, hal ini disesuaikan dengan jumlah umat Katolik yang di sekitar paroki atau gereja Katolik dan jumlah Pastor yang bertugas pelayanan di paroki tersebut. Jadwal misa harian maupun minggu ditentukan oleh masing-masing kepada Pastoral dan Pastor-Pastor yang bertugas pelayanan di paroki tersebut. Di dalam penyampain suatu informasi seputar jadwal misa kepada umat Katolik masih kurang efektif, dikarenakan media yang dipakaian untuk menyebarkan informasi hanya dengan menempelkan selebaran kertas pengumuman di papan pengumuman, maka tidak semua umat Katolik mengetahui apabila ada perubahan pada jadwal misa atau pada jadwal misa besar seperti perayaan Paskah dan Natal. Dengan dibangun suatu aplikasi *mobile* berbasis Android, dengan penyebaran kuesioner kepada umat Katolik dapat diambil suatu kesimpulan bahwa untuk saat ini banyak menggunakan *smartphone* berbasis Android. Dengan pertimbangan inilah akan dibangun suatu aplikasi sebagai solusi untuk mengatasi permasalahan tersebut. Untuk melakukan proses tahapan yang digunakan untuk permasalahan yang terjadi, maka metodologi penelitian deskriptif dianggap paling tepat, karena penelitian ini bertujuan supaya fenomena yang sedang terjadi dapat jelas tergambar. Dengan metode pengumpulan data dengan studi literature dengan mempelajari sumber kepustakaan dan referensi buku, jurnal dan hasil penelitian lain. Wawancara dimana tahap ini melakukan tanya jawab langsung dengan pihak yang terkait dengan penelitian yang dilakukan. Penelitian yang menjadi acuan dalam menyelesaikan permasalahan yang terjadi ini yaitu aplikasi *chatbot* untuk informasi objek wisata kota Bandung dengan menggunakan metode *NLP (Natural Language Processing)*. Dimana aplikasi yang dibangun dengan menggunakan bahasa sehari-hari atau bahasa alami

(*Natural Language*) untuk media berkomunikasi antar sesama [2].

2. LANDASAN TEORI

2.1 Chatbot

Chatbot adalah suatu program komputer yang diprogram untuk dapat berinteraksi antara manusia dengan komputer dengan menggunakan bahasa sehari-hari. Contoh kongkritnya seperti *Help Bot* pada Yahoo! Mesengger dan ALICE (*Artificial Linguistic Internet Computer Entity*) yang dikembangkan oleh Dr. Richard S. Wallace. *Chatbot* merupakan *QA system* atau *question answering system*, yang memeberikan kemampuan pada sebuah komputer untuk menginterpretasikan bahasa alami untuk melakukan percakapan dengan manusia atau pengguna, percakapan yang terjadi seperti dua manusia sedang berdialog [3].

2.2 Dialogflow API

Dialogflow API ialah suatu teknologi *Google* yang dapat berinteraksi manusia dengan komputer yang berbasis pada percakapan suara dan teks yang didukung dengan kecerdasan buatan (*Artificial Intelegence*). *Dialogflow API* merupakan milik *Google* teknologi pengembangan interaksi antara manusia atau pengguna dengan komputer dengan menggunakan bahasa manusia atau dengan bahasa sehari-hari. *Dialogflow API* menyediakan satu *platform* yang dapat memungkinkan pengembang untuk merancangn dan mengimpementasikan antarmuka percakapan yang dapat disematkan aplikasi eksternal seperti *bot* [4].

2.3 Natural Language Processing (NLP)

Natural Language Processing (NLP) adalah suatu bentuk representasi dari suatu pesan yang ini dikomunikasikan antar manusia atau pengguna. Salah bentuk dari respresentasinya yakni berupa bentuk suara atau ucapan (*spoken language*), akan tetapi sering diaplikasikan ke dalam bentuk teks atau tulisan berupa pertanyaan dari pengguna. Dalam bahasa dapat dibedakan yaitu bahasa alami dan bahasa buatan. Bahasa alami adalah bahasa yang sering digunakan dalam sehari-hari, dan bahasa buatan yaitu bahasa yang dibuat secara khusus untuk memenuhi kebutuhan tertentu seperti bahasa pemrograman [3].

