

# Pemanfaatan *Google Assistant* dan Sistem Rekomendasi untuk Belajar Alkitab

David Kristian<sup>1</sup>, Henry Novianus Palit<sup>2</sup>, Leo Willyanto Santoso<sup>3</sup>

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

E-mail: dkrist8@gmail.com<sup>1</sup>, hnpalit@petra.ac.id<sup>2</sup>, leow@petra.ac.id<sup>3</sup>

## ABSTRAK

Terdapat beberapa jenis kecerdasan buatan yang telah diciptakan oleh manusia. Kecerdasan buatan untuk menghitung, menentukan keputusan, menterjemahkan bahasa yang satu ke bahasa lainnya. Selain itu berbagai perusahaan teknologi besar seperti *Google* dan *Apple* juga sudah menciptakan kecerdasan buatanya masing-masing yaitu *Google Assistant* dan *Siri*. Tentunya segmen pasar dari dua perusahaan ini bersaing satu dengan yang lain, dimana *Google Assistant* yang lebih mendukung kepada Android lebih banyak digunakan daripada *Siri*.

*Google Assistant* merupakan sebuah kecerdasan buatan yang mampu membantu dengan menjadi asisten pribadi dan terintegrasi dalam banyak hal seperti *Smartwatch*, Handphone Android, *Google Home*, mobil, TV, dan lain-lain. Para developer pihak ketiga dapat mengintegrasikan *Apps* yang dimiliki dengan *Google Assistant*.

Aplikasi ini menggunakan *Google Assistant*, *Dialogflow* dan *Neo4j*. Aplikasi ini membantu proses pendalaman alkitab dan melihat konteks biblikal dengan lebih baik lagi menggunakan sistem rekomendasi, selain itu dengan kecerdasan buatan dapat membuat percakapan yang dilakukan menjadi lebih natural. Pengguna akan dapat melakukan pencarian berbagai data alkitabiah mulai dari pencarian ayat, garis besar ayat, nomor strong, devosional, kamus tentang data alkitab.

**Kata Kunci:** Kecerdasan Buatan, *Google Assistant*, *Dialogflow*, *Neo4j*, Konteks Biblikal, Sistem Rekomendasi, Percakapan Natural, Data Biblikal.

## ABSTRACT

*There are several types of artificial intelligence that have been created by humans. Artificial intelligence is used to calculate, determine decisions, translate one language to another. In addition, various large technology companies such as Google and Apple have also created their own artificial intelligence, namely the Google Assistant and Siri. Of course the market segments of these two companies compete with each other, and so far Google Assistant which is more supportive of Android is more widely used than Siri.*

*Google Assistant is an artificial intelligence that can help as a personal Assistant and integrated is in many ways with Smartwatch, Android Mobile, Google Home, car, TV, and others. Third-party developers can integrate their Apps with Google Assistant.*

*This application uses Google Assistant, Dialogflow and Neo4j. This application helps study bible and see Biblical context better by*

*using a recommendation system, and further the artificial Intelligence could make the conversation more natural. Users will be able to search various biblical data ranging from verse searches, outline verses, strong numbers, devotional, dictionaries about biblical data.*

**Keywords:** *Artificial Intelligence, Google Assistant, Dialogflow, Neo4j, Biblical Contextual, Recommendation System, Natural Conversation, Biblical Data.*

## 1. PENDAHULUAN

Dalam perkembangan jaman yang mulai didominasi generasi Z ini, perkembangan teknologi sangat cepat terjadi. Beberapa teknologi yang sebelumnya belum pernah terpikirkan mulai berhadiran satu dengan yang lainnya. Tentunya kemajuan teknologi ini ditujukan untuk mempermudah hidup manusia, semakin majunya teknologi semakin pula banyaknya diciptakan alat untuk membantu manusia menjalani kehidupannya sehari-hari. Salah satu perkembangan teknologi yang memiliki tujuan untuk mengembangkan hidup manusia adalah kecerdasan buatan.

Terdapat beberapa jenis kecerdasan buatan yang telah diciptakan oleh manusia. Kecerdasan buatan untuk menghitung, menentukan keputusan, menterjemahkan bahasa yang satu ke bahasa lainnya. Selain itu berbagai perusahaan teknologi besar seperti *Google* dan *Apple* juga sudah menciptakan kecerdasan buatanya masing-masing yaitu *Google Assistant* dan *Siri*. Tentunya segmen pasar dari dua perusahaan ini bersaing satu dengan yang lain, dimana *Google Assistant* yang lebih mendukung kepada Android lebih banyak digunakan daripada *Siri*.

*Google Assistant* merupakan sebuah kecerdasan yang mampu membantu dengan menjadi Assistant pribadi dan terintegrasi dalam banyak hal seperti *Smartwatch*, Handphone Android, *Google Home*, mobil, TV, dan lain-lain. *Google Assistant* bisa mengirim pesan secara pribadi, menjalankan playlist music, mematikan lampu, mengatur suhu ruangan, melakukan pesanan meja untuk dinner, dan lain-lain. *Google Assistant* sendiri juga terintegrasi dengan beberapa app ternama seperti kaskus, klik dokter, *youtube*, *uber*, *spotify*, *joox*, dan lain-lain. Para developer pihak ketiga dapat mengintegrasikan *Apps* yang dimiliki dengan *Google Assistant*.

