**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 sederajat dalam mencari informasi Perguruan Tinggi menggunakan cara mengunjungi setiap website Perguruan Tinggi atau mendatangi tempat Perguruan Tinggi satu persatu yang diinginkan untuk mendapatkan informasinya. Tentunya penyampaian atau pencarian informasi tersebut kurang efisien , karena jika kita akan mencari infomasi dari kampus lain harus mengunjungi website atau mengunjungi tempat nya satu persatu , hal ini kurang efisien.**

**Berdasarkan data kuisoner terhadap 100 responden siswa SMA sederajat yang akan melanjutkan sekolahnya ke Perguruan Tinggi, terdapat 81% siswa yang masih mengalami kesulitan dalam menentukan Perguruan Tinggi yang diminatinya. Lalu berdasarkan data kuisoner yang sama, faktor peminat siswa SMA sederajat dalam mencari Perguruan Tinggi diantaranya berdasarkan 85% Akreditasi, 52% Lokasi, 76% Biaya perkuliahan serta 75 % Jurusan yang diminatinya .**

**Oleh sebab itu dibutuhkan sebuah media penyampaian informasi yang lebih interaktif dan lebih lengkap 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. Dengan cara menggunakan chatbot memanfaatkan media sosial LINE atau LINE chatbot dengan menggunakan metode forward chaining.**

**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, gambar [4]. Pemanfaatan LINE Chatbot merupakan cara yang interaktif dalam mencari informasi Perguruan Tinggi.**

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 chatbot LINE ini akan berguna bagi siswa SMA sederajat 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 sederajat agar tidak kesulitan dalam mencari informasi Perguruan Tinggi yang ada di kota Bandung dengan menggunakan metode forward chaining yang dapat terintegrasi dengan LINE Chatbot . Aplikasi ini dibangun agar siswa SMA sederajat dapat mencari tahu tentang informasi kampus yang diinginkannya dengan cara tanya jawab kepada chatbot atau memilih menu yang sudah disediakan. 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 sederajat mengalami kesulitan dalam mencari informasi seputar Perguruan Tinggi diminatinya yang ada di Kota Bandung.
2. Belum adanya aplikasi media informasi Perguruaan Tinggi pada media sosial LINE.

# Maksud dan Tujuan

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

1. Membantu siswa sekolah SMA sederajat dalam mencari informasi seputar Perguruan Tinggi yang diminatinya yang ada di Kota Bandung.
2. Menghasilkan media informasi Perguruan Tinggi pada media sosal Line.

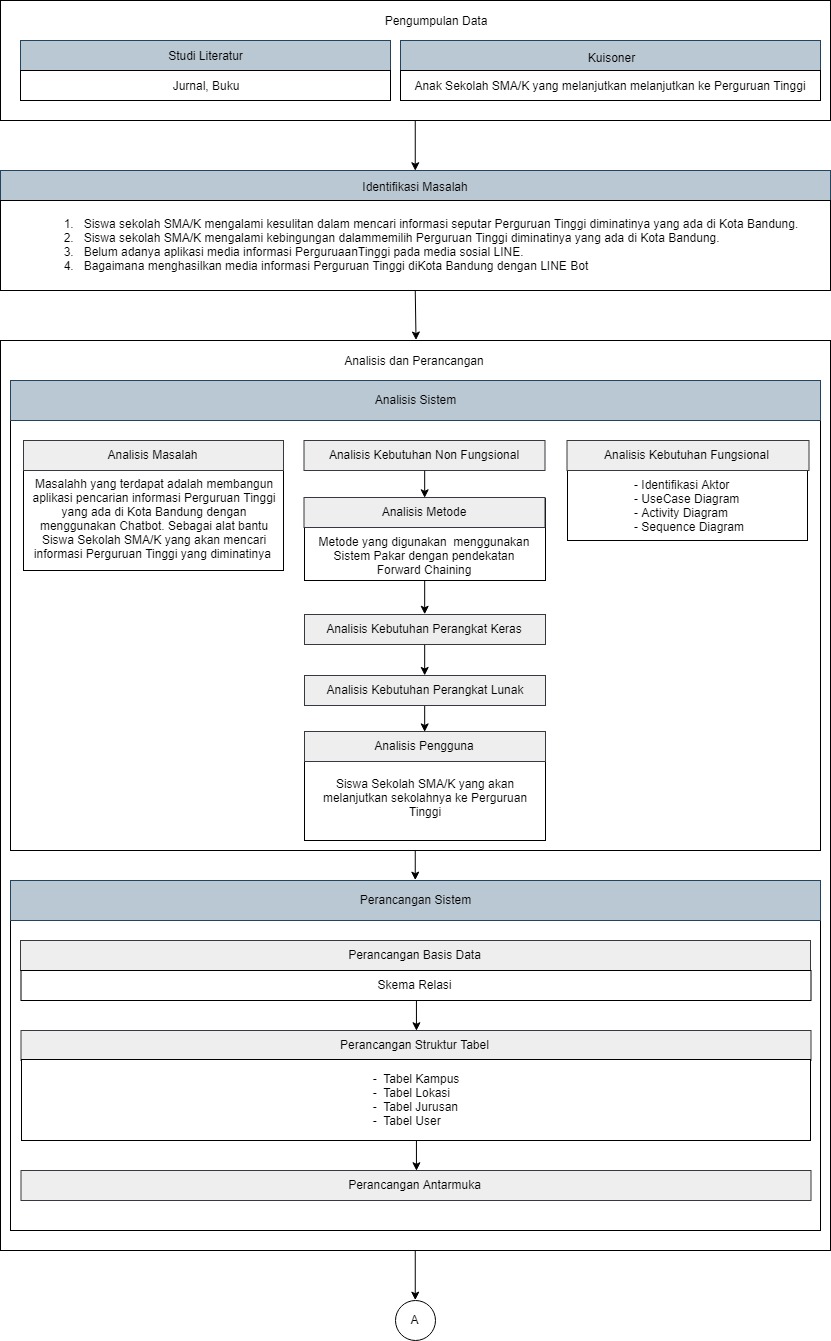
# Batasan Masalah

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

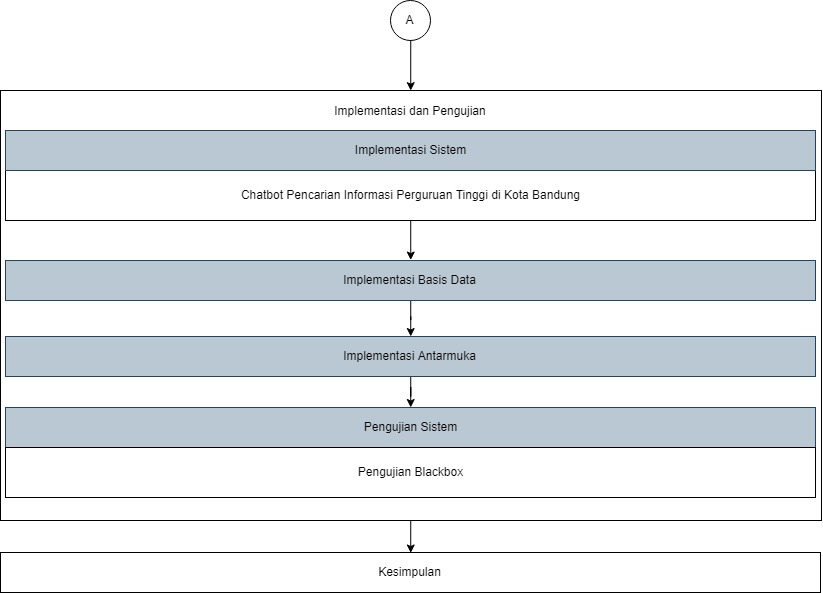
1. Informasi data Perguruan Tinggi masih di sekitar Kota Bandung yang berstatus Negeri dan Swasta. Seperti ITB, UNPAD, UNISBA, UNIKOM, ITHB, ITENAS, Tel-U dst.
2. Informasi jurusan yang ditampilkan hanya S1
3. Informasi biaya jurusan yang ditampilkan tiap semester sebagian dihitung berdasarkan rata-rata dari total biaya 8 semester .
4. Informasi biaya jurusan yang sesuai dapat dilihat di brosur yang sudah disediakan.
5. Informasi jurusan, biaya semester yang ditampilkan hanya kelas reguler dan bukan kelas karyawan.
6. Pengguna adalah siswa SMA sederajat yang akan melanjutkan ke Perguruan Tinggi di kota Bandung.
7. Metode yang digunakan pada penelitian ini adalah metode forward chaining.
8. Penelitian ini tidak terlalu mendalami mengenai AI
9. Layanan pada aplikasi yang akan dibangun yakni: pencarian kampus, pencarian lokasi kampus, pencarian biaya semester, dan pencarian lokasi.
10. Masukan (input) dapat diberikan melalui teks, location dan rich menu.
11. Keluaran (output) yang diberikan oleh bot berupa teks dan gambar dalam bentuk menu dan konten
12. Metode perancangan dan analisis perangkat lunak yang digunakan adalah metode Object Oriented Analysis & Design (OOAD) dengan menggunakan pemodelan Unified Modelling Language (UML).

# Metodologi Penelitian

Metodologi penelitian yang dimaksud adalah segala hal yang berhubungan dengan metode-metode yang digunakan dalam membangun aplikasi ini dengan cara melakukan pendekatan terhadap metode-metode yang telah ada. Adapun pelaksanaannya diatur dalam beberapa tahap, yaitu :



Gambar I.2 Alur Penelitian



Gambar I.3 Alur Penelitian lanjutan 1

Berikut merupakan penjelasan tahapan penelitian berdasarkan gambar pada alur metodologi penelitian yang akan dilakukan :

## Metode Pengumpulan Data

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

1. Studi Literatur

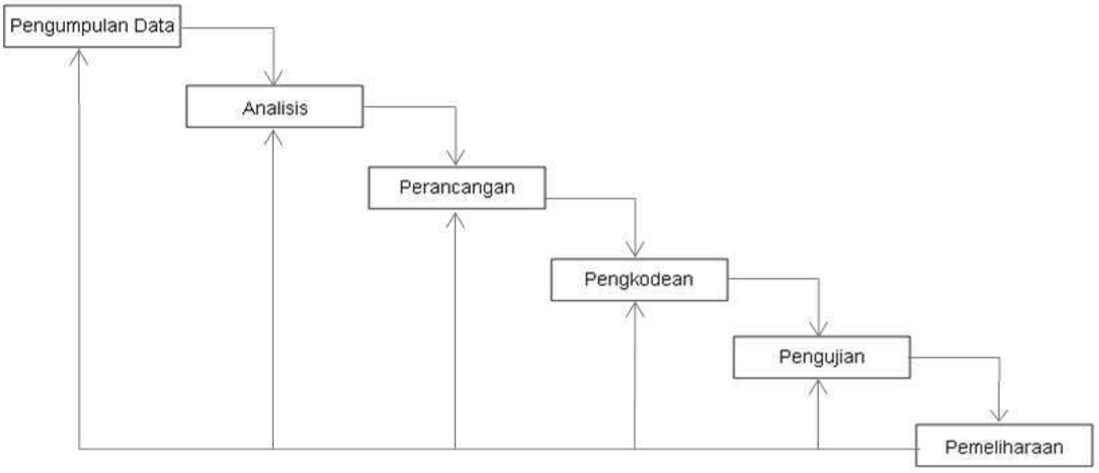
Pengumupulan data dengan cara mengumpulkan literatur – literatur yang berkaitan dengan judul penelitian

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 biaya, penambahan kampus 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 Perguruan Tinggi**

Perguruan tinggi merupakan kelanjutan pendidikan menengah yang diselenggarakan untuk mempersiapkan peserta didik untuk menjadi anggota masyarakat yang memiliki kemampuan akademis dan profesional yang dapat menerapkan, mengembangkan dan menciptakan ilmu pengetahuan, teknologi dan kesenian (UU 2 tahun 1989, pasal 16, ayat (1)).