2.3 Android

Android ialah suatu sistem operasi untuk *Smartphone* dan tablet. Sistem operasi yang dapat diilustrasikan sebagai jembatan antara piranti (*device*) dan penggunaanya, sehingga dapat berinteraksi dengan *devicenya* dan menjalankan aplikasi-aplikasi yang tersedia pada *device*. Di dunia personal komputer, sistem operasi yang banyak dipakai adalah *Windows*, *MAC*, dan *Linux* [5].

2.4 Komponen Aplikasi Android

Pemrograman yang menggunakan bahasa pemrograman Java, adalah bahasa pemrograman yang digunakan untuk penulisan bahasa pemrograman dalam membangun suatu aplikasi *android*. Kode-kode Java akan dikompilasi dengan semua *file-file* yang dibutuhkan yang nantinya dimana proses tersebut dikemas oleh alat yang bernama *APT tools* sehingga akan menghasilkan *file* dengan ekstensi *apk* pada paket *android* itu. *File apk* itu yang disebut adalah aplikasi yang nantinya dapat dilakukan proses pemasangan pada perangkat *android mobile* [6]

2.5 Bahasa Pemrograman Java

Java merupakan suatu bahasa pemrograman yang cukup terkenal. Sebagai bahasa pemrograman, *Java* dapat digunakan untuk menulis suatu program. Sebagaimana diketahui, program ialah kumpulan intruksi yang diajukan untuk komputer. Dengan melalui program, komputer dapat diatur agar melaksanakan tugas tertentu sesuai dengan yang ditentukan oleh pemrogram (*programmer*). Bahasa *Java* dikembangkan di *Sun Microsystems* dan mulai diliris kepada masyarakat luas pada tahun 1995. Contoh bahasa pemrogram yang berorientasi objek antara lain *C++* dan *Java*. *Java* dapat memudahkan dalam pembangunan suatu aplikasi dengan skala besar [7]. Perkembangan selanjutnya dari bahasa *Java* adalah sebagai bahasa untuk menyaingin gagasan *Mircrosoft* dalam *Distrbuted (Objects) Application* pada proyek DNA (*Distributed interNet Application*), *Sun* yang dibantu oleh perusahaan asing pesaing *Microsoft* lain telah mengeluarkan *J2EE (Java Platform 2rd Enterprise Edition)*. *Java* dimaksudkan sebagai bahasa Inggris untuk suatu pembangunan *Java Applet*, *Java Script*, *Java Servlet*, *Java Server Pages (JSP)*, khusus pada arsitektur *N-tier* yang dikembangkan untuk aplikasi tersebar dengan skala *enterprise* [8].

2.6 Database

Database ialah suatu struktur yang umumnya terbagi dalam dua hal, yakni sebuah *database flat* dan sebuah *database relasional*. *Database relasional* lebih mudah dipahami daripada *database flat* karena *database relasional* mempunyai bentuk yang sederhana serta mudah dilakukan operasi data. *MySQL* sendiri adalah sebuah *datbase relasional*. *Database* yang memiliki struktur relasional terdapat tabel-tabel untuk menyimpan data. Setiap tabel terdiri dari kolom dan baris serta sebuah kolom untuk mendefinisikan jenis informasi apa yang harus disimpan [9].

2.6 Unified Modeling Language (UML)

UML (Unified Modeling Language) adalah bahasa grafis untuk mendokumentasikan, mespesdifikasikan dan membangun sistem perangkat lunak. *UML* beroorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan

teknologi, pemanduan beberapa notasi di beragam metodologi, usaha bersama dari pihak, didukung oleh kakas-kakas yang diintegrasikan lewat XML (XMI). Standar UML dikelola oleh OMB (*Object Management Group*).

