

Rancang Bangun Aplikasi Bot Sebagai Media Informasi Khonghucu Pada Pesan Instan Line

Seminar Skripsi

Reviewer

Ednawati Rainarli, S.Si., M.Si.

Pembimbing

Irawan Afrianto, S.T., M.T.

10115253

Raka Suryaardi Widjaja

**Fakultas Teknik dan Ilmu Komputer
Program Studi Teknik Informatika
Universitas Komputer Indonesia**



Khonghucu?



Agama Khonghucu atau Khong Kau (*Khong Jiao*), istilah aslinya disebut Ji Kau (*Ru Jiao*) yang artinya agama dari orang-orang yang lembut hati, yang terbimbing atau terpelajar [1][2].

Agama Khonghucu mempunyai Kitab Suci tersendiri, diantaranya: *Kitab Sishu* (**Kitab Yang Empat**), *Kitab Wujing* (Lima Kitab), dan *Kitab Xiaojing* (Kitab Bakti).



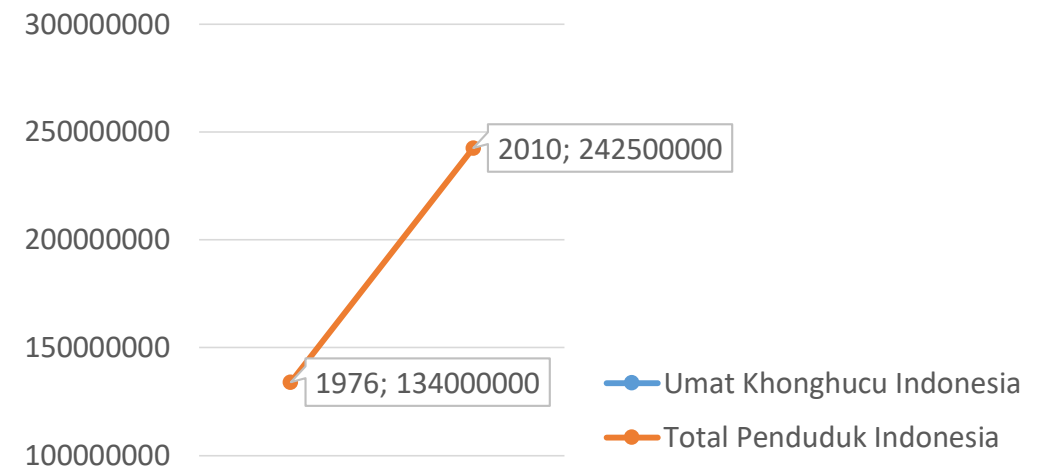
Latar Belakang Masalah



Pada tahun 1976 jumlah penduduk Indonesia yang beragama Khonghucu mencapai 0,7% (lebih dari satu juta jiwa) [3] namun berdasarkan sensus penduduk pada tahun 2010 jumlah pemeluk agama Khonghucu berkurang menjadi 117.091 jiwa [4].

Hal ini terjadi akibat bentuk diskriminasi dari segi hukum dan pelayanan publik yang terjadi mulai dari masa orde baru hingga masa reformasi sehingga membuat perkembangan agama Khonghucu menjadi terhambat dan mengalami keterputusan [3].

Data Umat Khonghucu di Indonesia



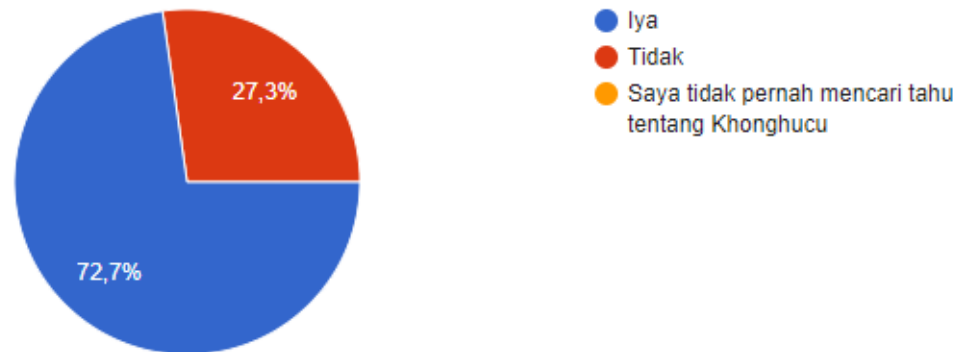
	1976	2010
Umat Khonghucu Indonesia	1000000	117091
Total Penduduk Indonesia	1340000000	2425000000

Latar Belakang Masalah



Apakah anda pernah mengalami kesulitan dalam mencari informasi apapun mengenai Khonghucu (khususnya yang menggunakan bahasa Indonesia)?

44 tanggapan



Berdasarkan survei yang penulis lakukan melalui kuesioner terhadap 44 responden yang 42 diantaranya menyebutkan sebagai umat Khonghucu, 32 dari 44 (72,7%) responden mengalami kesulitan dalam mencari informasi mengenai agama Khonghucu.

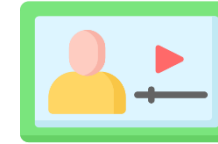


Bot

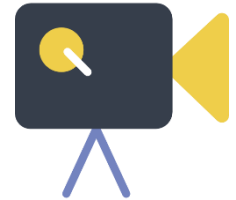
Bot merupakan program komputer yang dirancang untuk memberikan respon sesuai dengan permintaan yang diberikan [7]. Bot dapat memberikan respon berdasarkan permintaan berupa teks, suara, gambar, dan permintaan lainnya berdasarkan pengetahuan yang sudah dimiliki sebelumnya. Bot dapat dibuat pada berbagai macam platform, untuk berbagai macam tujuan dan dengan berbagai macam metode.



Informasi Layanan
Kampus



Informasi Wisata



Informasi Jadwal
Film



Promosi Produk

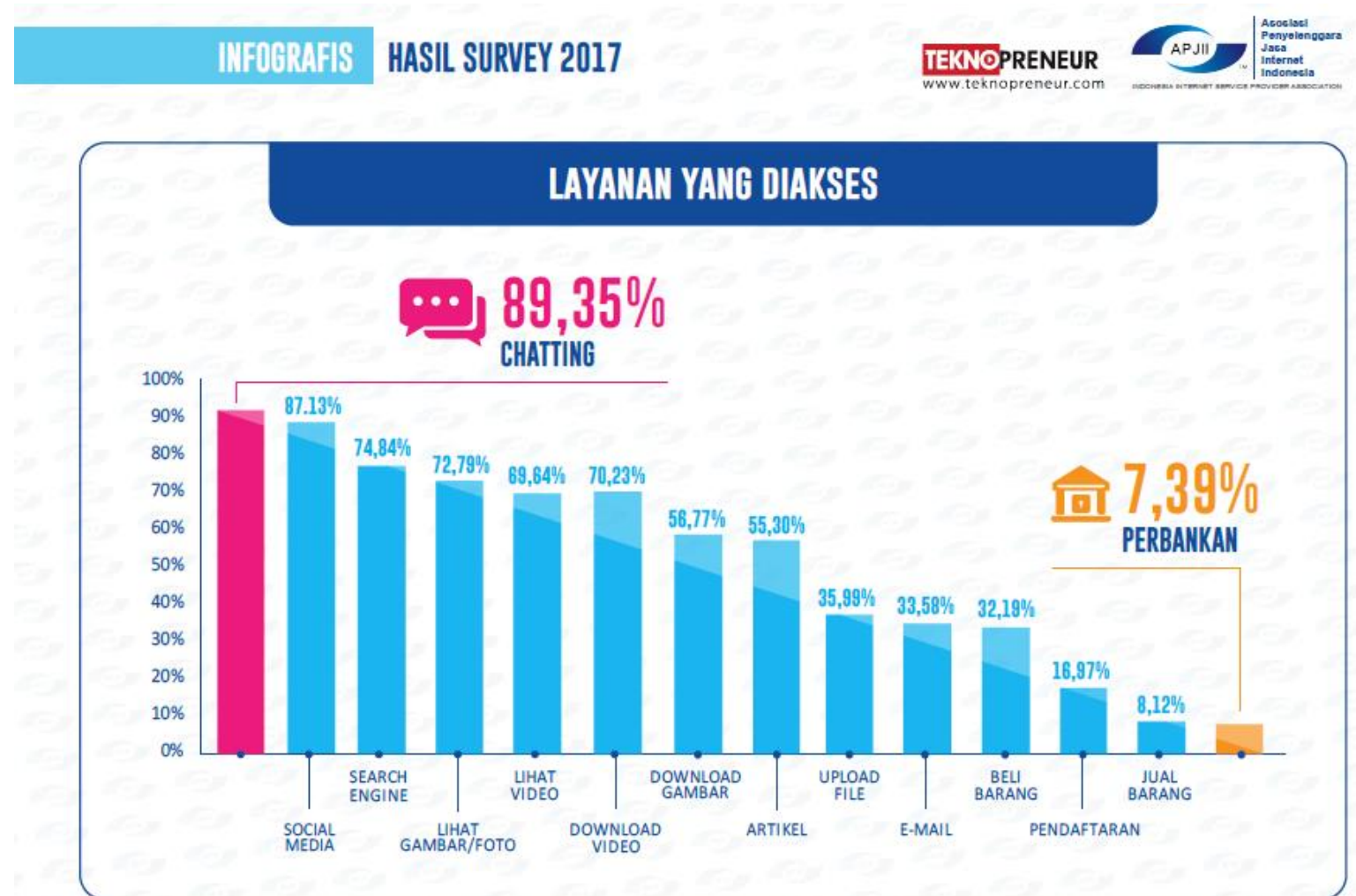


Monitoring
Keamanan Web
dan Jaringan



Survei APJII 2017

Berdasarkan survei yang dilakukan oleh Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) pada tahun 2017 diketahui jenis layanan yang paling banyak diakses di Internet diantaranya: **Chatting** (89,35%), *Social Media* (87,13%) dan *Search Engine* (74,84%) [5].



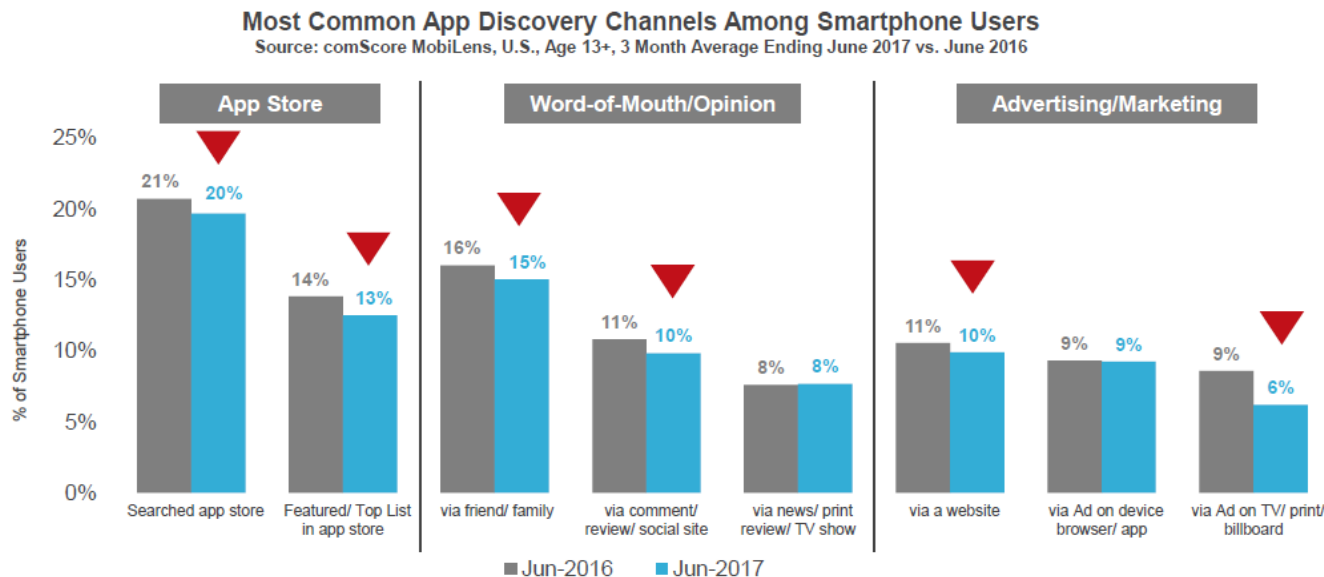


Laporan comScore 2017

App discovery is down across several channels, including the app store, word-of-mouth and advertising. Is interest in new apps waning?

Laporan comScore pada Januari 2017 menunjukkan sedikitnya penggunaan **aplikasi baru** dibandingkan aplikasi yang sudah lama populer.

Dibandingkan menarik perhatian dengan membuat aplikasi baru, membuat bot pada layanan pesan instan yang sudah ada lebih **mudah dalam pengembangan dan pengintegrasianya** [6].

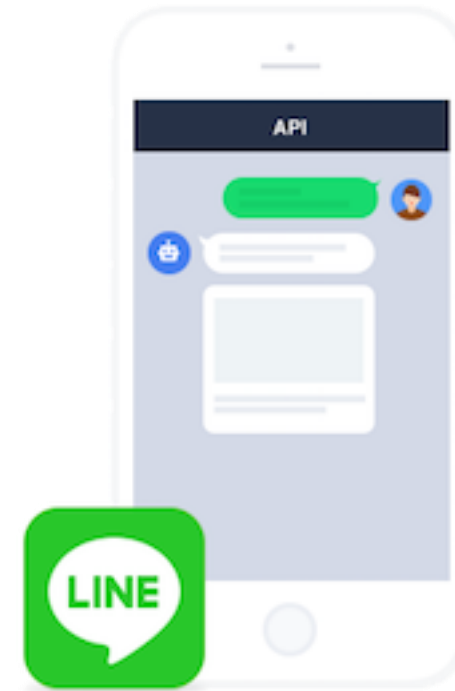




Pengguna Line 2018

Line merupakan aplikasi pesan instan yang banyak digunakan oleh masyarakat di Indonesia. Berdasarkan data pada tahun 2018 jumlah **pengguna Line di Indonesia** tercatat sebanyak 90 juta pengguna yang 80% diantaranya didominasi oleh anak muda [10].

