

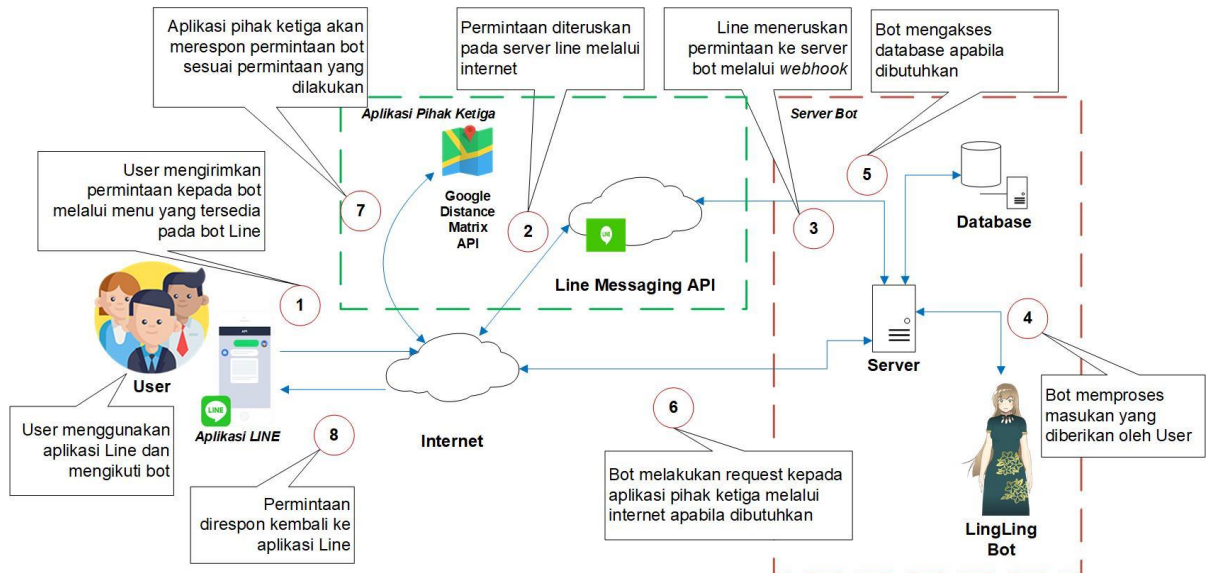
3.2 Analisis Solusi Yang Ditawarkan

Berdasarkan analisis terhadap masalah yang ada maka diperlukan aplikasi untuk menyediakan informasi mengenai Nyanyian Rohani Khonghucu, Kitab Sishu dan Lokasi tempat ibadah Khonghucu. Teknologi bot dipilih berdasarkan karakteristik kemudahan penggunaan dan pembangunan. Adapun karakter bot yang dibangun adalah:

1. User dapat berinteraksi dengan bot untuk melakukan pencarian informasi mengenai Nyanyian Rohani Khonghucu, Kitab Sishu dan Lokasi tempat ibadah Khonghucu.
2. User dapat berinteraksi dengan bot dengan cara melakukan pencarian melalui menu yang sudah disiapkan serta melakukan pencarian dengan mengetikkan pesan berupa pertanyaan kepada bot dalam bahasa Indonesia. Selain itu, user dapat mencari lokasi tempat ibadah Khonghucu dengan cara mengirimkan lokasi tempat user berada.
3. User dapat melihat statistik penggunaan aplikasi yang telah dilakukan yakni berupa statistik pengiriman pesan dan *bookmark* konten yang disukai.

3.3 Analisis Deskripsi Aplikasi

Aplikasi yang dibangun berupa bot dimana user dapat melakukan pencarian informasi dengan cara melalui menu yang sudah disediakan dan dengan cara mengetikkan pesan berupa pertanyaan. Adapun cara kerja aplikasi secara umum ditunjukkan pada Gambar III.1.



Gambar III.1. Analisis deskripsi aplikasi yang dibangun

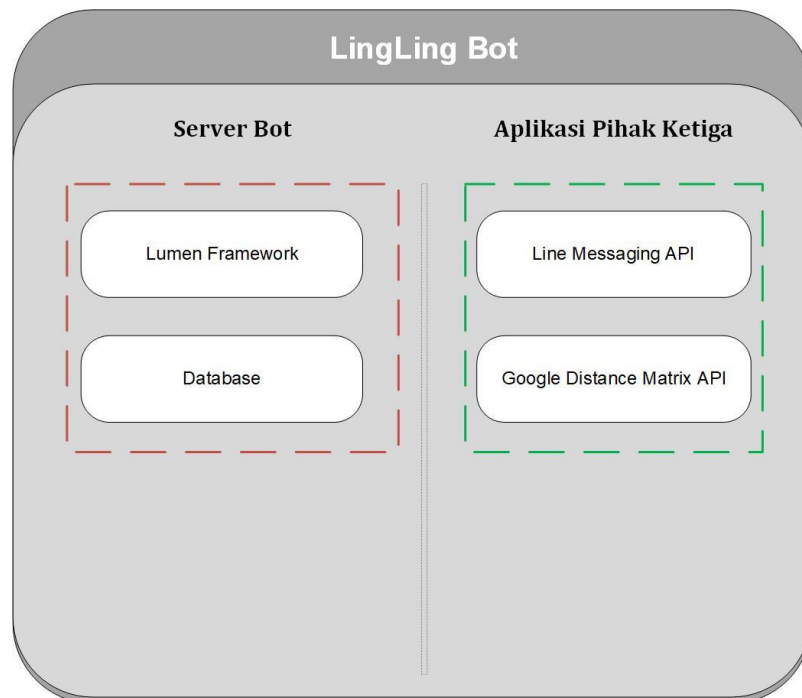
Secara umum aplikasi yang dibangun akan menggunakan *platform* Line sebagai mediasi antara pengguna dengan aplikasi. Alur penggunaan aplikasi adalah sebagai berikut:

1. User mengakses aplikasi Line yang tersedia pada perangkat Desktop atau Mobile. Untuk melakukan permintaan kepada Bot Line, user harus sudah mempunyai akun Line dan terlebih dahulu menambahkan Bot Line sebagai teman.
2. Permintaan user selanjutnya diteruskan ke server line melalui internet.
3. Line meneruskan permintaan yang dilakukan User ke Server Bot melalui Webhook yang sebelumnya sudah diatur.
4. Bot memproses masukan yang diberikan oleh User. Masukan yang diberikan oleh user dapat berupa masukan teks, masukan lokasi serta masukan menu yang dipilih.
5. Pada beberapa aksi yang memerlukan data dari database, bot akan mengakses data yang terdapat pada database untuk menyediakan kebutuhan data tersebut.
6. Pada beberapa aksi yang memerlukan data dari pihak luar, bot akan melakukan request kepada aplikasi pihak ketiga melalui internet apabila dibutuhkan.
7. Aplikasi pihak ketiga akan merespon permintaan bot sesuai permintaan yang dilakukan. Aplikasi pihak ketiga yang digunakan pada penelitian ini diantaranya: Google Distance Matrix API.

8. Hasil pemrosesan permintaan User diteruskan kembali kepada User melalui internet dan ditampilkan kembali pada Aplikasi Line milik User.

3.4 Analisis Arsitektur Aplikasi Yang Dibangun

Aplikasi terdiri dari dua bagian utama yakni Server Bot dan Aplikasi Pihak Ketiga (Google Distance Matrix API dan Line Messaging API). Gambar III.2 merupakan gambaran mengenai arsitektur aplikasi yang dibangun.



Gambar III.2 Analisis arsitektur aplikasi yang dibangun

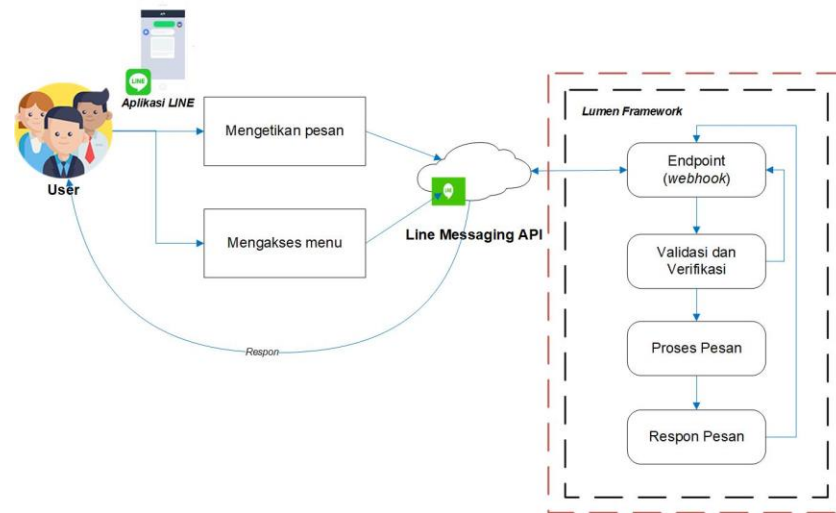
Penjelasan untuk setiap bagian aplikasi adalah sebagai berikut:

1. Server Bot

- a. Lumen Framework

Lumen framework digunakan untuk menangani *request* dan *response* yang dilakukan oleh user melalui Line Messaging API. User akan melakukan *request* berupa akses menu atau mengetikan masukan berupa pesan teks pada aplikasi Line. Lalu request tersebut diteruskan melalui Line Messaging API untuk kemudian diteruskan kembali ke server bot dan kemudian diterima oleh Lumen Framework melalui satu *endpoint* yang dinamakan dengan *webhook* yang sebelumnya sudah diatur pada Line

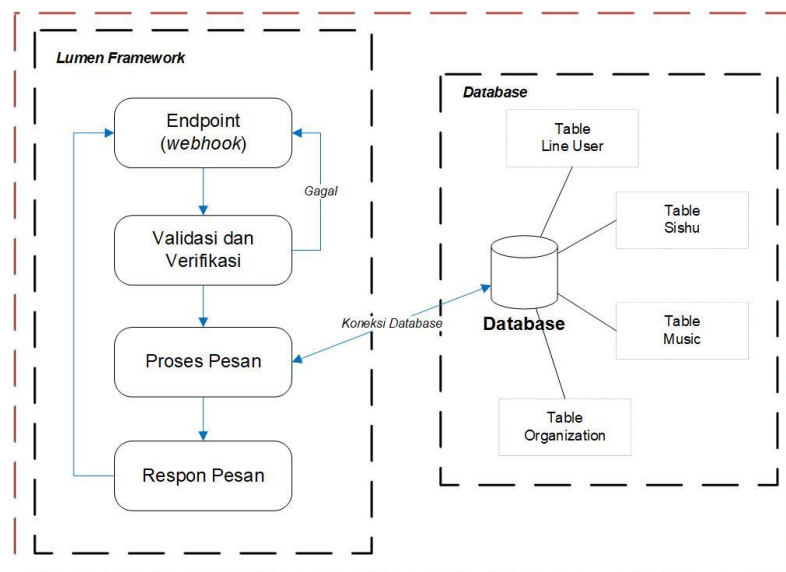
Developer Console. Gambar III.3 menunjukkan alur pada lumen framework.



Gambar III.3 Alur lumen framework

b. Database

Aplikasi membutuhkan akses terhadap data baik untuk melakukan pembacaan data ataupun penulisan data. Pada beberapa aksi yang perlu untuk melakukan pembacaan atau penulisan data, aplikasi akan melakukan akses ke database lalu kemudian data tersebut diolah pada Lumen Framework untuk kemudian diteruskan kembali kepada user melalui Line Messaging API. Gambar III.4 menunjukkan alur database.