Tujuan pendidikan tinggi adalah :

1. Mempersiapkan peserta didik menjadi anggota masyarakat yang memiliki kemampuan akademik dan profesional yang dapat menerapkan, mengembangkan dan menciptakan ilmu pengetahuan, teknologi dan kesenian.
2. Mengembangkan dan menyebar luaskan ilmu pengetahuan, teknologi dan kesenian serta mengoptimalkan penggunaannya untuk meningkatkan taraf hidup masyarakat dan memperkaya kebudayaan nasional ( UU 2 tahun 1989, Pasal 16, Ayat (1) ; PP 30 Tahun 1990, Pasal 2, Ayat (1) ).

**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.

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. [8]

**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

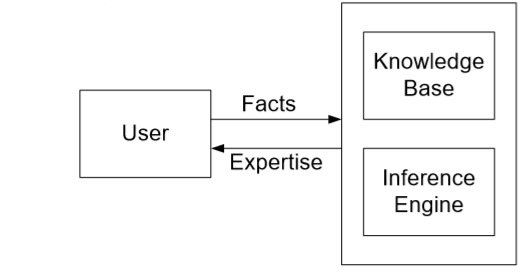
Hal yang unik dari sistem pakar adalah kemampuan untuk menjelaskan dimana keahlian tersimpan dalam basis pengetahuan. Kemampuan komputer untuk mengambil kesimpulan dilakukan oleh komponen yang dikenal dengan mesin inferensi yaitu meliputi prosedur tentang pemecahan masalah.

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

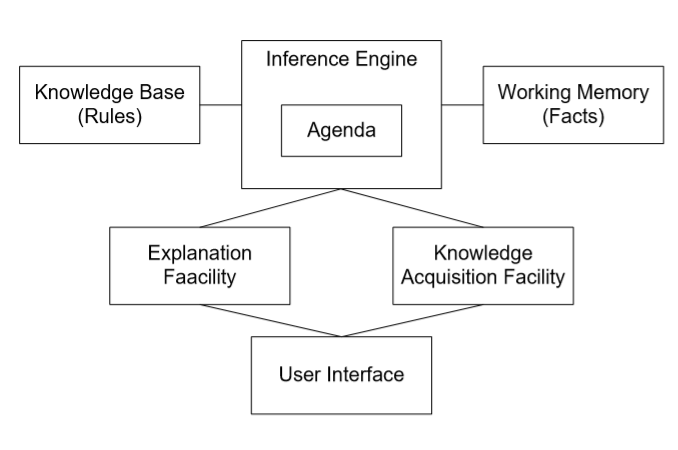
Keunikan lain dari sistem pakar adalah kemampuan dalam menjelaskan atau memberi saran rekomendasi serta juga menjelaskan mengapa beberapa tindakan saran tidak direkomendasikan.



Gambar II.2 Konsep Dasar Fungsi Sistem Pakar

Gambar II.2 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 pengetahuan pengetahuan yang dibutuhkan untuk memahami, memformulasikan dan menyelesaikan masalah.

**2.3.3 Basis Aturan Sistem Pakar**

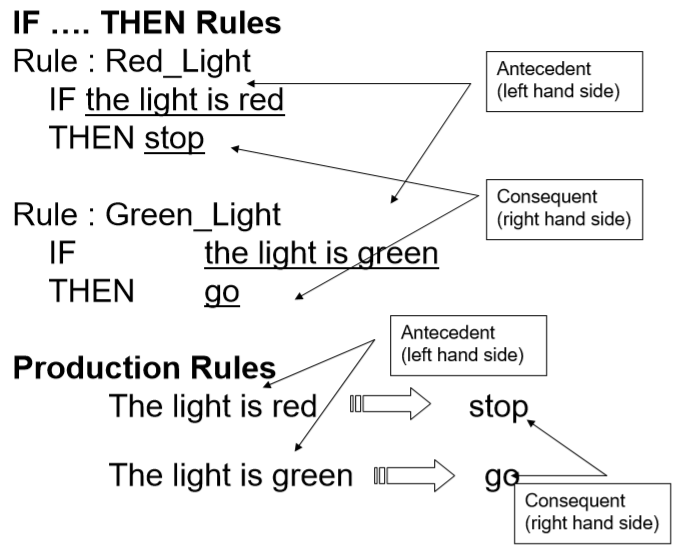
Adapun basis aturan dalam Sistem Pakar adalah sebagai berikut :

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 Basis Aturan Sistem Pakar

**2.3.3 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.

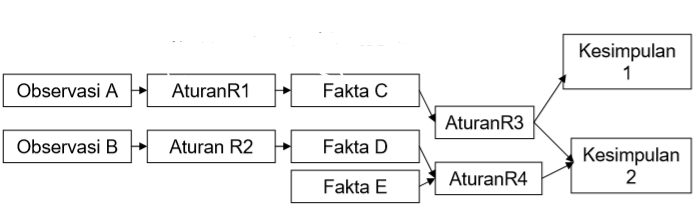
**2.4 Penarikan Kesimpulan (Inferencing)**

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

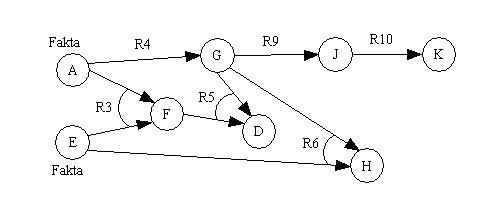
R7 : if C and H then I

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

Tabel II.1 Hasil Metode 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.

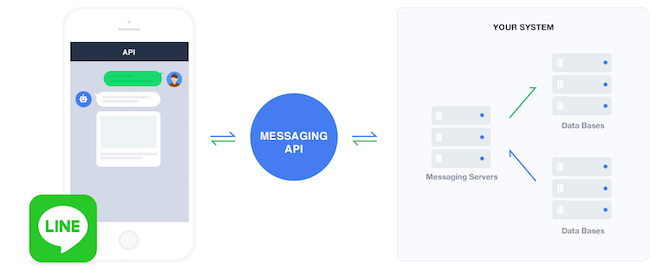
Bot biasa digunakan dalam IRC (Internet Chat Groups) di mana bot tersebut melakukan berbagai fungsi seperti menyapa partisipan baru dan memonitor penggunaan kata-kata yang tidak pantas. Sekarang ini, bot dapat mengumpulkan informasi, mengolah informasi tersebut, dan memberikan respon terhadap pengguna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

**2.5.1 Cara Kerja Chatbot**

Pada dasarnya bot 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 “Tolong carikan saya kampus Negeri”. Disini anda hanya perlu memberikan perintah atau melakukan sebuah percakapan baik itu dalam bentuk teks maupun menu, dan bot akan mencari data yang anda butuhkan sesuai dengan kata kunci yang anda berikan.

**2.5.2 LINE Messaging API**

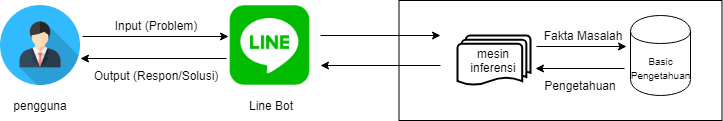
Api (*Application Programming Interface*) merupakan kumpulan dari fungsi-fungsi yang digunakan oleh *programmer* untuk membangun sebuah aplikasi. Pada line *chatbot* ini, api yang digunakan yaitu *messaging api* pada line yang berfungsi untuk saling mengirimkan data antar aplikasi line dengan sistem yang digunakan yaitu web server melalui server line [10]



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 II.8 dibawah ini :

Gambar II.8 Skema Knowledge Chatbot

Dari gambar II.8 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 suatu keywoard atau database.
4. Program chat bot memberikan respon ke pengguna.

**2.5.4 Fungsi Strpos**

Fungsi strpos() adalah fungsi bawaan PHP yang bisa digunakan untuk mencari posisi sebuah karakter atau sebuah string di dalam string lainnya. Hasil akhir fungsi ini adalah angka yang menunjukkan posisi karakter/string yang ingin dicari.

**2.6 Metode Jaro-Winkler Distance**

Jaro-Winkler *distance* merupakan varian dari Jaro *distance* metrik yaitu sebuah algoritma untuk mengukur kesamaan antara dua string, biasanya algoritma ini digunakan di dalam pendeteksian duplikat. Semakin tinggi Jaro-Winkler distance untuk dua string, semakin mirip dengan string tersebut. Jaro-Winkler distance terbaik dan cocok untuk digunakan dalam perbandingan string singkat seperti nama orang.

Rumus :

Dimana :

= Jumlah karakter yang sama persis

= Panjang string 1

= Panjang string 2

= Jumlah transposisi

Menghitung Jarak

Menghitung *Jaro-Winkler Distance*

Dimana :

= *Jaro* *distance* untuk string dan string

= Panjang *prefix* (panjang karakter yang sama sebelum ditemukan

ketidaksamaan) nilai maksimum 4 karakter

= Konstanta *scaling factor* (Nilai standar untuk konstanta ini menurut *Winkler* adalah 𝑝 = 0.1)

**2.6 Web Server**

**2.6.1 Hosting**

Hosting adalah layanan berbasis internet yang menyediakan sumber daya atau *resource* untuk disewakan sebagai tempat menyimpan data atau tempat menjalankan aplikasi atau website ditempat terpusat yang disebut dengan *server* sehingga memungkinkan organisasi atau individu menempatkan informasi di internet berupa *HTTP, FTP, EMAIL*, atau *DNS*. Dari pengertian hosting diatas dapat disimpulkan bahwa hosting adalah ibarat sebuah rumah yang disewakan dengan ukuran tertentu dan setiap bangunan rumah pasti memiliki ukuran yang berbeda. Masing-masing penyedia jasa web host menawarkan jenis hosting yang berbeda agar sesuai dengan keinginan dan kebutuhan klien. Berikut contoh hosting yang sering digunakan:

*Shared Hosting*

*Shared hosting* merupakan tipe web hosting yang paling banyak digunakan. Paket ini menjadi pilihan yang pas untuk blog dan website sederhana. Dengan shared hosting, dapat berbagi server dengan klien lainnya di provider hosting yang sama. Website yang dionlinekan di server yang sama berbagi semua *resource, seperti memori, computing power, disk space,* dan lain-lain.

Kelebihan:

Murah

Mudah digunakan oleh pemula

Pre-configured server

Control panel yang mudah digunakan

Maintenance dan administrasi server dilakukan oleh penyedia hosting

Kekurangan:

* 1. Sedikit atau sama sekali tidak dapat mengontrol konfigurasi server
  2. Apabila traffic tinggi di website pengguna lain, maka hal tersebut akan memengaruhi kinerja situs

*VPS (Virtual Private Server) Hosting*

Dengan menggunakan VPS (*Virtual Private Server*) hosting, masih berbagi server dengan pengguna lainnya. Akan tetapi, penyedia web hosting telah mengalokasikan beberapa bagian sendiri di server. Hal ini berarti akan mendapatkan space server yang dedicated dan memperoleh sejumlah power computing dan memory. Selain itu, VPS hosting merupakan pilihan yang tepat bagi pengguna website bisnis yang trafficnya tinggi.

Kelebihan:

Space server yang dedicated (tanpa harus membeli server dedicated)

Peningkatan traffic di website lain tidak berdampak bagi performa situs

Akses root ke server

Scalability yang mudah

Dapat dikelola sendiri

Kekurangan:

* + 1. Sangat mahal jika dibandingkan dengan shared hosting
    2. Harus menguasai informasi dan pengetahuan teknis terkai manajemen server

*Cloud Hosting*

Saat ini, cloud hosting merupakan paket yang paling banyak disukai di pasaran karena literally tanpa *downtime*. Dengan cloud hosting, dapat menggunakan sekelompok server. File dan resource direplikasi di setiap server. Ketika salah satu server cloud sibuk atau bermasalah, maka traffic situs akan secara otomatis di-route ke server lainnya dalam cluster.

