**BAB I**

**PENDAHULUAN**

# Latar Belakang Masalah

**Perguruan Tinggi di kota Bandung pada tahun 2017 berjumlah 107 untuk Swasta diantaranya 20 Universitas, 2 Institut, 50 Sekolah Tinggi, 25 Akademi, dan 10 Politeknik [1]. Sedangkan untuk Perguruan Tinggi Negeri di Kota Bandung berjumlah 11 [2]. Karena jumlah Perguruan Tinggi yang banyak, untuk mendapatkan Informasi Perguruan Tinggi yang diminatinya saat ini siswa SMA/K dalam mencari informasi Perguruan Tinggi menggunakan cara mengunjungi setiap website Perguruan Tinggi satu persatu yang diinginkan untuk mendapatkan informasi mengenai Perguruan Tinggi. Tentunya penyampaian atau pencarian informasi tersebut tidak efektif karena jika kita akan mencari infomasi dari kampus lain harus mengunjungi website satu persatu , hal ini sangat tidak efektif dan efisien.**

**Berdasarkan data kuisoner terhadap 68 responden siswa SMA/K yang akan melanjutkan sekolahnya ke Perguruan Tinggi, terdapat 86% siswa yang masih mengalami kebingungan dalam menentukan Perguruan Tinggi yang diminatinya. Lalu berdasarkan data kuisoner yang sama, faktor peminat siswa SMA/K dalam mencari Perguruan Tinggi diantaranya berdasarkan Akreditasi, Lokasi, Biaya perkuliahan serta Jurusan yang diminatinya . Oleh sebab itu dibutuhkan sebuah media yang lebih efektif dan interaktif untuk mencari suatu informasi Perguruan Tinggi di yang ada di kota Bandung yang dapat menampilkan informasi seperti Akreditasi, Biaya perkuliahan, Jurusan, serta Lokasi kampus yang berada di kota Bandung.**

**Karena itu diperlukan satu alternatif penyampaian informasi yang terintegrasi di satu tempat yang bisa memberikan informasi, sesuai dengan informasi Perguruan Tinggi di kota Bandung secara lengkap dan interaktif. Dengan cara menggunakan chatbot memanfaatkan media sosial LINE atau LINE chatbot dengan menggunakan metode Webhook.**

**LINE merupakan aplikasi pesan instan yang banyak digunakan oleh masyarakat di Indonesia. Berdasarkan data dari LINE, penggunanya di Indonesia berjumlah 90 juta pengguna, 80% diantaranya didominasi pengguna anak muda[3]. Salah satu produk LINE yang saat ini sedang diminati pelanggan yakni LINE Chatbot. Chatbot merupakan program komputer yang dirancang untuk mensimulasikan sebuah percakapan atau komunikasi yang interaktif kepada user (manusia) melalui bentuk teks, audio, maupun video[4]. Pemanfaatan LINE Chatbot merupakan cara yang efektif dalam mencari informasi secara interaktif .**

Pada penelitian sebelumnya yaitu menerapkan dan membuat bot yang dapat membantu pekerjaan dan aktivitas harian manusia, menemukan informasi serta mengumpulkan informasi, dengan menggunakan platform Telegram membuat sebuah bot yang diberi nama Bot Telegram. Dengan Bot Telegram ini bisa membantu tata usaha/dosen dalam menyampaikan informasi mengenai perkuliahan kepada mahasiswa secara tepat dan cepat. Hal ini juga bisa membantu dosen dalam menyampaikan materi perkuliahan [5]. Berbeda dengan penelitian penulis lakukan mengenai pembuatan bot LINE, yang akan menyampaikan informasi mengenai Perguruan Tinggi yang ada di Kota Bandung. Nantinya bot LINE ini akan akan berguna bagi siswa SMA/K yang ingin mencari informasi Perguruan Tinggi yang diminatinya.

Berdasarkan permasalah yang ada sebagai solusi dalam penelitian ini diusulkan membuat aplikasi yang dapat membantu siswa SMA/K agar tidak kebingungan dalam mencari informasi Perguruan Tinggi yang ada di kota Bandung dengan menggunakan metode webhook yang dapat terintegrasi dengan LINE Chatbot . Aplikasi ini dibangun agar siswa dapat mencari tahu tentang informasi kampus yang diinginkannya dengan cara tanya jawab kepada chatbot. Siswa memasukan pertanyaan seputar kampus berupa teks, kemudian chatbot akan memberikan respon berupa jawaban atas pertanyaan pengguna[6]. Maka sebab itu, penulis membuat penelitian mengenai “PEMBANGUNAN APLIKASI VIRTUAL ASISTEN UNTUK MEDIA INFORMASI PERGURUAN TINGGI MENGGUNAKAN LINE CHATBOT DI KOTA BANDUNG”.

# Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan, penulis mengidentifikasi masalah yang timbul yakni :

1. Siswa sekolah SMA/K mengalami kebingungan dalam mencari informasi seputar Perguruan Tinggi diminatinya yang ada di Kota Bandung.
2. Belum adanya aplikasi media informasi Perguruaan Tinggi pada media sosial LINE.
3. Bagaimana menghasilkan media informasi Perguruan Tinggi di Kota Bandung dengan LINE Bot

# Maksud dan Tujuan

Berdasarkan uraian permasalahan, maka maksud dari penelitian ini adalah menerapkan LINE Chatbot agar dapat memberikan informasi Perguruan Tinggi yang ada di kota Bandung . Sedangkan tujuan dilakukannya penelitian ini adalah sebagai berikut :

1. Membangun aplikasi yang dapat membantu siswa sekolah SMA/K dalam mencari informasi seputar Perguruan Tinggi yang diminatinya yang ada di Kota Bandung.
2. Membangun aplikasi media informasi Perguruan Tinggi pada media sosial LINE.
3. Membangun aplikasi Line Bot untuk menyampaikan informasi Perguruan Tinggi di Kota Bandung.

# Batasan Masalah

Adapun batasan – batasan yang terdapat pada penelitian ini adalah sebagai berikut :

1. Informasi data Perguruan Tinggi masih di sekitar Kota Bandung yang berstatus Negeri dan Swasta. Seperti UPI, ITB, UNPAD, UNISBA, UNIKOM, ITHB, ITENAS dll.
2. Pengguna adalah siswa SMA/K yang akan melanjutkan ke Perguruan Tinggi di kota Bandung.
3. Bahasa pemograman yang digunakan dalam pembangunan aplikasi adalah PHP, CSS, dan Javascript.
4. Konten pada aplikasi yang akan dibuat berbahasa Indonesia.
5. Metode pencocokan pola yang digunakan menggunakan metode Fulltext Search Boolean Mode tanpa menggunakan operator.
6. Pencocokan masukan berupa teks pada bot yang dibangun tidak menggunakan konsep learning melainkan menggunakan konsep searching karena terbatas dengan metode pencocokan yang digunakan.
7. Format data yang digunakan saat mengirim dan menerima pesan antara bot dengan line adalah JSON.
8. Chatbot ini dijalankan pada aplikasi media sosial LINE
9. Keluaran yang dihasilkan berupa teks dan gambar
10. Infromasi yang dimiliki oleh chatbot sudah disimpan didalam di database

# Metodologi Penelitian

Metodologi yang digunakan dalam penulisan proposal ini adalah sebagai berikut :

## Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut :

1. Studi Literatur

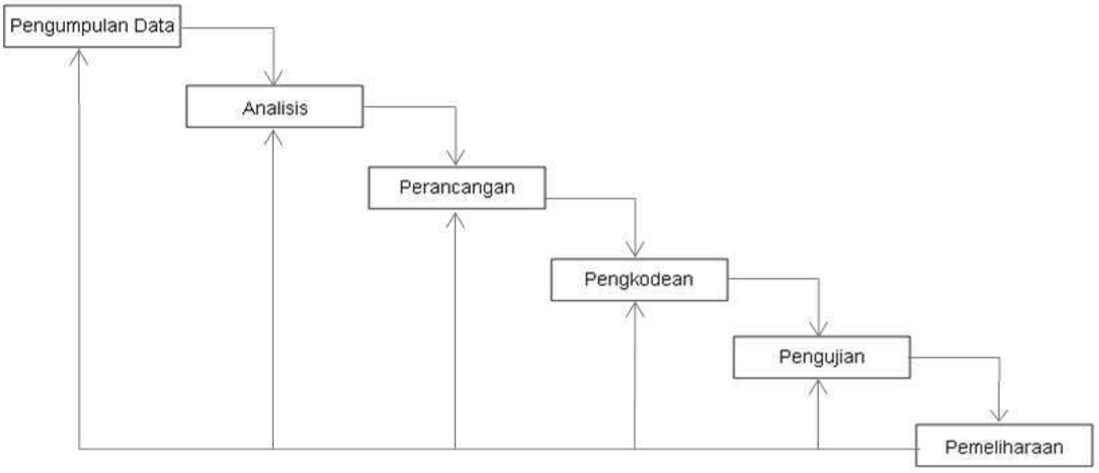
Pengumupulan data dengan cara mengumpulkan literatur – literatur yang berkaitan dengan judul penelitian

1. Kuisoner.

Teknik pengumpulan data dengan cara membuat sebuah pertanyaan untuk dijawab kepada siswa yang terkait dengan judul penelitian

## Metode Pembangunan Perangkat Lunak

Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah metode pembangunan perangkat lunak waterfall. Metode waterfall digunakan dalam penelitian ini dikarenakan kebutuhan perangkat lunak sudah diketahui di awal penelitian dan dijadikan sebagai batasan penelitian yang terdapat pada bagian batasan masalah. Gambar 2 merupakan tahapan pengembangan perangkat lunak waterfall :



Gambar I.1 Metode Waterfall

1. Pengumpulan data

Kegiatan yang dilakukan adalah mengumpulkan kebutuhan secara lengkap untuk selanjutnya menuju tahap analisis.

1. Analisis (Analysis)

Setelah data terkumpul, dilakukan analisis yang berkaitan dengan proses dan data yang diperlukan oleh sistem serta keterkaitannya.

1. Perancangan

Pada tahap ini diberikan gambaran umum yang jelas dan rancang bangun yang lengkap tentang sistem yang akan dikembangkan kepada pihak-pihak yang terlibat dalam pengembangan sistem

1. Pengkodean (Coding)

Setelah tahap perancangan sistem, selanjutnya dilakukan penterjemahan kode program yang dimengerti oleh mesin. Pada tahap ini dilakukan pembuatan komponen-komponen sistem yang meliputi implementasi modul-modul program, antarmuka dan basis data.

1. Pengujian (Testing)

Melakukan pengujian secara formal terhadap kode program yang telah dibuat untuk memastikan semua kebutuhan atau persyaratan telah terpenuhi.

1. Perbaikan (Maintenance)

Setelah menerapkan langkah-langkah tersebut, dilakukan evaluasi terhadap system yang baru untuk melihat apakah sistem telah memenuhi tujuan yang ingin dicapai. Perubahan bisa terjadi karena adanya error, perubahan lingkungan sistem,dsb

# Sistematika Penulisan

Penyusunan proposal ini dibagi ke dalam beberapa bab secara sistematis sesuai dengan pokok-pokok permasalahan yang dibahas. Adapun sistematika penulisan secara umum adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang masalah, perumusan masalah, maksud dan tujuan, batasan masalah, metodologi penelitian, serta sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini akan menjelaskan mengenai objek dari penelitain, dan teori – teori pendukung yang berhubungan dengan masalah yang dibahas.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tentang deskripsi sistem, analisis kebutuhan yang diperlukan dalam pengembangan sistem, serta perancangan untuk pengembangan sistem yang akan dibuat.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini membahas tentang implementasi dari hasil tahapan analisis dan perancangan aplikasi yang dibangun.Serta berisi uji coba dan hasil pengujian sistem.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang diperoleh dari hasil pengujian sistem dan dan berisi saran untuk pengembangan selanjutnya.

**BAB II**

**LANDASAN TEORI**

**2.1 Landasan Teori**

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan aplikasi media informasi Perguruan Tinggi menggunakan Chatbot Line di Kota Bandung.

**2.2 AI (Artificial Intelligence) atau Kecerdasan Buatan**

Pada dasarnya AI adalah suatu pengetahuan yang membuat komputer dapat meniru kecerdasan manusia sehingga komputer dapat melakukan hal-hal yang dikerjakan manusia dimana membutuhkan suatu kecerdasan; misalkan melakukan analisa penalaran untuk mengambil suatu kesimpulan atau keputusan atau penerjemahan dari satu bahasa ke bahasa lain .

Kecerdasan Buatan adalah salah satu cabang Ilmu pengetahuan berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi. Hal Ini biasanya dilakukan dengan mengikutI atau mencontoh karakteristik dan analogi berpikir dari kecerdasan atau Inteligensia manusia, dan menerapkannya sebagai algoritma yang dikenal oleh komputer. Dengan suatu pendekatan yang kurang lebih fleksibel dan efisien dapat diambil tergantung dari keperluan, yang mempengaruhi bagaimana wujud dari perilaku kecerdasan buatan. AI biasanya dihubungkan dengan Ilmu Komputer, akan tetapi juga terkait erat dengan bidang lain seperti Matematika, Psikologi, Pengamatan, Biologi, Filosofi, dan yang lainnya. Kemampuan untuk mengkombinasikan pengetahuan dari semua bidang ini pada akhirnya akan bermanfaat bagi kemajuan dalam upaya menciptakan suatu kecerdasan buatan.

Menurut beberapa ahli kecerdasan buatan didefinisikan sebagai berikut :

1. Menurut Rich and knight [7] : ”Kecerdasan buatan (artificial intellegence) merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia”.
2. Menurut Andri Kristanto [8]: Kecerdasan buatan merupakan bagian dari ilmu pengetahuan komputer yang khusus ditujukan dalam perancangan otomatisasi tingah laku cerdas dalam sistem kecerdasan komputer. Kecerdasan buatan dilihat dari berbagai sudut pandang adalah sebagai berikut :
3. Sudut pandang Kecerdasan (Intellegence)

Kecerdasan buatan adalah bagaimana membuat mesin yang ”cerdas” dan dapat melakukan hal-hal yang sebelumnya dapat dilakukan oleh manusia.

1. Sudut pandang Penelitian

Studi bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dilakukan oleh manusia.

**2.2.1 Konsep Dasar AI atau Kecerdasan Buatan**

Secara Garis besar Kecerdasan Buatan dibagi menjadi 4 kategori diantaranya :

1. Thinking humanly

Pendekatan ini dilakukan dengan dua cara :

1. *Melalui introspeksi*, mencoba menangkap pemikiran kita sendiri saat kita berfikir. “how do you know that you understand?”.
2. Melalui penelitian-penelitian dari segi psikologi.
3. Acting humanly
4. Tahun 1950, Alan Turing merancang suatu ujian bagi komputer yang berintelijensia (bot Cerdas) untuk menguji apakah komputer tersebut mampu mengelabuhi seorang manusia/ interrogator melalui komunikasi berbasis teks jarak jauh.
5. Tentunya komputer tersebut harus memiliki kemampuan, Natural Language Processing, Knowledge Representation, Automated Reasoning, Machine Learning, Computer Vision, Robotics.
6. Thinking rationally

Terdapat dua masalah dalam pendekatan ini diantaranya :

1. Tidak mudah membuat pengetahuan informal, lalu menyatakan dalam formal term dengan notasi-notasi logika.
2. Terdapat perbedaan besar antara dapat memecahkan masalah “secara prinsip” dan memecahkannya “dalam dunia nyata”.
3. Acting rationally (the Rational agent approach)
4. Membuat inferensi logis merupakan bagian dari suatu rational agent. Karena untuk melakukan aksi secara rasional adalah dengan menalar secara logis.
5. Dengan menalar secara logis, maka bisa didapatkan kesimpulan bahwa aksi yang dilakukan akan mencapai tujuan atau tidak.

Selain dari keempat kategori Kecerdasan Buatan diatas, ada beberapa konsep yang harus dipahami dalam Kecerdasan Buatan diantaranya [9]:

1. Turing Test – Metode Pengujian Kecerdasan

Turing Test merupakan sebuah metode pengujian kecerdasan yang dibuat oleh Alan Turing. Proses uji ini melibatkan seorang penanya (manusia) dan dua obyek yang ditanyai. Yang satu adalah seorang manusia dan yang satunya adalah sebuah mesin yang akan diuji. Penanya tidak dapat melihat langsung kepada obyek yang ditanyai. Penanya diminta untuk membedakan mana jawaban komputer dan mana jawaban manusia berdasarkan jawaban kedua obyek tersebut. Jika penanya tidak dapat membedakan mana jawaban mesin dan mana jawaban manusia maka Turing berpendapat bahwa mesin yang diuji tersebut dapat diasumsikan CERDAS.