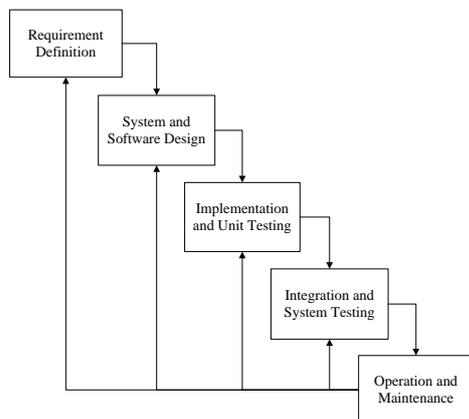
UML adalah bahasa pemodelan untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan artefak-artefak dari sistem [8]

Untuk memvisualisasi, menspesifikasikan, membangun dan mendokumentasikan aspek statik dari sistem. Diagram struktur di UML terdiri dari :

1. *Class Diagram* menunjukkan sekumpulan *class*, *interface* dan kolaborasi dan keterhubungannya. Diagram kelas ditunjukkan untuk pandangan statik terhadap sistem.
2. *Use Case Diagram* menunjukkan sekumpulan objek dan keterhubungannya. Diagram ini menunjukkan potongan statik dari instan-instan yang ada di diagram kelas. Diagram ini untuk memperlihatkan suatu protipe atau kasus tertentu yang mungkin terjadi.
3. *Sequence Diagram* menunjukkan interaksi yang terjadi antar objek. Diagram ini merupakan pandangan dinamis terhadap sistem. Diagram ini menekankan pada basis keberurutan waktu dari pesan-pesan yang terjadi.
4. *Activity Diagram* menunjukkan aliran aktivitas di sistem. Diagram ini adalah pandangan dinamis terhadap sistem. Diagram ini penting untuk memodelkan fungsi sistem dan menekankan pada aliran kendali di antara objek-objek [8].

3. METODE PENELITIAN

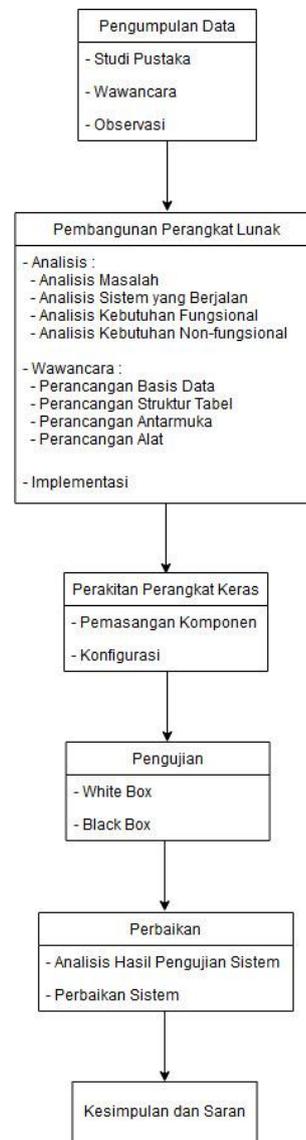
Metodologi yang digunakan dalam penelitian ini adalah deskriptif. Metode deskriptif merupakan suatu metode penelitian yang bertujuan untuk mendapatkan gambaran yang jelas tentang hal-hal yang dibutuhkan dan berusaha menggambarkan serta menginterpretasi objek yang sesuai dengan fakta secara sistematis, faktual dan akurat. Metode penelitian dapat dilihat pada gambar 1 berikut :



Gambar 1 Waterfall Model Ian Sommerville [10]

Metode pengumpulan data pada penelitian ini adalah yang pertama dengan menggunakan studi literature yang merupakan metode pengumpulan data dengan mempelajari dari sumber kepustakaan seperti buku referensi, hasil penelitian lain dan jurnal-jurnal terkait. Dan yang kedua adalah dengan melakukan wawancara, dimana pada tahap ini mengumpulkan data-data dengan cara melakukan tanya jawab secara langsung dengan pihak terkait mengenai permasalahan yang dibahas.