Aplikasi membaca alkitab sendiri sudah banyak diciptakan untuk membantu mempermudah hidup manusia yang dulunya lebih banyak membawa buku namun dengan adanya aplikasi, sekarang hanya perlu membawa handphone untuk dapat mengakses alkitab. SABDA merupakan salah satu Lembaga yang bergerak dalam bidang data-data alkitab dan studi bible merupakan salah satu pengembang dari aplikasi alkitab. Dengan berkembangnya *Google Assistant* maka SABDA ingin memaksimalkan kecerdasan buatan

milik *Google* ini karena merupakan salah satu sarana yang bagus untuk menjangkau baik generasi Z dan orang-orang lainnya. Dengan menggunakan *Google Assistant* yang sudah dimiliki banyak pengguna dan mempersiapkan masa depan, selain itu juga dapat menjadi salah satu inisiator untuk para pengembang studi alkitab ke depannya.

## 2. LANDASAN TEORI

### 2.1 Sistem Rekomendasi

Sistem Rekomendasi adalah sistem yang digunakan untuk melakukan penyaringan informasi yang berurusan dengan masalah yang ditimbulkan dari kelebihan informasi [7]. Cara kerjanya adalah menyaring potongan informasi penting dari informasi dinamis yang dalam jumlah besar dihasilkan berdasarkan pilihan dari pengguna, berupa keinginan atau perilaku pengguna. Sistem rekomendasi dapat sangat berguna untuk pengguna karena sistem rekomendasi sendiri dapat membantu pengguna mencari data yang mungkin pengguna ingin cari namun belum mengetahui kata kunci atau data yang dibutuhkan. Selain itu sistem rekomendasi membantu pengguna untuk mencari data yang terkait sehingga pengguna dapat melakukan penelitian mengenai suatu data alkitabiah yang terkait baik dari jumlah keluarnya kata, atau ayat yang terkait.

### 2.2 Graph

*Graph* menggambarkan suatu entitas atau benda yang berupa *nodes* dan hubungan yang menghubungkan antar *nodes* adalah relasi [12]. *Graph* memungkinkan penambahan relasi baru tanpa mengurangi atau menghilangkan jaringan yang ada atau migrasi data, dengan kata lain tetap menjaga data utama. Relasi dalam *graph* akan secara natural membentuk suatu jalur. Karena data model memiliki sifat dasar dari yang berorientasi pada suatu jalur maka pada umumnya *graph database* yang menggunakan dasar jalur pada umumnya sangat selaras dalam hal penataan data, yang meningkatkan tingkat keefisienan data. *Graph Database* menyimpan data dalam bentuk *nodes*, relasi, dan *properties* [8]. Ketika membuat *node*, kita akan memberi *properties* dari *node* tersebut, kemudian relasi juga dapat memiliki *properties* tersendiri. Inilah yang membuat *graph* dari data yang terkait langsung ke data, daripada ke *rows* dalam join tables. *Graph* sendiri terdiri dari beberapa figur pembentuk [9], yaitu:

- *Nodes* : Entitas dalam *Graph* digambarkan sebagai *nodes*
- Label : Label digunakan sebagai representasi peran *nodes* di domain yang kita miliki atau bisa disebut sebagai nama dari *node*. *Node* dapat memiliki beberapa label sekaligus. Selain berfungsi untuk memberi arti baru kepada *node*, label juga digunakan untuk mendambahkan pembatas dan indeks kepada masing-masing label
- Relasi : Koneksi yang secara semantik relevan antara dua *node*. Relasi di *Neo4j* akan selalu memiliki asal, akhir, dan biasanya akan juga memiliki arah. Kita dapat menggunakan arah untuk secara spesifik melihat arah data, atau kita bisa tidak menggunakan arah sama sekali
- *Properties* : Menyimpan data tentang *node* atau relasi

Beberapa perbedaan dari *Relational Database Management System* (RDBMS) dan *Graph Database*, yaitu:

- Data terlihat lebih simpel dan ekspresif dan semakin menunjukkan relasi secara eksplisit. RDBMS menggunakan *foreign key* untuk menunjukkan relasi antara satu tabel dan tabel lainnya secara implisit. Dengan *graph* kita dapat dengan mudah menelusuri relasi untuk mencari informasi yang dibutuhkan.

- Kecepatan *query* dari *graph database* lebih cepat dari RDBMS.
- Memiliki skalabilitas data karena tidak perlu merombak semua *database*, namun bisa dengan mudah menambah atau mengurangi koneksi. Kita bisa mengganti *node*, relasi, property. Fleksibilitas ini menyebabkan tidak perlu untuk melakukan banyak migrasi
- Skema yang digunakan *whiteboard friendly*, berarti kita bisa membuat skema di papan tulis dan langsung membuatnya sebagai *graph*, tidak perlu melakukan normalisasi dan denormalisasi tabel untuk membuat *graph* data model.