1. Pemrosesan Simbolik

Komputer semula didesain untuk memproses bilangan atau angka - angka (pemrosesan numerik). Sementara manusia dalam berfikir dan menyelesaikan masalah lebih bersifat simbolik, tidak didasarkan pada sejumlah rumus atau melakukan komputasi matematika. Sifat penting dari AI adalah bahwa AI merupakan bagian dari ilmu komputer yang melakukan proses secara simbolik dan non-algoritmik dalam penyelesaian masalah.

1. Heuristic

Istilah heuristic diambil dari bahasa Yunani yang berarti menemukan. Heuristic merupakan suatu strategi untuk melakukan proses pencarian (search) ruang problem secara selektif, yang memandu proses pencarian yang kita lakukan disepanjang jalur yang memiliki kemungkinan sukses paling besar.

1. Penarikan Kesimpulan (Inferencing)

AI mencoba membuat mesin memiliki kemampuan berfikir atau mempertimbangkan (reasoning). Kemampuan berfikir (reasoning) termasuk didalamnya proses penarikan kesimpulan (inferencing) berdasarkan fakta-fakta dan aturan dengan menggunakan metode heuristic atau pencarian lainnya.

1. Pencocokan Pola (Pattern Matching)

AI bekerja dengan metode pencocokan pola (pattern matching) yang berusaha untuk menjelaskan objek, kejadian (event) atau proses, dalam hubungan logika atau komputasional.

**2.2.2 Lingkup Utama Dalam Kecerdasan Buatan**

Makin pesatnya perkembangan teknologi menyebabkan adanya perkembangan dan perluasan lingkup yang membutuhkan kehadiran kecerdasan buatan. Karakteristik ‘cerdas' sudah mulai dibutuhkan diberbagai disiplin ilmu dan teknologi. Kecerdasan buatan tidak hanya dominan di bidang ilmu komputer (informatika), namun juga sudah merambah di berbagai disiplin ilmu yang lain antara lain psikologi dan kecerdasan buatan melahirkan sebuah area yang dikenal dengan nama cognition & psycolinguistics.

Kecerdasan buatan juga memberikan kontribusi yang cukup besar dibidang manajemen. Adanya sistem pendukung keputusan, dan Sistem Informasi Manajemen juga tidak terlepas dari andil kecerdasan buatan. Adanya irisan penggunaan kecerdasan buatan di berbagai disiplin ilmu tersebut menyebabkan cukup rumitnya untuk mengklasifikasikan kecerdasan buatan menurut disiplin ilmu yang menggunakannya. Lingkup utama dalam kecerdasan buatan, antara lain:

1. Sistem Pakar (Expert System). Disini digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian komputer akar memiliki keahlian untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar.
2. Pengolahan Bahasa Alami Natural Language Processing). Dengan pengolahan bahasa alami ini diharapkan user dapat berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.
3. Pengenalan Ucapan (Speech Recognition). Melalui penalaran ucapan diharapkan manusia dapat berkomunikasi dengan komputer menggunakan suara.
4. Robotika & Sistem Sensor (Robotics & Sensory Systems)
5. Computer Vision, mencoba untuk dapat menginterpretasikan gambar atau obyek-obyek tampak melalui komputer.
6. IntelligenceComputer-aided Instruction. Komputer dapat digunakan sebaga tutor yang dapat melatih dan mengajar.
7. Game playing.

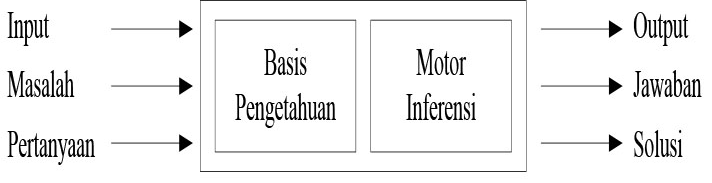
**2.2.3 Tujuan Kecerdasan Buatan**

Tujuan dari kecerdasan buatan menurut Winston dan Prendergast :

1. Membuat mesin menjadi lebih pintar (tujuan utama).
2. Memahami apa itu kecerdasan (tujuan ilmiah).
3. Membuat mesin lebih bermanfaat (tujuan entrepreneurial).

Dua bagian utama yang dibutuhkan untuk aplikasi kecerdasan buatan seperti yang terlihat pada Gambar x proses motor inferensi adalah :

1. Basis Pengetahuan (Knowledge Base) berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
2. Motor Inferensi (Inference Engine) adalah kemampuan menarik kesimpulan berdasarkan pengalaman.



Gambar II.1 Proses Motor Inferensi

**2.3 Sistem Pakar**

sistem pakar adalah suatu program yang bertindak sebagai penasehat atau konsultan pintar dengan mengambil pengetahuan yang disimpan dalam domain tertentu. Seorang pemakai yang belum berpengalaman dalam mendiagnosa suatu masalah dapat memecahkan masalah yang sulit dan mengambil keputusan yang benar. [9]

Secara umum sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan para ahli. Sistem pakar tidak untuk menggantikan kedudukan seorang pakar tetapi untuk memberikan pengetahuan dan pengalaman pakar tersebut. Sistem pakar dikembangkan pertama kali oleb komunitas AI tahun 1960an. Sistem pakar yang pertama adalah General Purpose Problem Solver (GPS) yang dikembangkan oleh Newel Simon. [9]

**2.3.1 Konsep Dasar Sistem Pakar**

Konsep dasar sistem pakar mengandung sebagai berikut :

1. Keahlian

Keahlian adalah suatu kelebihan penguasaan pengetahuan dibidang tertentu yang diperoleh dari pelatihan, membaca atau dari pengalaman Bentuk pengetahuan yang termasuk keahlian:

1. Fakta-fakta pada lingkup permasalahan tertentu.
2. Teori-teori pada lingkup permasalahan tertentu.
3. Aturan-aturan berkenaan dengan lingkup permasalahan tertentu.
4. Meta-knowledge (pengetahuan tentang pengetahuan).
5. Ahli/Pakar

Seorang ahli adalah seseorang yang mampu menjelaskan suatu tanggapan, mempelajari hal-hal baru seputar topik permasalahan, menyusun kembali pengetahuan jika dipandang perlu, memecahkan masalah dengan cepat dan tepat.

1. Pengalihan Keahlian

Tujuan dari sistem pakar adalah untuk mentransfer keahlian dari seorang pakar ke dalam komputer kemudian ke masyarakat. Proses ini meliputi 4 kegiatan.Yaitu perolehan pengetahuan (dari para ahli atau sumber-sumber lainnya). representasi pengetahuan ke komputer, kesimpulan dari pengetahuan dan pengalihan pengetahuan ke pengguna.

1. Mengambil Keputusan

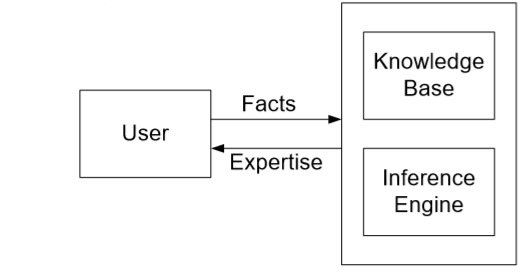
Hal yang unik dari sistem pakar adalah kemampuan untuk menjelaskan dimana keahlian tersimpan dalam basis pengetahuan. Kemampuan komputer untuk mengambil kesimpulan dilakukan oleh komponen yang dikenal dengan mesin inferensi yaitu meliputi prosedur tentang pemecahan masalah.

1. Aturan (*Rule*)

Sistem pakar dibuat merupakan sistem yang berdasarkan pada aturan-aturan dimana program disimpan dalam bentuk aturan-aturan sebagai prosedur pemecahan masalah. Aturan tersebut biasanya berbentuk IF-THEN.

1. Kemampuan menjelaskan

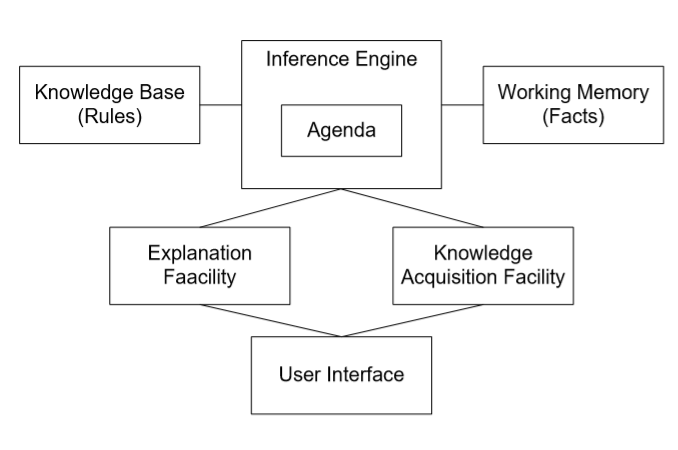
Keunikan lain dari sistem pakar adalah kemampuan dalam menjelaskan atau memberi saran rekomendasi serta juga menjelaskan mengapa beberapa tindakan saran tidak direkomendasikan.



Gambar II.2 Konsep Dasar Fungsi Sistem Pakar

Gambar 3 menggambarkan konsep dasar suatu sistem pakar knowledge base. Pengguna menyampaikan fakta atau informasi untuk sistem pakar dan kemudian menerima saran dari pakar atau jawaban ahlinya. Bagian dalam sistem pakar terdiri dari 2 komponen utama, yaitu *knowledge based* yang berisi pengetahuan kepakaran dan *inference engine* yang menggambarkan kesimpulan Kesimpulan tersebut merupakan respon dari sistem pakar atas permintaan pengguna

**2.3.2 Struktur Sistem Pakar**



Gambar II.3 Struktur Sistem Pakar

Berikut adalah penjelasan tiap struktur sistem pakar :

1. User Interface (antarmuka) : Mekanisme komunikasi antara user dan Sistem Pakar.
2. Explanation Facility (subsistem penjelasan) : digunakan untuk melacak respon dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif.
3. Working Memory : database global dari fakta yang digunakan dalam prosedur.
4. Agenda : Daftar prioritas prosedur yang dibuat oleh motor inferensi dan direkam dalam working memory
5. Inference engine (motor inferensi) : program yang berisi metodologi yang digunakan untuk melakukan penalaran terhadap informasi-informasi dalam basis pengetahuan untuk memformulasikan konklusi.
6. Knowledge acquisiton facility : berisi pengetahuanpengetahuan yang dibutuhkan untuk memahami, memformulasikan dan menyelesaikan masalah.

**2.3.3 Basis Aturan Sistem Pakar**

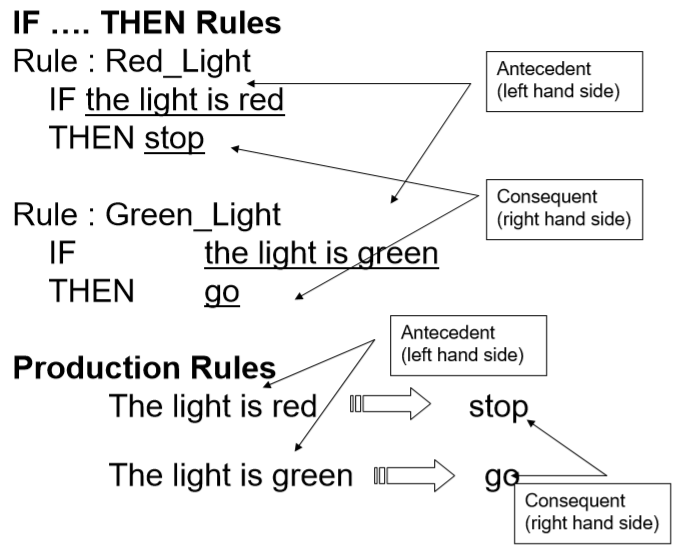
Adapun basis aturan dalam Sistem Pakar adalah sebagai berikut :

1. Pengetahuan dalam Sistem Pakar direpresentasikan dalam bentuk IF -THEN atau dalam bentuk Production Rules.
2. Motor inferensi menentukan aturan awal (rule antecedents) yang sesuai. Bagian sisi kiri harus cocok dengan fakta yang ada di memori kerja
3. Aturan yang sesuai ditempatkan di agenda dan dapat diaktivasi

- Aktivasi aturan akan membangkitkan fakta baru di sisi kanan

- Aktivasi dari satu aturan adalah bagian dari aktivasi aturan yang lain.

Contoh Basis Aturan (Rule Based) Sistem Pakar



Gambar II.4

**2.3.4 Keuntungan Sistem Pakar**

Keuntungan Sistem Pakar adalah sebagai berikut :

1. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli.
2. Bisa melakukan proses secara berulang secara otomatis.
3. Menyimpan pengetahuan dan kea hlian para pakar.
4. Meningkatkan output dan produktivitas.
5. Meningkatkan kualitas.
6. Mampu mengambil dan melestarikankeahlian para pakar.
7. Mampu beroperasi dalam lingkungan berbahaya.
8. Memiliki kemampuan untuk mengakses pengetahuan.
9. Memiliki realibilitas.
10. Meningkatkan kapabilitas system computer.
11. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidakpastian.
12. Sebagai media pelengkap dalam pelatihan.
13. Meningkatkan kapabilitas dalam penyelesaian masala.
14. Menghemat waktu dalam pengambilan keputusan.

**2.3.5 Kelemahan Sistem Pakar**

Adapun kelemahan dalam Sistem Pakar adalah sebagai berikut :

1. Biaya yang diperlukan untuk membuat dan memeliharanya sangat mahal.
2. Sulit dikembangkan. Hal ini erat kaitannya dengan ketersediaan pakar dalam bidangnya.
3. System pakar tidak 100% bernilai benar.

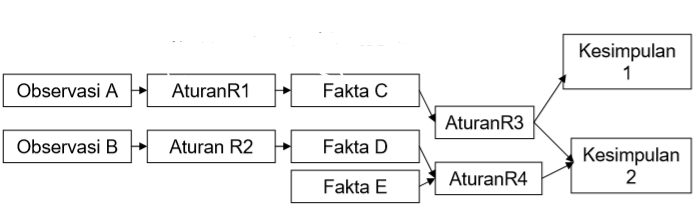
**2.4 Penarikan Kesimpulan (Inferencing)**

Kemampuan berfikir (reasoning) termasuk didalamnya proses penarikan kesimpulan (inferencing) berdasarkan fakta-fakta dan aturan dengan menggunakan metode heuristic atau pencarian lainnya.

**2.4.1 Metode Algoritma Forward Chaining**

Dalam sistem pakar, metode forward chaining merupakan metode yang melakukan pelacakan ke depan, dimulai dari sekumpulan fakta dan berakhir di kesimpulan. Metode forward chaining bermula dari fakta-fakta yang sudah diketahui atau ditetapkan dalam suatu sistem pakar. Kemudian menggunakan premis yang ditentukan oleh user, yang nantinya premis-premis itu akan disesuaikan dengan fakta-fakta tadi menggunakan suatu aturan tertentu. Hasil dari proses ini akan menghasilkan fakta baru, yang nantinya akan digunakan untuk melanjutkan proses dan mendapatkan kesimpulan akhir setelah tidak ada lagi aturan yang premisnya cocok dengan fakta.

Alasan mengapa menggunakan forward chaining adalah karena terdapat banyak cara atau aturan yang berbeda untuk mendapatkan kesimpulan yang sedikit, dan ingin mendapatkan kesimpulan dari fakta-fakta yang sudah ada sebelumnya.



Gambar II.5 Aturan Forward Chaining

Pencocokkan fakta atau pernyataan dimulai dari bagian sebelah kiri (IF dulu). Dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis.

Contoh :

Pada tabel di bawah ini terlihat 10 aturan yang tersimpan dalam basis pengetahuan. Fakta awal yang diberikan hanya A & F (artinya A dan F bernilai benar). Ingin dibuktikan apakah K bernilai benar (hipotesis : K) ?