Kelebihan:

Sedikit bahkan tidak ada downtime

Jika server error, sama sekali tidak akan memengaruhi situs

Alokasi resource sesuai permintaan

Bayar sesuai dengan yang digunakan

Lebih scalable daripada VPS

Kekurangan:

* + 1. Biaya sulit diperkirakan
    2. Akses root tidak selalu tersedia

*WordPress Hosting*

*WordPress hosting* merupakan bentuk lain dari *shared hosting* dan dibuat untuk pengguna website *WordPress. Server* dikonfigurasi secara khusus untuk *WordPress* dan situs dilengkapi dengan *pre-installed plugin* terhadap hal-hal penting, seperti *caching* dan keamanan. Karena konfigurasi sudah sangat optimal, maka situs akan loading lebih cepat dan terhindar dari masalah. Paket WordPress hosting menyertakan fitur tambahan terkait WordPress, tema WordPress yang *pre-designed, drag-and-drop page builder,* dan *tool developer* yang spesifik.

Kelebihan:

Murah (umumnya memiliki kesamaan harga dengan shared hosting)

Mudah digunakan oleh pemula

One-click WordPress installation

Performa yang baik untuk situs WordPress

Customer support menguasai hal-hal yang berhubungan dengan WordPress.

Terdapat plugin dan tema WordPress yang pre-installed.

Kekurangan:

Hanya bisa digunakan oleh website yang dibuat di WordPress (masalah bisa muncul jika ingin mengonlinekan banyak website di server, tetapi beberapa website tidak dibuat menggunakan WordPress).

*Dedicated Server Hosting*

Dengan adanya *dedicated hosting*, bisa memiliki server fisik sendiri yang secara khusus didedikasikan hanya untuk situs. Karena itulah, penggunaan dedicated hosting sangatlah fleksibel. Dapat mengonfigurasikan sistem operasi dan software yang ingin digunakan, serta setup keseluruhan aspek hosting sesuai dengan kebutuhan.

Menyewa dedicated server sama bagusnya dengan memiliki server sendiri, tetapi di dedicated server, dibantu oleh customer support yang andal.

*Kelebihan:*

Kontrol penuh terhadap konfigurasi server

Keandalan yang tinggi (tidak perlu berbagi resource server dengan pengguna lain)

Akses root ke server

Keamanan terjamin

*Kekurangan:*

1. Mahal
2. Harus menguasai pengetahui teknis terkait manajemen server

**2.6.2 Domain**

Domain adalah nama yang diberikan untuk mengidentifikasi sebuah jaringan tanpa menggunakan internet protokol (IP), mengingat biasanya IP menggunakan angka-angka yang sulit dihafal. Dalam dunia internet domain merupakan aspek penting dalam sebuah website. Fungsinya adalah sebagai alamat untuk memudahkan para pengunjung untuk mencari informasi yang diinginkan. Perkembangan domain saat ini sangat meningkat dari tahun ke tahun. Pada awalnya, [sejarah domain](https://dailysocial.id/post/apa-itu-domain) mulai digunakan pada tahun 1985 oleh perusahaan komputer asal Massachusetss, Amerika Serikat bernama Symbolics.com. Di akhir tahun 1985 baru ada enam domain yang terdaftar. Namun, sekarang ada ribuan bahkan jutaan domain yang ada di seluruh dunia.

Membuat website ada baiknya untuk mengenal lebih dalam apa itu domain dan seluk beluknya. Mulai dari jenis-jenis yang paling banyak digunakan, bagaimana memilih nama yang efektif hingga memilih jasa penyedia hosting untuk mendapatkan domain tersebut. Sebagai satu rangkaian dalam website, domain tidak berdiri sendiri. Setidaknya ada [tiga jenis domain](https://www.utopicomputers.com/pengertian-fungsi-contoh-jenis-domain-serta-cara-membeli-nama-domain/) yang paling sering digunakan, yaitu *top level domain*, *second level domain*, dan *third level domain*.

1. *Top Level Domain*

*Top level domain* adalah bagian dari nama domain itu sendiri yang berada di bagian paling kanan setelah titik. Contohnya seperti www.google.com, dari contoh tersebut yang dimaksud TLD-nya adalah .com. TLD sendiri terbagi dua jenis yaitu *Global Top Level Domain* dan *Country Code Level Domain*. Untuk kategori *Global Top Level Domain* biasanya menggunakan domain seperti .com, .net, .org, .edu dan lain sebagainya. Domain .com digunakan untuk website komersial, sedangkan .net digunakan untuk website jaringan, dan .edu untuk edukasi. Ada pula domain lain seperti .org untuk organisasi, .mil untuk situs angkatan bersenjata dan lain sebagainya.

Sementara itu, untuk *Country Code Level Domain* lebih sering digunakan untuk domain lokal di negara-negara tertentu. Semisalnya saja .id untuk domain Indonesia, .cn untuk negara Cina dan .my untuk Malaysia. Masing-masing negara tidaklah sama dan hanya diambil singkatannya saja sehingga lebih mudah untuk diingat.

1. *Second Level Domain*

Jenis domain selanjutnya adalah *Second Level Domain* atau biasa disebut dengan SLD. SLD ini merujuk pada sebuah nama yang digunakan. Bila mencontoh pada domain pada poin di atas, maka untuk SLD-nya adalah ‘google’. Di sini penting untuk membuat SLD yang mudah diingat atau diketik maupun dieja oleh pengunjung. Maka dari itu, harus menggunakan nama yang familiar dan tidak terlalu panjang. Domain SLD ini bisa dibuat dengan cara mendaftar dan membelinya. Jangan lupa, lakukan pembaruan secara berkala paling tidak selama satu tahun.

1. *Third Level Domain*

Secara umum *Third Level Domain* atau TLD ini adalah nama domain yang berikan sebelum TLS dan SLD. Contohnya saja bila membuat untuk keperluan email, bisa dengan menambahkan webmail.namadomain.com. Begitu pula untuk hal-hal lainnya seperti blog, promo, dan sebagainya. Dalam dunia internet, TLD ini juga sering disebut dengan subdomain. Nama tersebut bisa dibuat sesuai kebutuhan setelah memiliki SLD.

**2.6.3 cPanel**

cPanel adalah salah satu control panel berbasis Linux yang paling banyak digunakan di akun web hosting. Melalui panel ini, bisa dengan mudah mengelola semua layanan dalam satu tempat. Saat ini, cPanel merupakan panel standar yang paling banyak digunakan dan sebagian besar web developer sudah sangat familiar dengan tool ini. Selain intuitif dan mudah digunakan, cPanel juga memungkinkan untuk mengelola akun web hosting dengan maksimal. Membuat FTP user dan alamat email baru, memonitor resource, membuat subdomain, dan menginstall software merupakan beberapa kegunaan cPanel. Berikut kelebihan cPanel dan kekurangannya.

**Kelebihan cPanel:**

1. Mudah dipelajari
2. Mudah digunakan
3. Menghemat waktu dan uang
4. Telah melalui uji coba
5. Menawarkan software auto installer
6. Tersedia berbagai tutorial dan panduan online

**Kekurangan cPanel:**

1. Fitur yang terlalu banyak hanya akan memperlambat performa cPanel
2. Pengaturan yang penting bisa tiba-tiba berubah secara otomatis
3. Beberapa hosting menjalankan software yang tidak diupgrade
4. Biayanya mahal dan jarang dipasangkan di hosting gratis

**2.7** **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].

**2.8 Perangkat Lunak Pengembang (Tools)**

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

**2.8.1 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”);

?>

**2.8.2 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).

**2.8.3 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.

**2.8.4 LINE Bot Designer**

LINE Bot Designer adalah aplikasi yang memungkinkan kita untuk membuat prototipe sebuah chatbot untuk aplikasi perpesanan LINE dengan mudah tanpa memerlukan pengetahuan ngoding sedikitpun. Kita diberikan opsi untuk mendesain chatbot dalam skenario-skenario yang diinginkan.

Namun perlu diingat bahwa produk yang dihasilkan oleh LINE Bot Designer hanya bersifat prototipe dan belum bisa dijadikan chatbot sungguhan (yang bisa berinteraksi dengan pengguna lain). Tapi, prototipe ini bisa dimanfaatkan untuk mempercepat proses pembuatan bot aslinya atau untuk menunjukkan konsep bisnis yang ingin dilakukan dengan memanfaatkan aplikasi LINE dan layanan chatbot.

**2.8.5 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.

**2.8.5.1 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

**2.8.5.2 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).

**2.8.6 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.

**2.8.7 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.

**2.8.8 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.

**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].**

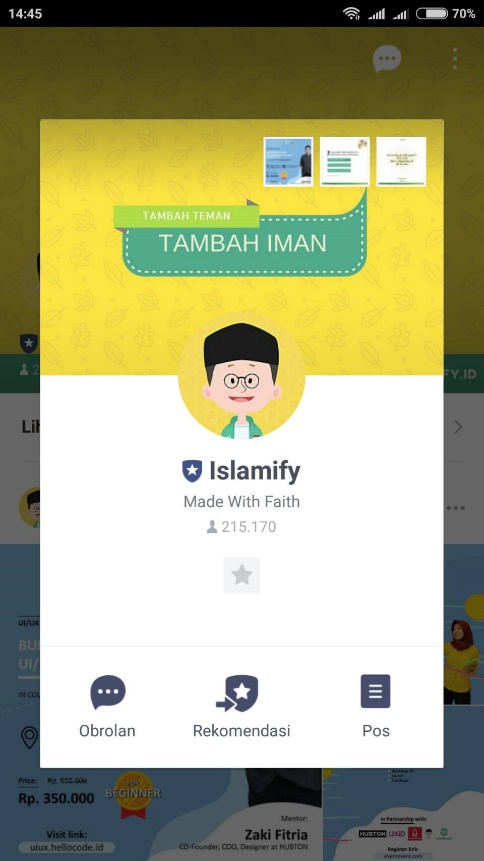
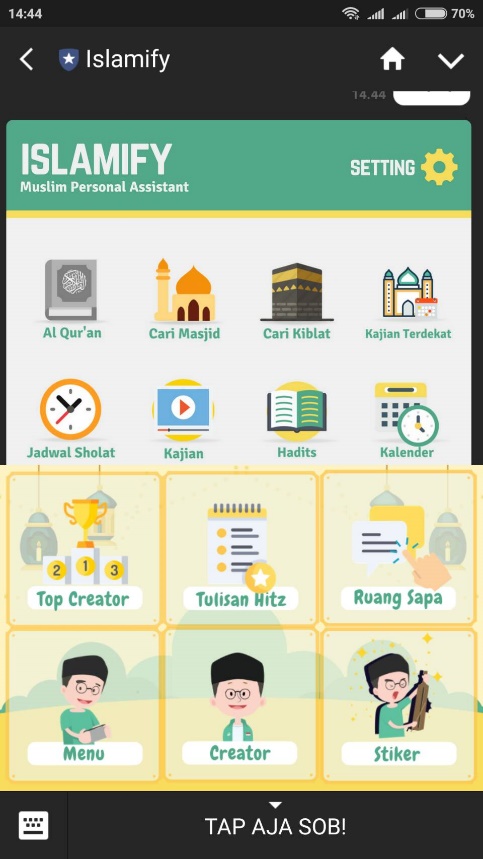
**Untuk mendapatkan Informasi Perguruan Tinggi yang diminatinya, saat ini siswa SMA sederajat dalam mencari informasi Perguruan Tinggi menggunakan cara mengunjungi setiap website Perguruan Tinggi satu persatu yang diinginkan untuk mendapatkan informasi mengenai Perguruan Tinggi. Berdasarkan data kuisoner terhadap 100 responden siswa SMA sederajat yang akan melanjutkan sekolahnya ke Perguruan Tinggi, terdapat 81% siswa yang masih mengalami kebingungan dalam menentukan Perguruan Tinggi yang diminatinya. Lalu berdasarkan data kuisoner yang sama, faktor peminat siswa SMA sederajat 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 :**