### 2.3 Rule-Based System

*Rule based systems* adalah hal yang berbeda dari prosedur standar atau pemrograman berorientasi objek yang tidak terdapat urutan dari jalannya kode [10]. Sebaliknya pengetahuan dari pakar di utarakan dalam satu set aturan, yang masing-masing mengkodekan pengetahuan pakar. Setiap peraturan memiliki kondisi untuk peraturan agar bisa berjalan dan apabila kondisi itu tercapai maka perintah akan dieksekusi. Kumpulan aturan ini akan berjalan satu demi satu sampai semua peraturan sudah selesai dieksekusi. Rules biasanya diimplementasi pada *rules engine*, yang memberikan suatu framework untuk menulis dan menjalankan *rules*.

*Rule-based system* terdiri dari seperangkat aturan “*If-Then*”, seperangkat fakta dan beberapa penerjemah yang mengontrol penerapan aturan. Gagasan sistem pakar adalah menggunakan pengetahuan dari sistem pakar dan menyandikannya ke dalam seperangkat aturan. Ketika terekspos pada data yang sama, sistem pakar akan melakukan dengan cara yang mirip dengan pakar. Sistem berbasis aturan adalah model yang sangat sederhana dan dapat diadaptasi dan diterapkan untuk masalah besar [5]. Persyaratannya adalah bahwa pengetahuan tentang area masalah dapat dinyatakan dalam bentuk aturan *if-then*. Area ini seharusnya tidak terlalu besar karena sejumlah aturan yang tinggi dapat membuat pemecah masalah (sistem pakar) tidak efisien.

### 2.4 Google Assistant

*Google Assistant* adalah komunikasi konversasional dua arah antara kita dan *Google* yang mengerti dunia kita dan membantu kita mencapai sesuatu [11]. Kita juga dapat menggunakan *Google Assistant* di device manapun dan dalam konteks apapun sebagai asisten pribadi kita. *Google Assistant* adalah kecerdasan buatan yang dibuat oleh *Google* yang semulanya merupakan upgrade atau ekstensi dari *Google Now* yang juga mengembangkan kontrol suara dari “*OK Google*” [1].

Banyak hal yang bisa dilakukan oleh *Google Assistant*. “*OK Google*” membuat kita bisa mengakses perintah suara, pencarian menggunakan suara, dan mengaktifkan *device* kita dengan suara, selain itu juga bisa mengirim pesan, mengecek janji dan lain-lain menggunakan perangkat android yang kita miliki.

Setelah memulai dengan “*Ok Google*” tidak perlu mengulangnya lagi untuk melakukan perintah berikutnya. *Google Assistant* dapat mengerti konteks dari pembicaraan dan menyaring apa yang akan diakses berdasarkan hal tersebut.

### 2.5 Google Action SDK

*Google Action SDK* merupakan platform untuk developer yang dapat membantu kita membuat software untuk mengembangkan fungsionalitas dari *Google Assistant* [4]. Berbeda dengan *Apps* dari mobile dan desktop yang tradisional, yang menggunakan paradigma komputer sentrik, pengguna dapat berinteraksi dengan *Actions* kepada Assistant melalui pembicaraan yang natural.

Cara kerja dari *Action* pada *Google Assistant* adalah kita membuat *Action* untuk Assistant dengan mendesain sebuah percakapan untuk berbagai permukaan, seperti percakapan *voice centric* untuk speaker yang diaktifkan melalui suara. Pengguna dapat meminta melalui berbagai permukaan seperti handphone Android. Semua *Action* berjalan pada *cloud*, bahkan ketika pengguna berkata kepada mereka pada handphone, *smart home*, atau jam mereka. Perangkat pengguna akan mengirimkan ucapan pengguna kepada *Assistant*, yang akan mengarahkannya ke layanan permohonan anda melalui permohonan *HTTP POST*. Akan dicari respon yang relevan dan akan dikirim kembali ke *Assistant*, yang akhirnya akan dikembalikan ke pengguna.

*Google Action SDK* terdiri dari:

- **Actions** : Pembungkus yang menggabungkan intent dan fulfillment untuk intent tersebut. Untuk app yang konversasional, terdapat beberapa *Action* dari *apps*.
- **Intents** : Menentukan apa yang pengguna *apps* ingin lakukan. Setiap *apps* memiliki *Actions.intent.Main* yang digunakan untuk menjalankan *apps*
- **Fulfillment** : Menghubungkan dengan *backend*. Dimana kita memproses data untuk output kepada pengguna. Fulfillment memberikan *Json* dan informasi berupa, teks yang dideteksi *Google*, intent yang dimaksud, parameter, id sesi, informasi tentang kemampuan dari alat

## 2.6 Dialogflow

*Dialogflow* memberikan pengguna cara baru untuk berinteraksi dengan produk yang dibuat agar dapat menggunakan percakapan yang memiliki *interface* berupa teks atau suara, seperti *voice Apps* dan chatbots, yang menggunakan kecerdasan buatan [2]. *Dialogflow* dikembangkan menggunakan *machine learning* milik *Google* dan didukung oleh produk-produk milik *Google* lainnya seperti *Google Cloud Speech to Text*. *Dialogflow* juga dioptimisasikan untuk *Google Assistant*.