R1 : if A and B then C

R2 : if C then D

R3 : if A and E then F

R4 : if A then G

R5 : if F and G then D

R6 : if G and E then H

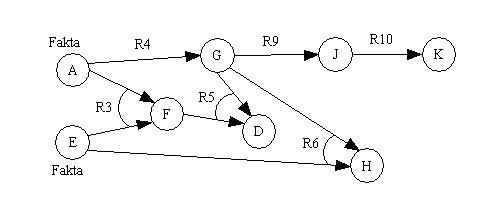
R7 : if C and H then I

R8 : if I and A then J

R9 : if G then J

R10 : if J then K

[Fakta](http://www.fakta.dk/) awal yang diberikan hanya A dan E, ingin membuktikan apakah K bernilai benar. Proses penalaran forward chaining terlihat pada gambar dibawah :



Gambar II.6 Forward Chaining

|  |  |
| --- | --- |
| **Aturan** | **Fakta Baru** |
| R-3 | F |
| R-4 | G |
| R-5 | D |
| R-6 | H |
| R-9 | J |
| R-10 | K (Terbukti) |

**2.5 Chatbot**

Chat Bot adalah sebuah program komputer yang dirancang untuk mensimulasikan sebuah percakapan atau komunikasi yang interaktif kepada user (manusia) melalui bentuk teks, suara, dan atau visual. Percakapan yang terjadi antara komputer dengan manusia merupakan bentuk respon dari program yang telah dideklarasikan pada database program pada komputer. Respon yang dihasilkan merupakan hasil pemindaian kata kunci pada inputan user dan menghasilkan respon balasan yang dianggap paling cocok, atau pola kata-kata yang dianggap paling mendekati, dari database tentunya. Dalam bahasa sehariharinya Chat Bot merupakan Aplikasi atau Program komputer yang dirancang untuk meniru manusia itu sendiri, batasan yang diambil dari Chat Bot adalah mampu meniru komunikasi manusia. Jadi jika manusia sedang bercakap-cakap dengan program ini, seakan-akan ada 2 pribadi manusia yang saling berkomunikasi. Nyatanya tidak, manusia berkomunikasi dengan Bot. Bot sudah dirancang untuk merespon segala jenis pertanyaan dan pernyataan yang diinputkan oleh manusia (user). Hal ini terjadi karena sebelumnya sudah dideklarasikan pada database, berupa entitas-entitas kata, pola kalimat, dan berbagai jenis pernyataan dan pertanyaan.

Chatbot adalah sebuah simulator percakapan yang berupa program komputer yang dapat berdialog dengan penggunanya dalam bahasa alami. Karena chatbot hanya sebuah program, dan bukan robot (chatbot tidak memiliki tubuh dan tidak memiliki mulut sehingga tidak dapat berbicara seperti manusia), maka yang dimaksud dengan dialog antar manusia sebagai pengguna dengan chatbot dilakukan dengan cara mengetik apa yang akan dibicarakan dan chatbot akan memberikan respon. Orang yang membuat dan mengembangkan program chatbot disebut bot master.

Kata bot diambil dari kata robot, yang dapat diartikan sebagai sebuah program komputer yang berfungsi mengumpulkan informasi atau memberikan layanan yang biasanya telah dijadwalkan sebelumnya. Sebuah bot (atau bisa juga dikatakan sebagai sebuah agent) biasanya mencari informasi di internet mengumpulkan informasi yang sesuai dengan yang kita inginkan, dan memberikannya pada kita secara periodik atau sesuai dengan perintah kita.

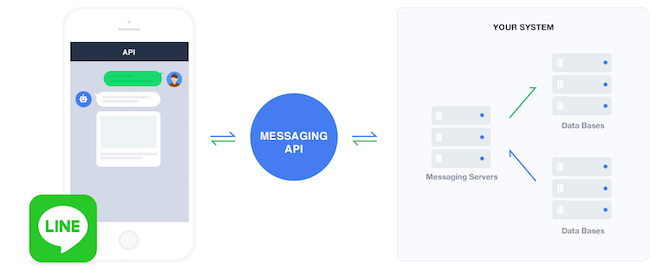
Bot biasa digunakan dalam IRC (Internet Chat Groups) di mana bot tersebut melakukan berbagai fungsi seperti menyapa partisipan baru dan memonitor penggunaan kata-kata yang tidak pantas. Sekarang ini, bot dapat mengumpulkan informasi, mengolah informasi tersebut, dan memberikan respon terhadap pengguna dan event yang terjadi di internet.

**2.5.1 Cara Kerja Chatbot**

Pada dasarnya bots bekerja dengan cara melihat kata kunci dalam data yang masuk dan membalasnya dengan kata kunci yang paling cocok, atau pola kata-kata yang paling mirip dari basis data tekstual. Artinya, jika pengguna mengirim suatu permintaan maka bot akan membalasnya dengan respon yang spesifik sesuai dengan kata kunci yang dikirim. Sebagai contoh, misalnya pengguna mengirim atau mengetikkan suatu pertanyaan “Jam berapa toko anda buka besok pagi?” maka dengan informasi yang sudah tersedia, bot akan segera merespon sesuai dengan pertanyaan tersebut “Toko kami besok buka jam 9 pagi dan tutup jam 5 sore”. Disini anda hanya perlu memberikan perintah atau melakukan sebuah percakapan baik itu dalam bentuk teks maupun audio, dan bot akan mencari data yang anda butuhkan sesuai dengan kata kunci yang anda berikan.

**2.5.2 LINE Chatbot**

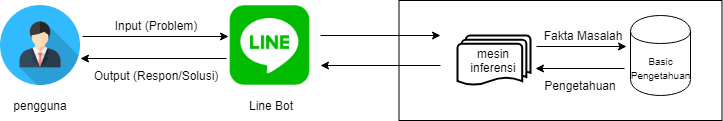
Line Chatbot merupakan salah satu layanan chatbot yang terintegrasi dengan layanan media sosial LINE . **LINE merupakan aplikasi pesan instan yang banyak digunakan oleh masyarakat di Indonesia. Berdasarkan data dari LINE, penggunanya di Indonesia berjumlah 90 juta pengguna, 80% diantaranya didominasi pengguna anak muda[3]. Salah satu produk LINE yang saat ini sedang diminati pelanggan yakni LINE Chatbot. Chatbot merupakan program komputer yang dirancang untuk mensimulasikan sebuah percakapan atau komunikasi yang interaktif kepada user (manusia) melalui bentuk teks, audio, maupun video[4]. Pemanfaatan LINE Chatbot merupakan cara yang efektif dalam mencari informasi secara interaktif . Berikut adalah gambar x cara kerja Messaging API yang dapat digunakan untuk membuat chatbot.**



Gambar II.7 Cara kerja Messaging API

Gambar di atas menjelaskan bagaimana LINE Messaging API bekerja. Pada dasarnya Messaging API memberi akses pada sistem kita (yang menjadi Chatbot Server) untuk dapat menerima dan mengirim pesan balasan ke pengguna melalui Messaging API.

**2.5.3 Skema Knowledge Chatbot**

 Adapun skema chatbot dapat dilihat pada gambar 8 dibawah ini :

Gambar II.8 Skema Knowledge Chatbot

Dari gambar 3 diatas dapat dijelaskan :

1. Pengguna menginputkan text ke program chat bot.
2. Program memproses data input dari pengguna menjadi bahasa yang dapat dimengerti oleh komputer.
3. Program komputer menyesuaikan input pengguna dengan pattern dari database.
4. Program chat bot memberikan respon ke pengguna.

**2.6** **Analisis Berorientasi Objek**

Analisis dan Desain Berorientasi Objek (*Object Oriented Analysis and Design*) adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep. Dasar pembuatannya sendiri adalah objek yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. Alasan mengapa harus memakai metode berorientasi objek yaitu karena perangkat lunak itu sendiri yang bersfat dinamis, di mana hal ini disebabkan karena kebutuhan pengguna berubah dengan cepat. Selain itu bertujuan untuk menghilangkan kompleksitas transisi antar tahap pada pengembangan perangkat lunak, karena pada pendekatan berorientasi objek, notasi yang digunakan pada tahap análisis perancangan dan impelemntasi relatif sama tidak seperti pendekatan konvensional yang dikarenakan notasi yang digunakan pada tahap analisanya berbeda-beda. Hal itu menyebabkan transisi antar tahap pengembangan menjadi kompleks. Di samping itu dengan pendekatan berorientasi objek membawa pengguna kepada abstraksi atau istilah yang lebih dekat dengan dunia nyata, karena di dunia nyata itu sendiri yang sering pengguna lihat adalah objeknya bukan fungsinya. Beda ceritanya dengan pendekatan terstruktur yang hanya mendukung abstraksi pada level fungsional. Adapun dalam pemrograman berorientasi objek menekankan berbagai konsep seperti: *Class, Object, Abstract, Encapsulation, Polymorphism, Inheritance* dan tentunya UML (*Unified Modeling Language*). UML (*Unified Modeling Language*) sendiri merupakan salah satu alat bantu yang dapat digunakan dalam Bahasa pemrograman berorientasi objek. Selain itu UML merupakan *standard modeling language* yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem dan software agar bias menyelesaikan tugas-tugas seperti: Spesifikas, Visualisasi, Desain Arsitektur, Konstruksi, Simulasi dan Testing. Dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah sebuah Bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesfikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis objek (*Object Oriented Programming*) [10].

Dokumentasi UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek yang 4 diantaranya adalah [11]:

1. *Activity Diagram*

*Activity Diagram* adalah sebuah tahapan yang lebih focus kepada menggambarkan proses bisnin dan urutan aktivitas dalam sebuah proses. Di mana biasanya dipakai pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. *Activity diagram* ini sendiri memiliki struktur yang mirip dengan *flowchart* atau data flow diagram pada perancangan terstruktur. *Activity diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram* [12].

1. *Use Case Diagram*

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Di dalam *use case diagram* ini sendiri lebih ditekankan kepada apa yang diperbuat sistem dan bagaimana sebuah sistem itu bekerja. Sebuah *use case* merepresentasikan sebuah interaksi antara actor dengan sistem. *Use case* merupakan bentuk dari sebuah pekerjaan tertentu, misalnya *login* ke dalam sistem, *posting* dan sebagainya, sedangkan seorang actor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu [12]. Adapaun komponen-komponen dalam *use case diagram* diantaranya:

1. Actor

*Actor* merupakan suatu entitas yang berkaitan dengan sistem tapi bukan dari bagian dalam sistem itu sendiri. *Actor* berada diluar sistem namum berkaitan erat dengan fungsionalitas di dalamnya. *Actor* memiliki hubungan secara langsung terhadap fungsi utama baik terhadap salah satu atau semua fungsionalitas utama. *Actor* juga dapat dibagi terhadap berbagai jenis atau tingkatan dengan cara digeneralisasi atau dispesifikasi tergantung kebutuhan sistemnya. *Actor* biasanya dapat berupapengguna atau *database* yang secara pandang berada dalam suatu ruang lingkup sistema.

1. Use Case

*Use case* merupakan gambaran umum dari fungsi atau proses utama yang menggambarkan tentang salah satu perilaku sistem. Perilaku sistem ini terdefinisi dari proses bisnis sistem yang akan dimodelkan. Tidak semua proses bisnis digambarkan secara fungsional pada *use case*, tetapi yang digambarkan hanya fungsionalitas utama yang berkaitan dengan sistem. *Use case* menitik-beratkan bagaimana sautu sistem dapat berinteraksi baik antar sistem maupun di luar sistem.

1. *Sequance Diagram*

*Sequance diagram* digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram jenis ini memberikan kejelasan sejumlah objek dan pesan-pesan yang diletakkan diantaranya di dalam sebuah *use case*. Komponen utamanya adalah objek yang digambarkan dengan kotak segi empat atau bulat, *message* yang digambarkan dengan gari penus dan waktu yang ditunjukkan dengan *progress vertical*. Manfaat dari *sequence* diagram adalah memberikan gambaran detail dari setiap *use case diagram* yang dibuat sebelumnya [12].

1. *Class Diagram*

*Class diagram* adalah sebuah *class*  yang menggambarkan struktur dan penjelasan *class,* paketdan objek serta hubungan satu sama lain. *Class diagram* juga menjelaskan hubungan antar *class* secara keseluruhan di dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan [12].

**2.7 Perangkat Lunak Pengembang (Tools)**

Perangkat Lunak Pengembang merupakan perangkat lunak yang membantu dalam pembuatan aplikasi sistem yang akan digunakan.

**2.7.1 PHP**

PHP adalah bahasa scripting yang menyatu dengan HTML dan dijalankan pada server side. Artinya semua sintaks yang kita berikan akan sepenuhnya dijalankan pada server sedangkan yang dikirimkan ke browser hanya hasilnya saja. contoh struktur syntax PHP sebagai berikut :

<?php

echo(“Hallo apakabar? Nama saya PHP script”);

?>

**2.7.2 MySQL**

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Dengan menggunakan SQL, proses akses database menjadi lebih user - friendly dibandingkan dengan menggunakan dBASE atau Clipper yang masih menggunakan perintah - perintah pemrograman.MySQL merupakan software database yang paling populer di lingkungan Linux, kepopuleran ini karena ditunjang performa query dari databasenya yang saat ini bisa dikatakan paling cepat dan jarang bermasalah. MySQL ini juga sudah dapat berjalan pada lingkungan Windows.

Perintah untuk mengelola database dibagi menjadi 3 (tiga ) kelompok, diantaranya :

1. Perintah untuk mendefinisikan data/DDL (Data Definition Language).
2. Perintah untuk memanipulasi data/DML (Data Manipulation Language).
3. Perintah untuk mengendalikan data/DCL (Data Control Language).

**2.7.3 Basis Data**

Basis data terdiri dari dua kata yaitu basis dan data. Basis dapat diartikan sebagai gudang tempat berkumpul. Sedangkan data adalah represebtasi fakta dunia nyata yang mewakili objek seperti manusia, barang, hewan, peristiwa konsep, yang nyatakan dalam bentuk angka, huruf, simbol teks, gambar, bunyi atau kombinasinya [4].

Basis data dapat didefinisikan dari beberapa sudut pandang diantaranya sebagai berikut :

1. Sekumpulan data persistence (data disimpan defile sekunder atau data yang tahan lama) yang saling terkait, menggambarkan suatu organisasi (enterprise).
2. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
3. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikina rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
4. Kumpulan file atau arsip yang saling berhubungan yang disimpan dalam

media penyimpanan elektronis.

**2.7.3.1 Tujuan Basis Data**

Tujuan Basis Data adalah sebagai berikut :

1. Kemudahan dan kecepatan dalam pengambilan kembali data atau arsip.
2. Efisiensi ruang dan waktu
3. Keakuratan data
4. Ketersediaan untuk proses pengambilan data yang diperlukan setiap saat
5. Kelengkapan data-data yang diperlukan atau yang tersimpan
6. Keamanan data
7. Kebersamaan

**2.7.3.2 Keuntungan Basis data**

Keuntungan basis data adalah sebagai berikut :

1. Mereduksi redudansi yang akibatnya mengurangi inkonsistensi.
2. Data dapat dishare antar aplikasi.
3. Standarisasi data dapat dilakukan.
4. Batasan security dapat diterapkan.
5. Mengelola integritas (keterjaminan akurasi) data.
6. Menyeimbangkan kebutuhan yang saling konflik.
7. Independensi data (objektif DBS) : kekebalan aplikasi terhadap perubahan struktur penyimpanan dan teknik pengaksesan data (basis data harus dapat berkembang tanpa mempengaruhi aplikasi yang telah ada).

**2.7.4 LINE**

LINE adalah sebuah aplikasi pengirim pesan yang dapat digunakan pada berbagai platform seperti smartphone, tablet, dan komputer. Dan jika pengguna ingin menggunakan LINE pastikan smartphone terhubung dengan jaringan internet, sehingga pengguna dapat mengirim pesan teks, mengirim gambar, video, pesan suara, dan lain – lain.

**2.7.5 API**

API adalah singkatan dari ***Application Programming Interface***, dan memungkinkan developer untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. API terdiri dari berbagai elemen seperti function, protocols, dan tools lainnya yang memungkinkan developer untuk membuat aplikasi. Tujuan penggunaan API adalah untuk mempercepat proses development dengan menyediakan function secara terpisah sehingga developer tidak perlu membuat fitur yang serupa. Penerapan API akan sangat terasa jika fitur yang diinginkan sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya. Misalnya: integrasi dengan payment gateway. Terdapat berbagai jenis sistem API yang dapat digunakan, termasuk sistem operasi, library, dan web.