Dan tahapan-tahapan dalam perancangan aplikasi yang akan dibangun dapat dilihat pada gambar 2. Pada gambar alir tersebut juga telah dicantumkan ringkasan mengenai langkah dari tahapan-tahapan yang akan dilakukan. Dimana dalam perancangan aplikasi yang akan dibangun memiliki tahap-tahap yang sistematis agar pembangunan aplikasi terarah. Berikut adalah tahapan-tahapan tersebut :



Gambar 2 Tahapan Perancangan Aplikasi

4. HASIL DAN PEMBAHASAN

Tujuan utama penelitian ini yaitu hasil dari pemecahan masalah yang didapat. Juga pembahasan-pembahasan yang berkaitan dengan berbagai analisis yang dilakukan, pemodelan, perancangan dan implementasi sistem yang dibangun. Lalu langkah-langkah yang dilakukan beserta hasil pengujian dari penelitian ini akan dipaparkan secara ringkas.

Yang pertama adalah analisis masalah, analisis – analisis masalah yang didapat dengan mengevaluasi pada aplikasi yang akan dibangun nantinya. Permasalahan-permasalahan yang didapat akan menjadi latar belakang dari Aplikasi *Chatbot* Menggunakan *Dialogflow API* untuk Informasi Jadwal Misa di Gereja Katolik Berbasis Android ini. Dan berikut ini permasalahan dari analisis yang telah dilakukan, media yang digunakan dalam menyampaikan informasi jadwal misa kepada umat Katolik yang masih kurang efektif.

Untuk analisis kebutuhan perangkat keras, adalah hal penting karena proses analisis pemanfaatan perangkat keras yang dapat menjalankan terhadap aplikasi yang akan dibangun. Dan untuk spesifikasi perangkat keras yang dibutuhkan dapat dilihat pada table 1 sebagai berikut.

Tabel 1 Spesifikasi Perangkat Keras Sistem *Mobile*

No.	Perangkat Keras	Kebutuhan Minimal
1	Processor	800 MHz
2	Memori	512 MB
3	Layar	4 inch
4	Perangkat Lain	Koneksi Internet

Analisis kebutuhan perangkat lunak, adalah hal yang penting pula dalam membangun aplikasi. Perangkat lunak merupakan juga suatu instruksi kepada perangkat keras agar aplikasi tersebut dapat berinteraksi diantara keduanya. Untuk spesifikasi kebutuhna perangkat lunak yang digunakan dapa dilihat pada table 2 di bawah ini :

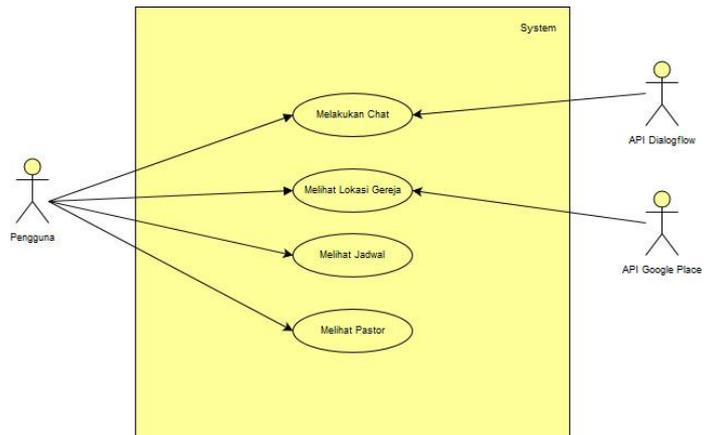
Tabel 2 Spesifikasi Kebutuhan Perangkat Lunak

Perangkat Lunak	Spesifikasi
IDE (<i>Intergrated Development Environment</i>)	Android Studio
Android SDK (<i>Software Development Kit</i>)	Versi 4.4 (API 19)
JDK (<i>Java Development Kit</i>)	Versi 7
Sistem Operasi	Windows 7
Web Browser	Windows 10

Analisis kebutuhan fungsional dimana didalamnya mencakup *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *Activity Diagram*, dan *Sequence Diagram*. Ini juga merupakan penggambaran proses kegiatan yang akan diterapkan kedalam aplikasi, apa saja kebutuhan yang diperlukan agar sistem berjalan baik.