*Dialogflow* memungkinkan kita untuk membuat interface dari konversasi pada produk dan servis milik kita. *Dialogflow* sendiri menggunakan mesin Natural Language Understanding (NLU) untuk memproses dan memahami *input* Bahasa natural. *Dialogflow* menggunakan *agents* sebagai media kita untuk memahami bahasa manusia yang sangat bervariasi dan menerjemahkannya menjadi arti yang terstruktur sehingga memungkinkan *apps* dan *service* yang kita gunakan untuk mengetahui apa yang dimaksudkan oleh pengguna.

Agent dari *Dialogflow* membantu memproses masukan dari pengguna kepada data yang sudah terstruktur yang dapat digunakan untuk memberikan timbal balik yang tepat [6]. Semua yang dikerjakan oleh agent akan dibantu oleh intent. Intent sendiri terdiri dari beberapa komponen, yaitu:

- **Training phrases** : Memberikan contoh kepada *Dialogflow* apa yang pengguna akan sampaikan. *Dialogflow* menggunakan frasa-frasa tersebut untuk melihat konteks dan model Bahasa agar dapat dikaitkan dengan frasa lain yang setipe. Melalui *training* dan *machine learning* yang berkelanjutan *Dialogflow* membuat model bahasa yang lebih bervariasi untuk mencocokkan *input* dari pengguna
- **Action dan Parameter** : Untuk mengembangkan model bahasa dari *intent*, kita bisa melakukan anotasi pada *training phrases* kita dengan sebuah entitas atau kategori yang ingin dicocokkan oleh *Dialogflow*. Hal tersebut memungkinkan *Dialogflow* mengetahui bahwa kita menginginkan tipe *input* tersebut untuk

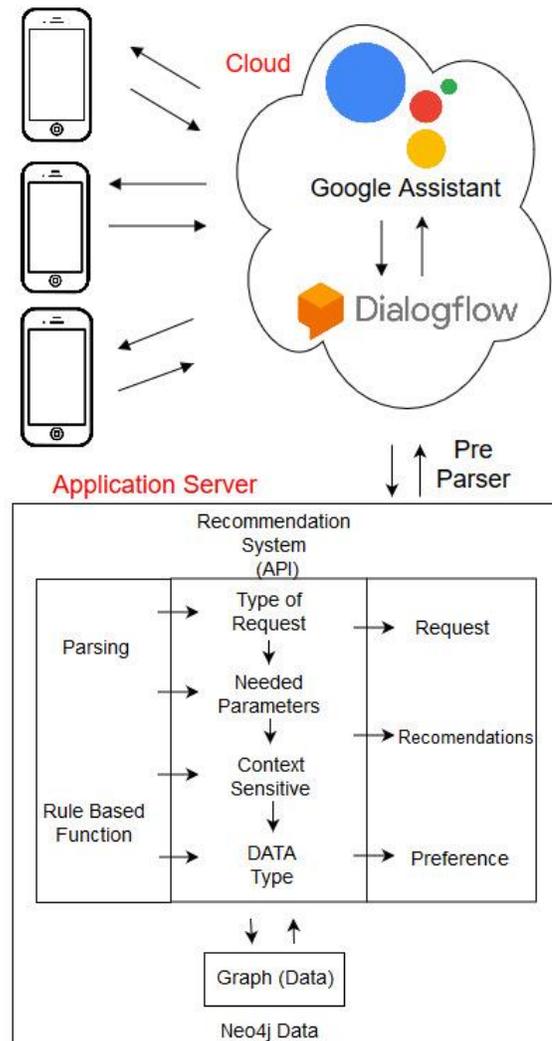
dikaitkan dengan entitas sebagai parameter dari *training phrases* kita. Kita bisa memproses parameter dalam sebuah logika yang disebut *fulfillment* untuk semakin menyesuaikan respons terhadap *user*

- **Respon**: Menjelaskan teks, perkataan atau repon bergambar kepada pengguna, yang memungkinkan pengguna untuk mengetahui apa yang harus dikatakan selanjutnya atau menyelesaikan percakapan. Untuk mengirim respon kita dapat menggunakan *response handler* yang sudah di *built-in* oleh *Dialogflow* atau meminta *fulfillment* untuk memproses data yang sudah diekstraksi dan melakukan respon balik kepada *dialogflow*.

*Fulfillment* dari *Dialogflow* memungkinkan repons yang berbeda dari respon yang statis dan kurang memiliki logika. [3] *Fulfillment* dapat membantu memproses *intent* dan kemudian memberikan repons yang lebih pintar. *Fulfillment* adalah logika yang kita implementasi sebagai *webhook* yang dibutuhkan oleh servis, memproses data yang didapat dari *webhook* tersebut, dan kemudian memberikan respon balasan.