Seperti namanya, Web API dalam diakses melalui protokol ***HTTP***, ini adalah konsep bukan teknologi. Kita bisa membuat Web API dengan menggunakan teknologi yang berbeda seperti PHP, Java, NET*,* dll.

**2.7.6 JSON**

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.

**2.7.7 Webhook**

Webhook adalah konsep API yang saat ini semakin populer digunakan. Semakin banyak yang dapat dilakukan di web, menjadikan webhook makin banyak digunakan. Selain itu, webhook sangat berguna dan mudah untuk diterapkan. Webhook atau yang biasa disebut callback adalah cara bagi suatu aplikasi untuk menyediakan aplikasi lain dengan informasi  real-time. Lebih mudahnya, webhook adalah link URL yang ditambahkan agar data yang dikirim dapat langsung diterima di waktu sama dengan link URL yang sudah ditentukan. Webhook merupakan satu cara yang efisien bagi provider maupun bagi konsumen. Satu-satunya kelemahan webhook adalah kesulitan di awal saat mengaturnya.

**2.8 Pengujian Black box**

Pengujian adalah proses untuk menemukan error pada perangkat lunak sebelum dikirim kepada pengguna. Pengujian perangkat lunak adalah kegiatan yang ditujukan untuk mengevaluasi atribut atau kemampuan program dan memastikan bahwa itu memenuhi hasil yang dicari, atau suatu investigasi yang dilakukan untuk mendapatkan informasi mengenai kualitas dari produk atau layanan yang sedang diuji (under test), Pengujian perangkat lunak juga memberikan pandangan mengenai perangkat lunak secara obyektif dan independen, yang bermanfaat dalam operasional bisnis untuk memahami tingkat risiko pada implementasinya.

**BAB III**

**ANALISIS DAN PERANCANGAN**

## 3.1 Analisis Sistem

Analisis sistem bertujuan untuk mengidentifikasi permasalahan-permasalahan yang terdapat pada sistem serta menentukan kebutuhan-kebutuhan dari sistem yang akan dibangun. Analisis tersebut meliputi analisis masalah, spesifikasi kebutuhan perangkat lunak, analisis kebutuhan fungsional, analisis kebutuhan non fungsional.

**3.1.1 Analisis Masalah**

Perguruan Tinggi adalah satuan pendidikan yang menyelenggarakan pendidikan tinggi dan dapat berbentuk Akademi, Politeknik, Sekolah tinggi, Institut, atau Universitas. **Perguruan Tinggi di kota Bandung pada tahun 2017 berjumlah 107 untuk Swasta diantaranya 20 Universitas, 2 Institut, 50 Sekolah Tinggi, 25 Akademi, dan 10 Politeknik [1]. Sedangkan untuk Perguruan Tinggi Negeri di Kota Bandung berjumlah 11 [2].**

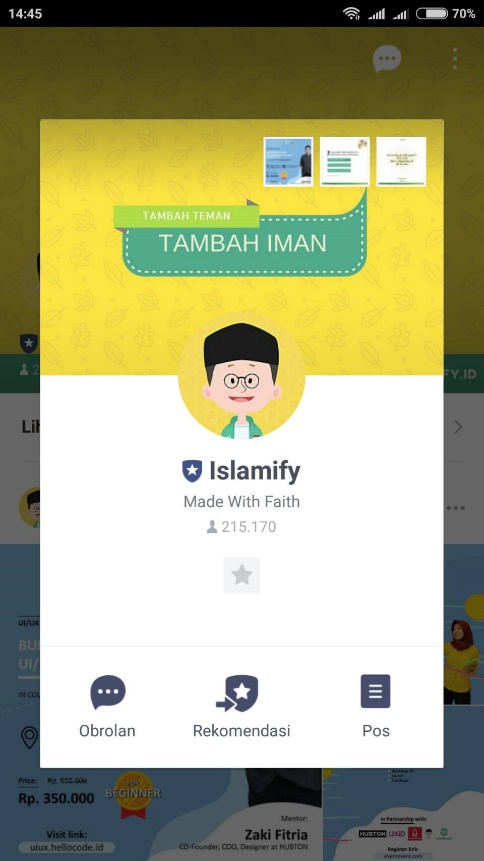
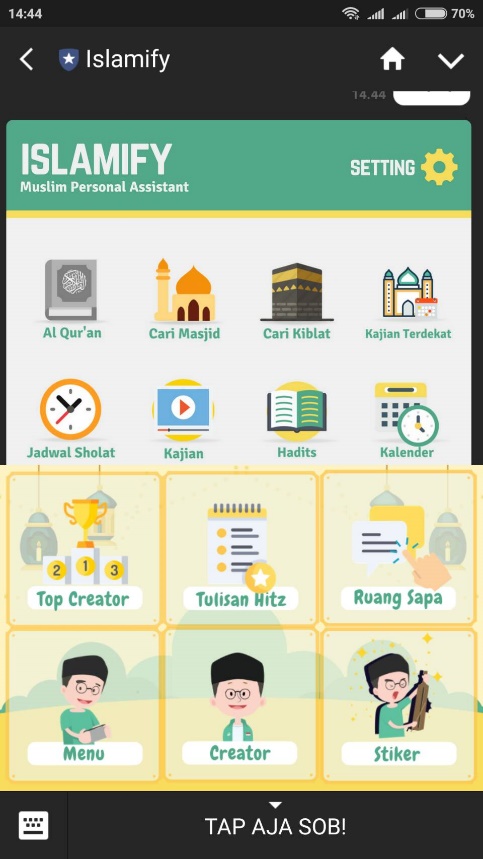
**Untuk mendapatkan Informasi Perguruan Tinggi yang diminatinya, saat ini siswa SMA/K dalam mencari informasi Perguruan Tinggi menggunakan cara mengunjungi setiap website Perguruan Tinggi satu persatu yang diinginkan untuk mendapatkan informasi mengenai Perguruan Tinggi. Berdasarkan data kuisoner terhadap 68 responden siswa SMA/K yang akan melanjutkan sekolahnya ke Perguruan Tinggi, terdapat 86% siswa yang masih mengalami kebingungan dalam menentukan Perguruan Tinggi yang diminatinya. Lalu berdasarkan data kuisoner yang sama, faktor peminat siswa SMA/K dalam mencari Perguruan Tinggi diantaranya berdasarkan Akreditasi, Lokasi, Biaya perkuliahan serta Jurusan yang diminatinya .**

**3.1.2 Analisis Aplikasi Sejenis**

**Berdasarkan analisis yang dilakukan terhadap beberapa bot yang dilakukan oleh penulis diperoleh beberapa masukan yang digunakan dalam penelitian ini yakni sebagai berikut :**

1. **Islamify**

**Islamify merupakan bot pada aplikasi pesan instal Line yang menyediakan informasi mengenai ayat Al Qur’an, lokasi masjid, kiblat, jadwal sholat dan kajian agama. Gambar 3.1 merupakan tampilan bot Islamify :**

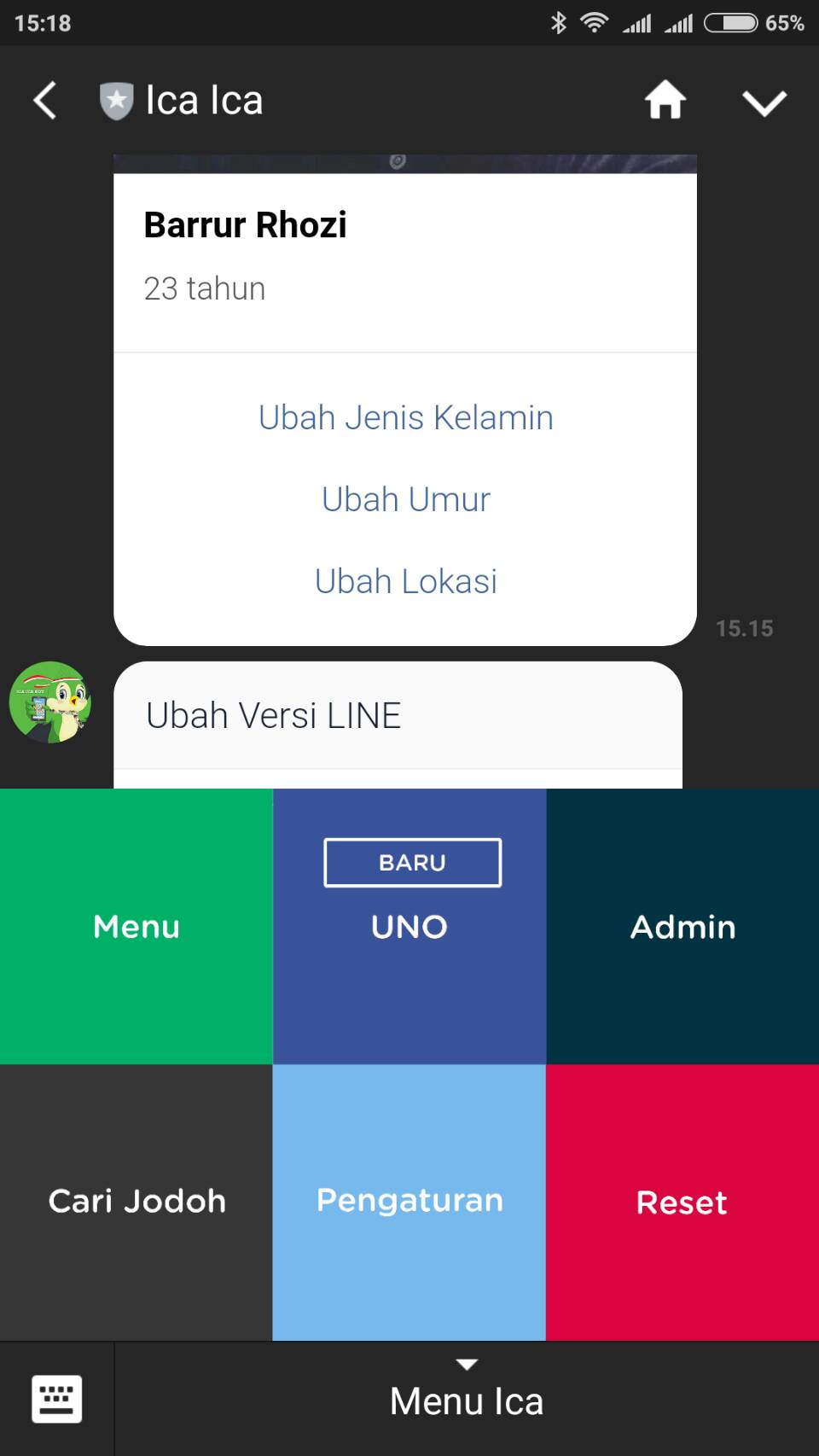
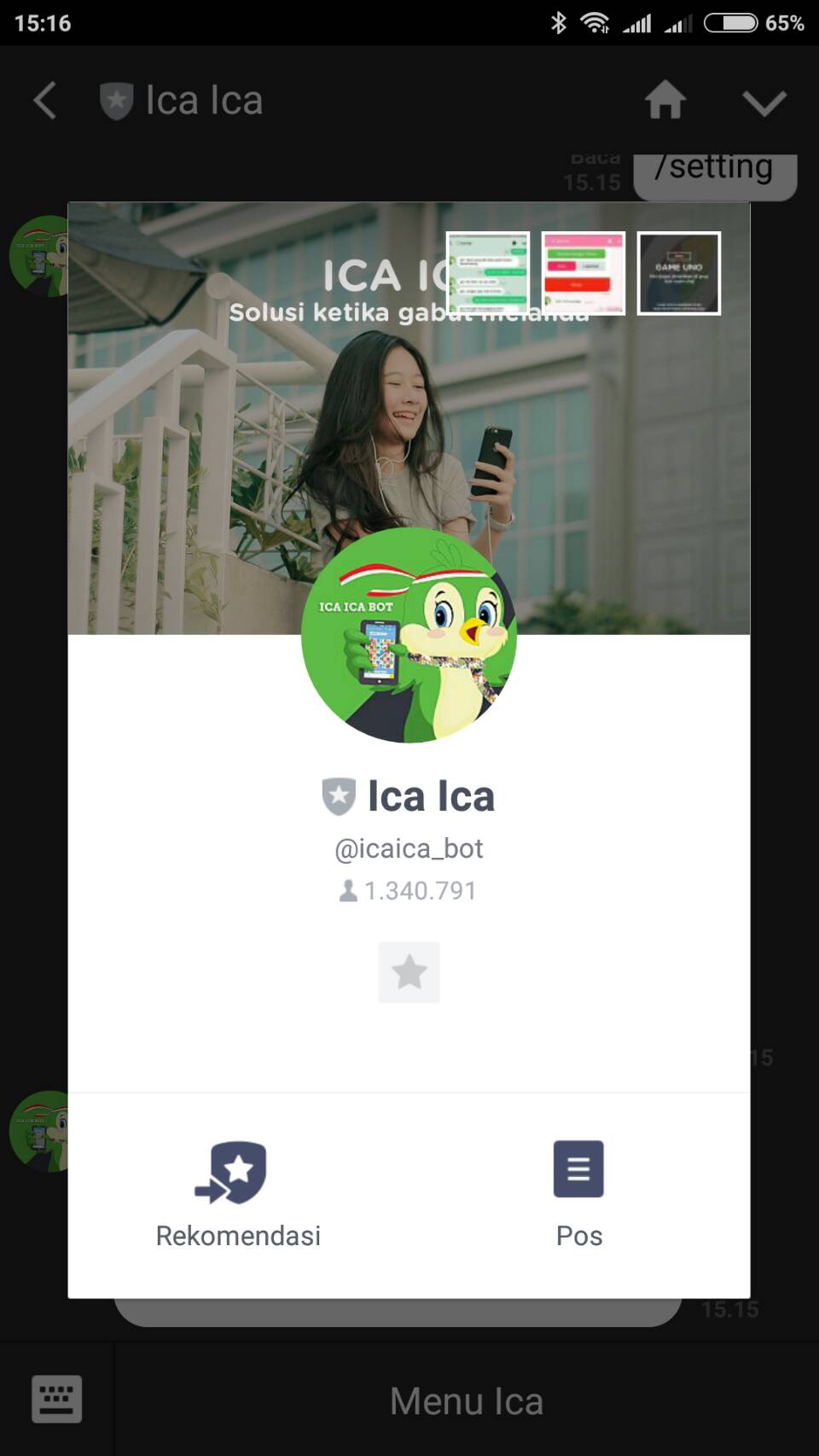
****

**Gambar III.1 Bot Islamify**

**Islamify dijadikan sebagai contoh acuan pengembangan pada penelitian ini. Beberapa acuan yang diterapkan diantaranya yakni konsep konten yang digunakan di dalam bot (mengenai pencarian lokasi).**

1. **Ica Ica**

**Ica Ica merupakan bot pada pesan instan Line yang meyediakan layanan bermain game dan melakukan pencarian jodoh. Gambar III.2 merupakan tampilan bot Ica Ica.**