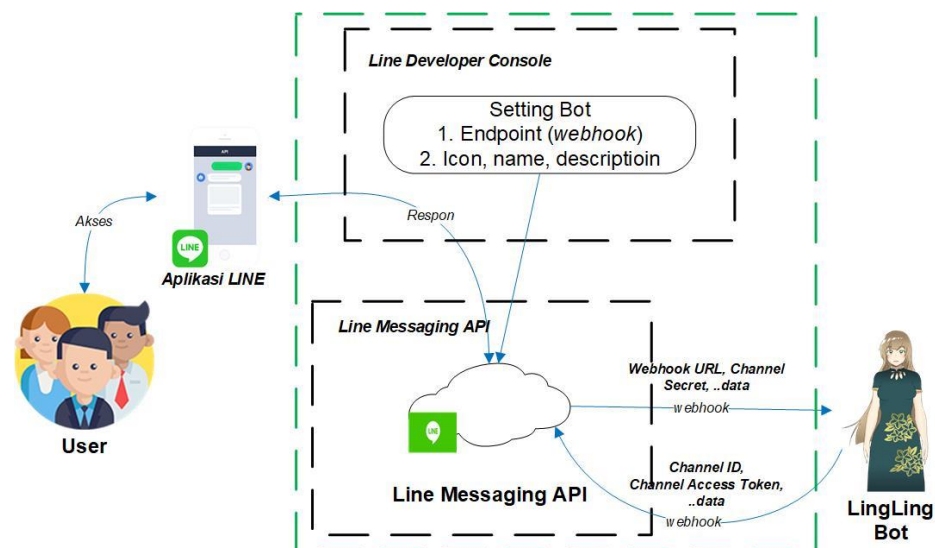


Gambar III.4 Alur database

2. Aplikasi Pihak Ketiga

a. Line Messaging API

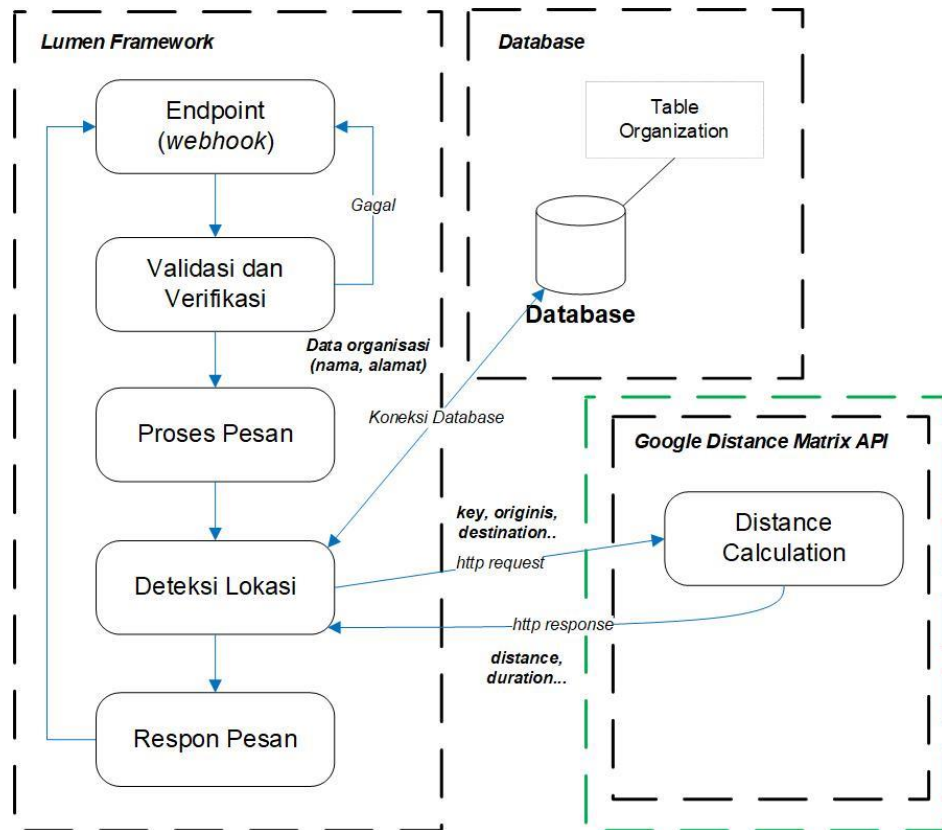
Line Messaging API digunakan untuk bisa terhubung dengan user melalui aplikasi Line. Untuk bisa terhubung melalui line messaging api, perlu dilakukan pengaturan *webhook* pada Line Developer Console dan pengaturan *channel secret* dan *channel token* pada aplikasi bot. Hal ini dilakukan untuk bisa mengatur setiap pesan yang dikirimkan oleh user akan diteruskan kepada bot yang sesuai. Gambar III.5 menunjukkan alur line messaging api.



Gambar III.5 Alur line messaging api

b. Google Distance Matrix API

Google Distance Matrix API digunakan untuk bisa mengetahui lokasi terdekat antara lokasi user dengan lokasi tempat ibadah Khonghucu. User akan mengirimkan lokasinya kepada bot, lokasi tersebut berupa lokasi saat ini berada ataupun lokasi yang dipilih oleh user. Bot kemudian akan memproses permintaan tersebut lalu melakukan *request* kepada Google Distance Matrix API untuk menghitung lokasi terdekat antara lokasi user dengan seluruh data lokasi tempat ibadah Khonghucu yang sebelumnya sudah ada di database. Gambar III.6 menunjukkan alur google distance matrix api.



Gambar III.6 Alur google distance matrix api

3.5 Analisis Teknologi Yang Digunakan

Dalam pembangunan aplikasi bot ini, penulis memanfaatkan teknologi Framework PHP Lumen, Database MariaDB (MySQL), aplikasi pihak ketiga Google Distance Matrix API serta Line Bot SDK.