Line memiliki beberapa fitur yang cukup digemari salah satunya yakni fitur bot yang dapat dibuat dan digunakan pada aplikasi Line untuk berbagai macam kebutuhan seperti diantaranya pencarian informasi, sosialisasi dan pemasaran produk, bermain game, dsb.



Batasan Masalah



1. Aplikasi yang akan dibangun merupakan aplikasi bot berbasis pesan instan.
2. Layanan pesan instan yang digunakan adalah Line.
3. API Line yang digunakan versi tidak berbayar.
4. Layanan pada aplikasi yang akan dibangun yakni: Lagu Rohani Khonghucu, Kitab Sishu, dan Lokasi tempat ibadah Khonghucu.
5. Data pokok yang digunakan berupa lagu rohani, kitab sishu, dan lokasi tempat ibadah yang digunakan diasumsikan tidak akan ditambah dalam penelitian ini agar lebih fokus dalam pembangunan aplikasi bot.
6. Konten pada aplikasi yang akan dibuat berbahasa Indonesia dan diasumsikan tidak akan diubah untuk kepentingan penelitian.
7. Masukan (input) dapat diberikan melalui teks, location dan rich menu.
8. Keluaran (output) yang diberikan oleh bot berupa teks dan gambar dalam bentuk menu dan konten.
9. Database yang digunakan adalah database MySQL (MariaDB) dengan engine InnoDB.

Batasan Masalah



10. Metode pencocokan pola yang digunakan menggunakan metode Fulltext Search Boolean Mode tanpa menggunakan operator.
11. Pencocokan masukan berupa teks pada bot yang dibangun tidak menggunakan konsep learning melainkan menggunakan konsep searching karena terbatas dengan metode pencocokan yang digunakan.
12. Protokol yang digunakan untuk mengirimkan dan menerima pesan antara bot dengan line adalah protokol HTTPS.
13. Format data yang digunakan saat mengirim dan menerima pesan antara bot dengan line adalah JSON.
14. Metode pengembangan perangkat lunak yang digunakan adalah metode Waterfall.
15. Metode perancangan dan analisis perangkat lunak yang digunakan adalah metode Object Oriented Analysis & Design (OOAD) dengan menggunakan pemodelan Unified Modelling Language (UML).



Analisis Masalah

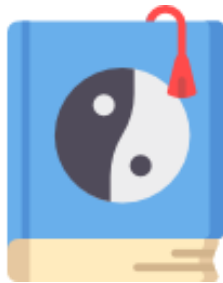
72,7%

32 dari 44 responden
kesulitan mencari informasi
mengenai Khonghucu



79,5%

Nyanyian rohani
Khonghucu



65,9%

Kitab Sishu



45,5%

Lokasi tempat
ibadah
Khonghucu

Berdasarkan masalah yang diketahui maka analisis masalah dijabarkan sebagai berikut:

1. Dibutuhkannya media informasi mengenai Khonghucu untuk mempermudah pencarian informasi mengenai Khonghucu. Berdasarkan kuesioner diketahui bahwa umat Khonghucu membutuhkan informasi mengenai **Nyanyian Rohani Khonghucu, Kitab Sishu** dan **Lokasi tempat ibadah Khonghucu**.
2. Solusi yang ditawarkan untuk mengatasi masalah ini ialah membuat aplikasi bot pada layanan pesan instan Line sebagai media informasi Khonghucu.



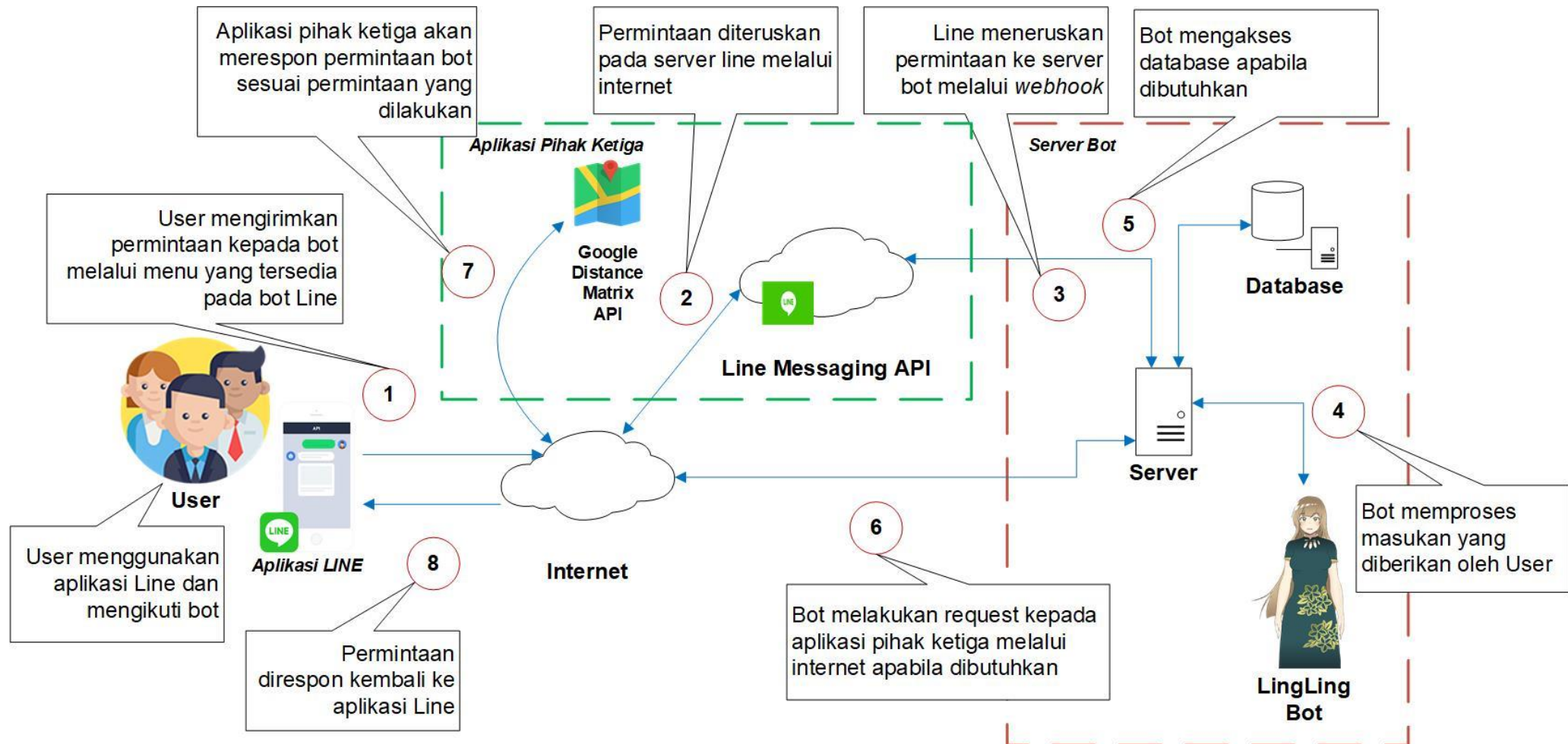
Analisis Aplikasi Sejenis

Hasil analisis aplikasi bot

Parameter	Islamify	Musleem	Icalca	Tel-U	LingLing
Konten/Fitur	a. Al Quran b. Lokasi Masjid c. Cari Kiblat d. Jadwal Sholat e. Kajian f. Hadits g. Kalender h. Stiker	a. Baca Quran b. Kajian c. Masjid Terdekat d. Event Muslim e. Jadwal Sholat f. Kalender	a. Permainan (Cari kata, tebak kata, adu penalti, dsb) b. Cari Jodoh	a. Akademik Kampus b. Lowongan Pekerjaan	a. Kitab Sishu b. Nyanyian Pujian c. Tempat Ibadah Khonghucu d. Stiker
Menyimpan Informasi Pengguna	a. Jenis Kelamin	Tidak	a. Jenis Kelamin b. Tanggal Lahir c. Lokasi d. Versi LINE	a. Todo List	a. Akun Pengguna Line (Nama, Gambar Profil) b. Statistik Penggunaan c. Konten Yang Disukai
Menggunakan Rich Menu	Iya	Iya	Iya	Iya	Iya