## 3. DESAIN SISTEM

### 3.1 Desain Sistem

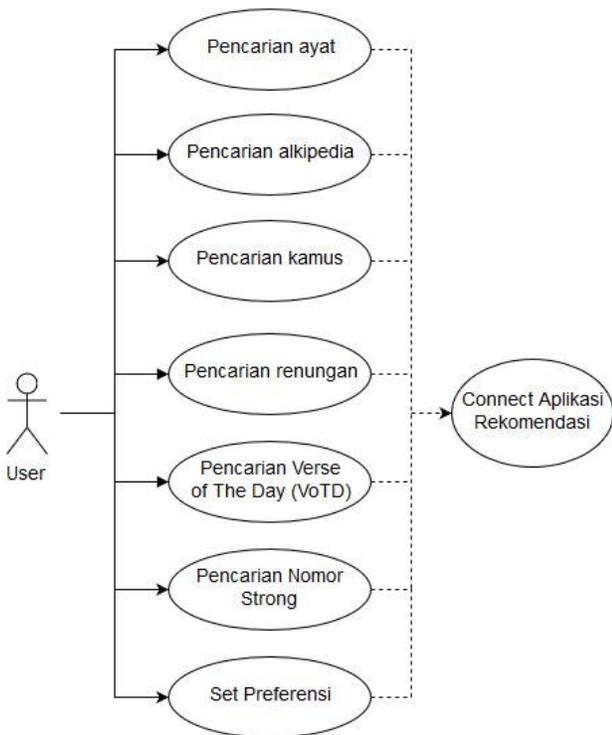


Gambar 1. Arsitektur Sistem

Gambar 1 menunjukkan rangkaian sistem dirancang untuk memudahkan pengguna dalam belajar alkitab. Pengguna dapat belajar alkitab melalui Agent *Dialogflow* yaitu Alkitab Rekomendais yang dapat diakses melalui *Google Assistant* yang terinstall di handphone masing-masing. Ketika pengguna memasukkan *input* pencarian, maka Agent *Dialogflow* akan melakukan proses parsing kata dan pengambilan parameter yang dibutuhkan untuk berbagai jenis perintah yang berbeda melalui *Dialogflow* dan API yang juga berupa recommendation system akan menentukan informasi yang akan ditampilkan dan rekomendasi dari informasi tersebut kemudian akan mengakses SABDA Data dan akan menampilkan informasi tersebut kepada pengguna.

### 3.2 Use Case Diagram

Gambar 2 menunjukkan *Use Case Diagram* yang menggambarkan interaksi-interaksi yang dapat dilakukan setiap pengguna yang berhubungan dengan sistem.

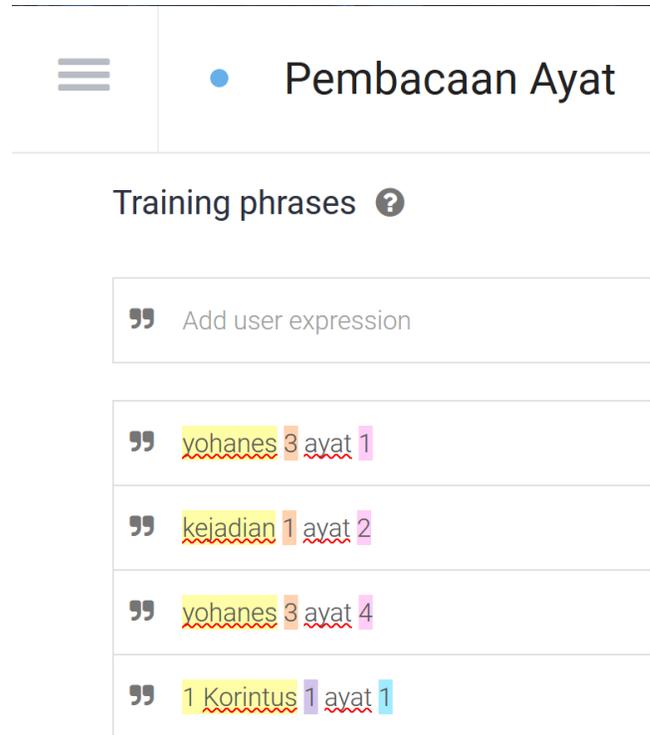


Gambar 2. Use Case Diagram

Actor yang berperan adalah pengguna aplikasi. Data dari pengguna akan disimpan melalui akun *gmail* utama dari pengguna, yang terhubung dengan handphone pengguna. Pengguna mempunyai akses untuk melakukan pencarian yang berdasarkan tiga kriteria utama dalam pencarian yaitu alamat ayat, kata, dan tanggal saat ini. Masing-masing dari tiga kriteria utama pencarian tersebut menawarkan hasil data yang berbeda antara satu dengan yang lain. Dalam pencarian dengan ayat pengguna akan mendapatkan hasil berupa ayat yang dicari. Apabila pengguna melakukan pencarian dengan kata maka akan ditampilkan kamus atau arti dan penjelasan dari kata yang telah diinput oleh pengguna. Pengguna juga dapat melakukan pencarian dengan memanfaatkan tanggal saat ini untuk mencari ayat maupun renungan harian. Fitur pencarian sendiri akan dapat menampilkan data-data berupa ayat, topik, kamus, penjelasan atau garis besar pasal (alkipedia), tokoh, nama dan tempat, verse of the day, renungan, nomor strong

### 3.3 Training phrase

*Training phrase* yang dilakukan pada *Dialogflow* dilakukan pada setiap *intent* (fungsi) yang ingin dijalankan. *Training phrase* merupakan frasa kata yang memungkinkan pemanggilan suatu *intent* pada Agent *Dialogflow*. Gambar 3 menunjukkan contoh dari *training phrase*.