1. *Use Case Diagram*

Untuk identifikasi dari *use case* dari sistem yang akan dibangun, dapat dilihat pada gambar 3 di bawah ini:



Gambar 3 *Use Case Diagram*

Dan pada table 3 di bawah ini merupakan pemaparan dari identifikasi *use case* dari sistem yang akan dibangun, dapat dilihat sebagai berikut.

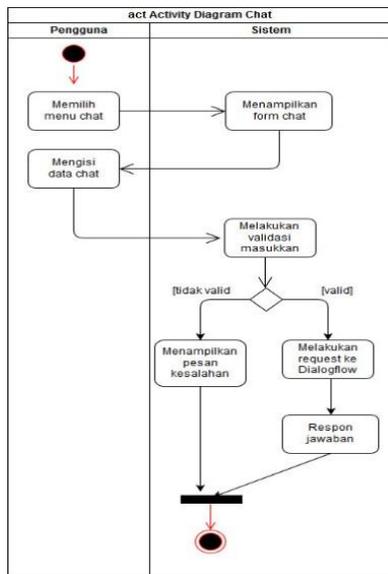
Tabel 3 Identifikasi *Use Case Diagram*

Use Case		
No	Use Case	Deskripsi
UC-01	Melakukan <i>Chat</i>	Fungsionalitas yang berguna untuk melakukan obrolan dengan <i>Bot</i>
UC-02	Melihat Lokasi Gereja	Fungsionalitas yang berguna untuk melihat lokasi gereja berdasarkan posisi pengguna
UC-03	Melihat Jadwal	Fungsionalitas yang berguna untuk melihat jadwal misa dan Pastor yang bertugas
UC-04	Melihat Pastor	Fungsionalitas yang berguna untuk melihat Pastor di setiap gereja atau paroki

2. *Activity Diagram*

Merupakan diagram yang bertujuan untuk menggambarkan proses bisnis dan alur kerjanya. Dan

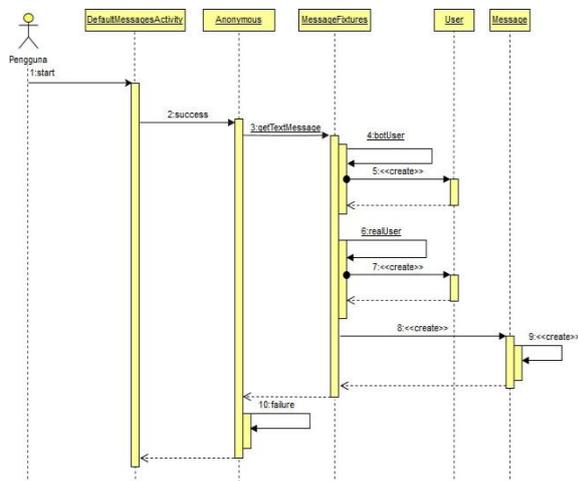
berikut ini salah satu contoh dari *activity diagram* melakukan *chat* pada aplikasi yang dibangun, dapat dilihat pada gambar 4 di bawah ini:



Gambar 4 Activity Diagram Melakukan Chat

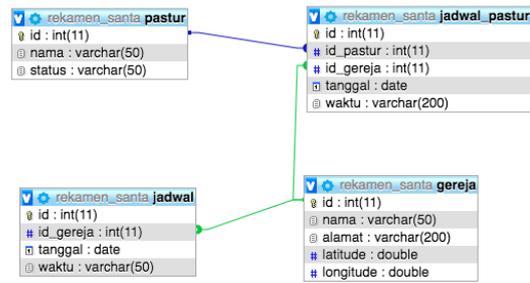
3. Sequence Diagram

Sequence diagram melakukan *chat* atau komunikasi menunjukkan rangkaian pesan yang dikirim objek pengguna dengan objek lainnya dalam proses melakukan *chat*. *Sequence diagram* melakukan *chat* dapat dilihat pada gambar 5.