Analisis Deskripsi Aplikasi

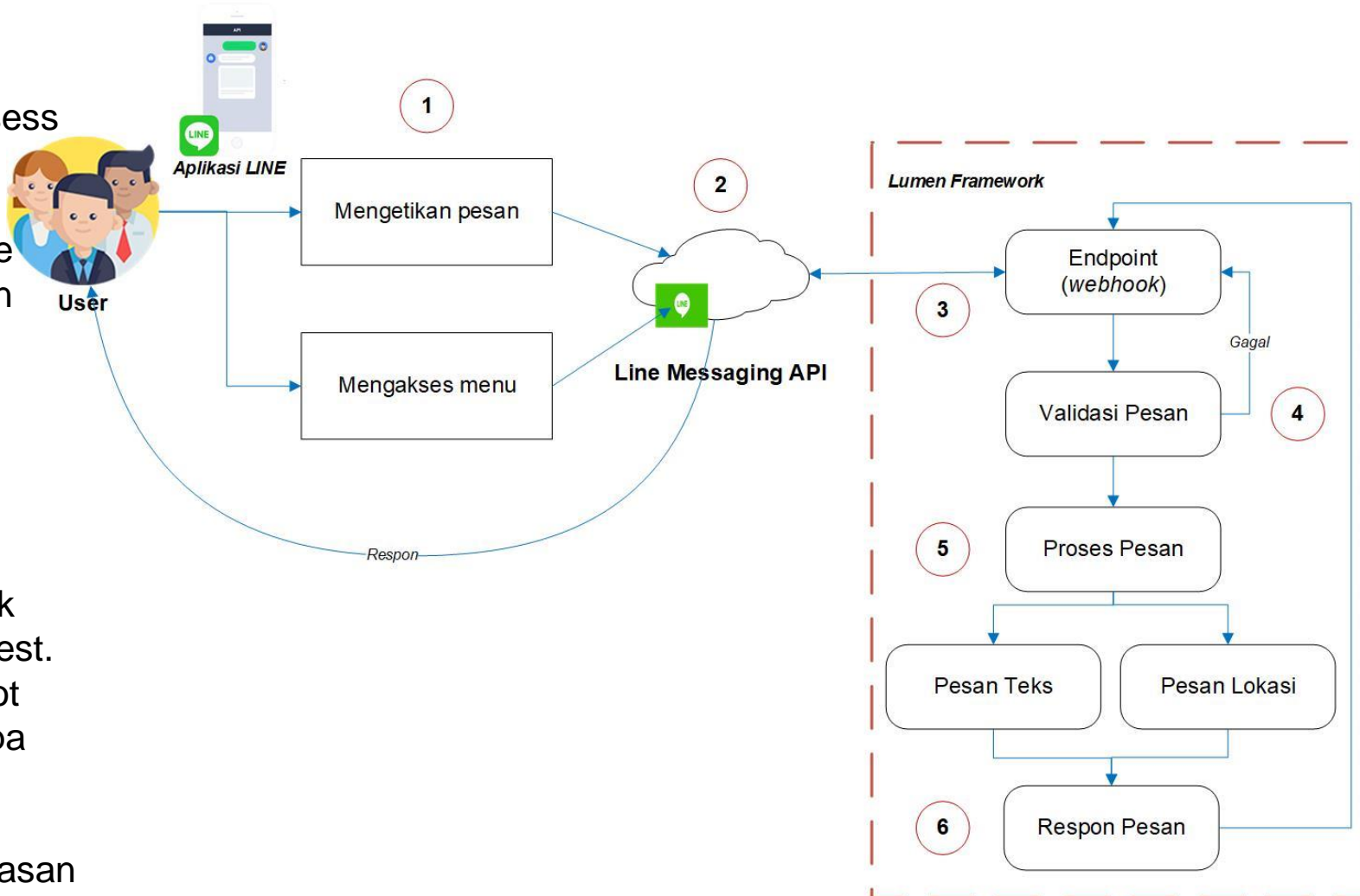




Analisis Arsitektur Aplikasi

Analisis Arsitektur Lumen Framework

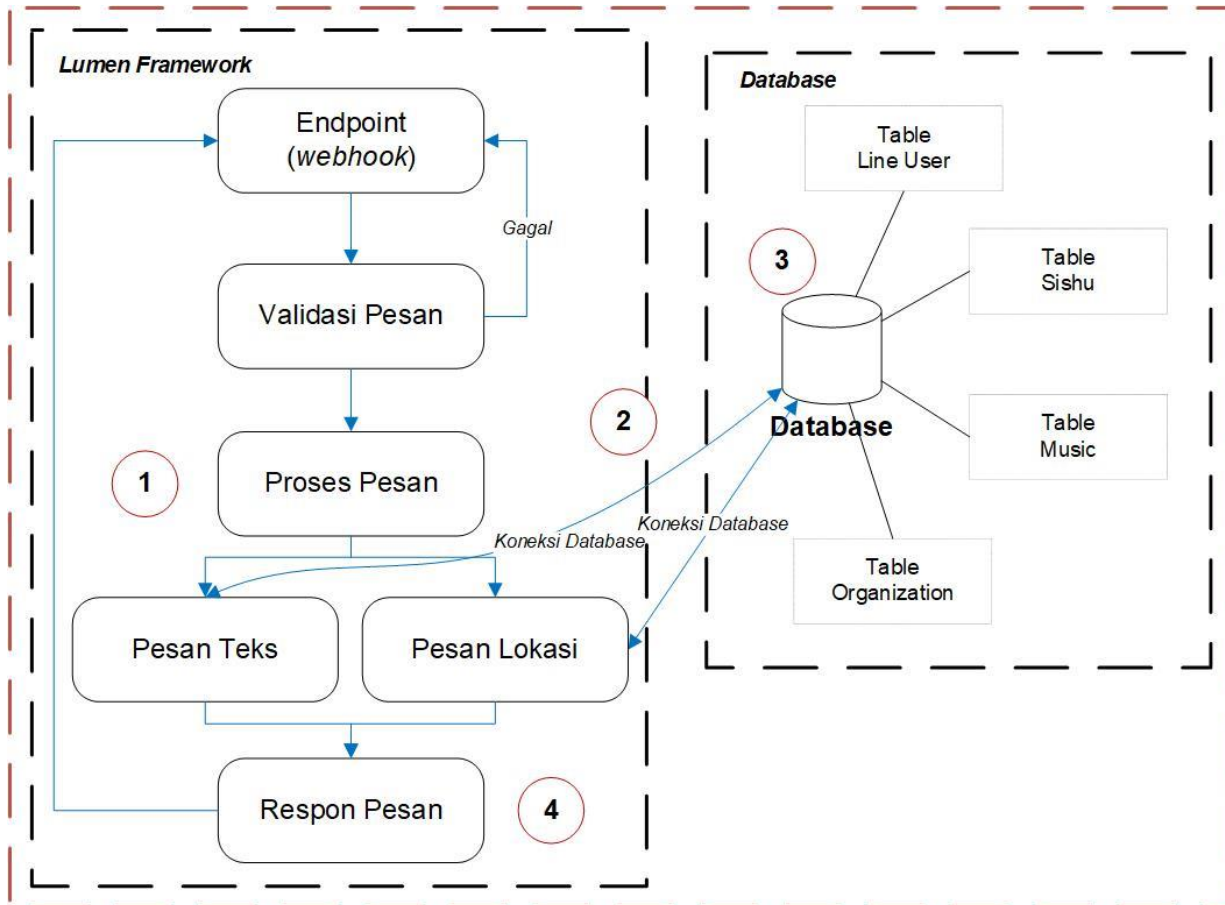
1. User akan melakukan request berupa akses menu atau mengetikan masukan berupa pesan teks pada aplikasi Line.
2. Request kemudian diteruskan melalui Line Messaging API untuk kemudian diteruskan kembali ke server bot
3. Request kemudian diterima oleh Lumen Framework melalui satu endpoint yang dinamakan dengan webhook yang sebelumnya sudah diatur pada Line Developer Console.
4. Request divalidasi oleh Lumen Framework dengan cara dikenali melalui header request.
5. Pesan yang dapat diproses oleh server bot yakni pesan berupa teks dan pesan berupa lokasi.
6. Bot kemudian akan memproses pesan dan mengembalikan respon berupa pesan balasan kepada user melalui Line Messaging API.





Analisis Arsitektur Aplikasi

Analisis Arsitektur Database



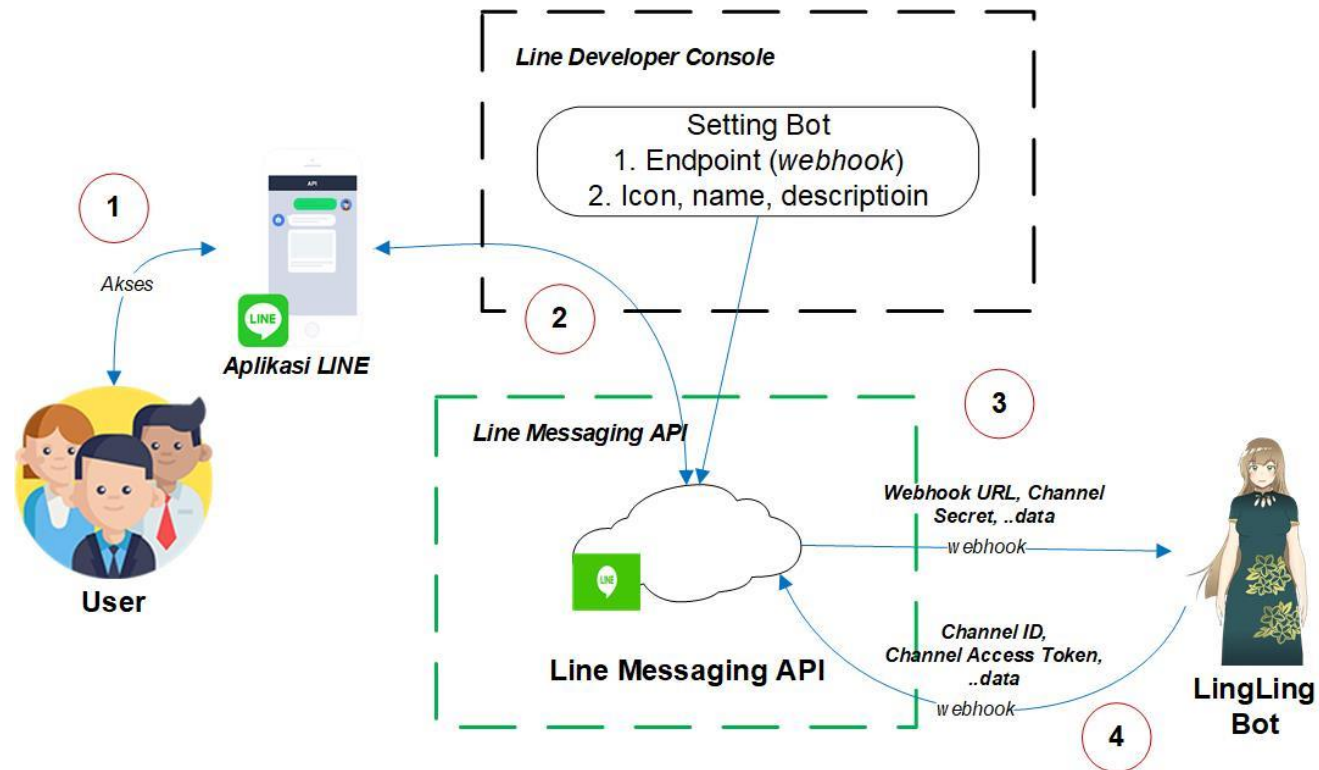
1. Setelah pesan yang dikirimkan oleh user divalidasi, pesan akan diproses sesuai dengan jenis pesan yang dikirimkan.
2. Pada pesan yang membutuhkan data yang terdapat pada database maka bot akan melakukan koneksi ke database untuk mencari data yang dibutuhkan tersebut.
3. Data yang terdapat pada database diantaranya data lokasi tempat ibadah (organisasi), data nyanyian pujian (musik), data sishu dan data pengguna aplikasi bot.
4. Jika data yang dicari ditemukan pada database maka bot akan merespon dengan pesan sesuai dengan data yang terdapat pada database. Namun jika data yang dicari tidak ditemukan pada database maka bot akan merespon dengan memberitahu pengguna bahwa data yang dicari tidak ditemukan.



Analisis Arsitektur Aplikasi

Analisis Arsitektur Line Messaging API

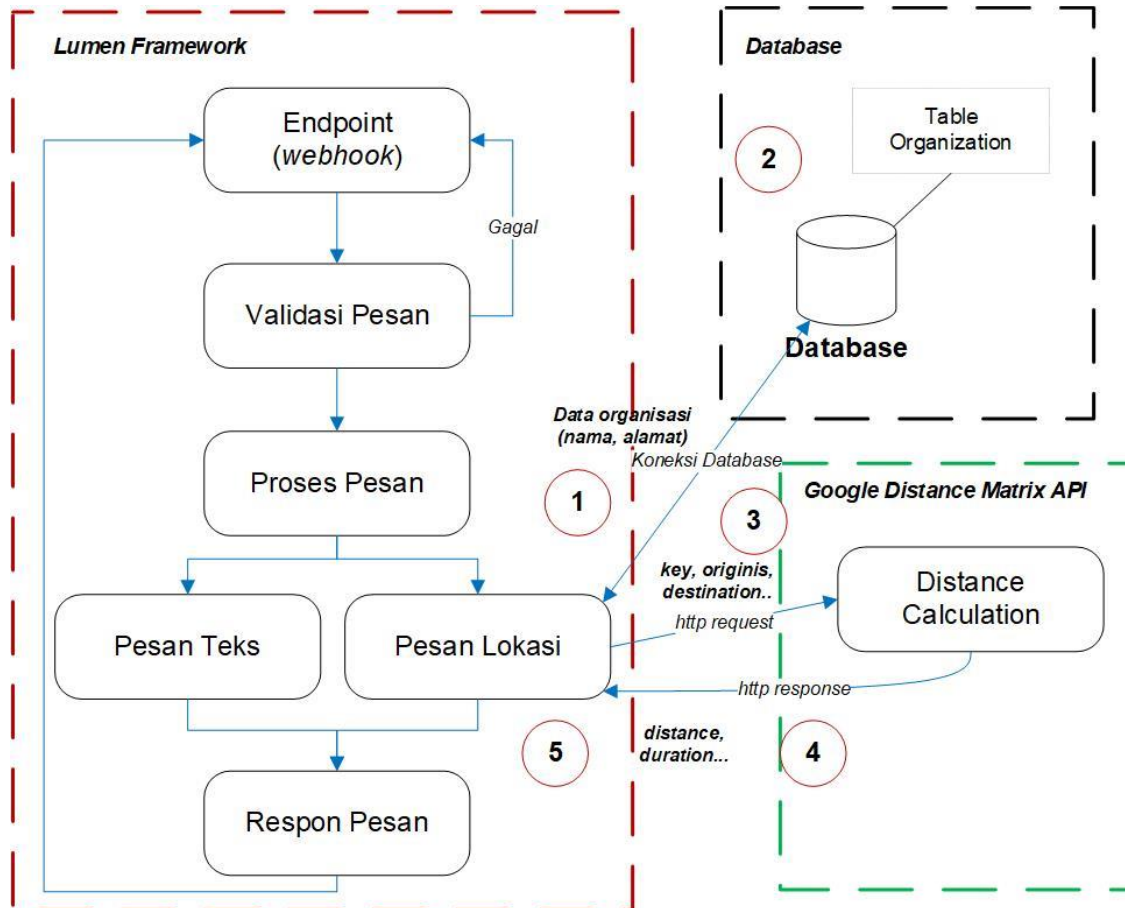
1. User mengirimkan dan menerima pesan melalui aplikasi Line.
2. Pesan kemudian dikirimkan oleh aplikasi Line melalui Line Messaging Api.
3. Pesan kemudian diteruskan melalui line messaging api kepada server bot. Line messaging api mengirimkan pesan yang dikirimkan oleh user melalui alamat server bot (webhook). Pesan yang dikirimkan oleh bot didalamnya sudah terdapat alamat channel dan informasi pesan yang dikirimkan oleh user.
4. Server bot kemudian akan memvalidasi, memproses dan merespon pesan sesuai dengan pesan yang dikirimkan oleh user. Pesan tersebut kemudian dikirimkan kembali melalui Line Messaging API dan ditampilkan kembali melalui aplikasi Line.





Analisis Arsitektur Aplikasi

Analisis Arsitektur Google Distance Matrix API



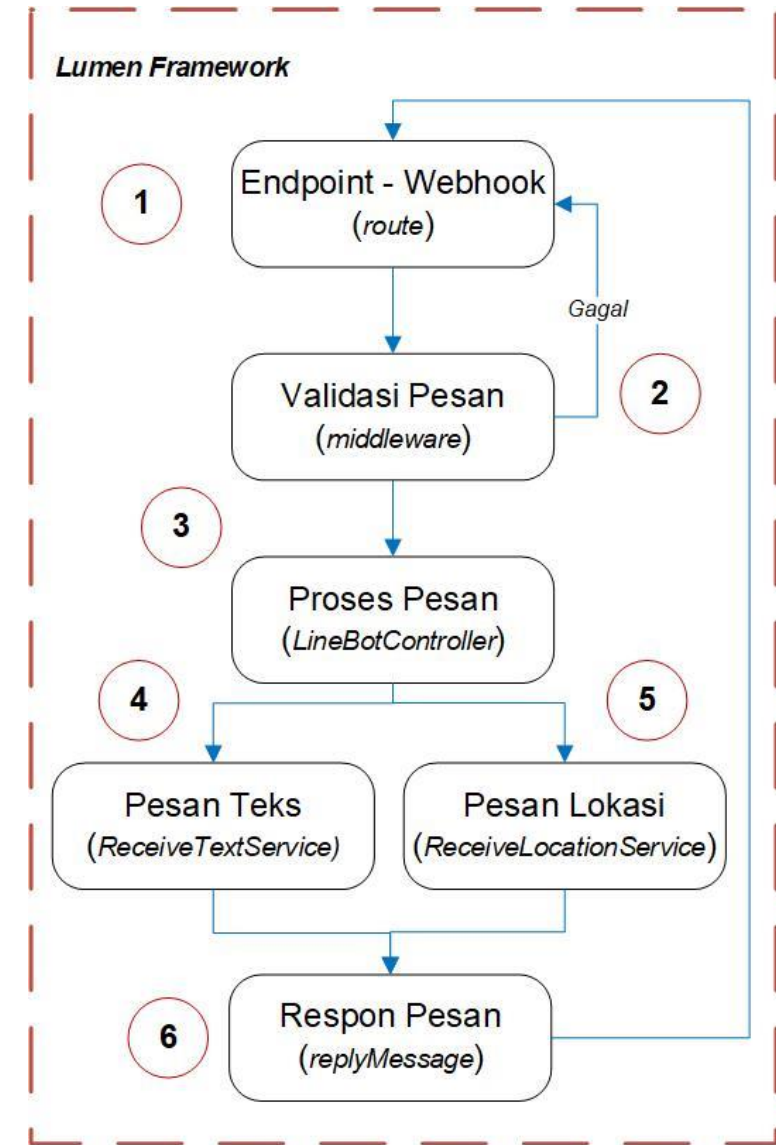
1. User akan mengirimkan lokasinya kepada bot, lokasi tersebut berupa lokasi saat ini berada ataupun lokasi yang dipilih oleh user.
2. Bot kemudian akan memproses permintaan tersebut lalu melakukan koneksi ke database untuk mendapatkan data seluruh tempat ibadah yang sudah tersedia pada database.
3. Setelah seluruh data tempat ibadah didapatkan, bot akan melakukan request kepada Google Distance Matrix API untuk menghitung lokasi terdekat antara lokasi user dengan seluruh data lokasi tempat ibadah.
4. Google Distance Matrix API kemudian akan menghitung jarak antara lokasi yang dikirimkan oleh user dengan lokasi tempat ibadah yang tersedia pada database.
5. Bot akan menerima respon yang dikirimkan oleh Google Distance Matrix API. Apabila hasil ditemukan, Bot kemudian akan mengurutkan hasil respon yang diberikan berdasarkan jarak yang terdekat dengan lokasi yang dikirimkan oleh user. Apabila tidak ditemukan maka bot akan merespon dengan pesan berupa informasi bahwa tidak ditemukan lokasi terdekat.