Gambar 3. Training phrase

*Training phrase* memungkinkan untuk mencontohkan pola yang akan dimasukan oleh pengguna sehingga Agent *Dialogflow* dapat mengerti pola dan parameter yang perlu dicari untuk masing-masing *intent*.

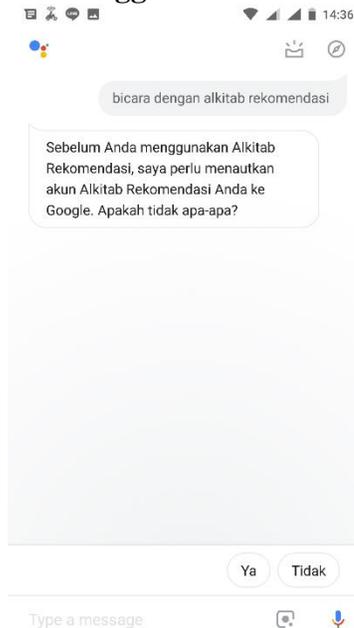
## 4. PENGUJIAN SISTEM

Pada bab ini akan dibahas tentang implementasi sistem sesuai dengan analisa dan desain sistem. Implementasi sistem meliputi pengaturan *Dialogflow*, pembuatan API dan implementasi Sistem Rekomendasi. Spesifikasi *device* Android yang digunakan pada pengujian sistem dapat dilihat pada Tabel 1:

Tabel 1. Spesifikasi *device* pengujian

Perangkat	Operating System	Display Size	CPU	Internal Memory
Nokia 8	Android 8.1.0 (Oreo)	1440 x 2560 pixels, 16:9 ratio (~554 ppi density)	Octa-core 4x2.5 GHz Kryo & 4x1.8 GHz Kryo	4 GB RAM
Samsung Galaxy A8	Android 7.1.1 (Nougat)	1080 x 2220 pixels, 18.5:9 ratio (~441 ppi density)	Octa-core (2x2.2 GHz Cortex-A73 & 6x1.6 GHz Cortex-A53)	4 GB RAM

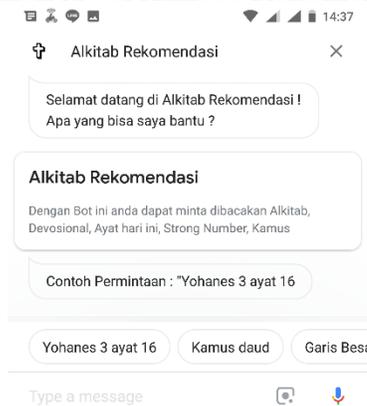
## 4.1 Halaman Panggil Alkitab Rekomendasi



Gambar 4. Halaman Panggil Alkitab Rekomendasi

Pada Gambar 4 pengguna memanggil Alkitab Rekomendasi dan diharuskan untuk mengaitkan akun *Gmail* dengan Alkitab Rekomendasi. Mengaitkan akun *Gmail* dilakukan untuk menyimpan preferensi dari pengguna pada kamus.

## 4.2 Halaman Awal Alkitab Rekomendasi

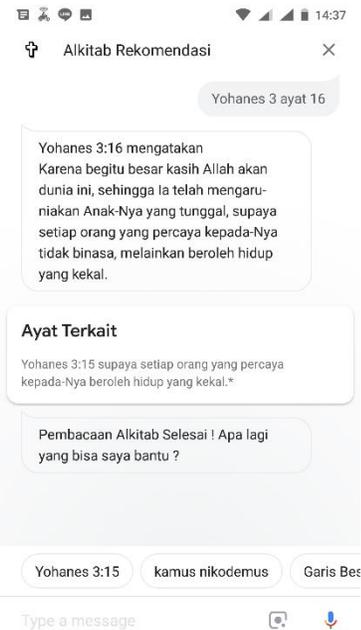


Gambar 5. Halaman Tampilan Awal

Pada Gambar 5 ditunjukkan tampilan awal dari Alkitab Rekomendasi dan hal yang bisa dilakukan oleh Alkitab Rekomendasi serta ditampilkan rekomendasi pencarian yang dapat dilakukan oleh Alkitab Rekomendasi.

## 4.3 Halaman Pencarian Ayat

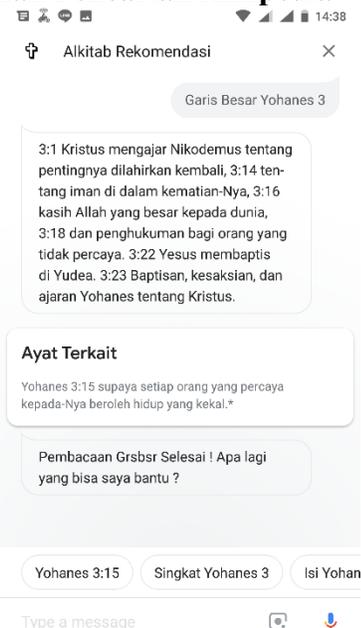
Halaman pencarian ayat menunjukkan tampilan pencarian alamat ayat oleh pengguna, akan ditampilkan ayat yang dicari, ayat terkait paling kuat dari ayat yang dicari, kamus, Alkipedia, Nomor Strong.