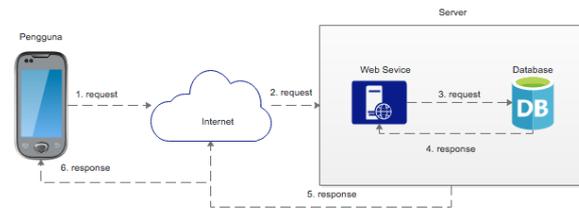
Gambar 5 Sequence Diagram Melakukan Chat

Skema Relasi, dimana ini merupakan rangkaian hubungan antara dua tabel dan atau lebih pada suatu sistem basis data. Untuk skema relasi yang dibangun pada perangkat lunak dapat dilihat pada gambar 5 sebagai berikut :



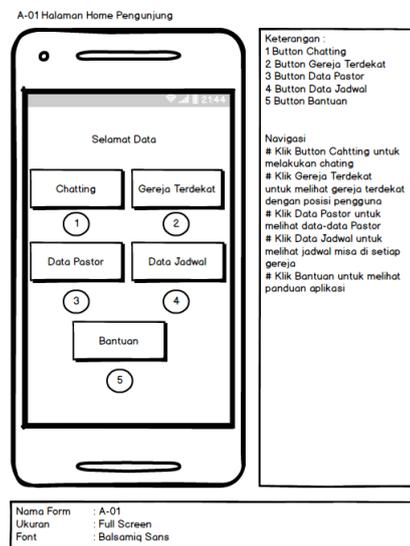
Gambar 6 Skema Relasi

Analisis arsitektur sistem dari aplikasi yang dibangun, dan dapat dilihat pada gambar 10 di bawah ini:



Gambar 7 Arsitektur Sistem Platform Mobile

Perancangan antarmuka dari aplikasi yang akan dibangun dan akan diambil salah satu contoh dari perancangan antarmuka tampilan halaman *home* dimana di dalamnya terdapat prakiraan menu dari aplikasi dan semua konten yang ada pada aplikasi tersebut, perancangan antarmuka *home* dapat dilihat pada gambar 11 di bawah ini sebagai berikut.



Gambar 8 Perancangan Antarmuka Home Utama

Implementasi aplikasi, yang bertujuan untuk menjelaskan tentang manualnya kepada pengguna yang akan menggunakan dan memanfaatkan dari aplikasi yang dibangun ini. Juga dapat memberi masukan kepada pengembang yang membangun aplikasi untuk dilakukan perbaikan yang bertujuan agar aplikasi menjadi lebih baik lagi. Di dalam

implementasi aplikasi terdapat beberapa hal yang harus diperhatikan antara lain:

1. Implementasi Perangkat Lunak, yaitu perangkat lunak yang akan menjalankan dari sistem aplikasi yang dibangun.
2. Implementasi Perangkat Keras, yaitu perangkat keras atau *hardware* yang akan digunakan untuk menjalankan dari sistem aplikasi yang dibangun.
3. Implementasi Data, yaitu perancangan berdasarkan basis data yang dibuat sebelumnya. Diimplementasikan menggunakan perangkat lunak yaitu MySQL.
4. Implementasi Antarmuka, yang berisi detail dari tampilan-tampilan yang ada pada perangkat lunak aplikasi yang dibangun tersebut yang terdiri dari nama antamuka atau *file* yang mewakilinya tersrbut.