Analisis Teknologi

Analisis Teknologi Lumen Framework

1. User akan mengirimkan pesan melalui aplikasi Line, aplikasi Line kemudian meneruskan pesan melalui Line Messaging API dan kemudian diteruskan ke server bot melalui webhook (route) yang sebelumnya sudah diatur.
2. Server bot kemudian melakukan validasi pesan yang diterima. Validasi dilakukan untuk memastikan bahwa pesan yang dikirimkan benar-benar dikirimkan melalui Line Messaging API. Validasi ini dilakukan dengan cara melihat dan membandingkan header HTTP_X_LINE_SIGNATURE.
3. Jika pesan berhasil divalidasi maka pesan akan diproses oleh LineBotController dengan menggunakan method index. Pesan kemudian akan diproses sesuai dengan jenis pesan yang dikirimkan.
4. Jika pesan yang dikirimkan berupa pesan teks, maka pesan akan diproses oleh ReceiveTextService. Pada pesan jenis teks, pesan dianggap sebagai kata kunci untuk melakukan pencarian.
5. Jika pesan yang dikirimkan berupa pesan lokasi, maka pesan akan diproses oleh ReceiveLocationService. Pada pesan jenis lokasi, pesan akan dianggap sebagai pesan untuk mencari lokasi tempat ibadah yang terdekat dengan lokasi yang dikirimkan oleh user.
6. Setiap pesan kemudian akan diproses dan hasil pemrosesan tersebut akan dikirimkan kembali kepada user melalui method replyMessage. Apabila terdapat data yang digunakan, maka pesan yang dikirimkan akan mengandung data yang diperoleh dari hasil pemrosesan sebelumnya.





Analisis Teknologi

Analisis Teknologi Fulltext Search

Fulltext Search adalah fitur yang dapat digunakan untuk melakukan pencarian pada database MySQL/MariaDB berdasarkan kata kunci dengan cara yang lebih spesifik dibandingkan pencarian dengan query LIKE [7]. Fulltext Search pada engine database InnoDB baru tersedia di MySQL versi 5.6 yang rilis pada tahun 2013.

Fulltext Search hanya dapat diterapkan pada engine database MyISAM dan InnoDB serta pada tipe data CHAR, VARCHAR, atau TEXT. Secara umum terdapat dua metode pencarian yang dapat digunakan dalam melakukan pencarian Fulltext.

1. Metode Natural Language

Metode Natural Language Fulltext Search adalah metode pencarian pada database MySQL yang dilakukan dengan pendekatan bahasa natural.

2. Metode Boolean

Metode Boolean Fulltext Search adalah metode pencarian pada database MySQL yang menerapkan metode yang sama dengan Natural Language Fulltext Search, namun pada metode ini dapat diterapkan operator dalam melakukan pencarian [22].



Analisis Teknologi

Analisis Teknologi Fulltext Search

Cari: **Buka Hati**

ID	Title (Judul)	Lyric (Lirik)
1	Menyelami Hati	Saat Nabi Di Puncak Tong San Nampak Kecillah Negeri Lo Saat Nabi Di Puncak Thai San Nampak Kecillah Dunia Ini Jika Telah Melihat Lautan Sedikitlah Air Di Sungai Yang Telah Di Dalam Gerbang Nabi Mengerti Tentang Makna Kata Ada Cara Memandang Air Lihatlah Buih Di Gelombangnya. Mentari Dan Bulan Bersinar Yang Tembus Cahaya Tersinar Laksana Air Terus Mengalir Memenuhi Suatu Tempat Bila Belum Cukup Belajar Janganlah Merasa Berhasil
2	Buka Hati	O Nabi Kini Kami Berdiri Buka Hati Terima Sabda Akhir Yang Dikau Ucapkan Bila Gunung Thai San Runtuhlah Betapa Pohon Dahan Semua Bila Sang Budiman Gugurlah O Betapa Sayu Sedih M’liput Hati Mengenang Kemangkatan Tetapi Ini Hanya Menambah Tekadku.
3	Menuntut Ilmu	Menuntut Ilmu Setiap Hari Janganlah Lupa Diulang Lagi Latih Dirimu Pada Masa Muda Bina Diri Slalu Ajaran Nabi Sabda Nabi Khongcu Bekal Hidupmu
4	Damai Di Dunia	Berdiri Kita Semua Di Dalam Sikap Pat Tik Menghadap Altar Nabi Khongcu Nabi Penyedat Hidup Berdoalah Bersama Dengan Hati Yang Suci Kepada Thian Yang Maha Esa Agar Damai Di Dunia



Analisis Teknologi

Analisis Teknologi Fulltext Search

Perhitungan nilai TF

ID	Kata	Term Frequency (TF)
1	Buka	0
	Hati	1
2	Buka	1
	Hati	1
3	Buka	0
	Hati	0
4	Buka	0
	Hati	0

Perhitungan nilai TF-IDF-IDF

ID	Kata	Relevansi (Title)
1	Buka	$(0) \times \log_{10}(4/1) \times \log_{10}(4/1) = 0$
	Hati	$(1) \times \log_{10}(4/2) \times \log_{10}(4/2) = 0,090619$
2	Buka	$(1) \times \log_{10}(4/1) \times \log_{10}(4/1) = 0,362476$
	Hati	$(1) \times \log_{10}(4/2) \times \log_{10}(4/2) = 0,090619$
3	Buka	$(0) \times \log_{10}(4/1) \times \log_{10}(4/1) = 0$
	Hati	$(0) \times \log_{10}(4/2) \times \log_{10}(4/2) = 0$
4	Buka	$(0) \times \log_{10}(4/1) \times \log_{10}(4/1) = 0$
	Hati	$(0) \times \log_{10}(4/2) \times \log_{10}(4/2) = 0$

$$\{IDF\} = \log_{10} \left(\frac{\{total\ records\}}{\{matching\ records\}} \right) \quad (1) \quad \{rank\} = \{TF\} * \{IDF\} * \{IDF\} \quad (2) \quad \{rank\} = \sum_{i=1}^n (\{TF\} * \{IDF\} * \{IDF\}) \quad (3)$$



Analisis Teknologi

Analisis Teknologi Fulltext Search

Hasil Perhitungan nilai TF-IDF-IDF

ID	Relevansi Buka	Relevansi Hati	Relevansi Total
1	0	0,090619	0,090619
2	0,362476	0,090619	0,453095
3	0	0	0
4	0	0	0

Hasil query fulltext

Showing rows 0 - 3 (4 total, Query took 0.0060 seconds.)

```
1 SELECT
2   id, title, lyric,
3   MATCH (title) AGAINST('buka hati' IN BOOLEAN MODE) relevancy
4 FROM `music`
5 ORDER BY relevancy DESC
```

☒ Enable foreign key checks

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#)

☐ Show all | Number of rows: 25 | Filter rows: Sort by key:

+ Options

	id	title	lyric	relevancy
<input type="checkbox"/> Edit Copy Delete	2	Buka Hati	O Nabi Kini Kami
 Berdiri Buka Hati <...	0.45309528708457947
<input type="checkbox"/> Edit Copy Delete	1	Menyelami Hati	Saat Nabi Di Puncak Tong San
 Nampak Ke...	0.0906190574169159
<input type="checkbox"/> Edit Copy Delete	3	Menuntut Ilmu	Menuntut Ilmu Setiap Hari
 Janganlah Lu...	0
<input type="checkbox"/> Edit Copy Delete	4	Damai Di Dunia	Berdiri Kita Semua
 Di Dalam Sikap Pat ...	0

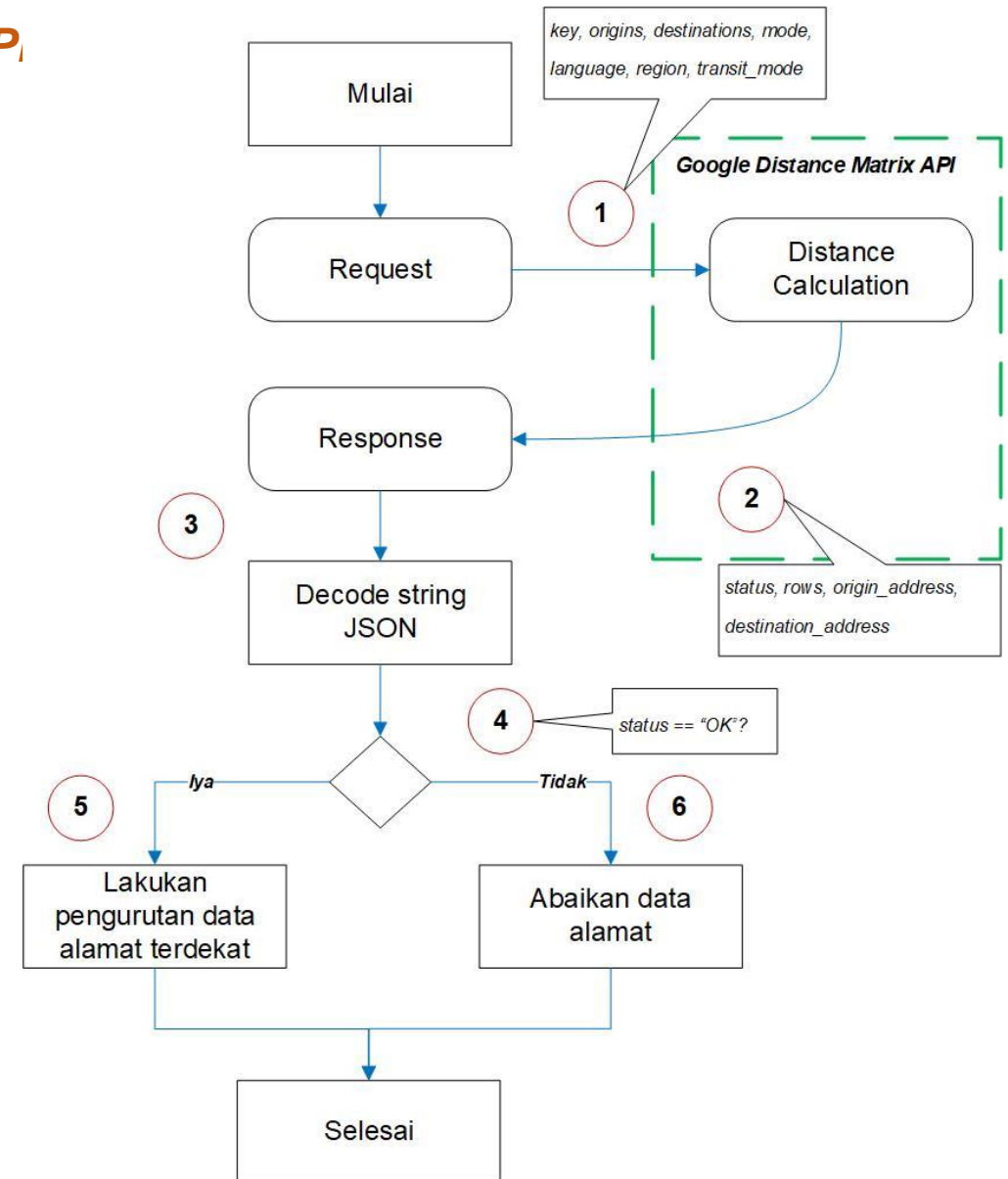
☐ Check all With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)



Analisis Teknologi

Analisis Teknologi Google Distance Matrix API

1. Request dilakukan dengan cara mengirimkan data berupa: key, origins, destinations, mode, language, region, transit_mode. Data wajib yang harus dikirimkan yakni: key, origins, dan destinations.
2. Google distance matrix api akan mengembalikan response berupa data dalam format json yang terdiri dari: origin_address, destination_address, rows, dan status.
3. Hasil response json berupa string kemudian diubah kedalam bentuk array atau objek untuk kemudian diproses didalam aplikasi.
4. Aplikasi akan memeriksa apakah array atau objek response status bernilai "OK"
5. Jika status bernilai "OK" maka aplikasi akan memproses dan mengolah data dengan cara mengurutkan data alamat berdasarkan alamat terdekat.
6. Jika status tidak bernilai "OK" maka aplikasi tidak akan memproses data alamat dan data alamat dianggap diabakan (tidak dihitung).