Gambar 6. Halaman Pencarian Ayat

Pencarian ayat pada Gambar 6 digunakan apabila ingin ditampilkan ayat alkitab dan data terkait dari masing-masing ayat yang dicari.

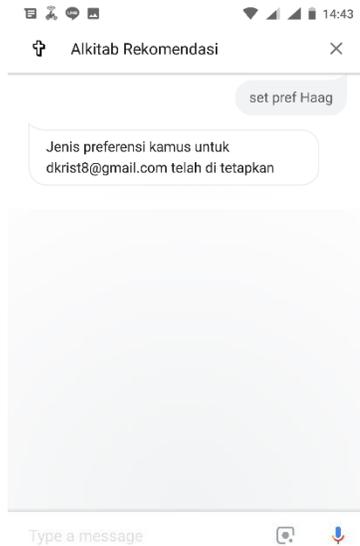
## 4.4 Halaman Pencarian Alkipedia



Gambar 7. Halaman Pencarian Alkipedia

Pada Gambar 7 terdapat tampilan pencarian alkipedia. Pengguna dapat mencari 8 jenis alkipedia, yaitu garis besar, fakta, isi, judul perikop, kesimpulan, nama, singkat, tokoh. apabila pengguna memberi *input* parameter berupa jenis alkipedia, injil dan pasal. Akan ditampilkan data alkipedia, dan ayat terkait. Rekomendasi jenis lain Alkipedia juga akan ditampilkan.

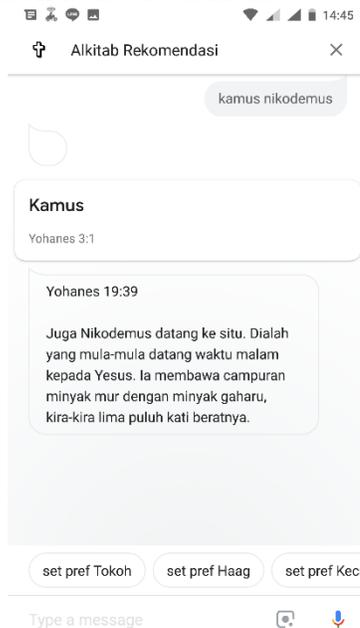
## 4.5 Halaman Set Preferensi Kamus



Gambar 8. Halaman Set Preferensi

Pada Gambar 8, *user* dapat melakukan *set* preferensi apabila pengguna memberi *input* parameter berupa jenis kamus. Terdapat 4 jenis kamus yaitu Tokoh, Kecil, *Haag* dan *Gering*. Alkitab Rekomendasi akan menyimpan preferensi pengguna menggunakan alamat email yang dikaitkan dengan Alkitab Rekomendasi.

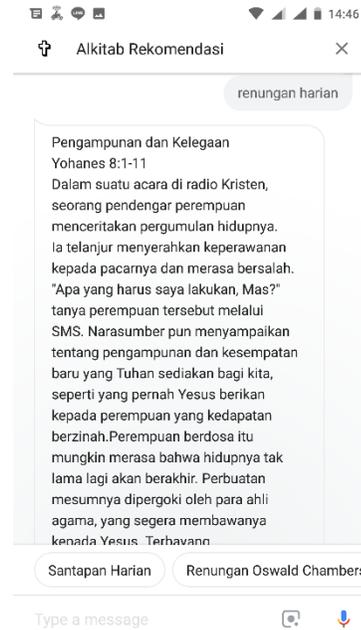
## 4.6 Halaman Pencarian Kamus



Gambar 9. Halaman Pencarian Kamus

Pada Gambar 9 pengguna ditampilkan pencarian kamus apabila pengguna memberi *input* parameter berupa istilah dan preferensi dari pengguna adalah kamus yang berupa *Haag*, Kecil, Tokoh, *Gering*. Akan ditampilkan kamus tentang istilah yang dicari, dan ayat terkait. Saran jenis lain untuk mengganti jenis preferensi kamus juga akan ditampilkan.

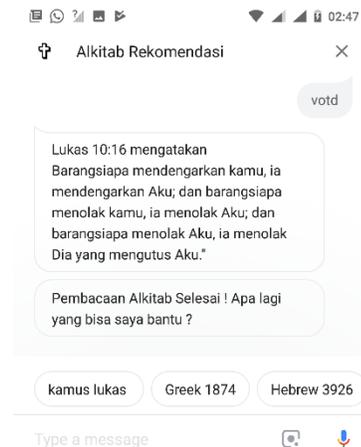
## 4.7 Halaman Pencarian Pencarian Devosional



Gambar 10. Halaman Pencarian Devosional

Pada Gambar 10 pengguna ditampilkan pencarian devosional apabila pengguna memberi *input* parameter berupa jenis devosional yaitu renungan harian, santapan harian, dan renungan Oswald Chambers. Akan ditampilkan devosional, dan ayat terkait. Saran jenis lain devosional juga akan ditampilkan.