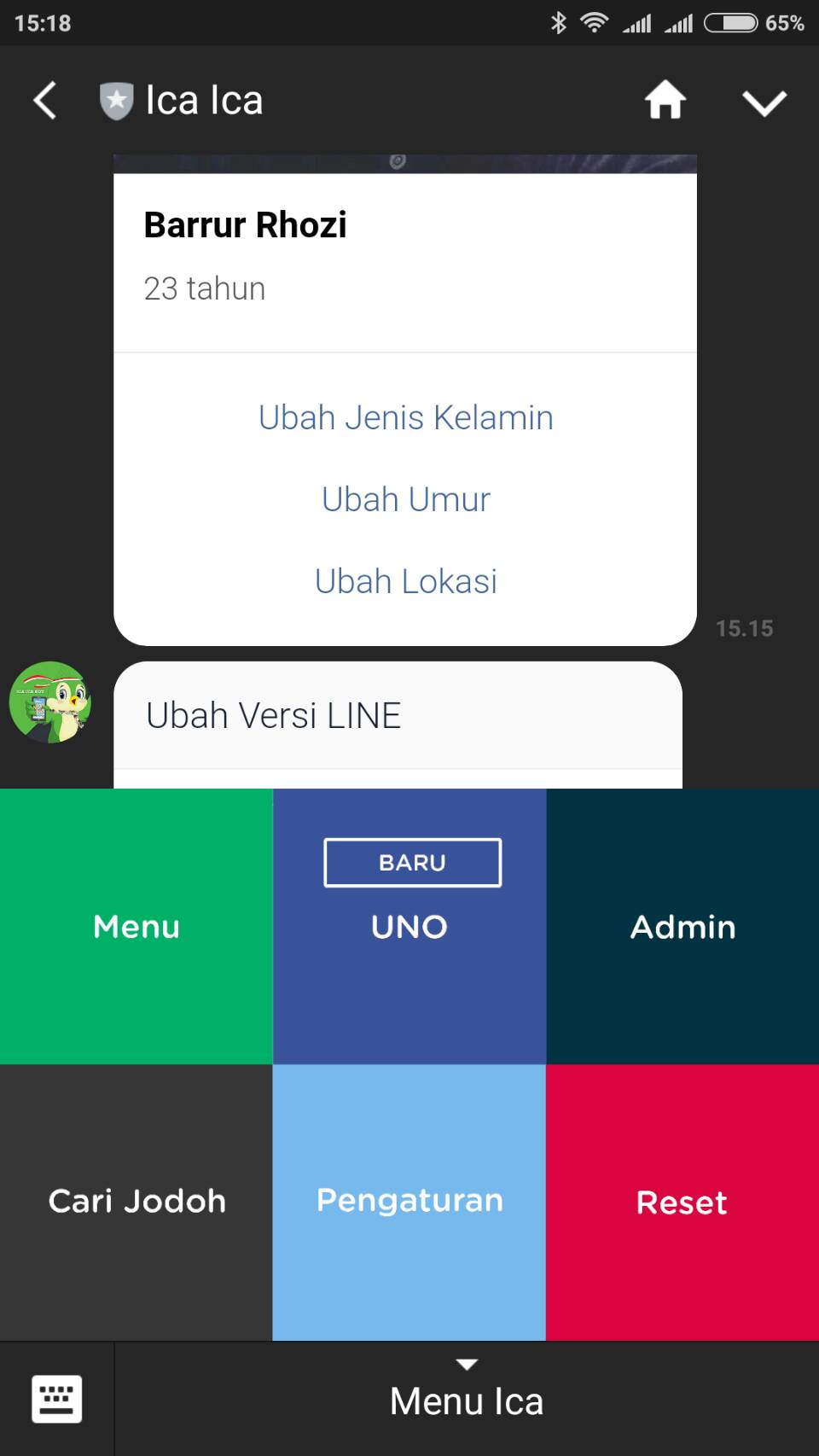
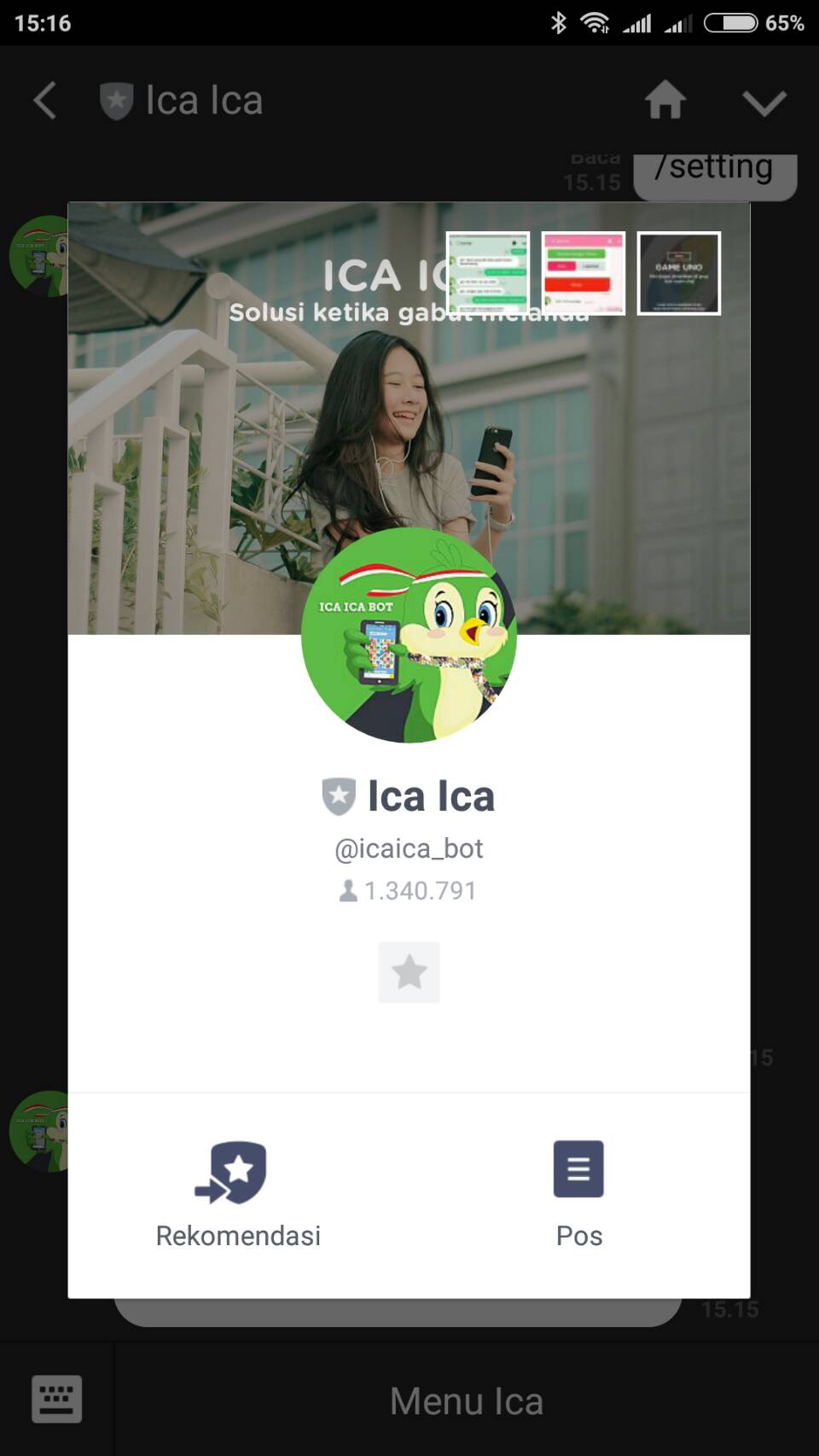
****