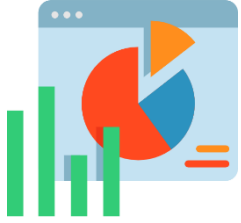
Analisis Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini yakni merupakan perangkat keras penunjang aplikasi bot berupa server untuk kebutuhan bot.



<http://bit.ly/CloudDigitalOcean>

No	Perangkat	Spesifikasi
1	Prosesor	1 CPU
2	Disk	10 GB
3	Memory	1 GB



Analisis Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam membangun dan mengimplementasikan aplikasi bot sebagai berikut.

A. User (Client)

1. Aplikasi Pesan Instan (LINE 5.9.1+)

B. Bot (Server)

1. Sistem Operasi Linux (Ubuntu 14.04)
2. Web Server (Nginx 1.13+)
3. Database server (MySQL 5.6.1+)
4. Interpreter PHP (PHP 7.1.3+)
5. PHP Framework (Lumen 5.6+)
6. PHP Package Depedency Manager (Composer 1.0+)



Analisis Perangkat Pikir

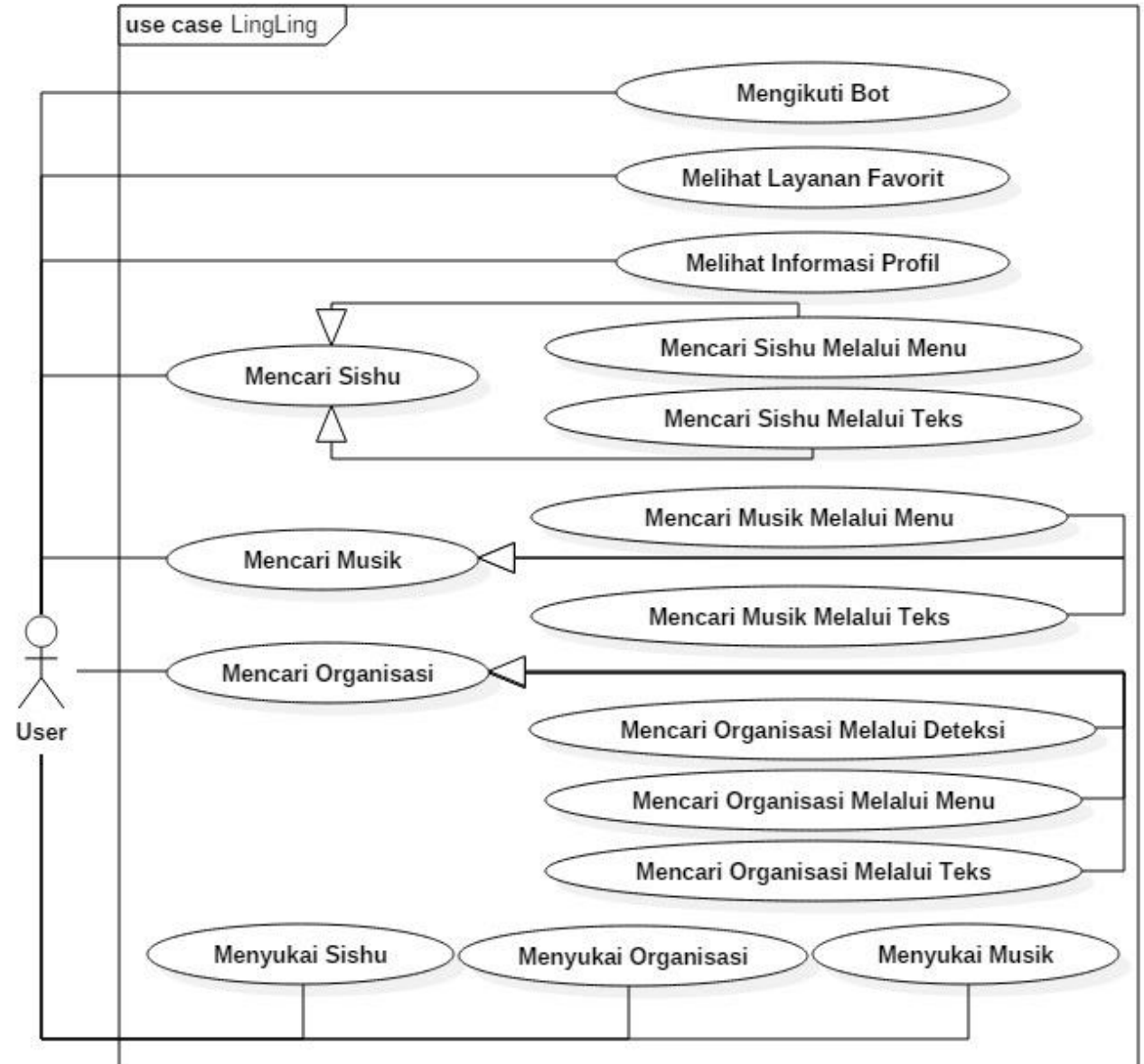
Analisis perangkat pikir dimaksudkan untuk mengetahui pengguna yang akan terlibat dalam menggunakan aplikasi Bot Line ini.

Stakeholder	Pengguna Line
Tingkat Keterampilan	<ul style="list-style-type: none">a. Mempunyai dan mampu menggunakan smartphoneb. Pernah mengakses layanan internet melalui smartphonec. Pernah menggunakan dan menambahkan teman pada aplikasi Lined. Pernah melakukan chatting menggunakan aplikasi Line



Use Case Diagram

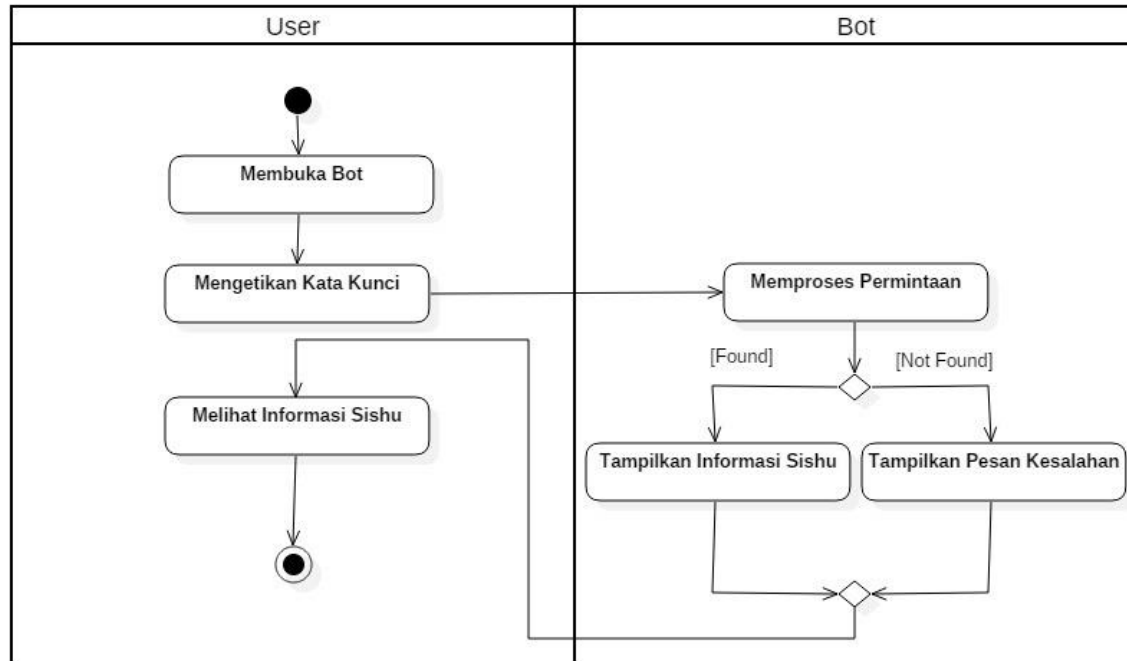
Fungsional yang terdapat didalam aplikasi dimodelkan melalui use case diagram.



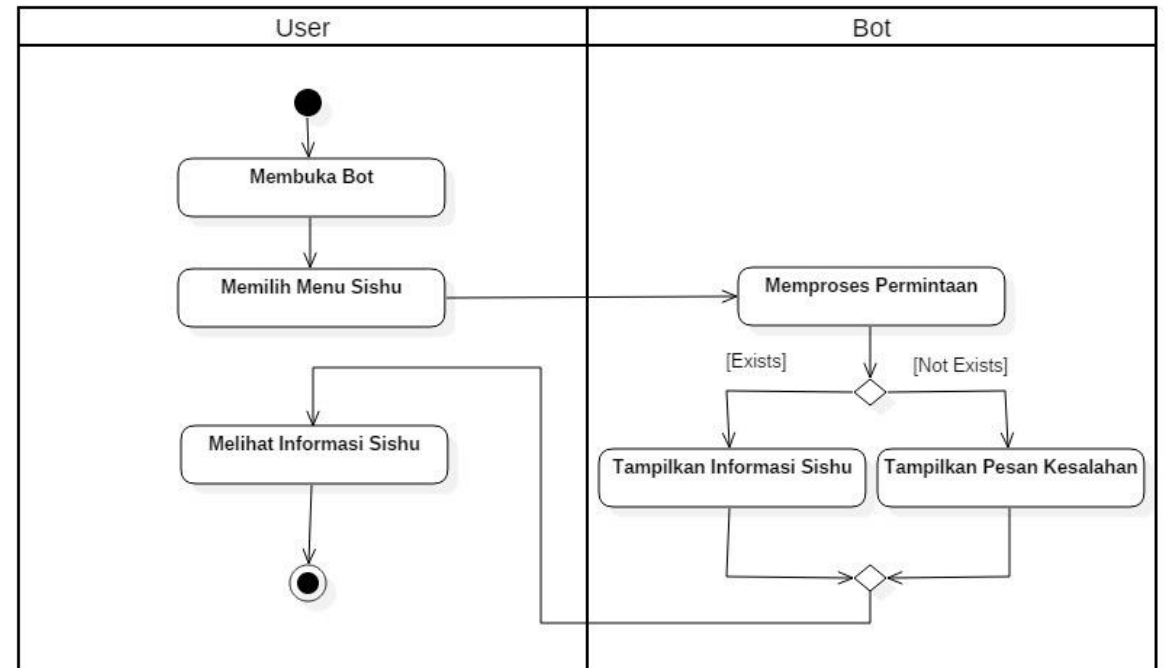


Activity Diagram

Activity Diagram Mencari Sishu Melalui Teks



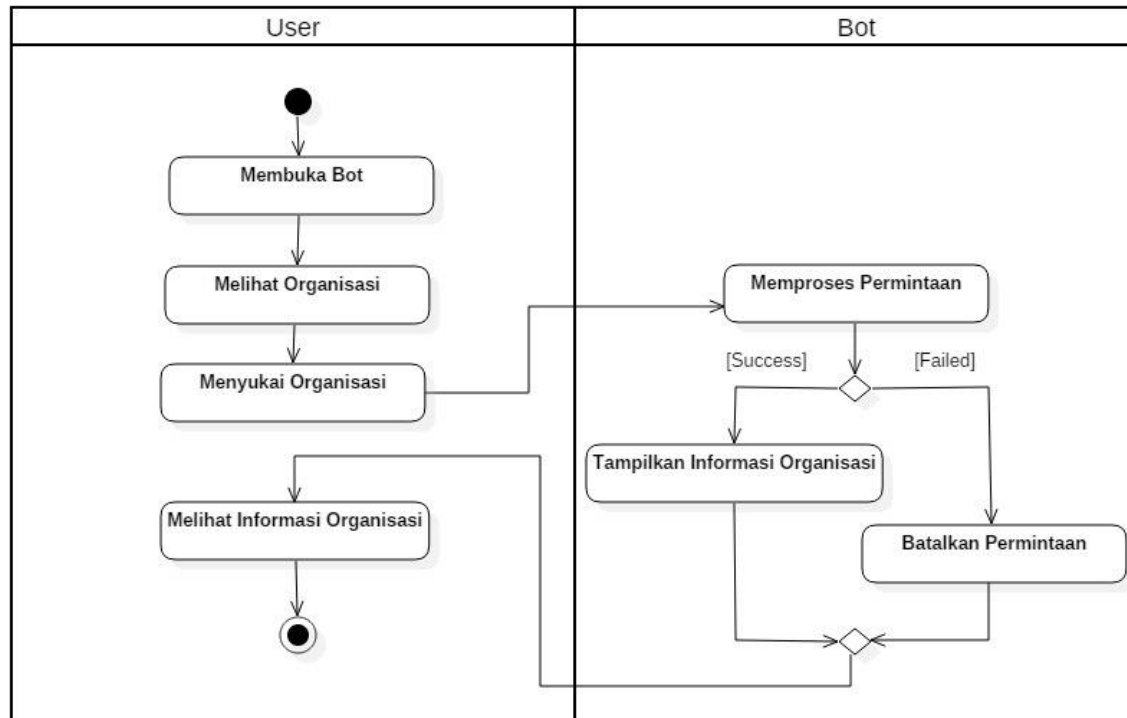
Activity Diagram Mencari Sishu Melalui Menu