Hingga yang terakhir adalah Pengujian Sistem, pengujian terhadap sistem secara fungsional *alpha* dan *beta*. Pertama untuk pengujian *alpha* tersebut menggunakan metode pengujian *blackbox* yang berfokus pada fungsional dari perangkat lunak yang dibangun. Untuk mengetahui apakah terjadi kesalahan dalam pengoperasiannya, kemampuan perangkat lunak dan tata laksana dari perangkat lunak itu. Apabila pengujian fungsionalitas sistem ini berjalan sesuai yang diharapkan, maka dapat dikatakan aplikasi yang dibuat telah benar. Dan untuk melihat skenario pengujian *alpha* dapat dilihat pada tabel 4 sebagai berikut.

Tabel 4 Rencana Pengujian *Alpha*

Kelas Uji	Poin Pengujian	Jenis Pengujian
Melakukan Chat	Input data chat	<i>Black Box</i>
	Validasi data chat	<i>Black Box</i>
Melihat Lokasi Gereja	Input parameter	<i>Black Box</i>
	Menampilkan data gereja	<i>Black Box</i>
Melihat Jadwal	Input parameter	<i>Black Box</i>
	Menampilkan data jadwal	<i>Black Box</i>
Melihat Pastor	Input parameter	<i>Black Box</i>
	Menampilkan data pastor	<i>Black Box</i>

Pengujian beta, pengujian yang dilakukan bertujuan mengetahui sejauh mana kualitar sistem aplikasi yang dibangun. Pengujian beta dilakukan kepada responden yang disebut adalah pengguna dari aplikasi dengan cara melakukan pengumpulan data dengan metode kuisisioner. Kuisisioner ini diberikan kepada sepuluh responden dengan memberikan empat pertanyaan dengan skala jawaban 1 sampai 5.

Skala jawaban kuisisioner dan pertanyaan yang diajukan kepada responden sebagai berikut.

- TS : Tidak Setuju
- KS : Kurang Setuju
- CS : Cukup Setuju
- S : Setuju
- SS : Sangat Setuju

Pertanyaan-pertanyaan yang diajukan, berikut adalah pertanyaan-pertanyaan yang diajukan kepada responden :

1. Saat membuka aplikasi, pengunjung mengerti apa yang harus dilakukan?
2. Pengguna (pengunjung) sebelumnya pernah mendengar istilah *chatbot*?
3. Saat berinteraksi dengan *chatbot*, apakah Anda berpikir bahwa yang menjawab pertanyaan Anda adalah manusia juga?
4. Berinteraksi dengan *chatbot* lebih menarik dibandingkan dengan aplikasi web biasa?
5. Setelah menggunakan aplikasi *chatbot* ini, apakah aplikasi ini dapat membantu pengunjung dalam hal pencarian informasi seputar jadwal misa di gereja Katolik di Keuskupan Bandung?
6. Informasi atau jawaban yang diberikan *chatbot* apakah sudah sesuai dengan yang ditanyakan atau dimaksud pengunjung?
7. Perlu pembaharuan untuk menyempurnakan sistem?

Berdasarkan pada hasil Kuisisioner yang diberikan kepada responden sebagai sample, maka dapat dihitung persentasenya menggunakan rumus :

$$Y = P/Q * 100\%$$

- Keterangan :
- Y : Nilai Prosentase
 - P : Total Skor
 - Q : Skor Tertinggi

Dan berikut adalah jawaban dari kuisisioner yang diberikan skor, dapat dilihat pada table 5 di bawah ini:

Tabel 5 Skor Jawaban Kuisisioner

Skala Jawaban	Keterangan	Skor
TS	Tidak Setuju	1
KS	Kurang Setuju	2
CS	Cukup Setuju	3
S	Setuju	4
SS	Sangat Setuju	5

Dari kuisisioner yang telah dilakukan, berikut ini adalah hasil dari kuisisioner yang telah diisi oleh sebanyak 26 responden sebagai contoh sampelnya. Dapat dilihat pada table 6 di bawah ini:

Tabel 6 Tabel Hasil Kuisisioner

No.	Pertanyaan	T S	K S	C S	S	S S
1	Saat Membuka aplikasi, pengunjung mengerti apa yang harus dilakukan?	0	0	6	14	6
2	Pengguna (pengunjung sebelumnya pernah mendengar istilah <i>chatbot</i> ?	0	2	6	12	6
3	Saat berinteraksi dengan <i>chatbot</i> , apakah Anda berpikir bahwa yang menjawab pertanyaan Anda adalah manusia juga?	0	0	6	13	7
4	Berinteraksi dengan <i>chatbot</i> lebih menarik dibandingkan dengan palikasi web biasa?	0	0	4	12	10
5	Setelah menggunakan aplikasi <i>chatbot</i> ini, apakah aplikasi ini dapat membantu pengunjung dalam hal pencarian informasi seputar jadwal misa di gereja Katolik di Keuskupan Bandung?	0	1	6	14	5
6	Informasi atau jawaban yang	0	0	3	18	5

	diberikan <i>chatbot</i> apakah sudah sesuai dengan yang ditanyakan atau dimaksud pengunjung?					
7	Perlu pembaharuan untuk menyempurnakan sistem?	0	0	5	8	13

Berdasarkan pengujian *beta* maka kesimpulan yang dapat diambil yaitu bahwa aplikasi yang dibangun mudah digunakan, memberikan manfaat dan dapat membantu pengguna dalam mencari informasi jadwal misa dan lokasi gereja di Keuskupan Bandung ini dengan fitur-fitur yang disediakan aplikasi yang dibangun ini, hal ini sesuai dengan jawaban dari responden pengguna terhadap pertanyaan kuisisioner yang diberikan.

5. KESIMPULAN

Setelah melakukan analisis-analisis, desain dan perancangan, dan juga implementasi dari perancangan perangkat lunak yang dibangun. Namun saran-saran diperlukan dan sangat penting demi dilakukannya perbaikan untuk pengembangan perangkat lunak untuk waktu yang akan datang. Kesimpulan secara umum dapat diringkas bahwa aplikasi telah dapat memberikan solusi dan membantu mengatasi dari permasalahan yang timbul dan dibahas pada bagian sebelumnya. Hasil itu didapat dari rata-rata jawaban responden dari 4 pertanyaan kuisisioner yang diberikan.

Untuk aplikasi *chatbot* untuk informasi jadwal misa ini memang masih jauh dari kesempurnaan, dan pastinya masih memiliki kekurangan. Maka diperlukanlah pembangunan dan atau penyempurnaan yang lebih jauh. Saran-saran yang harus diperhatikan seperti aplikasi dapat lebih natural dalam menjawab pertanyaan dari umat mengenai jadwal misa, lokasi gereja yang tersebar di Keuskupan Bandung maupun informasi pastor-pastor yang bertugas pelayanan di gereja-gereja di Keuskupan Bandung.

6. DAFTAR PUSTAKA

- [1] E. Martasudjita, Ekaristi : *Tinjauan Teologis, Liturgis, dan Pastoral*, Yogyakarta: Kanisius, 2005.
- [2] E. N. SCP, "*Rancang Bangun Aplikasi Chatbot Informasi Objek Wisata Kota*

Bandung dengan Pendekatan Natural Language Processing," Undergraduate Theses, 2013.

- [3] S. Kusumadewi, *Artificial Intelligence (Teknik dan Aplikasinya)*, Yogyakarta: Graha Ilmu, 2003.
- [4] S. Sannikova, "Chatbot Implementasi with Microsoft Framework," p. 8, 2018.
- [5] A. A. S, *Begining Android Programming with ADT Bundle*, Jakarta: Gramedia, 2014.
- [6] N. Safaat, *Aplikasi Berbasis Android, Bandung*: Informatika, 2015.
- [7] A. Kadir, *Buku Pertama Belajar Pemrograman Java untuk Pemula*, Yogyakarta: Mediakom, 2014.
- [8] B. Heriyanto, *Rekayasa Sistem Berorientasi Objek*, Bandung: Informatika, 2004.
- [9] K. W, *Panduan Belajar MySQL Database Server*, Jakarta: Mediakita, 2010.
- [10] S. Ian, *Software Engineering*, London: Dorling Kindersley, 2011.