**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.**

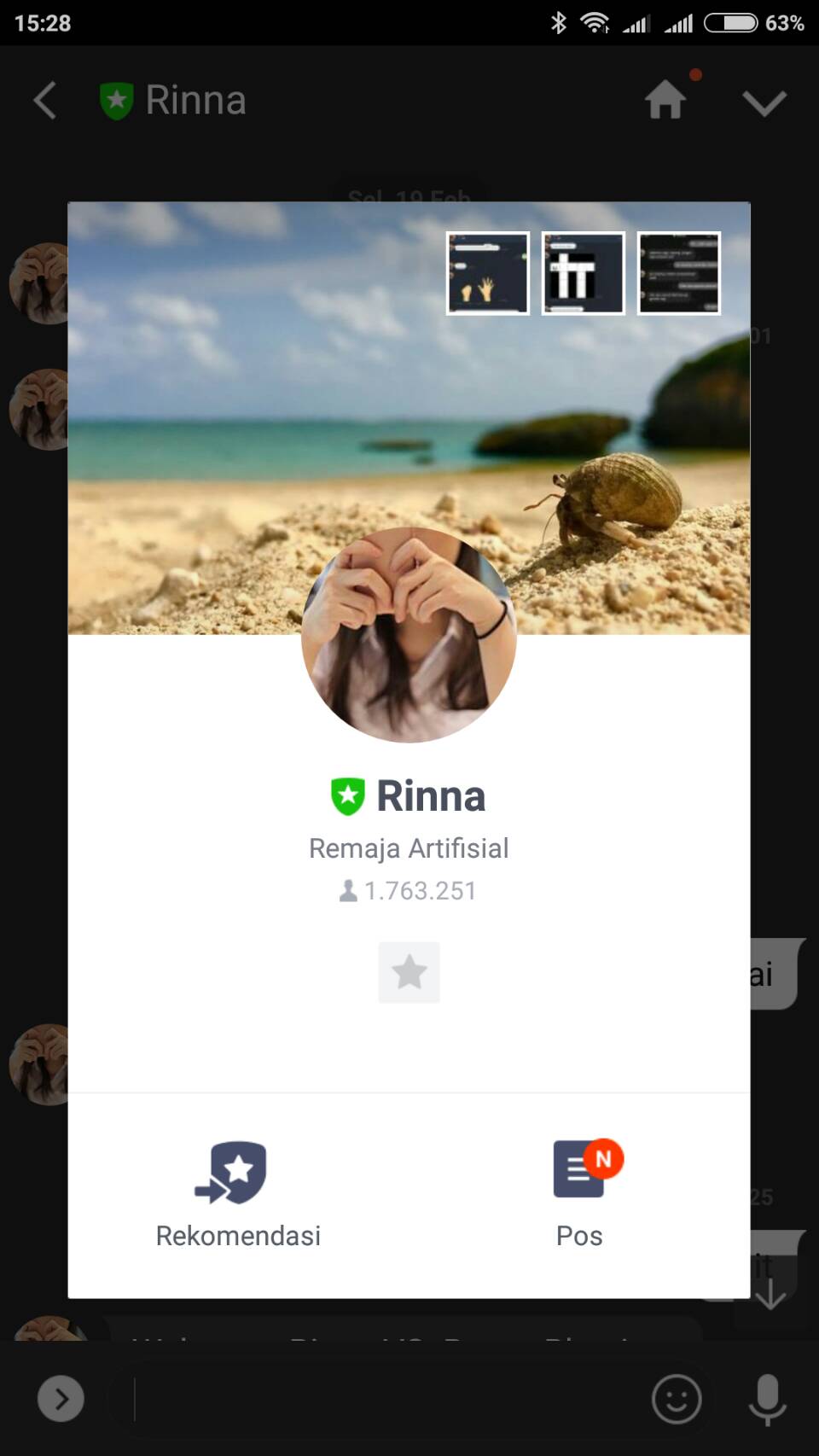


Gambar III.2 Ica Ica Bot

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.**

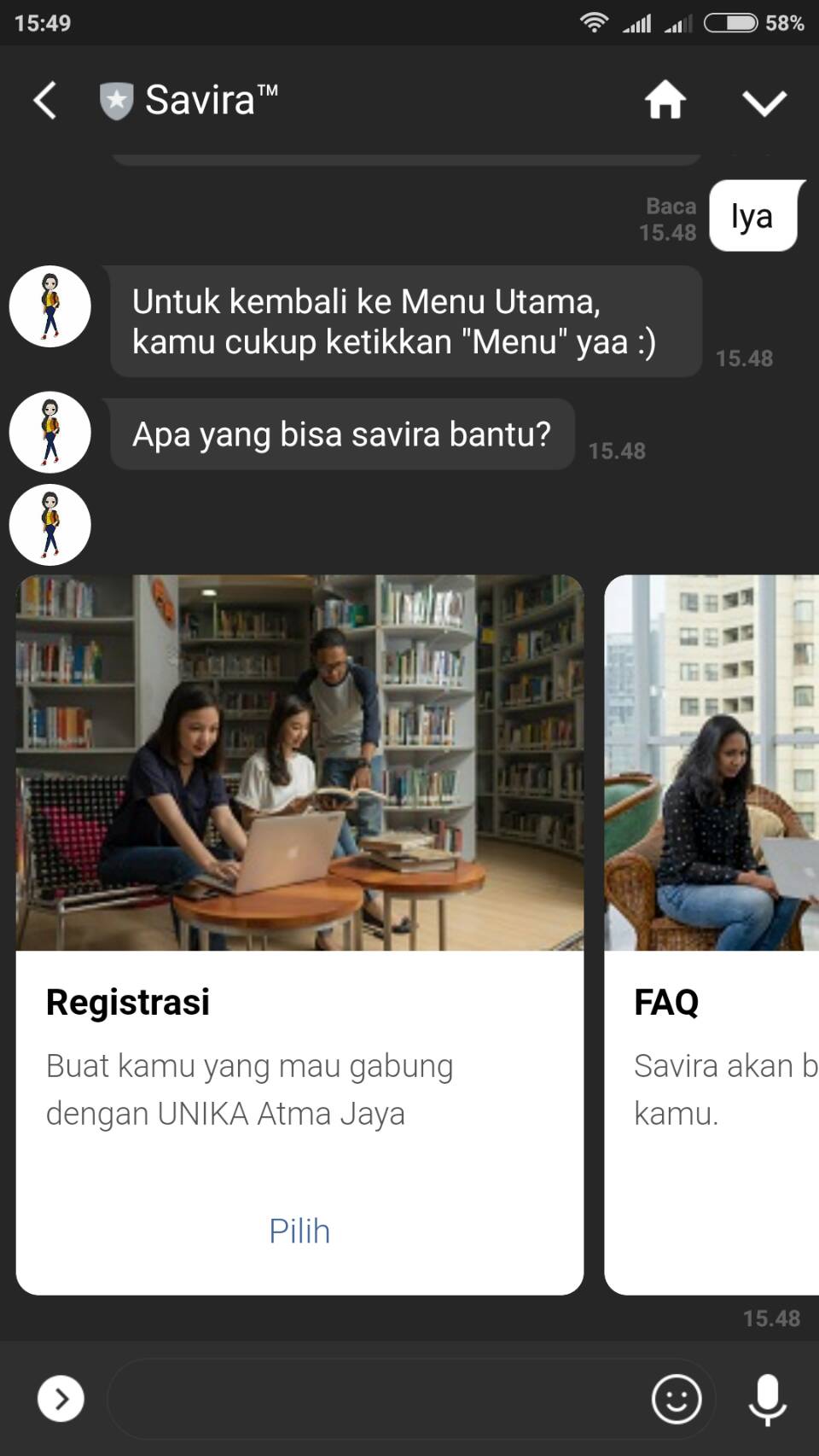
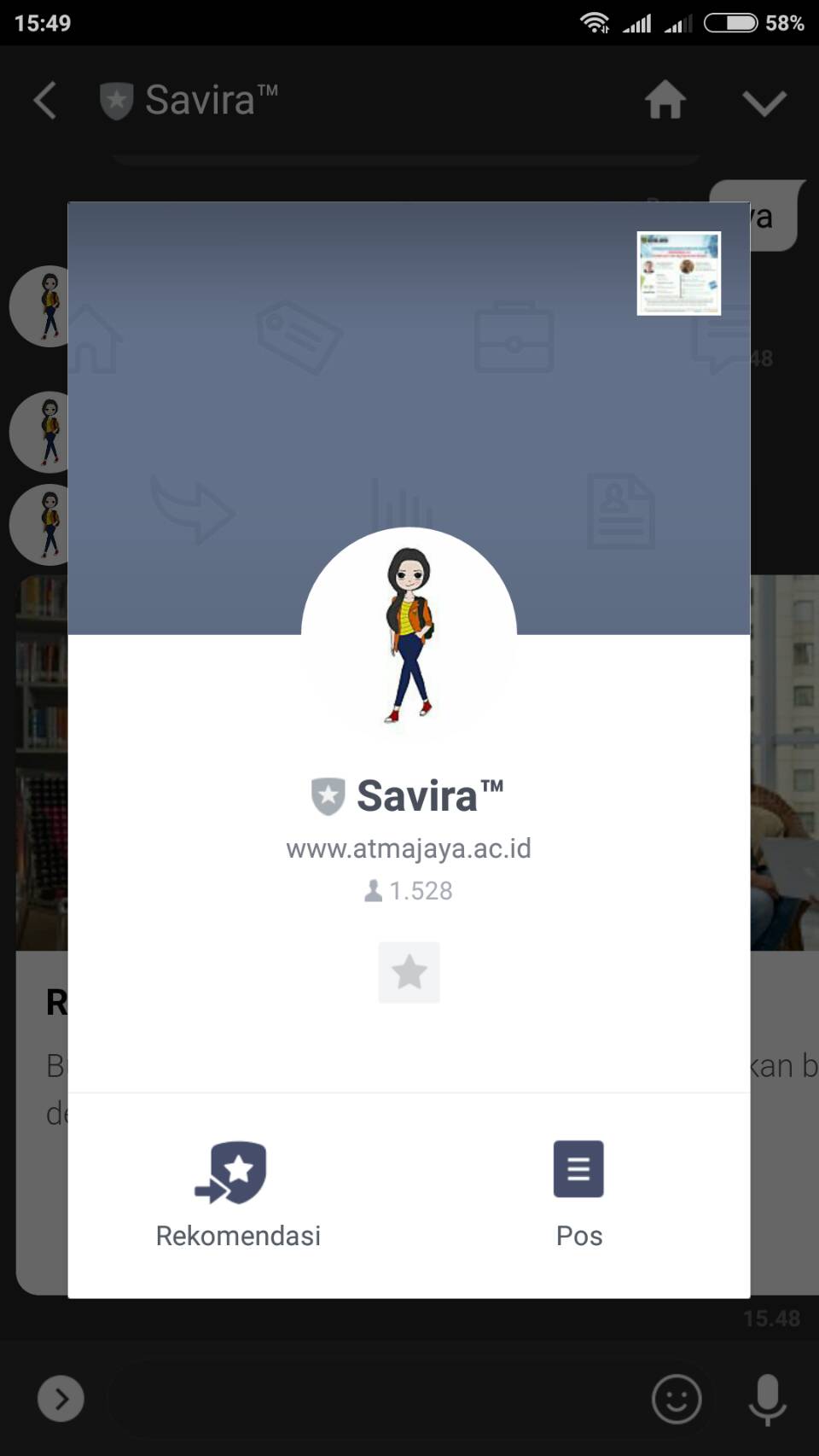


Gambar III.3 Bot Rinna

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.**

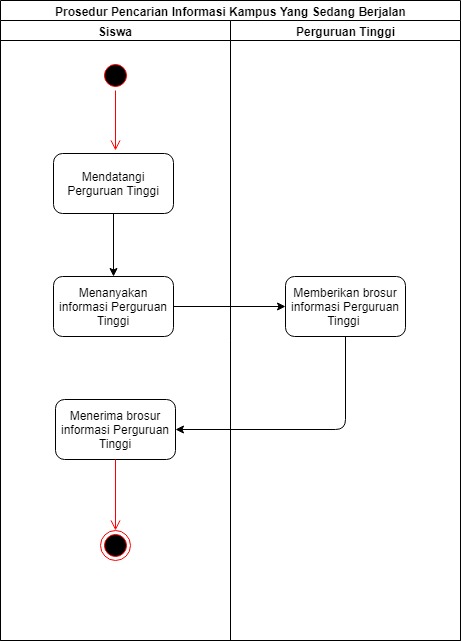
Berdasarkan berbagai aplikasi sejenis mengenai chatbot pada gambar diatas, maka dapat hasil analisis aplikasi bot dapat dilihat dibawah ini

**Tabel III.1 Analisis Aplikasi Sejensi**

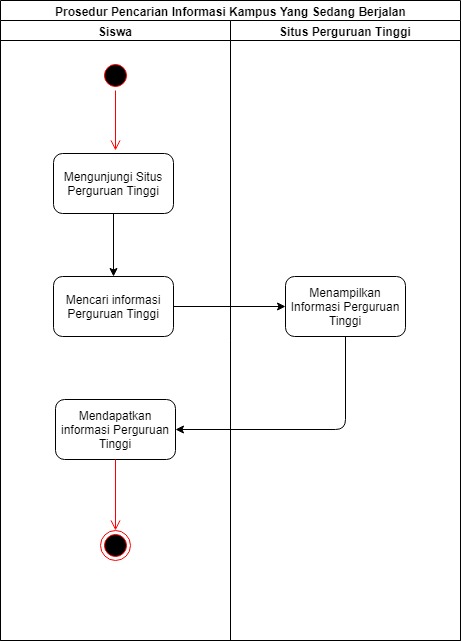
|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Konten/Fitur | Menyimpan Informasi Pengguna | Menggunakan Rich Menu |
| Islamify | 1. Al-Qur’an 2. Lokasi Masjid 3. Cari Kiblat 4. Jadwal Sholat 5. Kajian 6. Hadits 7. Kalender 8. Stiker | 1. Nama Pengguna 2. Jenis Kelamin | Ya |
| Icaica | 1. Permainan (Cari kata, tebak kata, adu pinalti, dsb) 2. Cari jodoh | 1. Jenis Kelamin 2. Tanggal Lahir 3. Lokasi 4. Versi LINE | Ya |
| Rinna | 1. Permainan (Gunting, Batu, Kertas, Escape Room) 2. Menggunakan Teknologi AI 3. Dapat bercakap dalam bahasa alami | 1. Nama pengguna 2. Jenis Kelamin | Tidak |
| Savira | 1. Registrasi kuliah 2. FAQ 3. Pindah prodi 4. Pindah kampus 5. Salah transfer 6. Payment Plan | 1. Nama pengguna 2. Asal sekolah 3. Email | Tidak |
| Akami Bot | 1. Cari Kampus 2. Cari Jurusan 3. Cari Biaya 4. Cari Lokasi 5. Profile 6. Panduan | 1. Nama Pengguna 2. User Id LINE 3. Foto pengguna | Ya |

**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 1



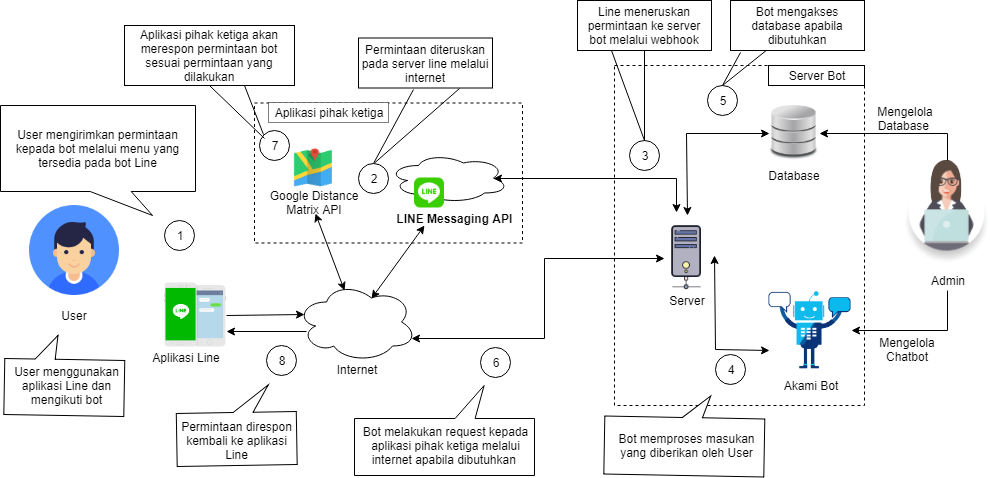
Gambar III.6 Prosedur sistem yang berjalan 2

**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 media sosial 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.