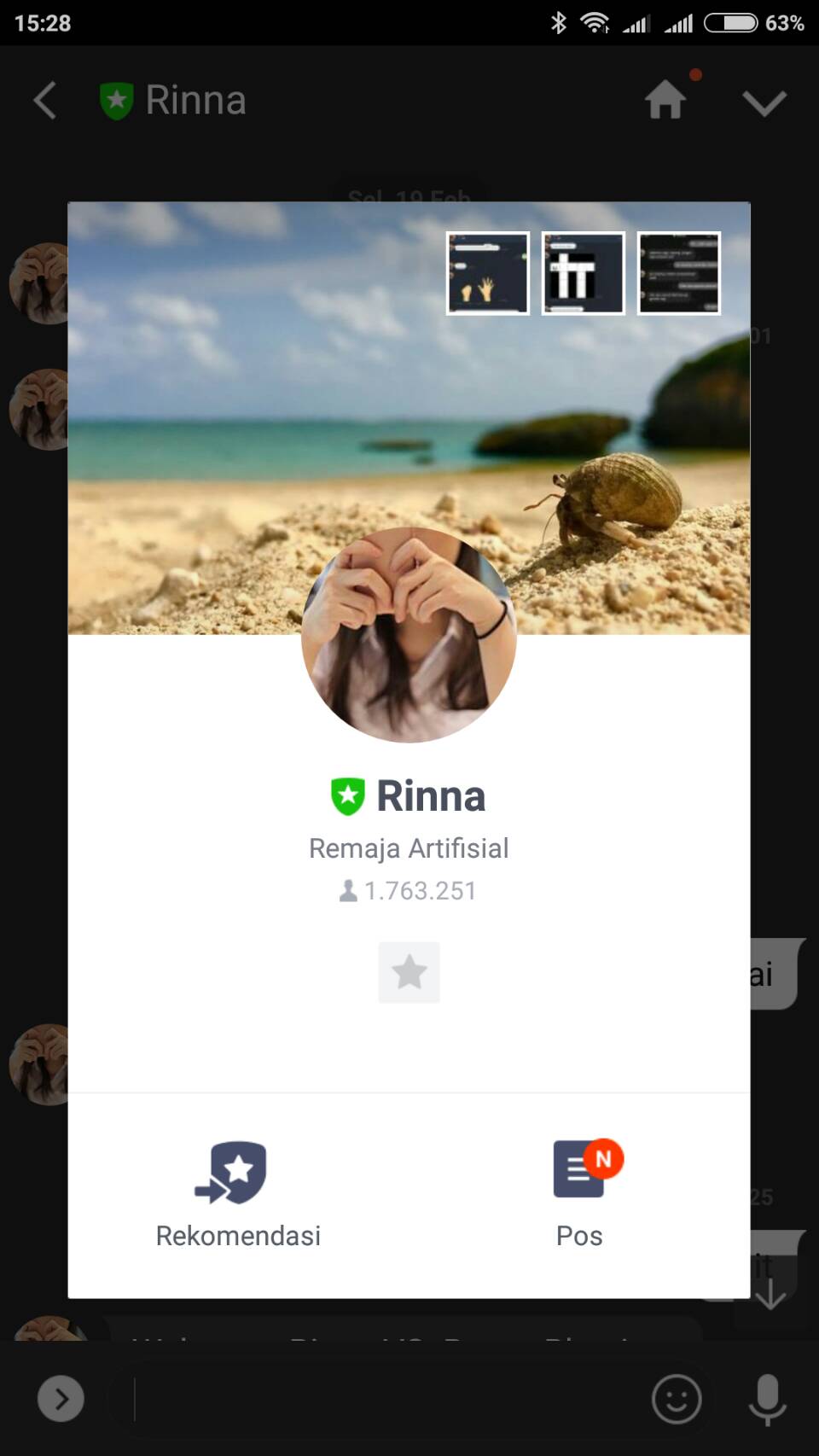


Gambar III.2 Ica Ica Bot

Ica Ica dijadikan acuan pengembangan dalam penelitan ini dalam bentuk fitur yang tersedia didalamnya (menu profil untuk melihat informasi umum mengenai akun Line yang digunakan oleh user dan tampilan menu dengan ukuran 2500x1686px yang didalamnya terdapat 6 menu).

1. Rinna

Rinna diciptakan oleh beberapa teknisi dari tim Microsoft Artificial Intelligence & Research dengan merancangnya sebagai Bot AI yang peka terhadap para pengguna. Rinna menggunakan bahasa seperti remaja pada umumnya, dan ia dapat membicarakan topik tentang wanita seperti gosip, artis, kadang-kadang film Korea, bahkan animasi atau fesyen. **Gambar III.3 merupakan tampilan mengenai bot Rinna.**

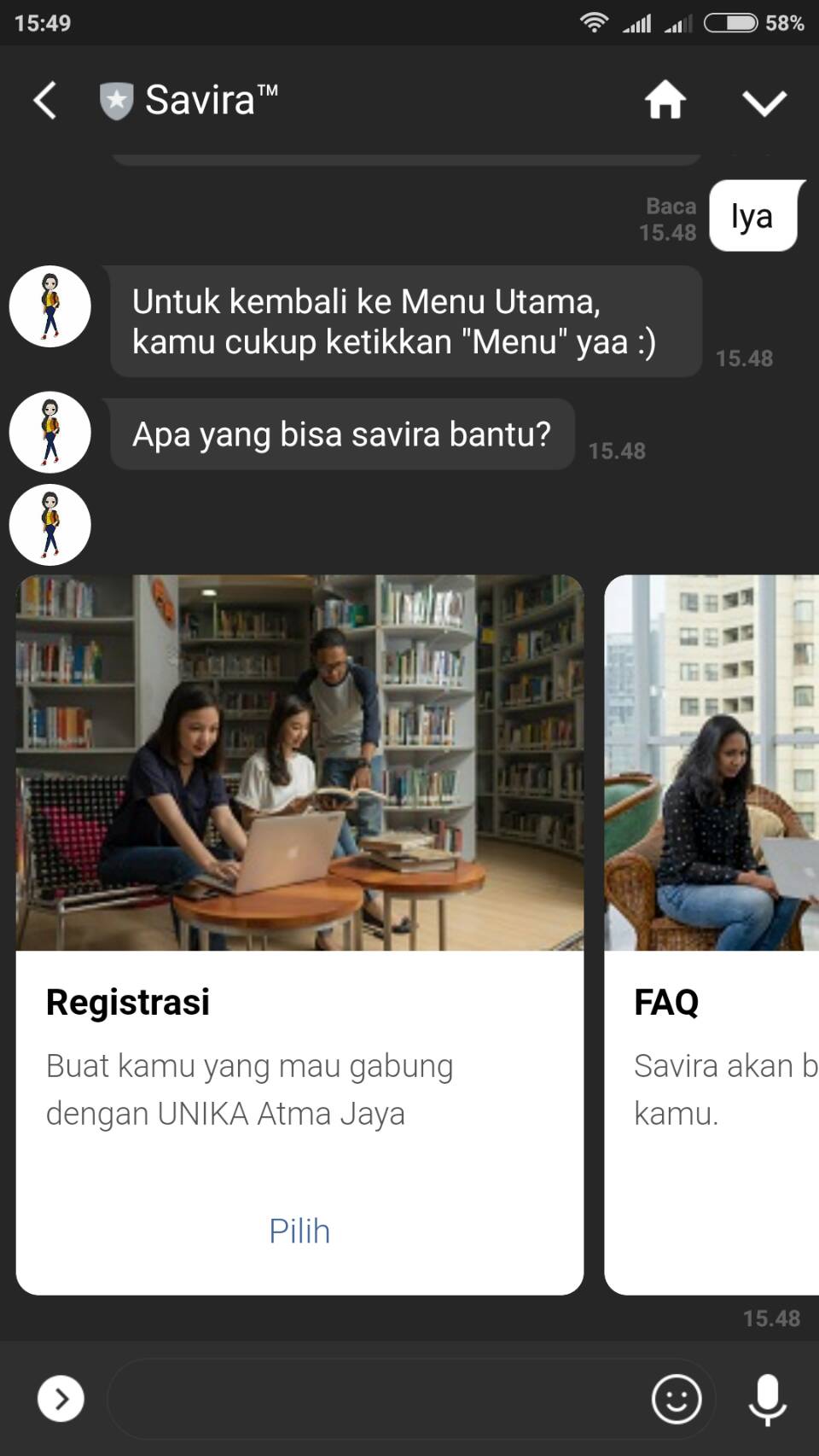
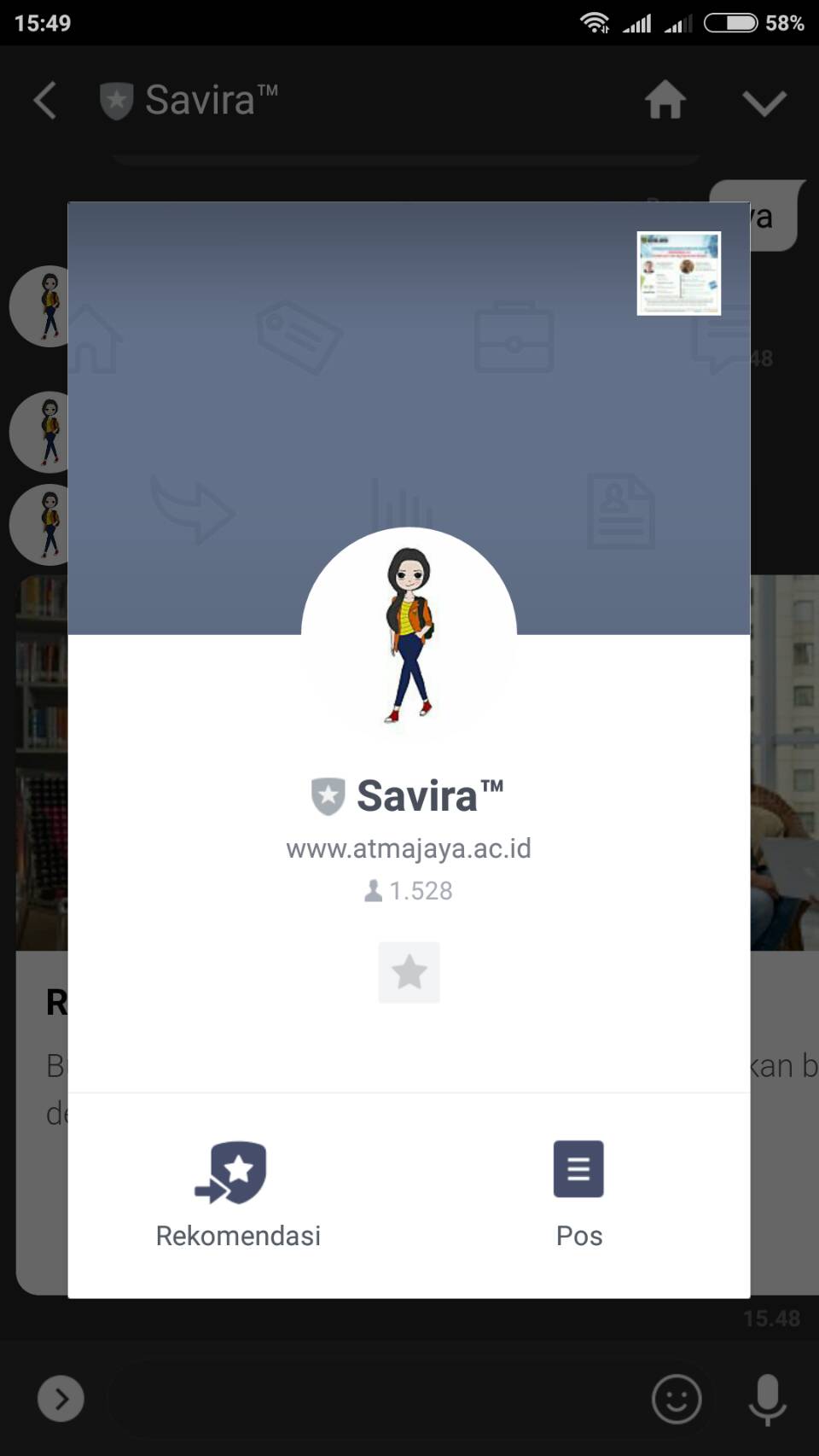


Gambar III.3 Bot Rinna

Bot Rinna dijadikan acuan pengembangan dalam penelitan ini dalam bentuk AI (Artificial Intelligence) atau kecerdasan buatan dalam pembuatan aplikasi chatbot yang akan diteliti.

1. SaviraTM

Savira atau Sahabat Virtual Atma Jaya adalah chatbot LINE yang dapat memberikan informasi seputar perkuliahan di Unika Atma Jaya. Calon mahasiswa dapat bertanya tentang pendaftaran mahasiswa baru, fasilitas, jurusan, prestasi, hingga terkait administrasi. Gambar III.4 merupakan tampilan mengenai Bot Savira.

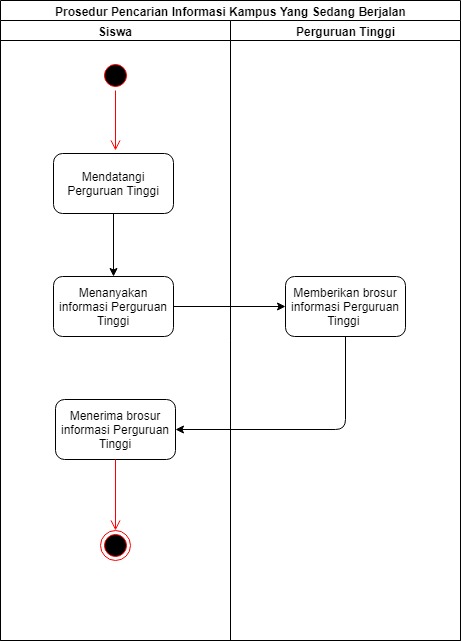


Gambar III.4 Bot Savira

**Bot Savira dijadikan sebagai contoh acuan pengembangan pada penelitian ini. Beberapa acuan yang diterapkan diantaranya yakni konsep penyediaan informasi mengenai kamus diantaranya informasi akreditasi, program studi, lokasi, dan biaya perkuliahan.**

**3.2 Analisis Prosedur Yang Berjalan**

Prosedur pencarian informasi Perguruan Tinggi yang terjadi pada siswa sekolah SMA/K, biasanya harus mencari melalui brosur perguruan tinggi atau mendatangi Perguruan Tinggi secara langsung untuk mendapatkan informasinya. Sehingga, hal ini sangat menyulitkan siswa sekolah SMA/K dalam mencari informasi Perguruan Tinggi yang diminatinya . Tentu saja hal ini sangat kurang efektif dan efisien dalam mencari informasi Peguruan Tinggi. Sedangkan untuk melakukan pencarian informasi lokasi Perguruan Tinggi, siswa sekolah SMA/SMK harus mencari secara manual baik melalui internet ataupun brosur yang didalamnya tercantum lokasi tempat Perguruan Tinggi.



Gambar III.5 Prosedur sistem yang berjalan

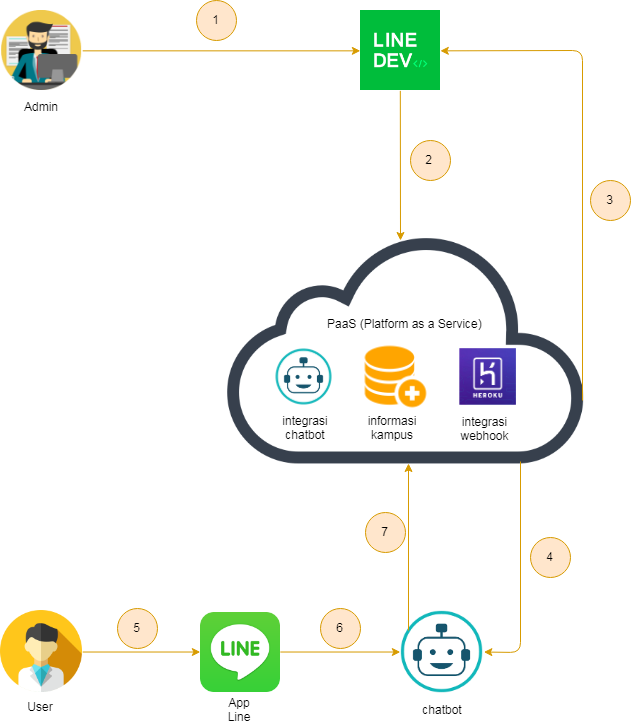
**3.3 Analisis Solusi Yang Ditawarkan**

Berdasarkan analisis terhadapat masalah yang ada, maka diperlukan aplikasi untuk menyediakan pencarian informasi Perguruan Tinggi di Kota Bandung dengan menggunakan LINE Chatbot. Teknologi LINE Chatbot dipilih berdasarkan data kuisoner terhadap pengguna yang sering menggunakan layanan chat, serta karakteristik kemudahan penggunaan dan pembangunan . Adapun karakter chatbot yang dibangun adalah sebagai berikut :

1. Pengguna dapat berinteraksi dengan bot untuk melakukan pencarian informasi Perguruan Tinggi mengenai akreditasi, biaya perkuliahan, jurusan, dan lokasi kampus yang ada di Kota Bandung.
2. Pengguna dapat berinteraksi dengan bot dengan cara melakukan pencarian melalui menu yang sudah disiapkan, serta melakukan pencarian dengan mengetikan pesan berupa pertanyaan kepada bot dalam bahasa Indonesia. Selain itu, pengguna dapat mencari lokasi tempat Perguruan Tinggi di Kota Bandung dengan cara mengirimkan lokasi tempat pengguna berada.