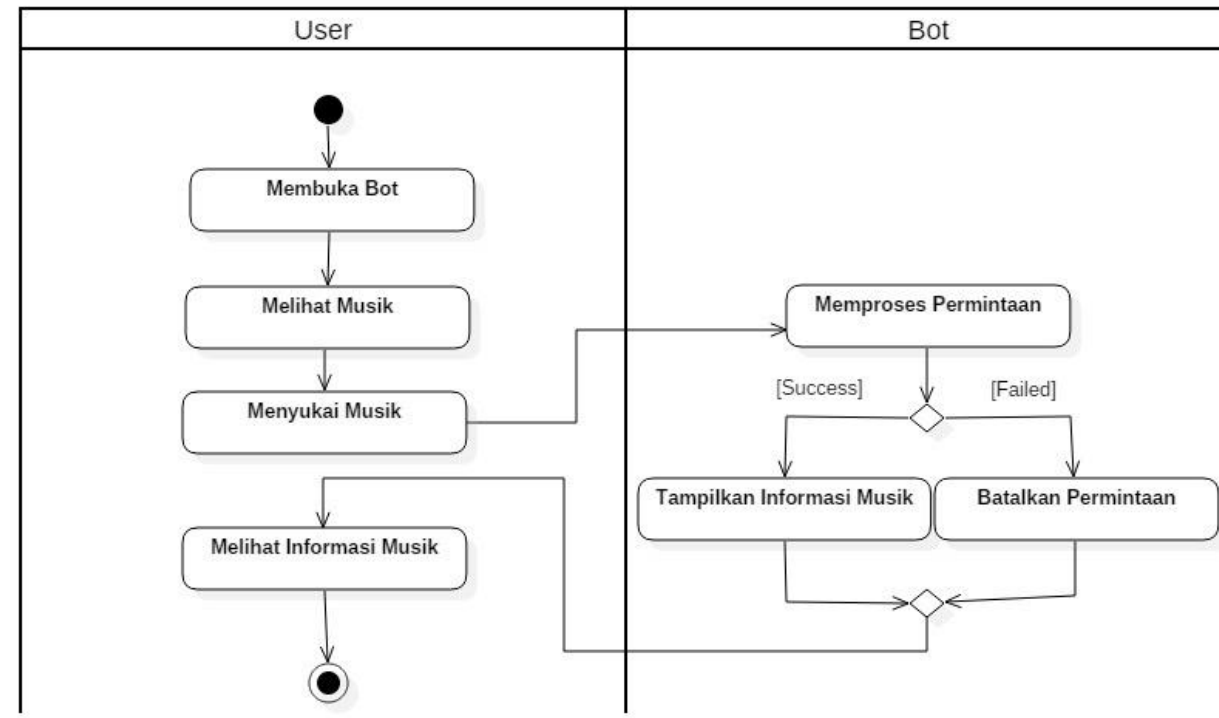


Activity Diagram

Activity Diagram Menyukai Organisasi



Activity Diagram Menyukai Musik





```

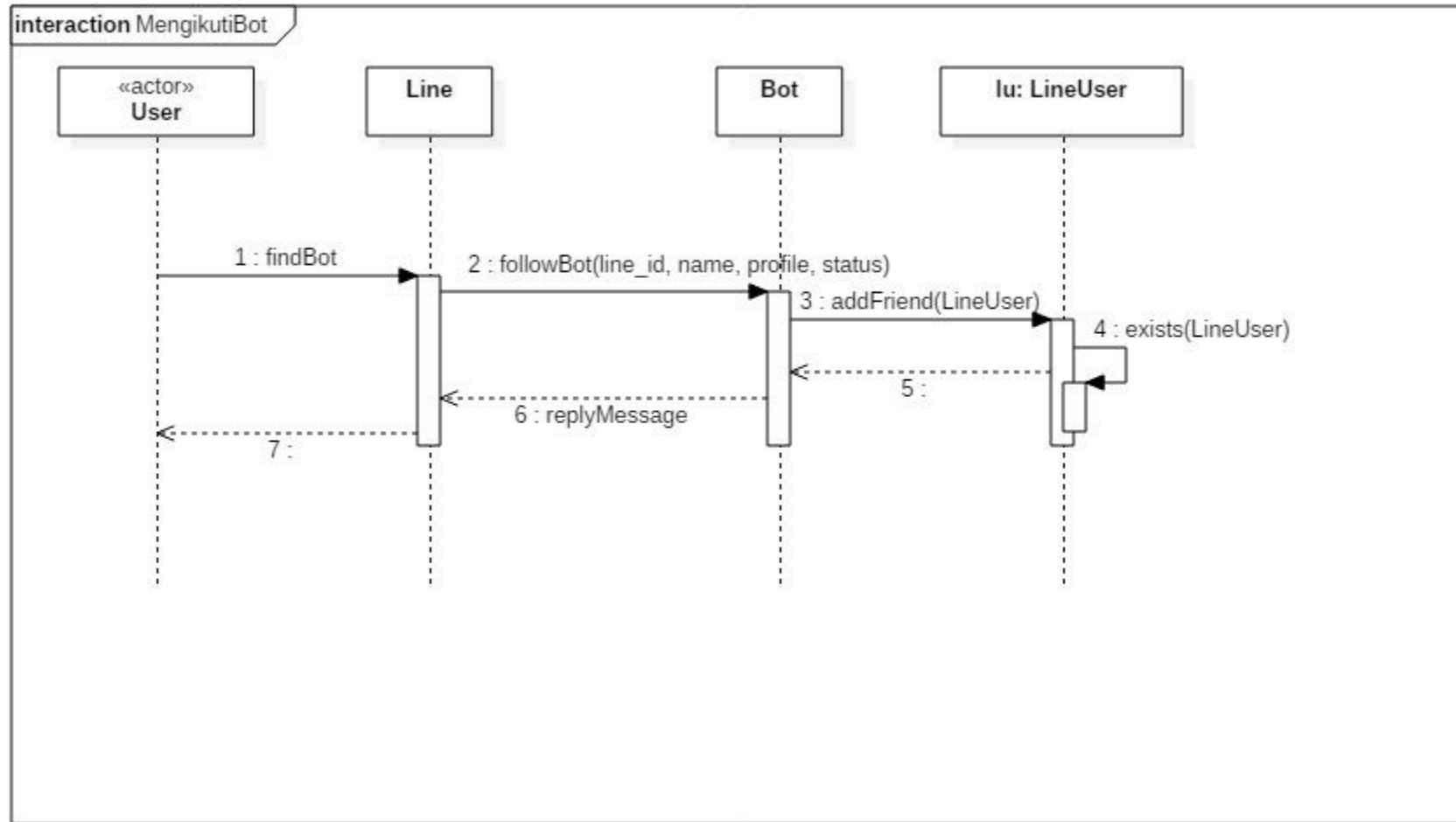
classDiagram
    class Province {
        +id: Integer
        +name: String
        +thumbnail: String
        +get(): Province
        +all(): Province[*]
    }
    class Organization {
        +id: Integer
        +name: String
        +slug: String
        +address: String
        +phone: String
        +email: String
        +province_id: Integer
        +created_at: Timestamp
        +updated_at: Timestamp
        +get(): Organization
        +all(): Organization[*]
    }
    class LineUserLoveMusic {
        +line_user_id: String
        +music_id: Integer
        +created_at: Timestamp
        +updated_at: Timestamp
        +get(): LineUserLoveMusic
        +all(): LineUserLoveMusic[*]
    }
    class LineUser {
        +id: Integer
        +user_id: String
        +line_id: String
        +name: String
        +picture: String
        +status: Boolean
        +created_at: Timestamp
        +updated_at: Timestamp
        +get(): LineUser
        +all(): LineUser[*]
    }
    class HeroMusicContribution {
        +hero_id: Integer
        +music_id: Integer
        +created_at: Timestamp
        +updated_at: Timestamp
        +get(): HeroMusicContribution
        +all(): HeroMusicContribution[*]
    }
    class Hero {
        +id: Integer
        +slug: String
        +name: String
        +honor_id: Integer
        +created_at: Timestamp
        +updated_at: Timestamp
        +get(): Hero
        +all(): Hero[*]
    }
    class SishuCategory {
        +id: String
        +english: String
        +indonesian: String
        +pinyin: String
        +zhongwen: String
        +slug: String
        +description: String
        +orders: Integer
        +get(): SishuCategory
        +all(): SishuCategory[*]
    }
    class LineUserLoveSishu {
        +line_user_id: String
        +sishu_section_id: String
        +created_at: Timestamp
        +updated_at: Timestamp
        +get(): LineUserLoveSishu
        +all(): LineUserLoveSishu[*]
    }
    class Honor {
        +id: Integer
        +title: String
        +name: String
        +get(): Honor
        +all(): Honor[*]
    }
    class HeroHonor {
        +hero_id: Integer
        +honor_id: Integer
        +created_at: Timestamp
        +updated_at: Timestamp
        +get(): HeroHonor
        +all(): HeroHonor[*]
    }
    class SishuChapter {
        +id: String
        +title: String
        +name: String
        +slug: String
        +orders: Integer
        +sishu_category_id: String
        +get(): SishuChapter
        +all(): SishuChapter[*]
    }
    class SishuSection {
        +id: String
        +title: String
        +slug: String
        +orders: Integer
        +sishu_chapter_id: String
        +get(): SishuSection
        +all(): SishuSection[*]
    }
    class Sishu {
        +id: String
        +title: String
        +content: String
        +sishu_section_id: String
        +get(): Sishu
        +all(): Sishu[*]
    }
    class OrganizationHandler {
        -data
        +execute()
        +menu(data)
        +province(data)
        +detect(data)
        +list(data)
        +view(data)
        +love(data)
    }
    class UserHandler {
        -data
        +execute()
        +menu(data)
        +favourite(data)
        +profile(data)
    }
    class LineBotController {
        +index(Request)
    }
    class SishuHandler {
        -data
        +execute()
        +menu(data)
        +chapter(data)
        +section(data)
        +view(data)
        +love(data)
    }
    class MusicHandler {
        -data
        +execute()
        +menu(data)
        +list(data)
        +view(data)
        +love(data)
    }
    class FollowService {
        +execute(FollowEvent)
        +addFriend(FollowEvent)
    }
    class PostbackService {
        -bot
        -user
        +execute(PostbackEvent)
        +menu(data, parameter = null)
    }
    class ReceiveTextService {
        +execute(TextMessage)
        +store(TextMessage)
    }
    class ReceiveLocationService {
        +execute(LocationMessage)
        +store(LocationMessage)
    }

    Province "1" -- "*" Organization
    Organization "*" -- "*" LineUserLoveOrganization
    LineUserLoveOrganization "*" -- "*" LineUser
    LineUser "*" -- "*" LineUserLoveMusic
    LineUser "*" -- "*" LineUserLoveSishu
    LineUserLoveMusic "*" -- "*" Music
    HeroMusicContribution "*" -- "*" Hero
    Hero "*" -- "*" HeroHonor
    SishuCategory "1" -- "*" SishuChapter
    SishuChapter "1" -- "*" SishuSection
    SishuSection "1" -- "*" Sishu
    LineUserLoveSishu "*" -- "*" LineUser
    LineUserLoveSishu "*" -- "*" SishuSection
    LineUserLoveSishu "*" -- "*" Sishu
    Honor "*" -- "*" HeroHonor
    SishuHandler --> OrganizationHandler
    SishuHandler --> UserHandler
    SishuHandler --> LineBotController
    SishuHandler --> SishuHandler
    SishuHandler --> MusicHandler
    SishuHandler --> FollowService
    SishuHandler --> PostbackService
    SishuHandler --> ReceiveTextService
    SishuHandler --> ReceiveLocationService
    
```