1. Lumen Framework

Lumen framework digunakan untuk menangani *request* dan *response* yang dilakukan oleh user melalui Line Messaging API. Line messaging api akan mengirimkan request kepada aplikasi lumen melalui *route (webhook)* yang sebelumnya sudah diatur.

```

$router->group(['middleware' => ['throttle:600,1'], 'prefix' => 'api'],
function () use ($router) {
    $router->group(['prefix' => 'v0'], function () use ($router) {
        $router->group(['middleware' => ['auth.line'], 'prefix' => 'line'],
function () use ($router) {
            $router->post('', ['as' => 'line.index', 'uses' =>
'Line\LineBotController@index']);
        });
    });
});
  
```

```

    });
});

```

Selanjutnya, seluruh *request* yang masuk melalui alamat *webhook* akan ditangani oleh *class* `LineBotController` dengan *method* `index`.

```

<?php

namespace App\Http\Controllers\Line;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

use App\Http\Controllers\Controller;

use LINE\LINEBot;

use App\Services\Line\Event\Action\FollowService;
use App\Services\Line\Event\Action\UnfollowService;
use App\Services\Line\Event\Action\JoinService;
use App\Services\Line\Event\Action\LeaveService;
use App\Services\Line\Event\Action\PostbackService;

use App\Services\Line\Event\Message\RecieveTextService;
use App\Services\Line\Event\Message\RecieveImageService;
use App\Services\Line\Event\Message\RecieveVideoService;
use App\Services\Line\Event\Message\RecieveAudioService;
use App\Services\Line\Event\Message\RecieveFileService;
use App\Services\Line\Event\Message\RecieveLocationService;
use App\Services\Line\Event\Message\RecieveStickerService;

class LineBotController extends Controller {

    public function index(Request $request) {
        $bot = app('line-bot');
        $events = $bot->parseEventRequest($request->getContent(),
$_SERVER['HTTP_X_LINE_SIGNATURE']); // $request->header(HTTPHeader::LINE_SIGNATURE)

        foreach ($events as $event) {
            $token = $event->getReplyToken();
            $message = 'Maaf lingling belum mengerti..';

            switch ($event->getType()) {
                case 'follow': // $event instanceof \LINE\LINEBot\Event\FollowEvent
                    $service = new FollowService($bot);
                    $message = $service->execute($event);
                    break;
                case 'unfollow':

```

```

        $service = new UnfollowService($bot);
        $message = $service->execute($event);
    break;
    case 'join':
        $service = new JoinService($bot);
        $message = $service->execute($event);
    break;
    case 'leave':
        $service = new LeaveService($bot);
        $message = $service->execute($event);
    break;
    case 'postback':
        $service = new PostbackService($bot);
        $message = $service->execute($event);
    break;
    case 'message':
        switch (true) {
            case ($event instanceof
\LINE\LINEBot\Event\MessageEvent\TextMessage):
                $service = new RecieveTextService($bot);
                $result = $service->execute($event);
                break;
            case ($event instanceof
\LINE\LINEBot\Event\MessageEvent\ImageMessage):
                $service = new RecieveImageService($bot);
                $message = $service->execute($event);
                break;
            case ($event instanceof
\LINE\LINEBot\Event\MessageEvent\VideoMessage):
                $service = new RecieveVideoService($bot);
                $message = $service->execute($event);
                break;
            case ($event instanceof
\LINE\LINEBot\Event\MessageEvent\AudioMessage):
                $service = new RecieveAudioService($bot);
                $message = $service->execute($event);
                break;
            case ($event instanceof
\LINE\LINEBot\Event\MessageEvent\FileMessage):
                $service = new RecieveFileService($bot);
                $message = $service->execute($event);
                break;
            case ($event instanceof
\LINE\LINEBot\Event\MessageEvent\LocationMessage):
                $service = new RecieveLocationService($bot);
                $message = $service->execute($event);
                break;
            case ($event instanceof
\LINE\LINEBot\Event\MessageEvent\StickerMessage):

```



```

        $service = new RecieveStickerService($bot);
        $result = $service->execute($event);
        break;
        default:
            Log::info('Unknown message');
    }
    break;
    default:
        // $body = $event->getEventBody();
    }

    if (!$event instanceof \LINE\LINEBot\Event\MessageEvent\StickerMessage
||
        !$event instanceof \LINE\LINEBot\Event\MessageEvent\ImageMessage
||
        !$event instanceof \LINE\LINEBot\Event\MessageEvent\TextMessage) {
        $result = $bot->replyText($token, $message);
    }
    return $result->getHTTPStatus();
}
}
}

```

Seluruh *request* akan ditangani oleh *method* `index()`. Method ini kemudian akan memproses setiap *request* sesuai dengan jenis request yang dilakukan.

2. MySQL (MariaDB) Fulltext Search

MySQL menyediakan fitur fulltext untuk dapat melakukan pencarian data bertipe TEXT, CHAR dan VARCHAR. Untuk melakukan pencarian dengan menggunakan fitur fulltext search, perlu ditambahkan index fulltext pada kolom yang akan dicari. Pada penelitian ini, data yang akan dicari diantaranya adalah data pada kitab sishu, musik dan organisasi (tempat ibadah). Kolom yang diindex merupakan kolom yang memiliki data yang akan dicari. Tabel III.1 merupakan contoh data nyanyian pujian khonghucu.

Tabel III.1 Contoh data nyanyian rohani khonghucu

ID	Title	Lyric
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<br
 Yang Dikau Ucapkan
 <br
 Bila Gunung Thai San Runtuhlah<br
 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 Penyedar Hidup