**3.4 Analisis Deskripsi Aplikasi**

Aplikasi yang dibangun berupa chatbot dimana pengguna dapat melakukan pencarian informasi dengan cara melalui menu yang sudah disediakan dan dengan cara mengetikan pesan berupa pertanyaan. Adapun cara kerja aplikasi secara umum dapat dilihat pada gambar dibawah ini.



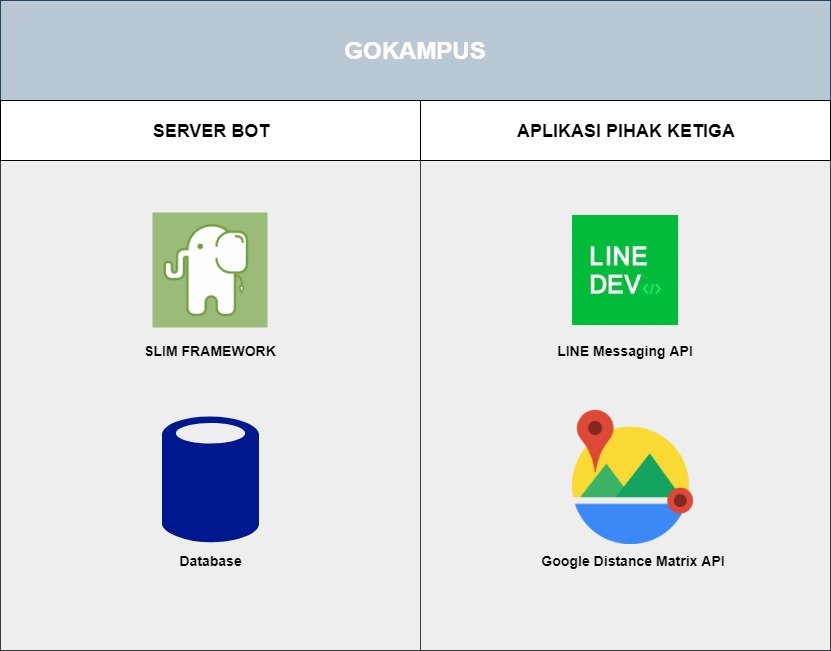
Gambar III.6 Analisis Deskripsi Aplikasi

Secara umum aplikasi yang dibangun akan menggunakan platform Line sebagai media antara pengguna dengan aplikasi. Adapun deskripsi aplikasi adalah sebagai berikut :

1. Admin atau Developer melakukan pendaftaran atau login di website LINE Developers (https://developers.line.biz), setelah dapat mengakses selanjutnya Developer membuat chatbot LINE dan menyalin data
2. Channel Secret dan Channel Access Token kemudian diintegrasikan dengan layanan Platform as a Service
3. Developer akan diberi subdomain webhook Heroku untuk dapat terintegrasi dengan Chatbot LINE, url subdomain kemudian di atur pada LINE Developers.
4. Setelah terintegrasi chatbot dapat digunakan pada aplikasi pesan instan LINE.
5. User membuka aplikasi LINE untuk dapat berkomunikasi dengan chatbot.
6. User melakukan komunikasi dengan chatbot untuk mencari informasi perguruan tinggi yang ada di Kota Bandung
7. User dapat melihat informasi perguruan tinggi sesuai percakapan kepada chatbot.

**3.5 Analisis Arsitektur Aplikasi Yang Dibangun**

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

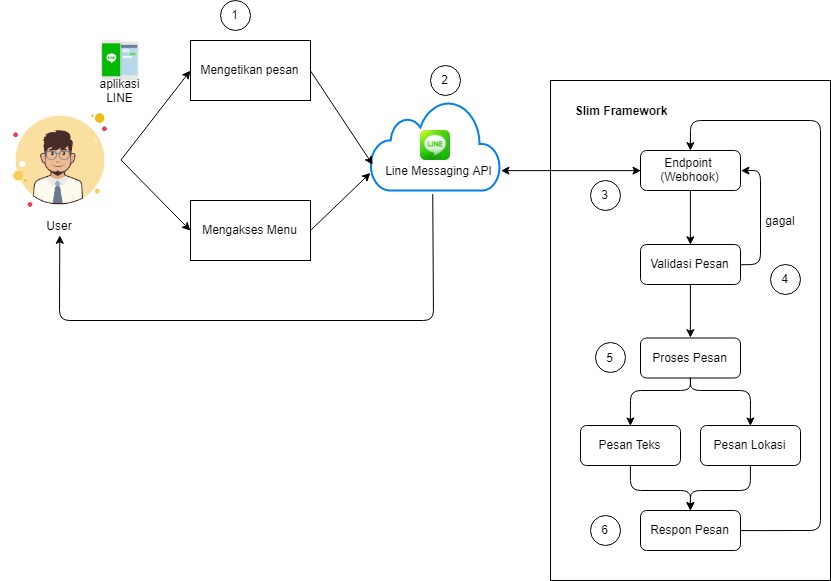


Gambar III.7 Analisis Arsitektur Aplikasi Yang Dibangun

Penjelasan untuk setiap bagian aplikasi adalah sebagai berikut:

1. Server Bot
2. Slim Framework

Slim framework digunakan untuk menangani request dan response yang dilakukan oleh user melalui Line Messaging API. User akan melakukan request berupa aksess 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 Slim Framework melalui satu endpoint yang dinamakan dengan webhook yang sebelumnya sudah diatur pada Line Developer Console.



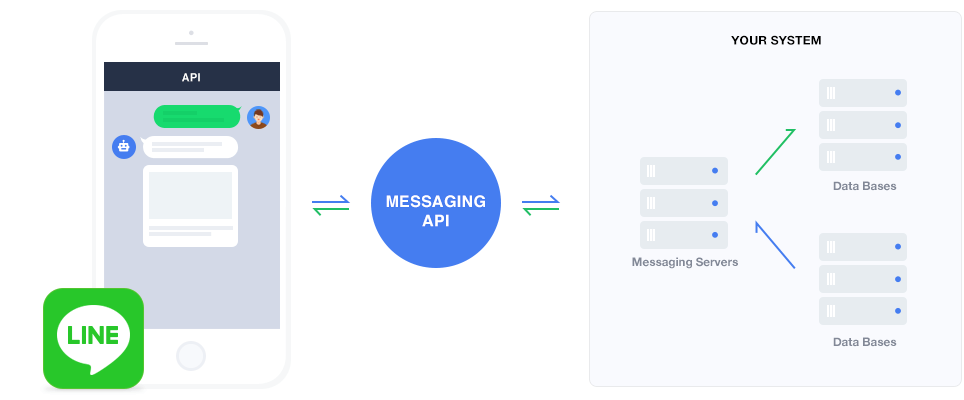
Gambar III.8 Alur Slim Framework

1. Database

Aplikasi membutuhkan akses terhadap data baik untuk melakukan pembacaan data atapun 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 Slim Framework untuk kemudian diteruskan kembali kepada user melalui Line Messaging API.

1. Aplikasi Pihak Ketiga
2. 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.9 Line Messaging API

1. Google Distance Matrix API

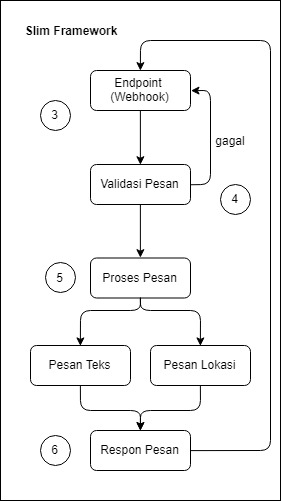
Google Distance Matrix API digunakan untuk bisa mengetahui lokasi terdekat antara lokasi pengguna dengan lokasi tempat Perguruan Tinggi. Pengguna akan mengirimkan lokasinya kepada bot, lokasi tersebut berupa lokasi saat ini berada ataupun lokasi yang dipilih oleh pengguna. Bot kemudian akan memproses permintaan tersebut lalu melakukan request kepada Google Distance Matrix API untuk menghitung lokasi terdekat antara lokasi pengguna dengan seluruh data lokasi tempat perguruan tinggi yang sebelumnya sudah ada di database.

**3.6** **Analisis Teknologi Yang Digunakan**

Dalam pembangunan aplikasi chatbot ini, penulis memanfaatkan teknologi Slim Framework, Deploy Heroku, aplikasi pihak ketiga Google Distance Matrix API serta Line Bot SDK.

1. Slim Framework

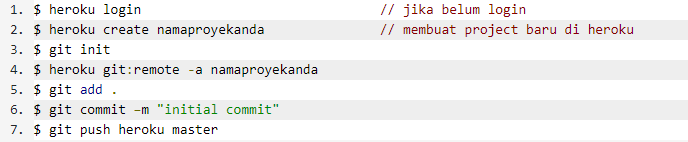
Slim framework digunakan untuk menangani request dan response yang dilakukan oleh user melalui Line Messaging API. Line messaging api akan mengirimkan request kepada aplikasi slim melalui route (webhook) yang sebelumnya sudah diatur.



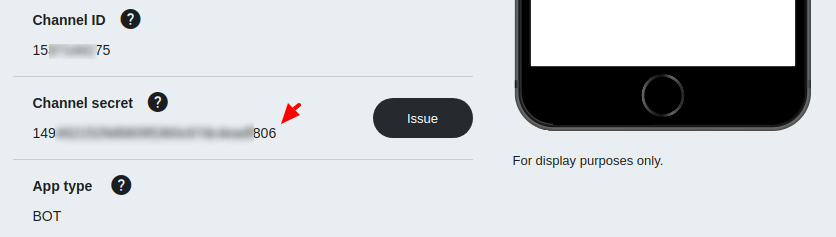
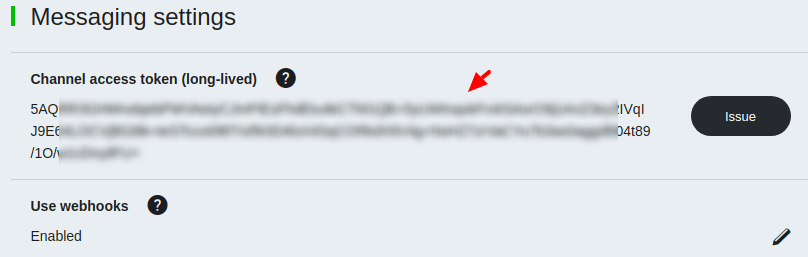
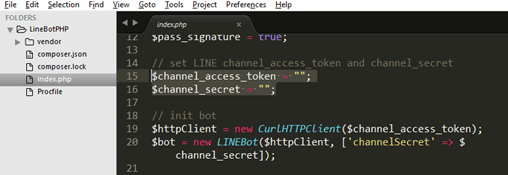
III.10 Alur Penggunaan Slim Framework

1. Deploy Heroku

Heroku adalah sebuah layanan ***PaaS* (**Platform as a Service**)** yang menyediakan web service  bagi para  developer aplikasi. Dengan menempatkan aplikasi mereka di ruang publik, diharapkan semua orang dapat mengaksesnya.

Setelah program sudah siap , maka selanjutnya mendeploy program ke heroku . Berikut adalah cara mendeploy program kita ke heroku.

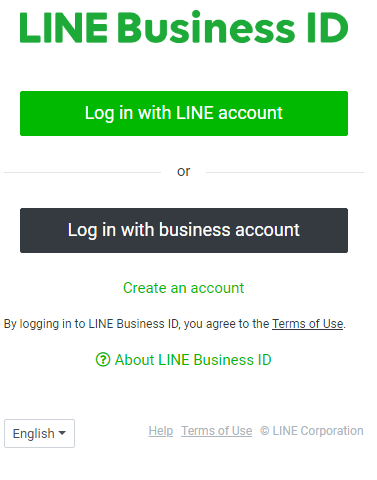
Setelah proses push berhasil, cek aplikasi yang dibuat di browser dengan alamat URL  [*https://namaproyekanda.herokuapp.com*](https://namaproyekanda.herokuapp.com)*.* Langkah selanjutnya menintegrasikan Line Channel terhadap webhook yang sudah dibuat sebelumnya.



Gambar III.4 Integrasi LINE Channel API

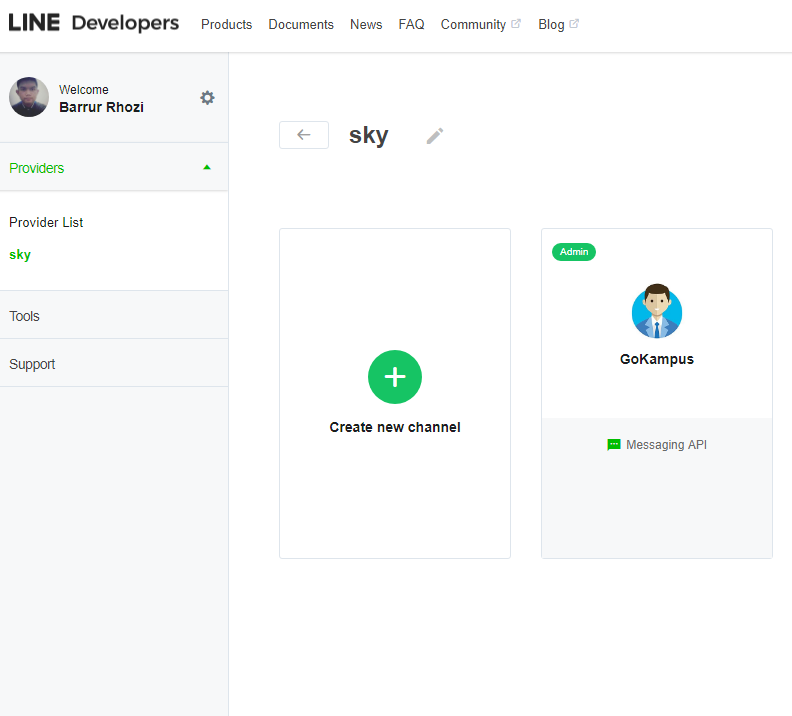
1. Line Messaging API

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.4 merupakan halaman masuk line developer.



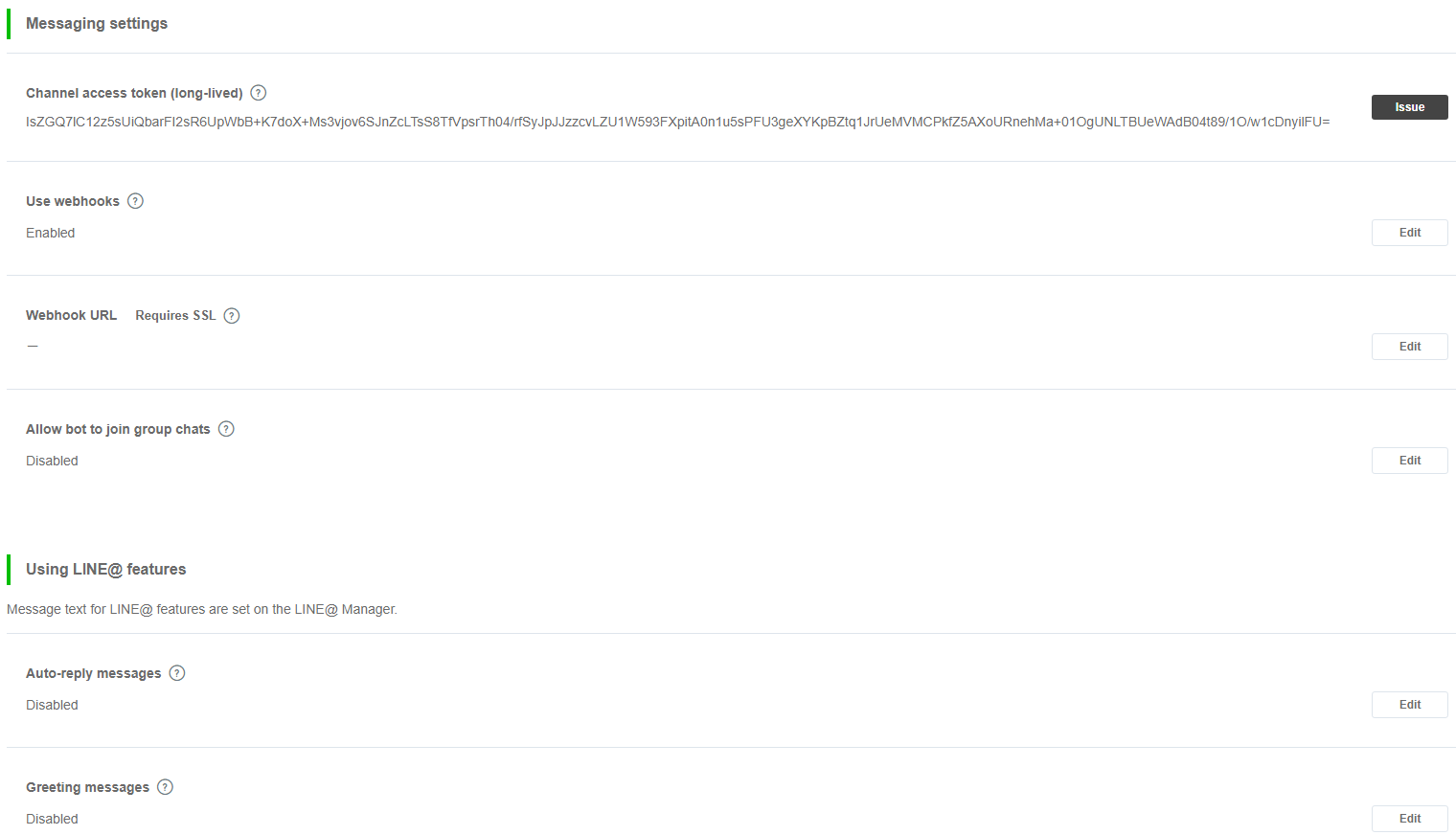
Gambar III.5 Halaman masuk line developer

Setelah masuk, developer dapat membuat channel messaging api untuk membuat bot. Gambar III.6 merupakan channel line messaging api.