Sequence Diagram

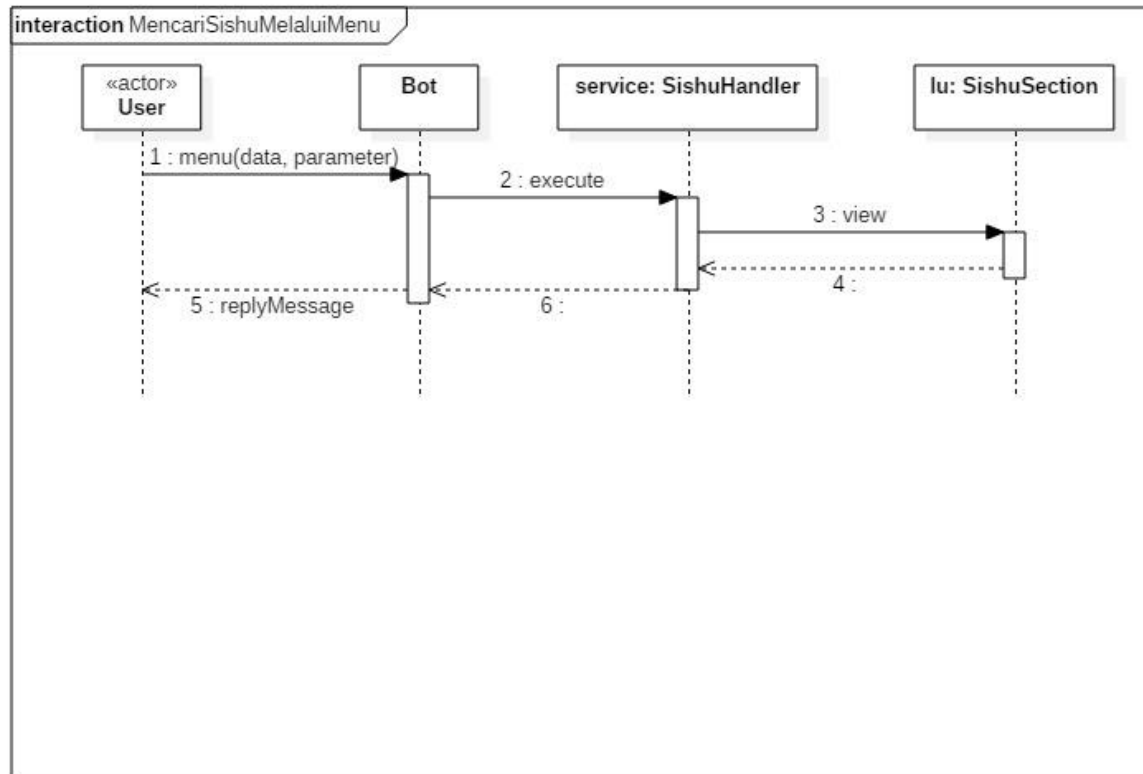
Sequence Diagram Mengikuti Bot



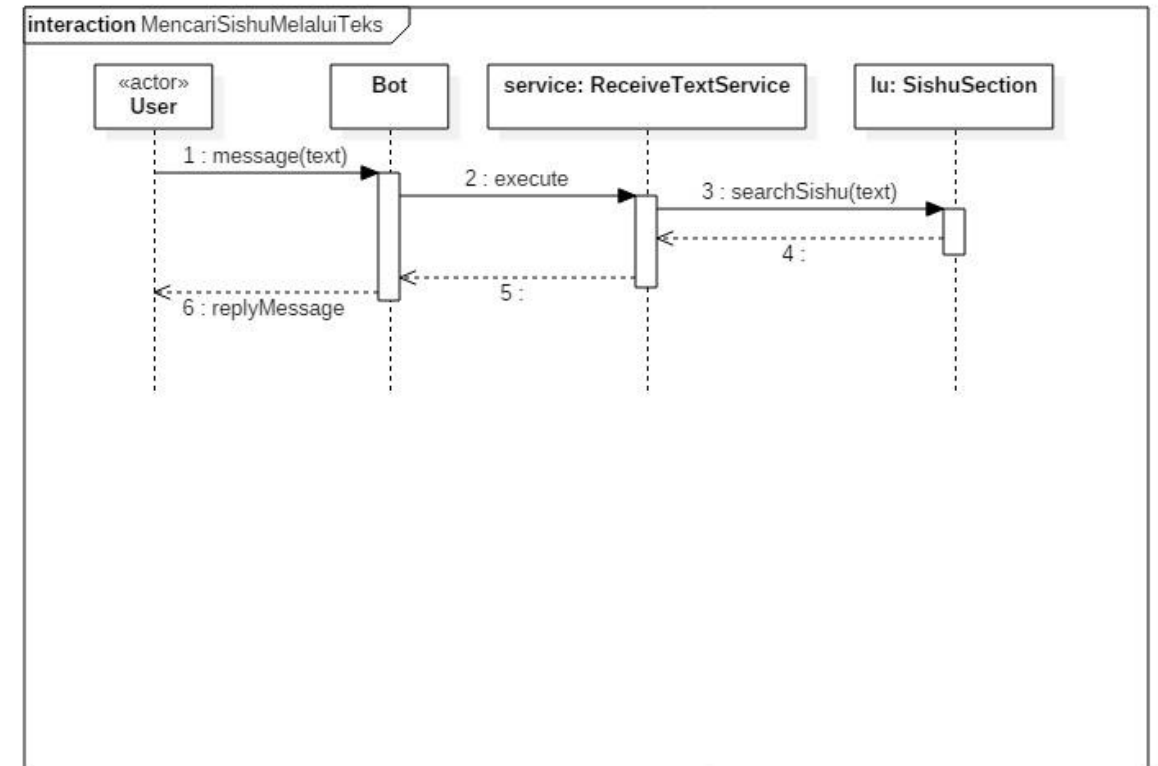


Sequence Diagram

Sequence Diagram Mencari Sishu Melalui Menu



Sequence Diagram Mencari Sishu Melalui Teks





Perancangan Skema Relasi

Skema relasi menggambarkan hubungan antara tabel yang digunakan aplikasi.

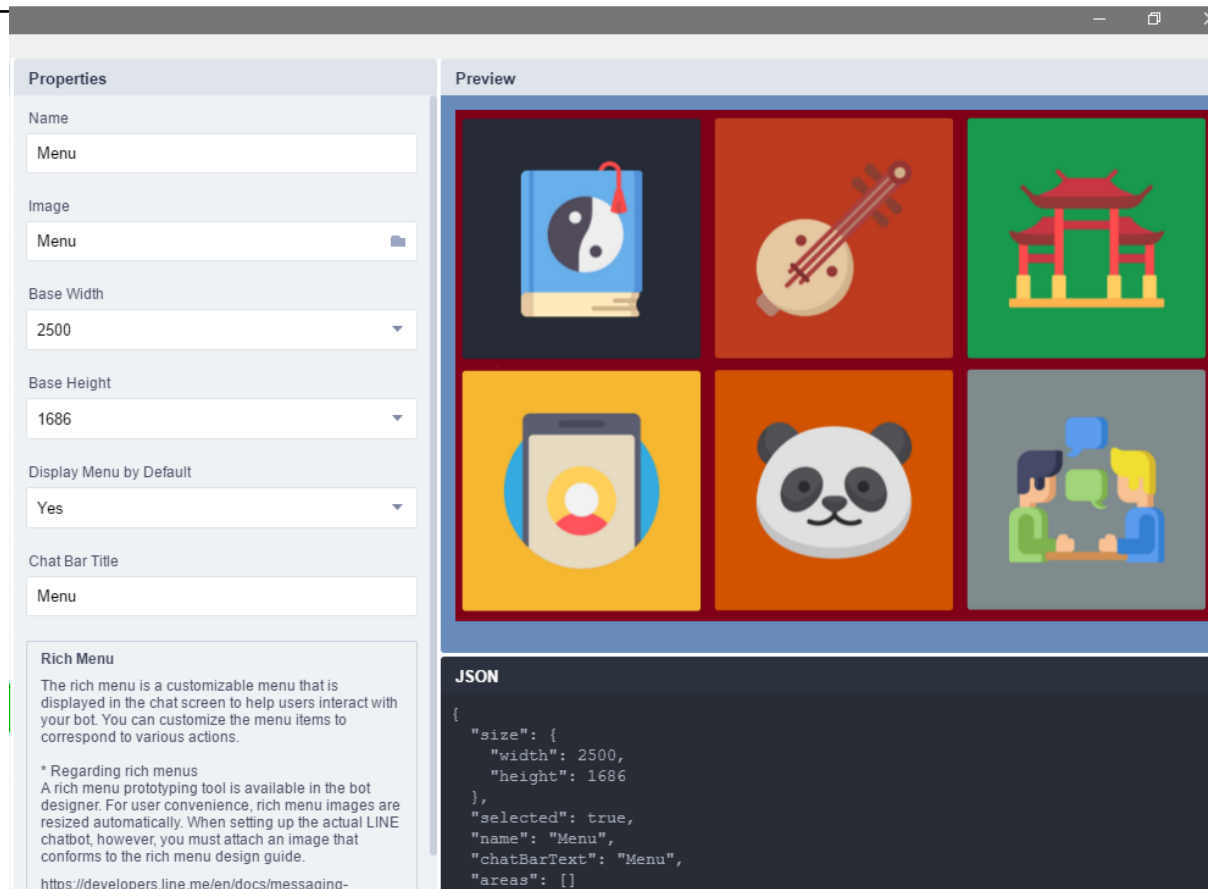




Perancangan Antarmuka

Rancangan antarmuka (A1 – Menu)

No: A1 (Menu)



1. Tekan Gambar Buku untuk masuk ke A2.
2. Tekan Gambar Alat Musik untuk masuk ke A7.
3. Tekan Gambar Bangunan untuk masuk ke A11.
4. Tekan Gambar Smartphone untuk masuk ke A16.
5. Tekan Gambar Panda untuk melihat stiker.
6. Tekan Gambar Orang untuk membagikan bot.

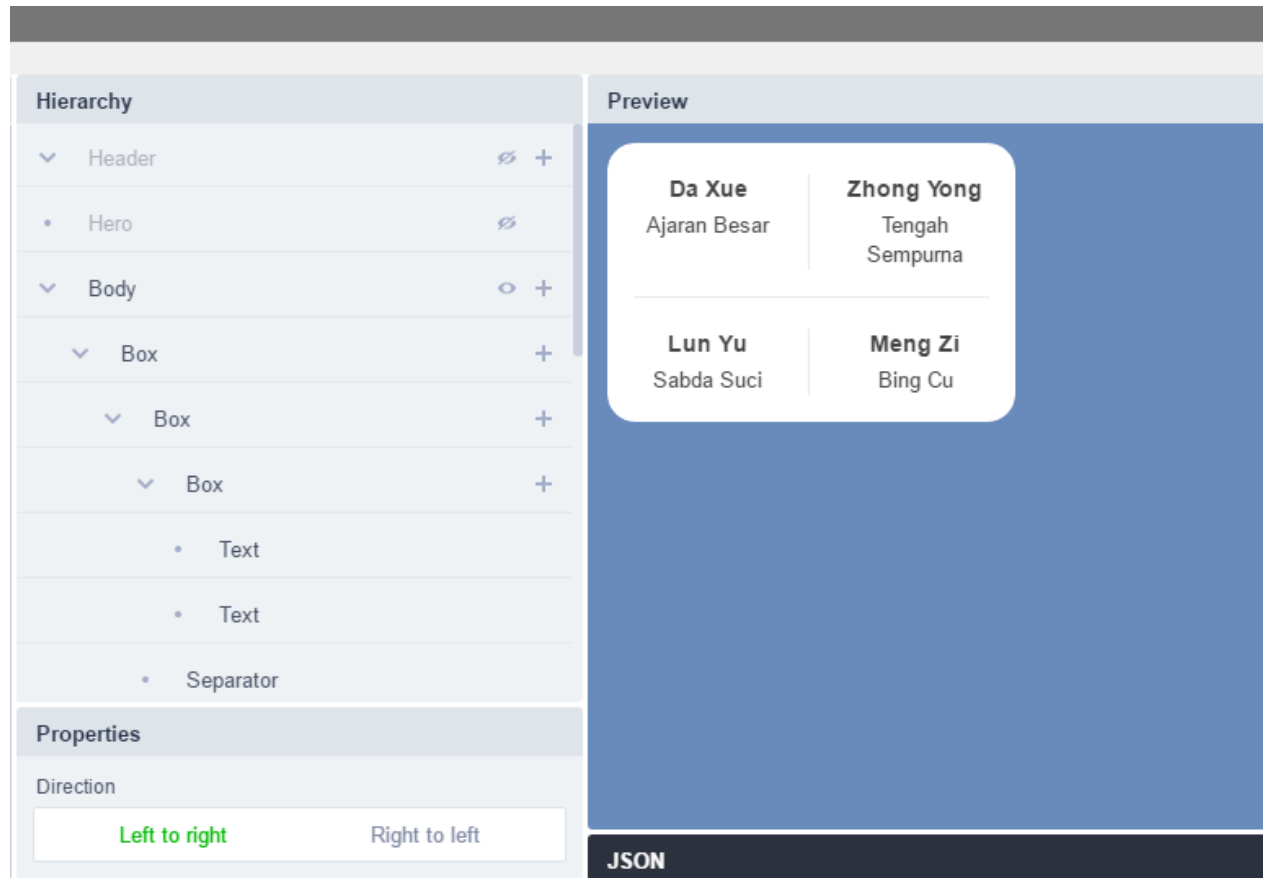
Ukuran: 2500x1686px



Perancangan Antarmuka

Rancangan antarmuka (A2 – Indeks Sishu)

No: A2 (Indeks Sishu)



1. Tekan Da Xue, Zhong Yong, Lun Yu atau Meng Zi untuk masuk ke A3.

Warna Latar Belakang: #ffffff

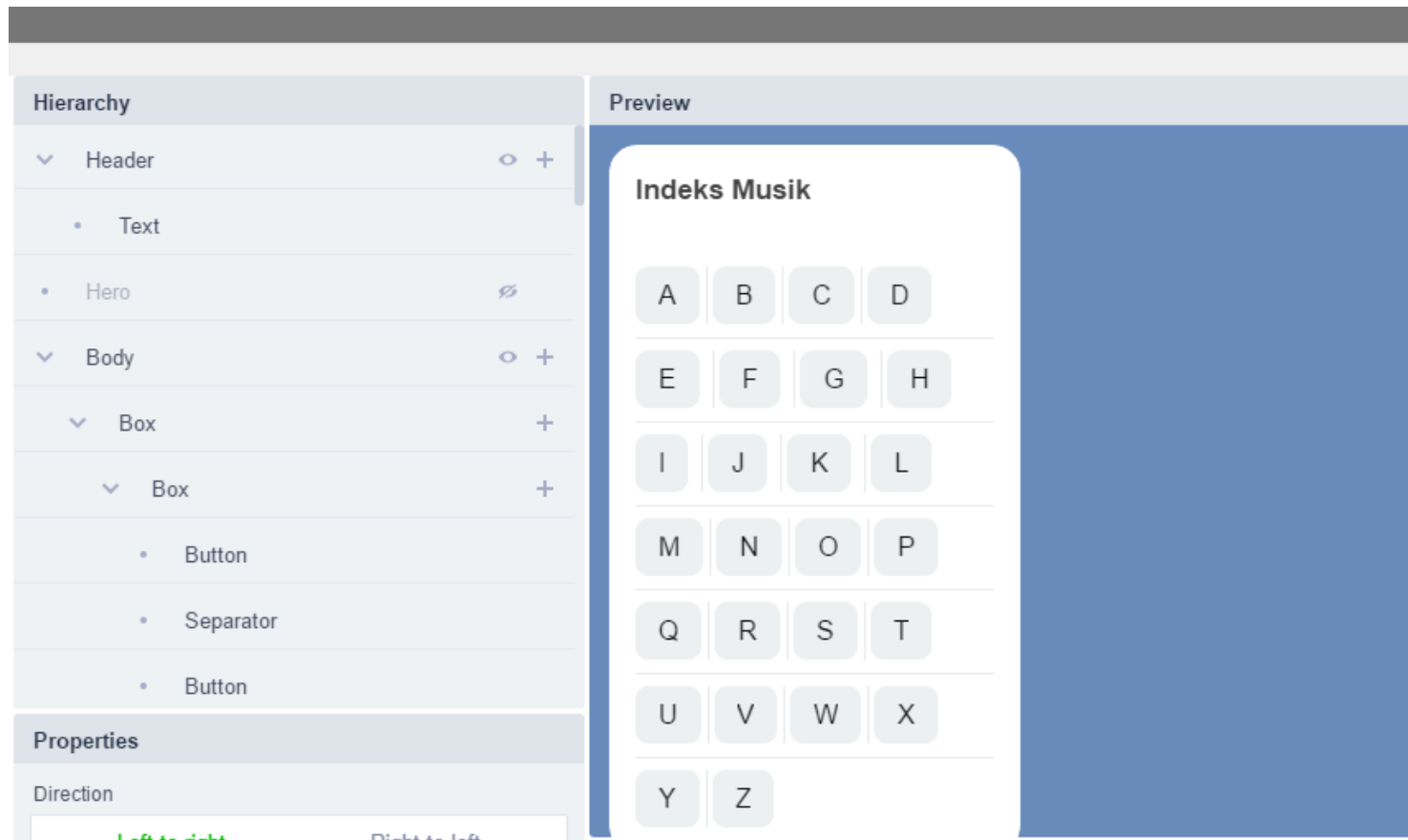


Perancangan Antarmuka

Rancangan antarmuka (A7 – Indeks Musik)