**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.

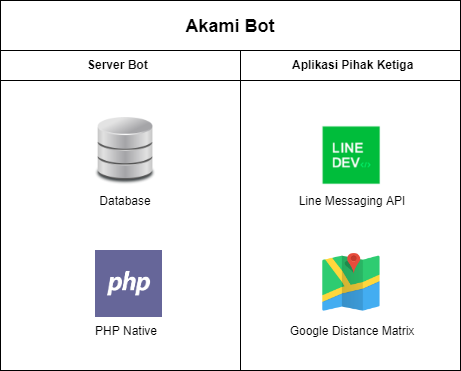
Gambar III.7 Analisis Deskripsi Aplikasi

Berikut adalah penjelasan dari arsitektur sistem pada chatbot yang dibangun :

1. User menggunakan aplikasi Line dan mengikuti bot untuk mengirimkan permintaan kepada bot melalui menu yang tersedia pada bot Line.
2. Permintaan diteruskan pada server line melalui internet.
3. Line meneruskan permintaan ke server bot melalui webhook.
4. Bot memproses masukan yang diberikan oleh User.
5. Bot mengakses database apabila dibutuhkan.
6. Bot melakukan request kepada aplikasi pihak ketiga melalui internet apabila dibutuhkan.
7. Aplikasi pihak ketiga akan merespon permintaan bot sesuai permintaan yang dilakukan.
8. Permintaan direspon kembali ke aplikasi Line

**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). Gambar III.2 dibawah ini merupakan gambaran mengenai arsitektur aplikasi yang dibangun.

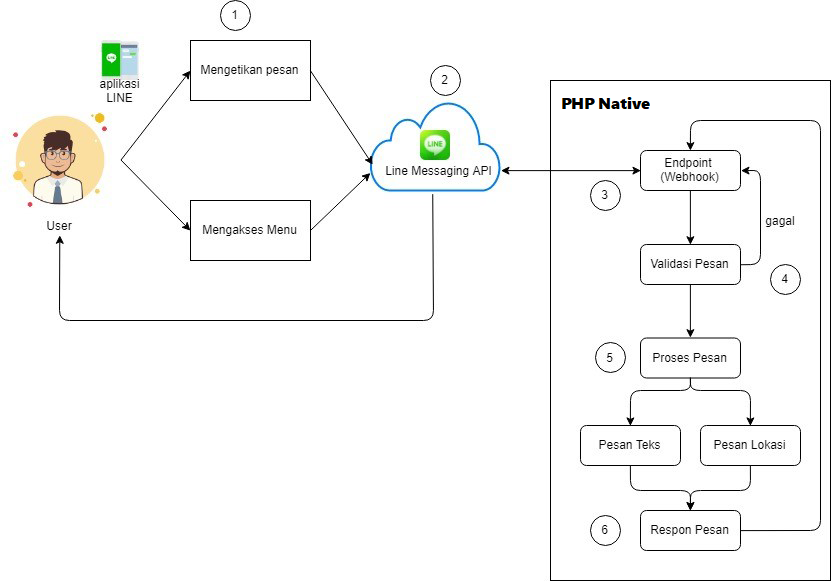


Gambar III.8 Analisis Arsitektur Aplikasi Yang Dibangun

Penjelasan untuk setiap bagian aplikasi adalah sebagai berikut :

1. Server Bot
2. PHP Native

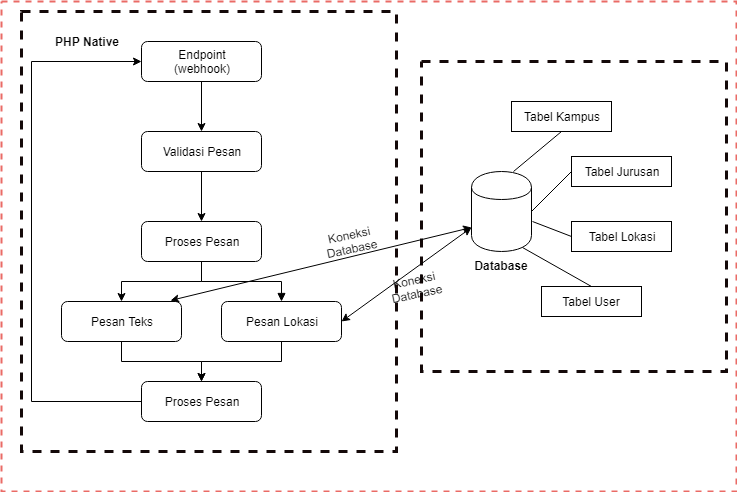
PHP Native digunakan untuk menangani request dan response yang dilakukan oleh user melalui Line Messaging API. User akan melakukan request berupa aksess menu atau mengetikan masukan berupa pesan teks pada aplikasi Line. Lalu request tersebut diteruskan melalui Line Messaging API untuk kemudian diteruskan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



Gambar III.9 Alur chatbot pada PHP Native

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 server untuk kemudian diteruskan kembali kepada user melalui Line Messaging API.



Gambar III.10 Analisis Arsitektur Database

1. Aplikasi Pihak Ketiga
2. Line Messaging API

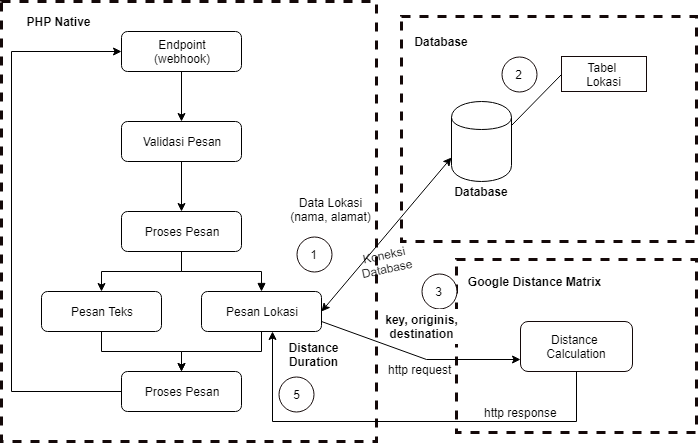
Line Messaging API digunakan untuk bisa terhubung dengan user melalui aplikasi Line. Untuk bisa terhubung melalui line messaging api, perlu dilakukan pengaturan webhook pada Line Developer Console dan pengaturan channel secret dan channel token pada aplikasi bot. Hal ini dilakukan untuk bisa mengatur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



Gambar III.10 Alur Analisis Arsitektur Line Messaging API

1. 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.

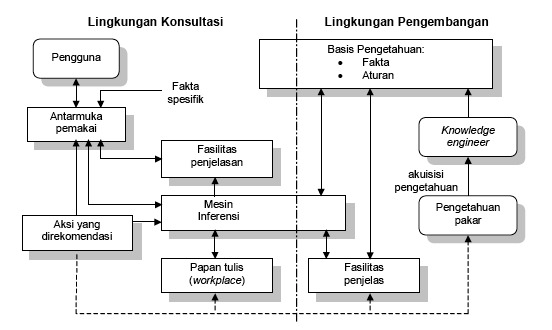


Gambar III.10 Alur Analisis Arsitektur Google Distance Matrix

## 3.6 Analisis Metode

Analisis metode adalah analisis metode-metode yang akan digunakan. Penelitian ini dibangun menggunakan sistem pakar dengan metode yang digunakan forward chaining .

**3.6.1 Diagram Struktur Sistem Pakar**

[](http://universitaspendidikan.com/wp-content/uploads/2013/11/diagram-sistem-pakar.png)

Gambar III.11 Diagram Struktur Sistem Pakar

**Struktur sistem pakar bisa dijelaskan seperti berikut :**

1. Basis pengetahuan : Berupa pengetahuan-pengetahuan yang dibutuhkan untuk memahami, memformulasi, dan memecahkan masalah. Diaman basis pengetahuan tersusun atas 2 elemen dasar:

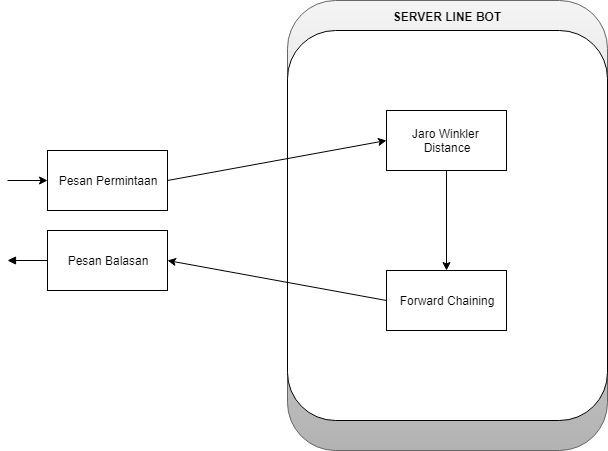
* Fakta, misalnya: situasi, kondisi, dan kenyataan dari permasalahan yang ada, serta teori dalam bidang itu
* Aturan, yang mengarahkan penggunaan  pengetahuan untuk memecahkan masalah yang spesifik dalam bidang yang khusus

Mengembangkan Sistem Pakar dapat dilakukan dengan 2 cara:

1. Membangun sendiri semua komponen di atas, atau
2. Memakai semua komponen yang sudah ada kecuali isi basis pengetahuan.  
   Tahap-tahap pembangunnan yaitu:
3. Pemilihan Masalah
4. Rekayasa Pengetahuan (Knowledge Engineering)
5. Partisipan Dalam Proses Pengembangan
6. Akuisisi Pengetahuan

Inferensi digunakan dalam sistem pakar untuk memperoleh informasi  
terbaru dari informasi yang sudah ada yaitu salah satunya forward chaining.

**3.6.2 Analisis Server Line Bot**

****

Gambar III.12 Alur Server Line Bot

Keterangan :

* + 1. Jaro Winkler Distance : Untuk meminimalisir kesalahan pengetikan *keywoard* yang dilakukan oleh pengguna
    2. Forward Chaining : Untuk memetakan alur komunikasi yang ada pada LINE Bot agar respon dapat diberikan sesuai dengan permintaan yang diberikan.