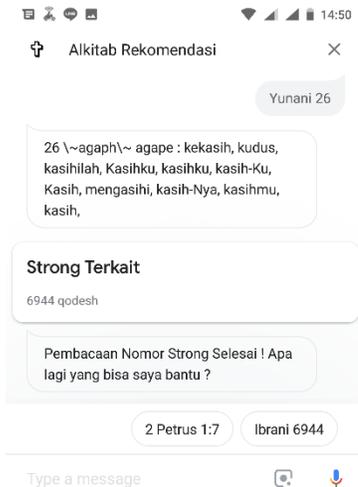
## 4.8 Halaman Pencarian Pencarian Verse of the Day



Gambar 11. Halaman Pencarian Verse of the Day

Pada Gambar 11 pengguna ditampilkan pencarian *Verse of the Day* (VOTD) apabila pengguna memberi *input* parameter berupa VOTD. Akan ditampilkan VOTD dan ayat terkait jika ada. Saran berupa ayat terkait, pencarian kamus dan nomor strong jika ada akan ditampilkan.

## 4.9 Halaman Pencarian Pencarian Nomor Strong



Gambar 12. Halaman Pencarian Devosional

Pada Gambar 12 pengguna ditampilkan pencarian nomor Strong Bahasa Yunani apabila pengguna memberi *input* parameter berupa Bahasa Yunani dan nomor Strong. Akan ditampilkan Nomor Strong, cara baca dalam Bahasa Yunani, kata dalam Bahasa Yunani dan arti Bahasa Indonesia dan juga nomor Strong terkait dalam Bahasa Ibrani. Saran berupa nomor Strong terkait, ayat terkait akan ditampilkan..

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dari hasil pemanfaatan *Google Assistant* dan sistem rekomendasi untuk Belajar Alkitab, dapat diambil kesimpulan antara lain:

1. Preferensi dapat dilakukan menggunakan *Google Sign In* sehingga *Action* dapat menyimpan email dari pengguna untuk melakukan pencarian jenis kamus yang sesuai preferensi pengguna, tanpa pengguna harus terus mengulang jenis pencarian kamus yang diinginkan.
2. *Dialogflow* menggunakan *machine learning* dalam *training phrase* berupa contoh *input* yang mungkin dilakukan pengguna dapat membantu membuat percakapan antara pengguna dan *Action* menjadi lebih natural.
3. Berdasarkan kuisisioner yang dibagikan, 91% menyatakan Alkitab Rekomendasi dapat membantu pembelajaran alkitab dengan baik.

### 5.2 Saran

Saran yang dapat diberikan untuk menyempurnakan dan mengembangkan aplikasi ini lebih lanjut antara lain:

1. Versi ayat lain dapat ditambahkan, untuk membantu membandingkan Bahasa lain sesuai dengan preferensi.
2. Mempunyai server dapat membantu perilsan *Action* pada *Google Assistant* daripada menggunakan *Ngrok* karena pada penggunaan *Ngrok* memerlukan komputer harus selalu menyala, dan kita tidak tahu kapan tim dari *Google* akan menguji *Action* kita.

## 6. REFERENCES

- [1] Betters, E., & Grabham, D. 2018, October 9. *What is Google Assistant, how does it work, and which devices offer it*. URI=<https://www.pocket-lint.com/Apps/news/Google/137722-what-is-Google-assistant-how-does-it-work-and-which-devices-offer-it>
- [2] Dialogflow. *Dialogflow home*. URI=<https://Dialogflow.com>
- [3] Dialogflow. *How does fulfillment work?* URI=<https://Dialogflow.com/docs/intro/fulfillment>
- [4] Google. 2018, October 2. *Overview: Google Action*. URI=<https://developers.Google.com/Actions/extending-the-assistant>
- [5] Grosan, C., & Abraham, A. 2011. *Intelligent Systems A Modern Approach*. Springer-Verlag Berlin Heidelberg.
- [6] *How do agents work?* URI=<https://Dialogflow.com/docs/intro/agents>
- [7] Isinkaye, F., Folajimi, Y., & Ojokoh, B. 2015. Recommendation systems: Principles, methods and. *Egyptian Informatics Journal*, 16(3), 261-273.
- [8] Kemper, C. 2015. *Beginning Neo4j*. New York: Springer Science+Business Media.
- [9] Lal, M. 2015. *Neo4j Graph Data Modeling*. Birmingham: Packt Publishing Ltd.
- [10] Lila, R. 2000. *Rule Based Systems*. URI=[http://www.ramalila.net/Adventures/AI/rule\\_based\\_sys\\_tems.html](http://www.ramalila.net/Adventures/AI/rule_based_sys_tems.html)
- [11] Pichai, S. 2016, May 18. *I/O: Building the next evolution of Google*. URI=<https://www.blog.Google/products/assistant/io-building-next-evolution-of-Google/>
- [12] Robinson, I., Webber, J., & Eifrem, E. 2015. *Graph Databases*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.