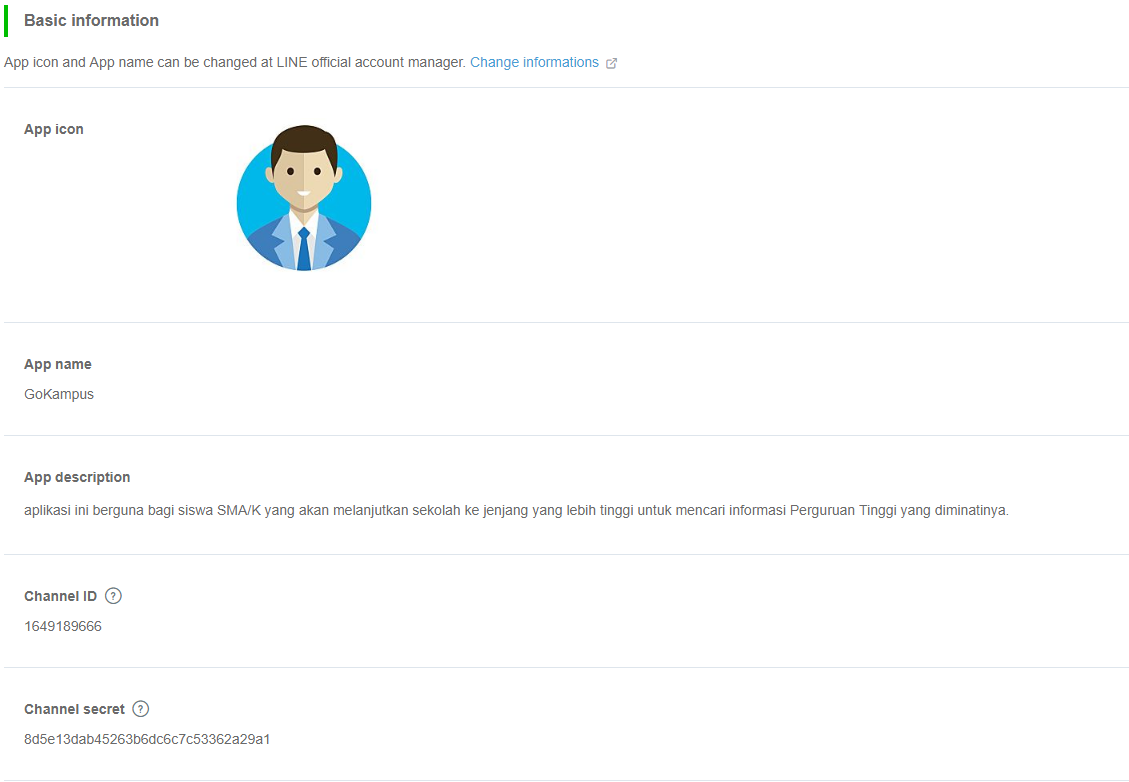
Gambar III.6 Channel Line Messaging API

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.7 menunjukkan pengaturan webhook pada situs line developer.



Gambar III.7 Pengaturan Webhook pada Line Developer

Developer harus mengatur token dan channel pada aplikasi bot untuk dapat mengembalikan respon berupa pesan kepada user. Gambar III.8 menunjukkan pengaturan umum yang dapat diatur oleh developer.



Gambar III.8 Pengaturan umum bot pada line developer

## 3.7 Analisis Spesifikasi Kebutuhan Perangkat Lunak

Berdasarkan hasil analisis masalah, analisis sistem aplikasi dan analisis arsitektur sistem didapatlah spesifikasi kebutuhan perangkat lunak yang dibangun dibagi menjadi dua kebutuhan yaitu spesifikasi kebutuhan nonfungsional dan spesifikasi kebutuhan fungsional.

**3.7.1 Analisis Kebutuhan Non Fungsional**

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

1. Analisis Perangkat Keras

Perangkat keras minimum yang harus diperlukan untuk mendukung kinerja aplikasi dapat dilihat pada tabel berikut :

Tabel III.1 Analisis Perangkat Keras

|  |  |  |
| --- | --- | --- |
| No | Perangkat | Spesifikasi |
| 1 | Prosesor | 1 CPU |
| 2 | Hardisk | 16 GB |
| 3 | RAM | 1 GB |

1. Analisis Perangkat Lunak

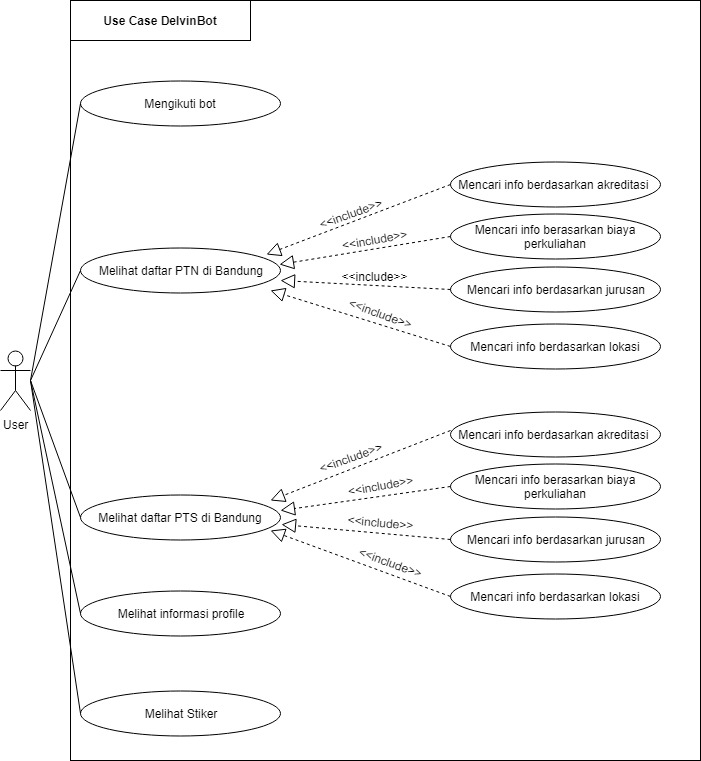
Berikut adalah spesifikasi perangkat lunak yang digunakan dalam membangun dan mengimplementasikan aplikasi chatbot sebagai berikut :

1. User (Client)
2. Aplikasi Pesan Instan (LINE v9.2.2)
3. Bot (Server)
4. Sistem Operasi Windows (7,8,10)
5. Database Server (MySQL 5.6.1)
6. Slim Framework
7. PHP (PHP v7.1.3)
8. Composer

**3.7.2 Analisis Kebutuhan Fungsional**

Identifikasi aktor dapat dilakukan dalam analisis berorientasi objek dengan menggunakan UML yaitu menentukan aktor atau user sistem. Aktor dalam konteks UML menampilkan peran pemain atau sesuatu diluar sistem yang dikembangkan dapat berupa perangkat keras, user, sistem yang lain dan sebagainya. Diagram yang akan digambarkan pada bagian ini adalah use case diagram, class diagram, dan sequence diagram.

1. **Use Case Diagram**



Gambar III.9 Use Case Diagram

Definisi aktor dari pembangunan perangkat lunak ini dapat dilihat pada tabel berikut:

Tabel II.2 Definisi Actor

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1 | User | *User* hanya dapat mengakses fiturnya sendiri |

* 1. **Definisi Use Case**

Definisi *Use case* dari pembangunan perangkat lunak ini dapat dilihat pada tabel berikut:

Tabel I.3 Definisi Use Case

|  |  |  |
| --- | --- | --- |
| **No.** | **Use Case** | **Deskripsi** |
| 1 | Mengikuti Bot | Mengikuti bot di aplikasi LINE |
| 2 | Melihat daftar PTN di Bandung | Melihat daftar PTN di kota Bandung |
| 3 | Melihat daftar PTS di Bandung | Melihat daftar PTS di kota Bandung |
| 4 | Melihat informasi profile | Melihat informasi profile |
| 5 | Mencari info berdasarkan akreditasi | Menampilkan informasi PTN atau PTS berdasarkan akreditasi |
| 6 | Mencari info berdasarkan biaya perkuliahan | Menampilkan informasi PTN atau PTS berdasarkan biaya perkuliahan |
| 7 | Mencari info berdasarkan jurusan | Menampilkan informasi PTN atau PTS berdasarkan jurusan |
| 8 | Mencari info berdasarkan lokasi | Menampilkan informasi PTN atau PTS berdasarkan lokasi |

* 1. **Use Case Description**

Keterangan lebih lengkap mengenai masing-masing usecase yang terdapat pada use case diagram ditulis dalam use case description yang terdapat pada Tabel.

Tabel III.4 Use Case Description Mengikuti Bot

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Mengikuti Bot | |
| **Related Requirements** |  | |
| **Goal in Context** | Mengikuti akun bot | |
| **Preconditions** | User sudah mempunyai dan masuk ke dalam aplikasi Line | |
| **Successful end Condition** | User berhasil mengikuti akun bot | |
| **Failed end Condition** | User gagal mengikuti akun bot | |
| **Primary Actors** | User | |
| **Secondary Actors** |  | |
| **Trigger** | User Meminta mengikuti bot | |
| **Main Flow** | **Step** | **Action** |
|  | 1 | User meminta untuk mengikuti bot |
|  | 2 | Bot memeriksa apakah pengguna sudah mengikuti bot |
|  | 3 | User berhasil mengikuti bot |
| **Extensions** | **Step** | **Branching Action** |
|  | 2.1 | Bot gagal memeriksa data pengguna |
|  | 2.2 | User gagal mengikuti bot |

Tabel III.5 Use Case Description Melihat Daftar PTN

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Melihat Daftar PTN | |
| **Related Requirements** |  | |
| **Goal in Context** | Melihat Daftar PTN di kota Bandung | |
| **Preconditions** | User sudah mengikuti bot | |
| **Successful end Condition** | User berhasil melihat daftar PTN di Kota Bandung | |
| **Failed end Condition** | User gagal melihat daftar PTN di Kota Bandung | |
| **Primary Actors** | User | |
| **Secondary Actors** |  | |
| **Trigger** | User meminta melihat daftar PTN di Kota Bandung | |
| **Main Flow** | **Step** | **Action** |
|  | 1 | User meminta untuk melihat daftar PTN |
|  | 2 | Bot memeriksa apakah pengguna sudah mengikuti bot |
|  | 3 | User berhasil melihat daftar PTN |
| **Extensions** | **Step** | **Branching Action** |
|  | 2.1 | Bot gagal memeriksa data PTN |
|  | 2.2 | User gagal melihat daftar PTN |

Tabel III.6 Use Case Description Melihat Daftar PTS

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Melihat Daftar PTS | |
| **Related Requirements** |  | |
| **Goal in Context** | Melihat Daftar PTS di kota Bandung | |
| **Preconditions** | User sudah mengikuti bot | |
| **Successful end Condition** | User berhasil melihat daftar PTS di Kota Bandung | |
| **Failed end Condition** | User gagal melihat daftar PTS di Kota Bandung | |
| **Primary Actors** | User | |
| **Secondary Actors** |  | |
| **Trigger** | User meminta melihat daftar PTS dii Kota Bandung | |
| **Main Flow** | **Step** | **Action** |
|  | 1 | User meminta untuk melihat daftar PTS |
|  | 2 | Bot memeriksa apakah pengguna sudah mengikuti bot |
|  | 3 | User berhasil melihat daftar PTS |
| **Extensions** | **Step** | **Branching Action** |
|  | 2.1 | Bot gagal memeriksa data PTS |
|  | 2.2 | User gagal melihat daftar PTS |

Tabel III.7 Use Case Description Melihat Informasi Profile

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Melihat Informasi Profile | |
| **Related Requirements** |  | |
| **Goal in Context** | Melihat Informasi Profile User | |
| **Preconditions** | User sudah mengikuti bot | |
| **Successful end Condition** | User berhasil melihat informasi profile | |
| **Failed end Condition** | User gagal melihat informasi profile | |
| **Primary Actors** | User | |
| **Secondary Actors** |  | |
| **Trigger** | User meminta melihat informasi profile | |
| **Main Flow** | **Step** | **Action** |
|  | 1 | User meminta untuk melihat informasi profile |
|  | 2 | Bot memeriksa data profile user |
|  | 3 | User berhasil melihat informasi profile |
| **Extensions** | **Step** | **Branching Action** |
|  | 2.1 | Bot gagal memeriksa data profile user |
|  | 2.2 | User gagal melihat informasi profile |

Tabel III.8 Use Case Description Mencari Informasi Berdasarkan Akreditasi

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Mencari informasi Akreditasi melalui menu | |
| **Related Requirements** |  | |
| **Goal in Context** | Mencari informasi berdasarkan akreditasi | |
| **Preconditions** | User sudah mengikuti bot | |
| **Successful end Condition** | User berhasil mencari informasi Akreditasi melalui menu | |
| **Failed end Condition** | User gagal mencari informasi Akreditasi melalui menu | |
| **Primary Actors** | User | |
| **Secondary Actors** |  | |
| **Trigger** | User meminta mencari informasi Akreditasi melalui menu | |
| **Main Flow** | **Step** | **Action** |
|  | 1 | User meminta untuk mencari informasi Akreditasi melalui menu |
|  | 2 | Bot memeriksa informasi Akreditasi yang diakses user |
|  | 3 | User berhasil melihat informasi Akreditasi |
| **Extensions** | **Step** | **Branching Action** |
|  | 2.1 | Bot gagal memeriksa informasi Akreditasi yang diminta |
|  | 2.2 | User gagal mencarti informasi Akreditasi |

Tabel III.9 Use Case Description Mencari Informasi Berdasarkan Biaya Perkuliahan

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Mencari informasi biaya perkuliahan melalui menu | |
| **Related Requirements** |  | |
| **Goal in Context** | Mencari informasi berdasarkan biaya perkuliahan | |
| **Preconditions** | User sudah mengikuti bot | |
| **Successful end Condition** | User berhasil mencari informasi biaya perkuliahan melalui menu | |
| **Failed end Condition** | User gagal mencari informasi biaya perkuliahan melalui menu | |
| **Primary Actors** | User | |
| **Secondary Actors** |  | |
| **Trigger** | User meminta mencari informasi biaya perkuliahan melalui menu | |
| **Main Flow** | **Step** | **Action** |
|  | 1 | User meminta untuk mencari informasi biaya perkuliahan melalui menu |
|  | 2 | Bot memeriksa informasi biaya perkuliahan yang diakses user |
|  | 3 | User berhasil melihat informasi biaya perkuliahan |
| **Extensions** | **Step** | **Branching Action** |
|  | 2.1 | Bot gagal memeriksa informasi biaya perkuliahan yang diminta |
|  | 2.2 | User gagal mencarti informasi biaya perkuliahan |