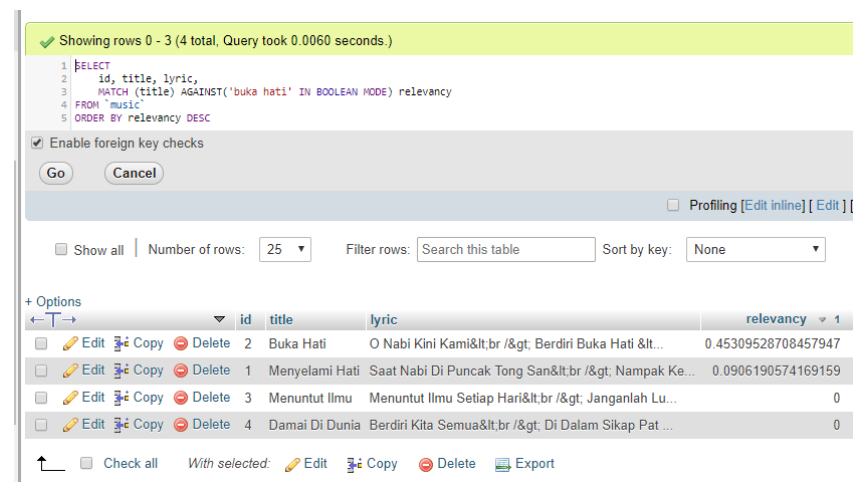
 Berdoalah

		Bersama Dengan Hati Yang Suci Kepada Thian Yang Maha Esa Agar Damai Di Dunia
--	--	--

Dalam kasus pencarian musik, maka kolom yang akan diindex yakni kolom *title*, *lyric* pada table *music* serta kolom *name* pada table *hero*. Berikut merupakan *query* untuk menambahkan index fulltext pada kolom *title* dan *lyric* yang berada pada tabel *music*.

```
ALTER TABLE music ADD FULLTEXT keyword(title, lyric);
ALTER TABLE hero ADD FULLTEXT keyword(name);
```

Untuk melakukan pencarian menggunakan fulltext search, query yang diketikan berbeda dengan pencarian biasa menggunakan perintah operator logika ataupun menggunakan query LIKE. Gambar III.7 menunjukkan contoh query untuk melakukan pencarian menggunakan fulltextsearch.



Gambar III.7 Contoh query fulltext search

Pencarian dilakukan dengan cara mencari dokumen yang paling relevan dengan kata kunci pencarian yang dimasukkan. Semakin relevan kata kunci yang dicari maka nilai relevansi akan semakin besar. Nilai relevansi merupakan hasil perhitungan pembobotan dokumen yang sudah secara otomatis dilakukan oleh database MySQL atau MariaDB. Tabel III.2 merupakan analisis kata kunci yang dilakukan dalam melakukan pencarian dengan menggunakan kata kunci '*buka hati*' hanya pada kolom *title*.

Tabel III.2 Analisis kata kunci pencarian

ID	Title	Lyric
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<br
 Yang Dikau Ucapkan
 <br
 Bila Gunung Thai San Runtuhlah<br
 Betapa Pohon Dahan Semua
 Bila Sang Budiman Gugurlah
 O Betapa

 Sayu Sedih M
rsquo;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
 &nbsp;
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

Berdasarkan kata kunci yang digunakan untuk melakukan pencarian dan berdasarkan data yang tersedia diketahui terdapat beberapa kata kunci yang cocok dengan data yang ada pada kolom *title*. Tabel III.3 merupakan analisis kemunculan kata kunci pada kolom *title*.

Tabel III.3 Analisis nilai TF (term frequency)

ID	Kata	TF
1	Buka	0
	Hati	1
2	Buka	1
	Hati	1
3	Buka	0
	Hati	0
4	Buka	0
	Hati	0

Meskipun terdapat kata yang sama pada kolom *lyric*, namun karena pencarian tidak mengikutsertakan kolom *lyric* maka kolom tersebut akan diabaikan (tidak ikut dihitung bobotnya). Tabel III.4 merupakan perhitungan nilai relevansi pembobotan kolom *title*.

Tabel III.4 Perhitungan nilai relevansi TF-IDF

ID	Kata	Relevancy Title
1	Buka	$(0) \times \log_{10} \left(\frac{4}{1} \right) \times \log_{10} \left(\frac{4}{1} \right) = 0$
	Hati	$(1) \times \log_{10} \left(\frac{4}{2} \right) \times \log_{10} \left(\frac{4}{2} \right) = 0,090619$
2	Buka	$(1) \times \log_{10} \left(\frac{4}{1} \right) \times \log_{10} \left(\frac{4}{1} \right) = 0,362476$
	Hati	$(1) \times \log_{10} \left(\frac{4}{2} \right) \times \log_{10} \left(\frac{4}{2} \right) = 0,090619$
3	Buka	$(0) \times \log_{10} \left(\frac{4}{1} \right) \times \log_{10} \left(\frac{4}{1} \right) = 0$
	Hati	$(0) \times \log_{10} \left(\frac{4}{2} \right) \times \log_{10} \left(\frac{4}{2} \right) = 0$
4	Buka	$(0) \times \log_{10} \left(\frac{4}{1} \right) \times \log_{10} \left(\frac{4}{1} \right) = 0$
	Hati	$(0) \times \log_{10} \left(\frac{4}{2} \right) \times \log_{10} \left(\frac{4}{2} \right) = 0$

Untuk setiap kata yang dicari pada kata kunci maka setiap kata tersebut akan dihitung dan diakumulasikan sehingga kemudian menghasilkan total nilai relevansi. Karena sebelumnya kata yang dicari adalah kata *buka* dan *hati* maka pembobotan dilakukan dengan cara menjumlahkan nilai relevansi kata *buka* dan kata *hati*. Tabel III.5 merupakan hasil perhitungan relevansi pencarian kata *buka hati*.

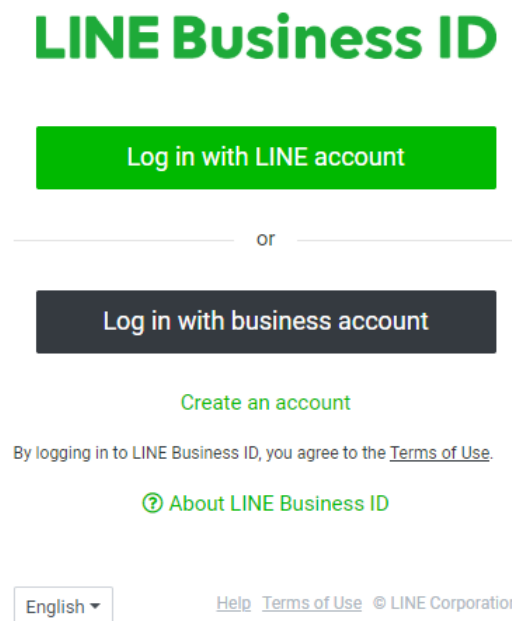
Tabel III.5 Hasil perhitungan nilai relevansi

ID	Relevancy <i>Buka</i>	Relevancy <i>Hati</i>	Relevancy Total
1	0	0,090619	0,090619
2	0,362476	0,090619	0,453095
3	0	0	0
4	0	0	0

Dari ide dasar inilah prinsip pencarian akan dilakukan. Pencarian akan dilakukan dengan cara menggabungkan kolom yang dianggap penting sehingga menambah nilai relevansi pada saat melakukan pencarian. Penggabungan kolom tidak hanya dilakukan pada tabel yang sama melainkan pada tabel yang berbeda dengan menggunakan *query JOIN* pada tabel tersebut.