No: A7 (Indeks Musik)

1. Tekan tombol A, B.. Z untuk masuk ke A8.



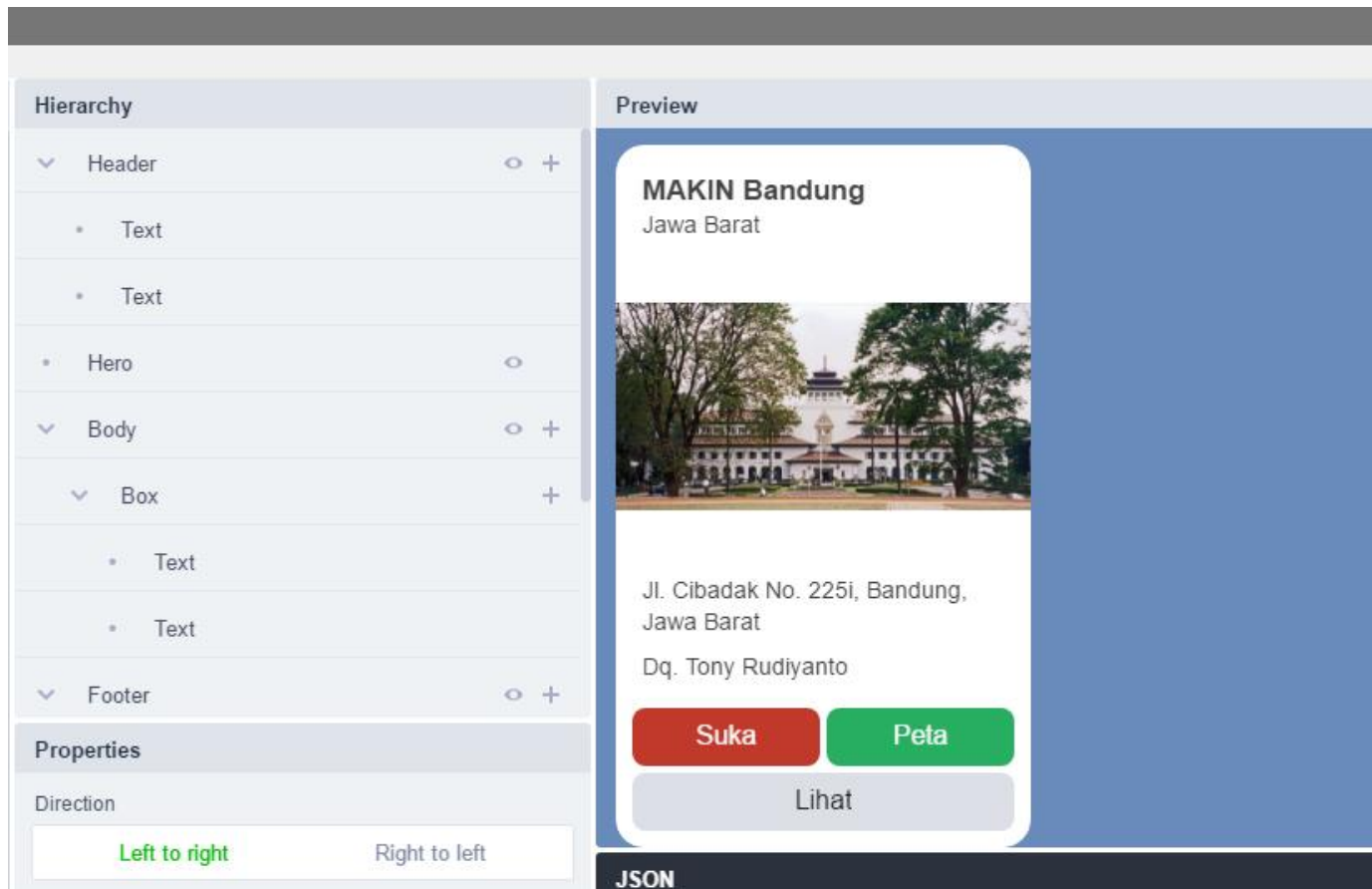
Warna Latar Belakang: #ffffff



Perancangan Antarmuka

Rancangan antarmuka (A14 – View Organisasi)

No: A14 (View Organisasi)



1. Tekan tombol Suka untuk masuk ke A14.
2. Tekan tombol Peta untuk untuk melihat alamat organisasi pada entitas luar (website).
3. Tekan tombol Lihat untuk melihat konten pada entitas luar (website).

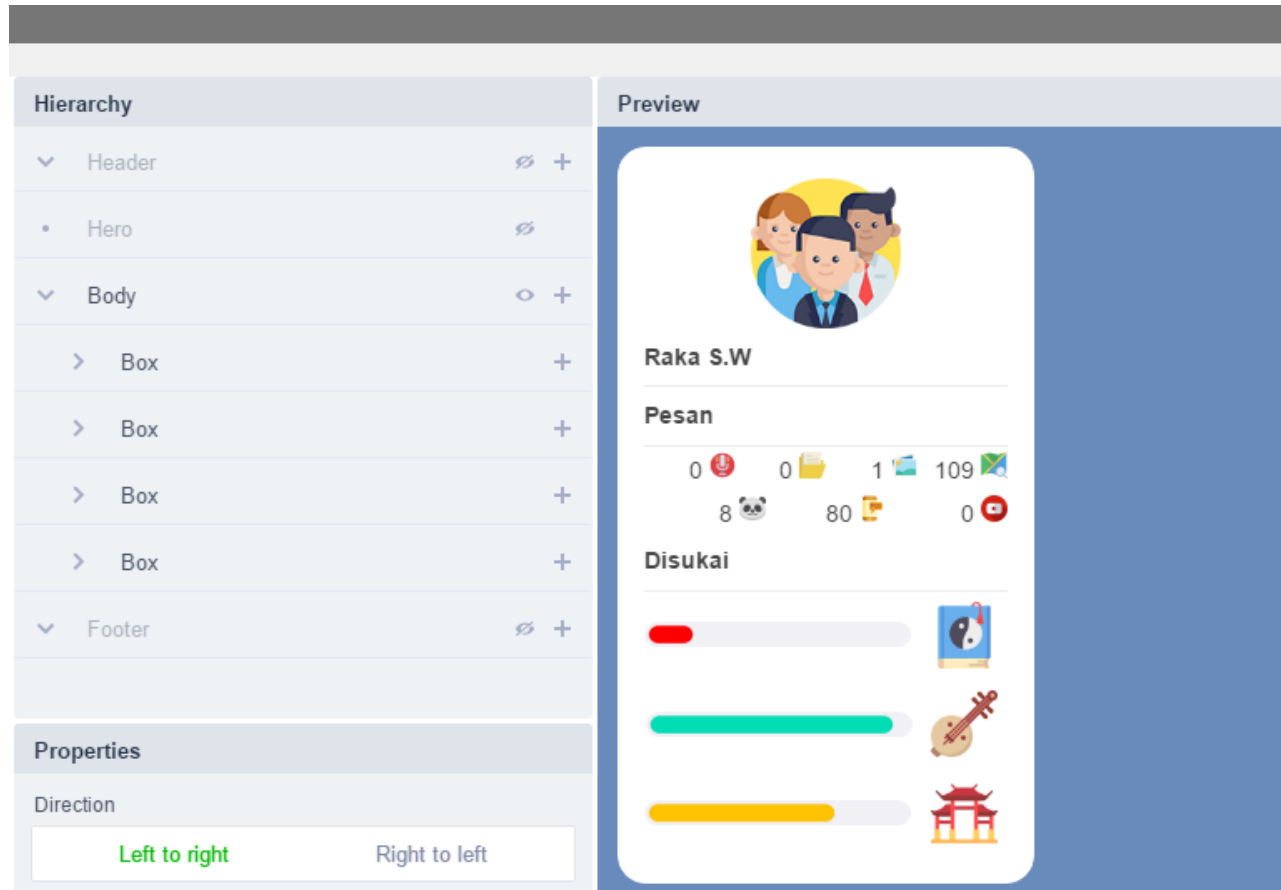
Warna Latar Belakang: #ffffff



Perancangan Antarmuka

Rancangan antarmuka (A18 – Profil User)

No: A18 (Profil User)

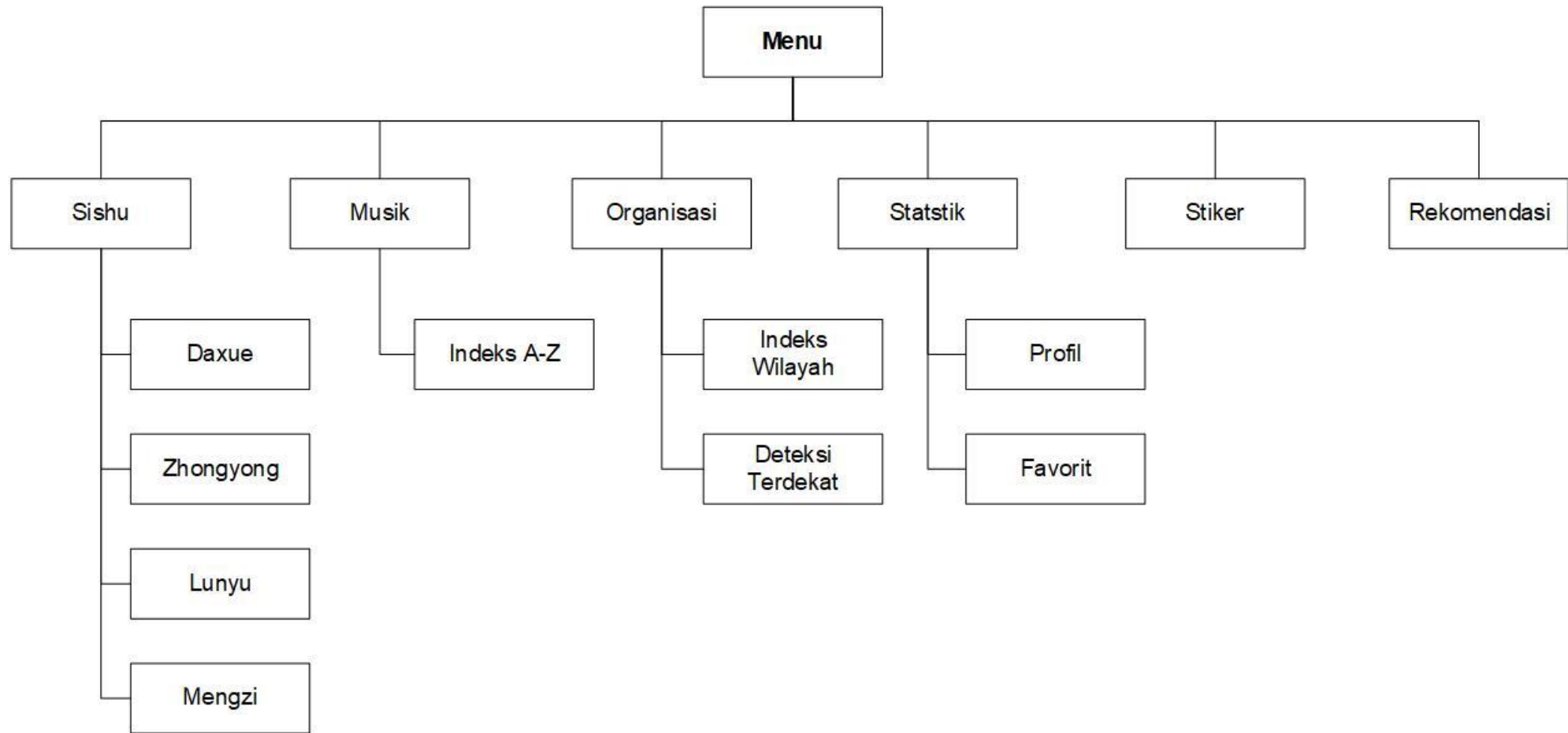


1. Tekan gambar Musik atau Bar Musik untuk masuk ke A7.
2. Tekan gambar Sishu atau Bar Sishu untuk masuk ke A2.
3. Tekan gambar Organisasi atau Bar Organisasi untuk masuk ke A11.

Warna Latar Belakang: #ffffff



Perancangan Struktur Menu





Kode	Antarmuka
A15	Card Organisasi
A16	Indeks User
A17	Layanan Favorit User
A18	Profil User
A19	Hasil Pencarian
Kode	Pesan
P1	Oops, aksi tidak diketahui
P2	Oops, menu tidak ditemukan
P3	Oops, konten tidak ditemukan