Tabel III.10 Use Case Description Mencari Informasi Berdasarkan Jurusan

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Mencari informasi Jurusan melalui menu | |
| **Related Requirements** |  | |
| **Goal in Context** | Mencari informasi berdasarkan jurusan | |
| **Preconditions** | User sudah mengikuti bot | |
| **Successful end Condition** | User berhasil mencari informasi jurusan melalui menu | |
| **Failed end Condition** | User gagal mencari informasi jurusan melalui menu | |
| **Primary Actors** | User | |
| **Secondary Actors** |  | |
| **Trigger** | User meminta mencari informasi jurusan melalui menu | |
| **Main Flow** | **Step** | **Action** |
|  | 1 | User meminta untuk mencari informasi jurusan melalui menu |
|  | 2 | Bot memeriksa informasi jurusan yang diakses user |
|  | 3 | User berhasil melihat informasi jurusan |
| **Extensions** | **Step** | **Branching Action** |
|  | 2.1 | Bot gagal memeriksa informasi jurusan yang diminta |
|  | 2.2 | User gagal mencarti informasi jurusan |

Tabel III.11 Mencari Informasi Berdasarkan Lokasi

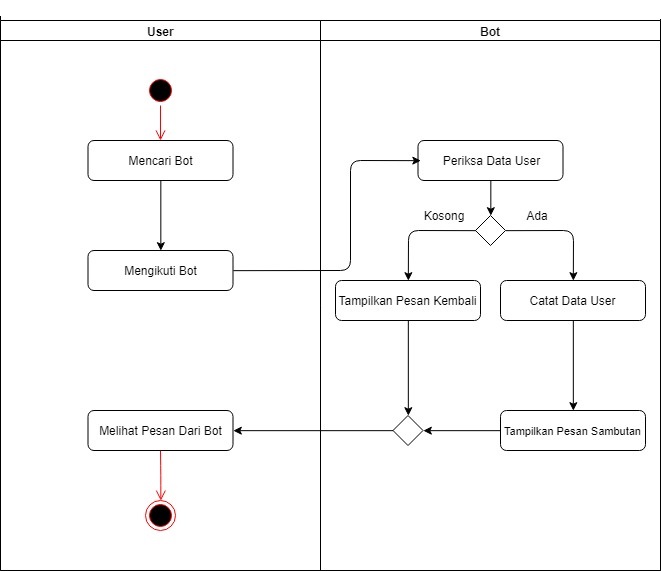
|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Mencari informasi lokasi melalui menu | |
| **Related Requirements** |  | |
| **Goal in Context** | Mencari informasi berdasarkan lokasi | |
| **Preconditions** | User sudah mengikuti bot | |
| **Successful end Condition** | User berhasil mencari informasi lokasi melalui menu | |
| **Failed end Condition** | User gagal mencari informasi lokasi melalui menu | |
| **Primary Actors** | User | |
| **Secondary Actors** |  | |
| **Trigger** | User meminta mencari informasi lokasi melalui menu | |
| **Main Flow** | **Step** | **Action** |
|  | 1 | User meminta untuk mencari informasi lokasi melalui menu |
|  | 2 | Bot memeriksa informasi lokasi yang diakses user |
|  | 3 | User berhasil melihat informasi lokasi |
| **Extensions** | **Step** | **Branching Action** |
|  | 2.1 | Bot gagal memeriksa informasi lokasi yang diminta |
|  | 2.2 | User gagal mencarti informasi lokasi |

1. Activity Diagram

Activity Diagram adalah diagram yang menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Aktivitas pengguna di dalam aplikasi pada setiap use case dimodelkan melalui activity diagram .

* 1. Activity Diagram Mengikuti Bot

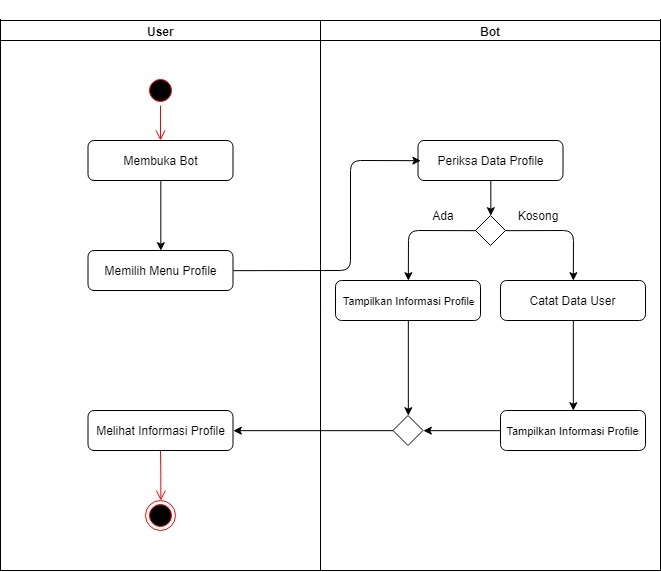
Activity diagram Mengikuti Bot menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mengikuti bot. Gambar III.10 menunjukkan activity diagram mengikuti bot :



Gambar III.10 Mengikuti Bot

* 1. Activity Diagram Melihat Informasi Profile

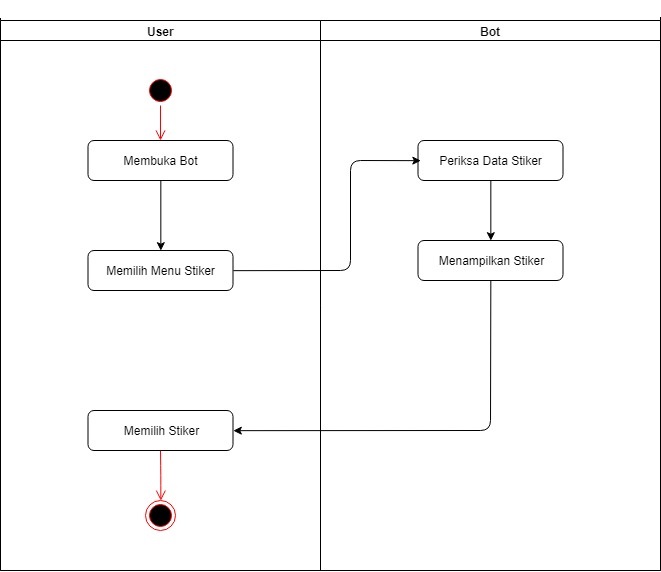
Activity diagram Melihat Informasi Profil menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna melihat informasi profil. Gambar III.11 menunjukkan activity diagram melihat informasi profil:



Gambar III.11 Activity Diagram Melihat informasi profil

* 1. Activity Diagram Melihat Stiker

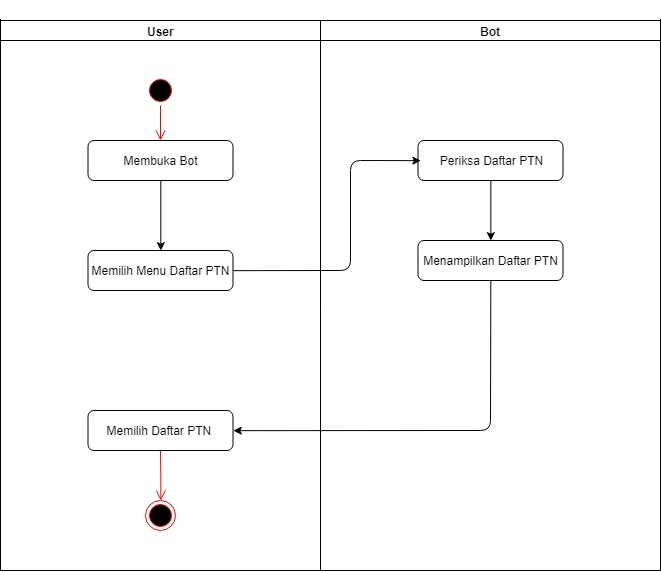
Activity diagram Melihat Stiker menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna melihat stiker. Gambar III.12 menunjukkan activity diagram melihat informasi stiker :



Gambar III.12 Activity Diagram Melihat Stiker

* 1. Activity Diagram Melihat Daftar PTN di Kota Bandung

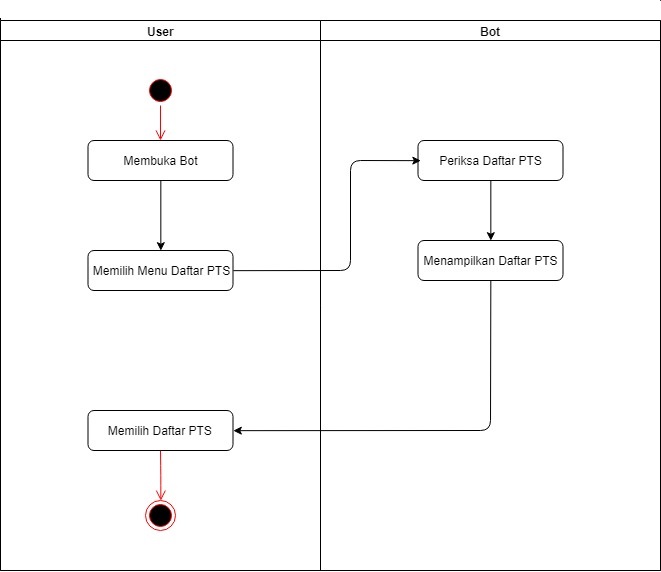
Activity diagram Melihat Daftar PTN di Kota Bandung menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna melihat Daftar PTN di Kota Bandung. Gambar III.13 menunjukkan activity diagram melihat Daftar PTN di Kota Bandung:



Gambar III.13 Activity diagram melihat Daftar PTN di Kota Bandung

* 1. Activity Diagram Melihat Daftar PTS di Kota Bandung

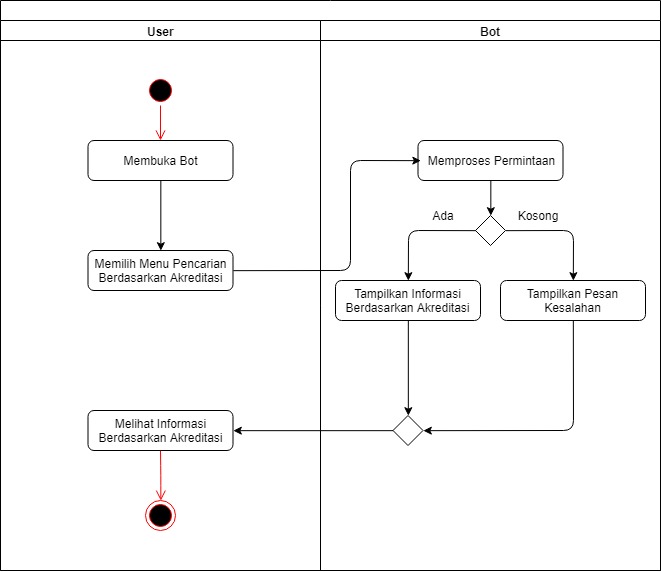
Activity diagram Melihat Daftar PTN di Kota Bandung menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna melihat Daftar PTN di Kota Bandung. Gambar III.13 menunjukkan activity diagram melihat Daftar PTN di Kota Bandung:



Gambar III.13 Activity diagram melihat Daftar PTS di Kota Bandung

* 1. Activity Diagram Mencari Info Berdasarkan Akreditasi

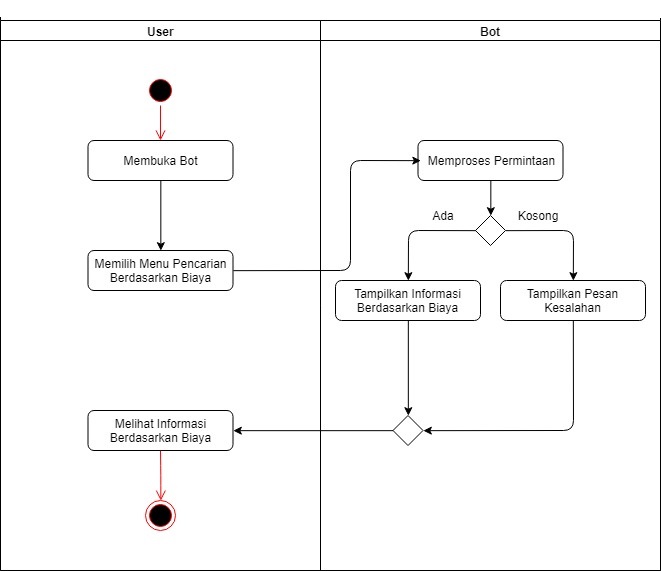
Activity diagram Mencari informasi berdasarkan akreditasi yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi kampus berdasarkan akreditasi. Gambar III.14 menunjukkan activity diagram mencari informasi berdasarkan akreditasi:



Gambar III.14 activity diagram mencari informasi berdasarkan akreditasi

* 1. Activity Diagram Mencari Info Berdasarkan Biaya Perkuliahan

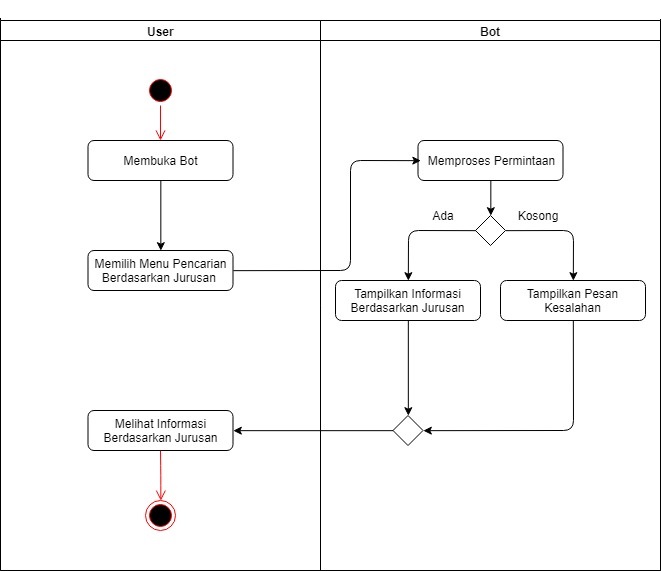
Activity diagram Mencari informasi berdasarkan biaya perkuliahan yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi kampus berdasarkan biaya perkuliahan. Gambar III.15 menunjukkan activity diagram mencari informasi berdasarkan biaya perkuliahan :



Gambar III.15 activity diagram mencari informasi berdasarkan biaya perkuliahan

* 1. Activity Diagram Mencari Info Berdasarkan Jurusan

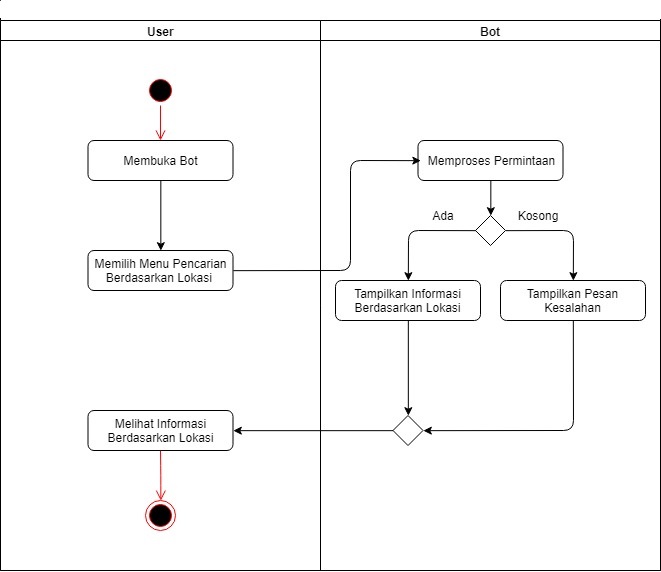
Activity diagram Mencari informasi berdasarkan jurusan yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi kampus berdasarkan jurusan. Gambar III.16 menunjukkan activity diagram mencari informasi berdasarkan biaya jurusan :



Gambar III.16 activity diagram mencari informasi berdasarkan biaya jurusan

* 1. Activity Diagram Mencari Info Berdasarkan Lokasi

Activity diagram Mencari informasi berdasarkan lokasi yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi kampus berdasarkan lokasi. Gambar III.17 menunjukkan activity diagram mencari informasi berdasarkan lokasi :



Gambar III.17 activity diagram mencari informasi berdasarkan biaya jurusan

1. Class Diagram
2. Sequence Diagram
3. Perancangan Data
4. Perancangan Struktur Menu
5. Perancangan Antarmuka
6. Perancangan Pesan Antarmuka
7. Perancangan Jaringan Semantik Antarmuka