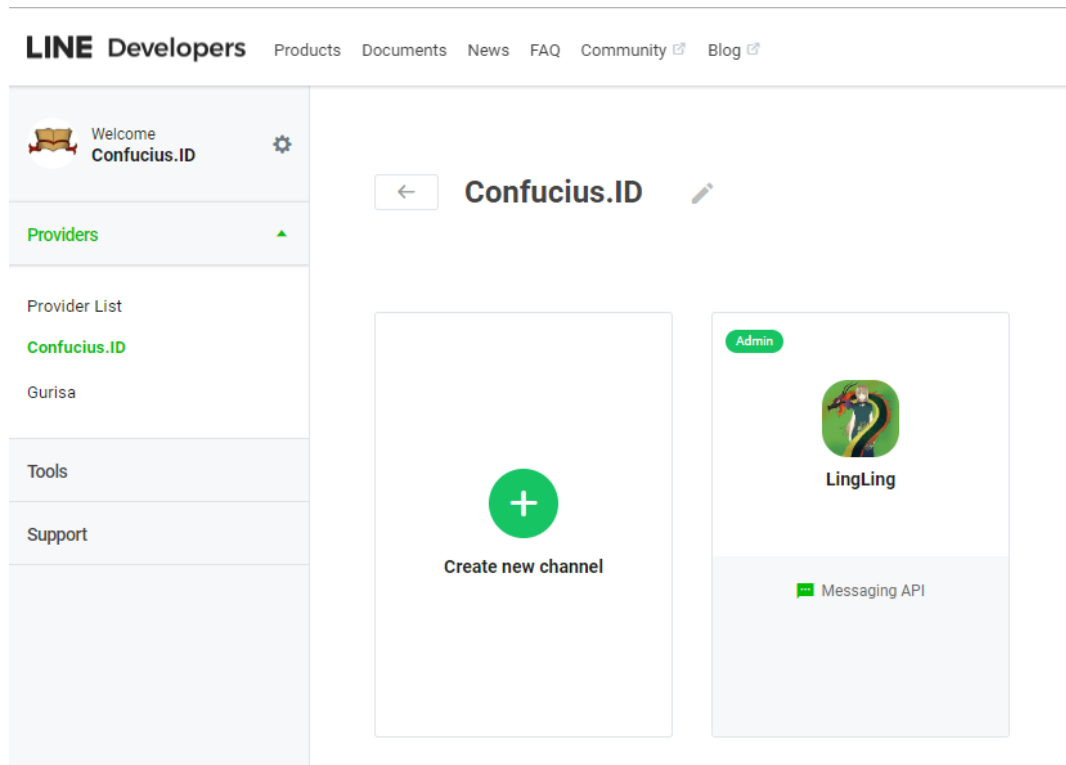
3. Line Messaging API

Untuk dapat memanfaatkan fitur line messaging api, developer harus membuat channel messaging api terlebih dahulu. Channel ini merupakan representasi bot yang akan dibuat pada aplikasi Line. Untuk membuat channel ini, developer harus mempunyai akun Line Developer terlebih dahulu dengan cara mendaftar pada situs line developer. Gambar III.8 merupakan halaman masuk line developer.



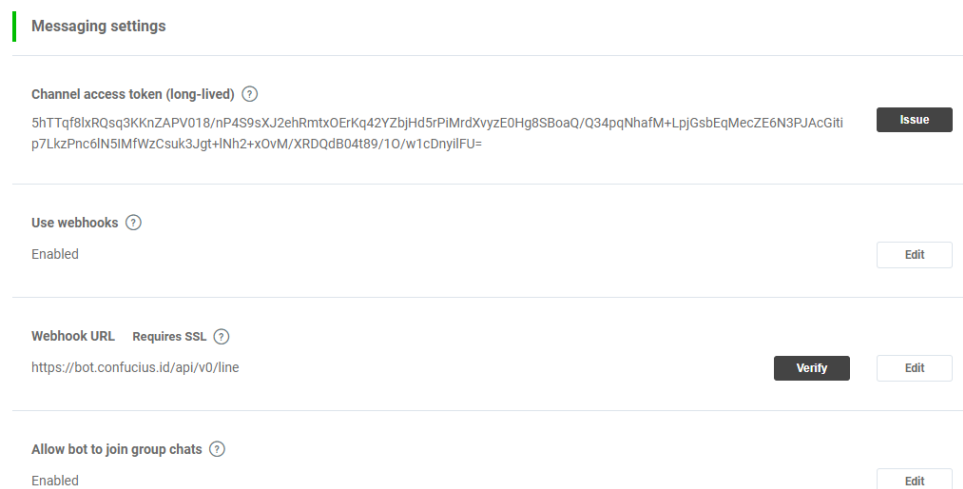
Gambar III.8 Halaman masuk line developer

Setelah masuk, developer dapat membuat channel messaging api untuk membuat bot. Gambar III.9 merupakan channel line messaging api.



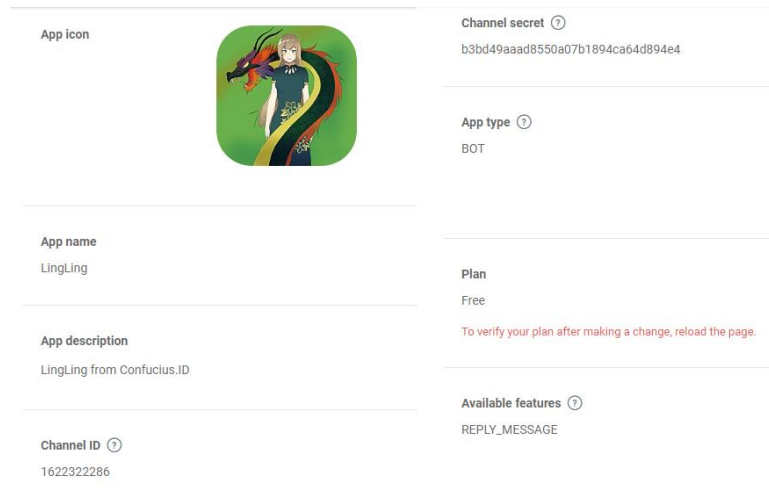
Gambar III.9 Channel line messaging api

Untuk dapat terhubung ke server bot, developer harus mengatur *webhook* (*endpoint*) pada line developer. Setiap *request* yang dikirimkan melalui bot akan dikirimkan ke alamat tersebut. Gambar III.10 menunjukkan pengaturan webhook pada situs line developer.



Gambar III.10 Pengaturan webhook pada line developer

Developer harus mengatur token dan channel pada aplikasi bot untuk dapat mengembalikan respon berupa pesan kepada user. Gambar III.11 menunjukkan pengaturan umum yang dapat diatur oleh developer.

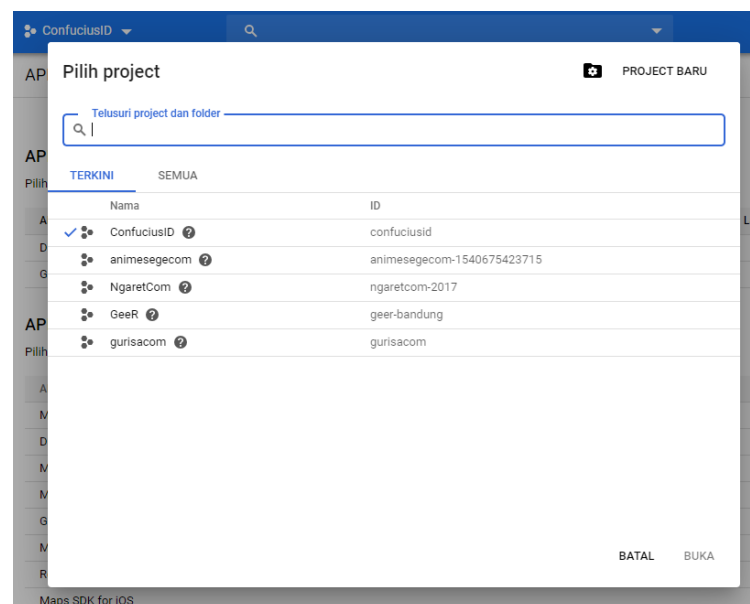


The screenshot displays the 'General' settings for a Line bot. The interface is divided into two columns. The left column contains fields for 'App icon' (a character icon), 'App name' (LingLing), 'App description' (LingLing from Confucius.ID), and 'Channel ID' (1622322286). The right column contains 'Channel secret' (b3bd49aad8550a07b1894ca64d894e4), 'App type' (BOT), 'Plan' (Free), and 'Available features' (REPLY_MESSAGE). A red message states: 'To verify your plan after making a change, reload the page.'

Gambar III.11 Pengaturan umum bot pada line developer

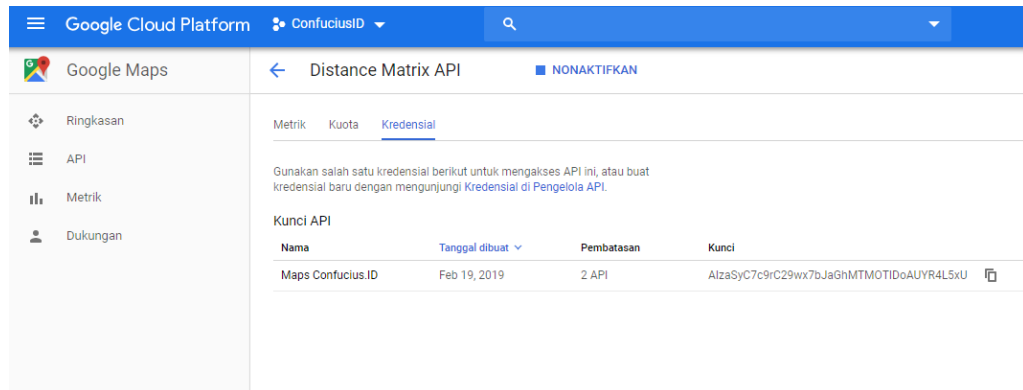
4. Google Distance Matrix API

Untuk dapat menggunakan layanan google distance matrix, developer harus memiliki kunci (*key*) yang digunakan saat melakukan *request* melalui *endpoint* google distance matrix api. Untuk memiliki *key* tersebut, developer harus membuat proyek baru pada situs Google Developer Console. Gambar III.12 menunjukkan proyek google console.



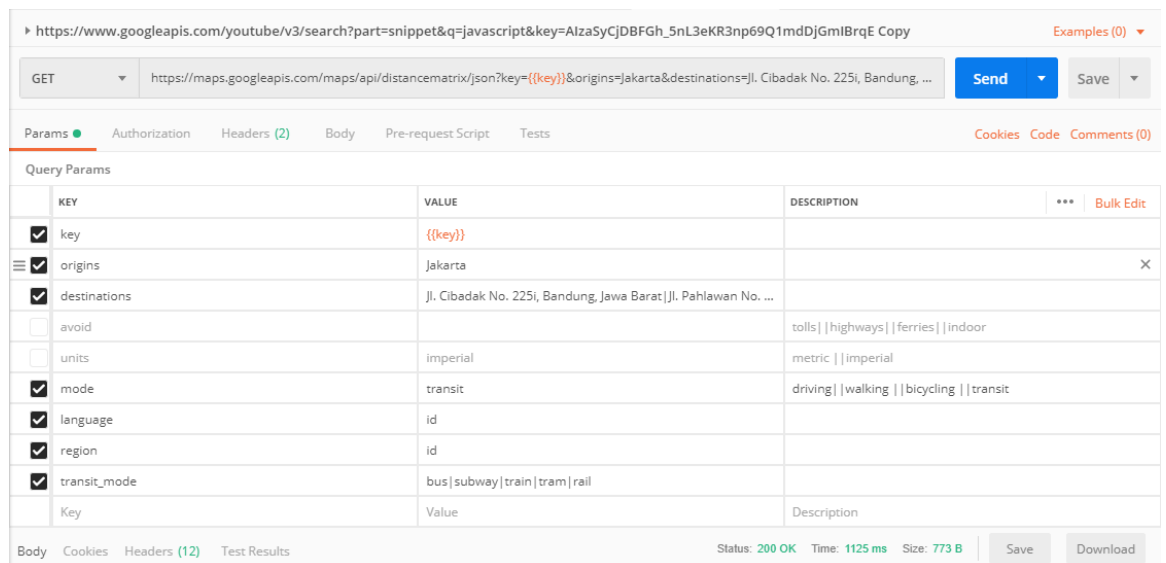
Gambar III.12 Membuat proyek google console

Setelah mempunyai proyek pada google developer console, developer harus mengaktifkan layanan google distance matrix untuk dapat menggunakan layanan tersebut. Developer juga harus membuat *key* untuk dapat menggunakan layanan google distance matrix. Gambar III.13 menunjukkan layanan google distance matrix api.



Gambar III.13 Google distance matrix api

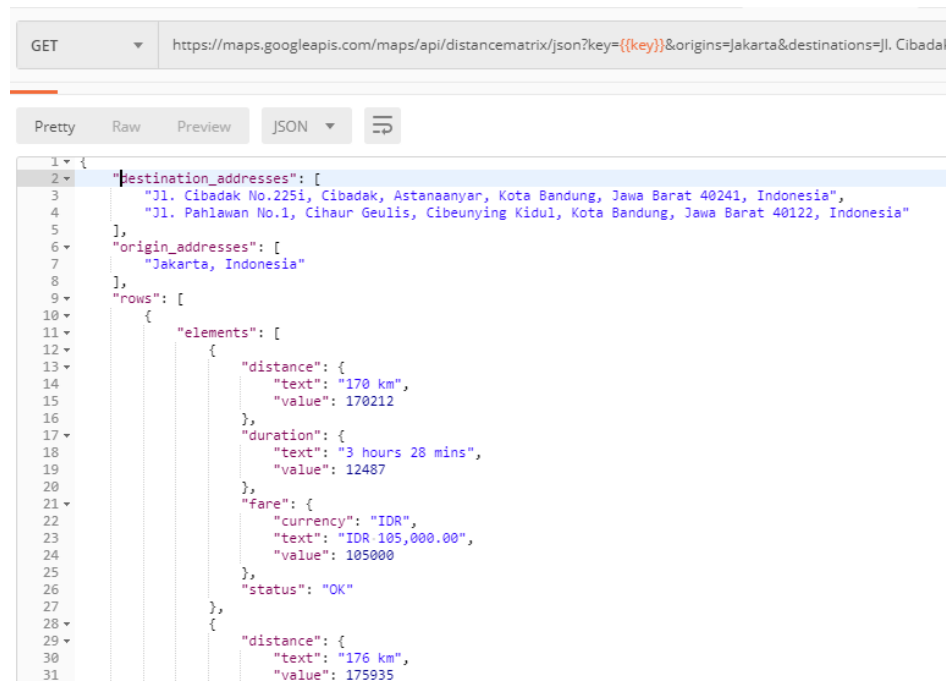
Setelah mendapatkan *key*, google distance matrix api dapat diakses melalui endpoint yang sudah ditentukan dengan memasukan key dan parameter yang dibutuhkan lainnya. Gambar III.14 menunjukkan contoh request layanan google distance matrix api



Gambar III.14 Contoh request google distance matrix api

Google distance matrix api kemudian akan mengembalikan *request* berupa *response* dalam format json. Berikut merupakan contoh *response* yang

dikembalikan melalui google distance matrix api. Gambar III.15 merupakan contoh respon yang diberikan oleh google distance matrix api.



Gambar III.15 Contoh respon google distance matrix api

3.6 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional yang dilakukan meliputi analisis perangkat keras, analisis perangkat lunak dan analisis perangkat pikir. Analisis yang dilakukan yakni sebagai berikut:

3.6.1 Analisis Perangkat Keras

Perangkat keras yang diperlukan untuk mendukung kinerja aplikasi tertulis pada Tabel III.6 sebagai berikut:

Tabel III.6. Analisis perangkat keras

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

3.6.2 Analisis Perangkat Lunak

Berikut adalah 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 Dependency Manager (Composer 1.0+)

3.6.3 Analisis Perangkat Pikir

Analisis perangkat pikir dimaksudkan untuk mengetahui siapa saja pengguna yang akan terlibat dalam menggunakan aplikasi Bot Line ini. Tabel III.7 menunjukkan karakteristik pengguna aplikasi:

Tabel III.7. Karakteristik pengguna aplikasi

Stakeholder	:	Pengguna Line
Umur	:	Lebih dari 10 Tahun
Tingkat Keterampilan	:	a. Pernah melakukan chatting menggunakan aplikasi LINE
Pengalaman Menggunakan Smartphone	:	Cukup

3.7 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam membangun sistem atau aplikasi agar dapat berjalan dengan baik serta sesuai dengan kebutuhan. Analisis kebutuhan fungsional ini meliputi Analisis