1. Analisis Algoritma Jaro Winlker Distance

Rumus :

Menghitung *Jaro-Winkler Distance*

Menghitung Jarak

Misalkan :

S1 = KAMPUS

S2 = KMPUS

Maka diketahui :

m = 5

| S1 | = 6

| S2 | = 5

Karakter yang tidak ada yaitu huruf A setelah huruf K, dalam string ini tidak terdapat transposisi sehingga t = 0. Sehingga Jaro Distance adalah :

0,944

Kemudian bila diperhatikan antara S1 dan S2 dapat diketahui nilai l = 1. Maka nilai Jaro-Winkler Distance adalah :

2) Analisis Algoritma Forward Chaining

**Contoh : Cari Kampus**

* Rule 1

**Jika** pengguna memilih menu cari kampus, **Maka** bot akan menampilkan pilihan kampus berdasarkan akreditasi, dan status.

* Rule 2

**Jika** pengguna memilih pilihan berdasarkan akreditasi, **Maka** bot akan menampilkan pilihan akreditasi A,B, dan C.

* Rule 3

**Jika** pengguna memilih pilihan berdasarkan status, **Maka** bot akan menampilkan pilihan status kampus Negeri atau Swasta.

* Rule 4

**Jika** pengguna memilih pilihan akreditasi A,B, atau C, **Maka** bot akan menampilkan informasi kampus berdasarkan akreditasi

* Rule 5

**Jika** pengguna memilih pilihan status kampus Negeri atau Swasta, **Maka** bot akan menampilkan informasi kampus Negeri atau Swasta yang dipilih.

## 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.3 Analisis Perangkat Keras

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

1. Analisis Perangkat Lunak

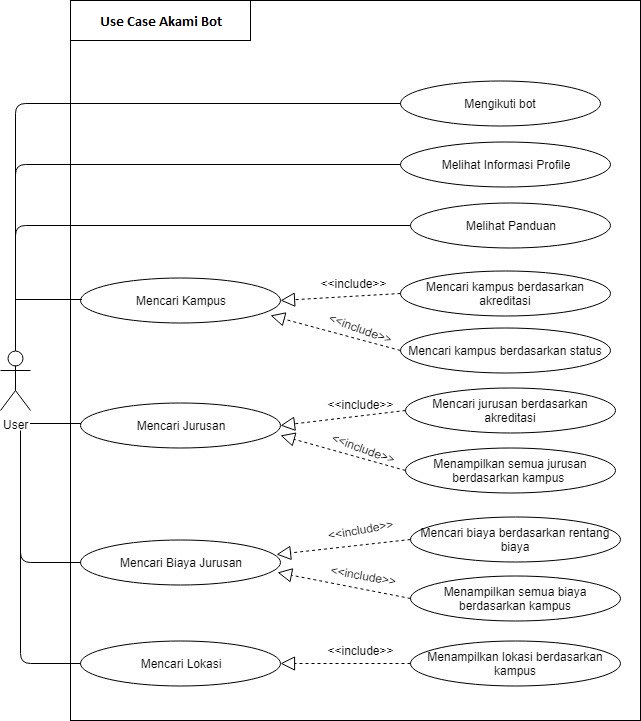
Berikut adalah spesifikasi perangkat lunak yang digunakan dalam membangun dan mengimplementasikan aplikasi chatbot sebagai berikut :

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. PHP (PHP v7.1.3)

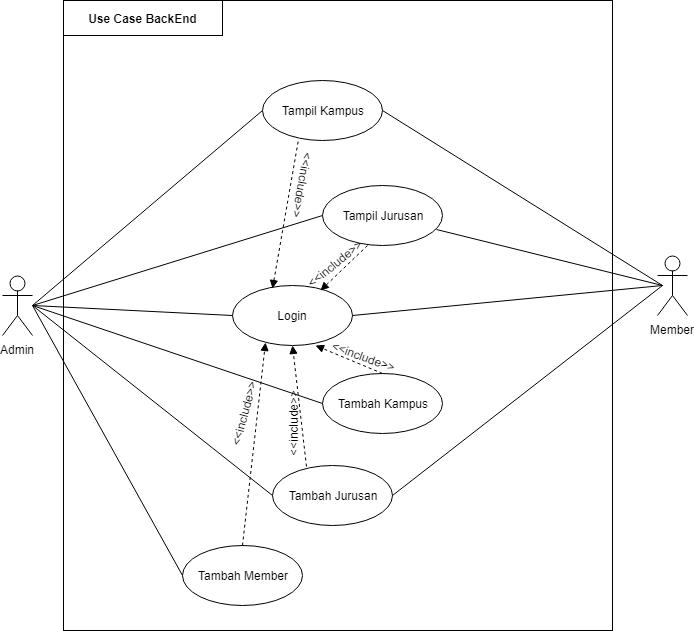
**3.7.2 Analisis Kebutuhan Fungsional**

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.15 Use Case Diagram FrontEnd



Gambar III.15 Use Case Diagram BackEnd

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

Tabel II.4 Definisi Actor

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1 | User | *User* hanya dapat mengakses fitur yang tersedia |

* + 1. **Definisi Use Case FrontEnd**

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

Tabel I.5 Definisi Use Case FrontEnd

|  |  |  |
| --- | --- | --- |
| **No.** | **Use Case** | **Deskripsi** |
| 1 | Mengikuti Bot | Mengikuti bot di aplikasi LINE |
| 2 | Melihat Informasi Profile | Melihat informasi profile pengguna LINE |
| 3 | Melihat Panduan | Melihat panduan pencarian informasi kampus |
| 4 | Mencari kampus berdasarkan akreditasi | Menampilkan kampus berdasarkan akreditasi |
| 5 | Mencari kampus berdasarkan status | Menampilkan kampus berdasarkan status Negeri atau Swasta |
| 6 | Mencari jurusan berdasarkan akreditasi | Menampilkan jurusan berdasarkan akreditasi |
| 7 | Menampilkan semua jurusan berdasarkan kampus | Menampilkan semua jurusan berdasarkan kampus yang dipilih |
| 8 | Mencari biaya berdasarkan rentang harga | Menampilkan biaya semester jurusan berdasarkan rentang harga |
| 9 | Menampilkan semua biaya berdasarkan kampus | Menampilkan semua biaya semester jurusan berdasarakan kampuus yang dipilih |
| 10 | Menampilkan lokasi berdasarkan kampus | Menampilkan lokasi berdasarkan kampus yang dipilih |

* + 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.6 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.7 Use Case Description Melihat Informasi Profile

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

Tabel III.8 Use Case Description Melihat Panduan

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

Tabel III.9 Use Case Description Mencari Kampus berdasarkan Akreditasi

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Mencari kampus berdasarkan akreditasi | |
| **Related Requirements** |  | |
| **Goal in Context** | Mencari kampus berdasarkan akreditasi di Kota Bandung | |
| **Preconditions** | User sudah mengikuti bot | |
| **Successful end Condition** | User berhasil mencari kampus berdasarkan akreditasi di Kota Bandung | |
| **Failed end Condition** | User gagal mencari kampus berdasarkan akreditasi di Kota Bandung | |
| **Primary Actors** | User | |
| **Secondary Actors** |  | |
| **Trigger** | User meminta mencari kampus berdasarkan akreditasi di Kota Bandung | |
| **Main Flow** | **Step** | **Action** |
|  | 1 | User meminta untuk mencari kampus berdasarkan akreditasi di Kota Bandung |
|  | 2 | Bot memeriksa apakah pengguna sudah mengikuti bo |
|  | 3 | User berhasil mencari kampus berdasarkan akreditasi di Kota Bandung |
| **Extensions** | **Step** | **Branching Action** |
|  | 2.1 | Bot gagal memeriksa informasi kampus berdasarkan akreditasi |
|  | 2.2 | User gagal melihat informasi kampus berdasarkan akreditasi |

Tabel III.10 Use Case Description Mencari Info Kampus berdasarkan status

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

Tabel III.11 Use Case Description Mencari Informasi Jurusan Berdasarkan Akreditasi

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

Tabel III.12 Use Case Description menampilkan Info semua jurusan Berdasarkan kampus

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

Tabel III.13 UseCase Description Mencari Info Biaya Berdasarkan rentang harga

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

Tabel III.14 Use Case Description menampilkan Info semua biaya Berdasarkan kampus

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

Tabel III.15 Use Case Description menampilkan Info lokasi Berdasarkan kampus

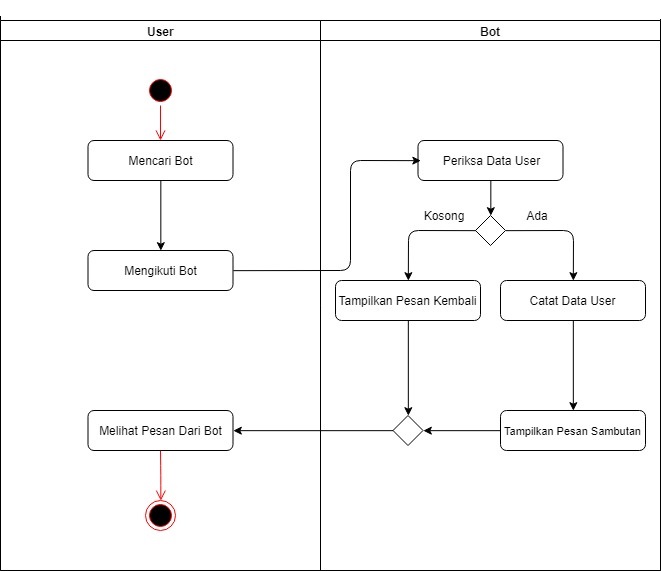
|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Menampilkan info lokasi berdasarkan kampus | |
| **Related Requirements** |  | |
| **Goal in Context** | Menampilkan informasi lokasi berdasarkan kampus | |
| **Preconditions** | User sudah mengikuti bot | |
| **Successful end Condition** | User berhasil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

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.16 menunjukkan activity diagram mengikuti bot :



Gambar III.16 Mengikuti Bot

* 1. Activity Diagram Melihat Informasi Profile

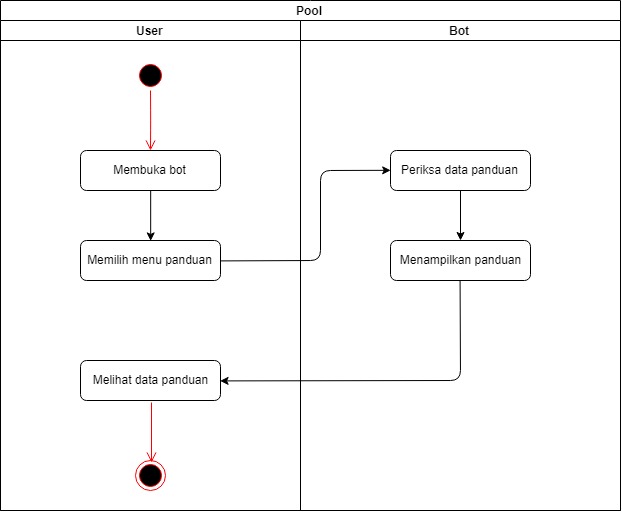
Activity diagram Melihat Informasi Profil menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna melihat informasi profil. Gambar III.17 menunjukkan activity diagram melihat informasi profil:



Gambar III.17 Activity Diagram Melihat Informasi Profil

* 1. Activity Diagram Melihat Panduan

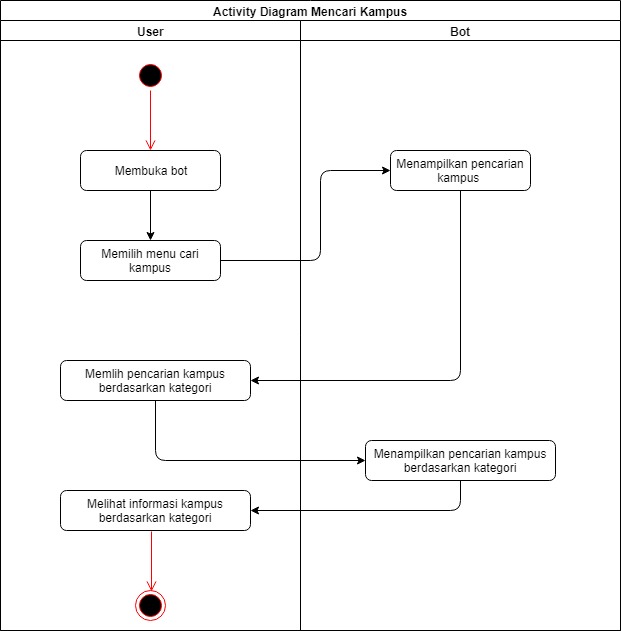
Activity diagram Melihat Stiker menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna melihat panduan. Gambar III.18 menunjukkan activity diagram melihat informasi panduan :



Gambar III.18 Activity Diagram Menu Melihat Panduan

* 1. Activity Diagram Menu Mencari Kampus

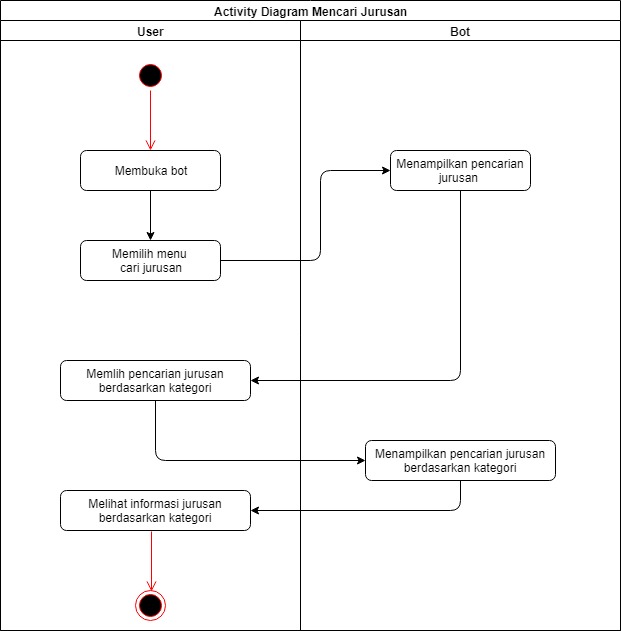
Activity diagram mencari kampus di Kota Bandung menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari kampus di Kota Bandung. Gambar III.19 menunjukkan activity diagram mencari kampus di Kota Bandung:



Gambar III.19 Activity Diagram Menu Cari Kampus

* 1. Activity Diagram menu cari jurusan

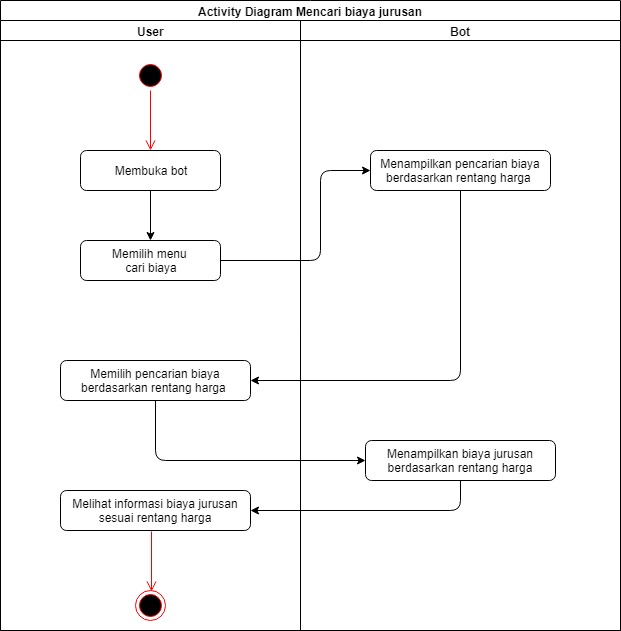
Activity diagram cari jurusan di Kota Bandung menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari jurusan di Kota Bandung. Gambar III.20 menunjukkan activity diagram mencari jurusan di Kota Bandung:



Gambar III.20 Activity Diagram Menu Cari Jurusan

* 1. Activity Diagram menu cari biaya

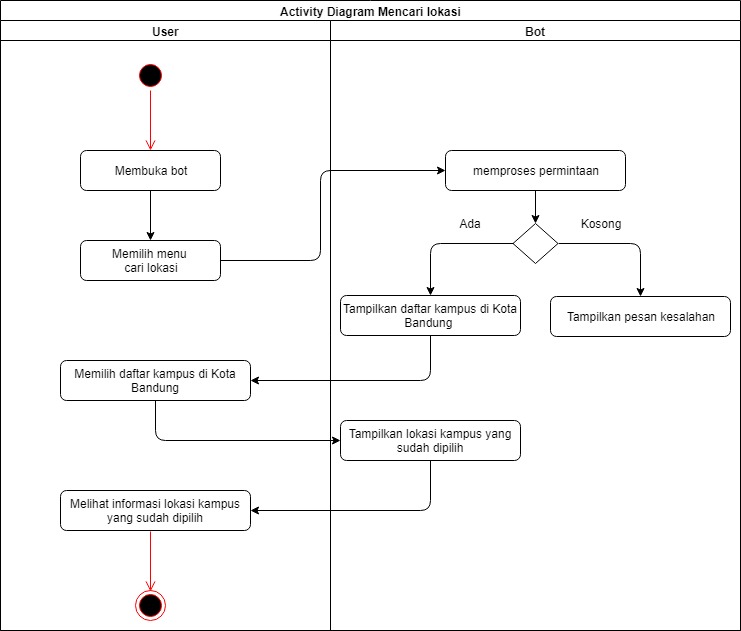
Activity diagram Mencari informasi pencarian biaya jurusan yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi cari biaya jurusan berdasarkan rentang harga. Gambar III.21 menunjukkan activity diagram mencari informasi biaya jurusan berdasarkan rentang harga:



Gambar III.21 Activity Diagram Menu Cari Biaya

* 1. Activity Diagram menu cari lokasi

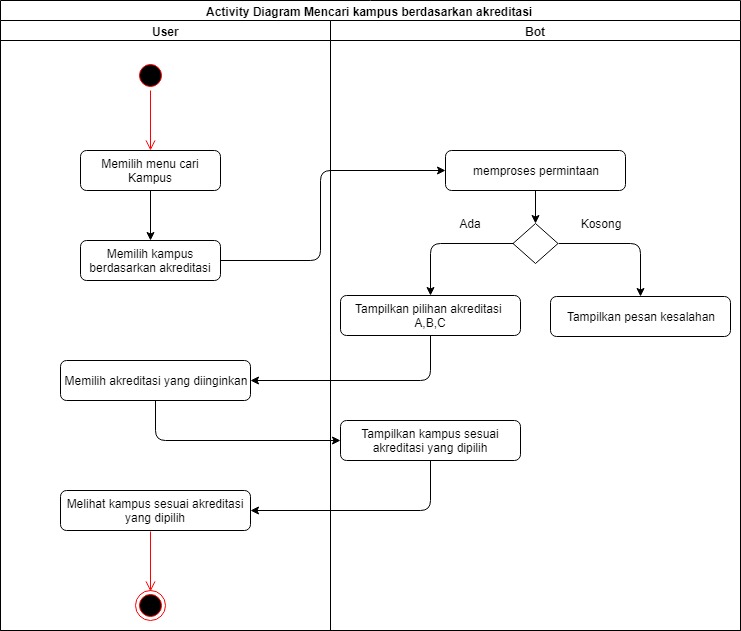
Activity diagram Mencari informasi pencari lokasi kampus yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi lokasi kampus. Gambar III.22 menunjukkan activity diagram mencari informasi lokasi kampus :



Gambar III.22 Activity Diagram Menu Cari Lokasi

* 1. Activity Diagram Mencari kampus berdasarkan akreditasi

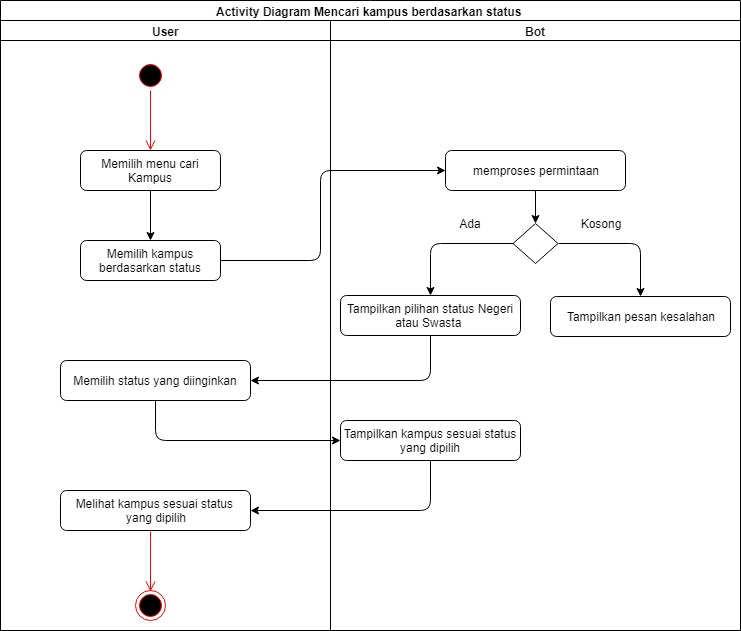
Activity diagram Mencari informasi kampus berdasarkan akreditasi yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi kampus berdasarkan akreditasi. Gambar III.23 menunjukkan activity diagram mencari informasi kampus berdasarkan akreditasi :



Gambar III.23 Activity Diagram Mencari Kampus Berdasarkan Akreditasi

* 1. Activity Diagram Mencari kampus berdasarkan status

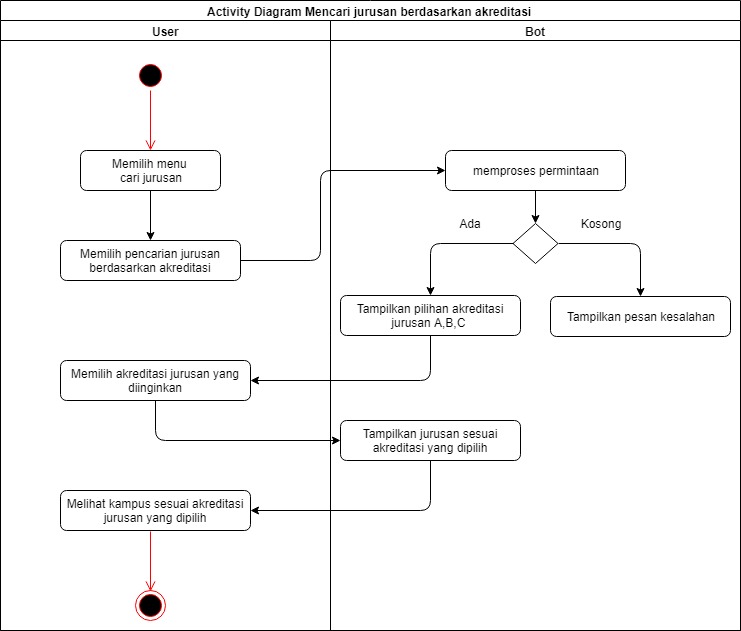
Activity diagram Mencari informasi kampus berdasarkan status yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi kampus berdasarkan status. Gambar III.24 menunjukkan activity diagram mencari informasi kampus berdasarkan status :



Gambar III.24 Activity Diagram Mencari Kampus Berdasarkan Status

* 1. Activity Diagram Mencari jurusan berdasarkan akreditasi

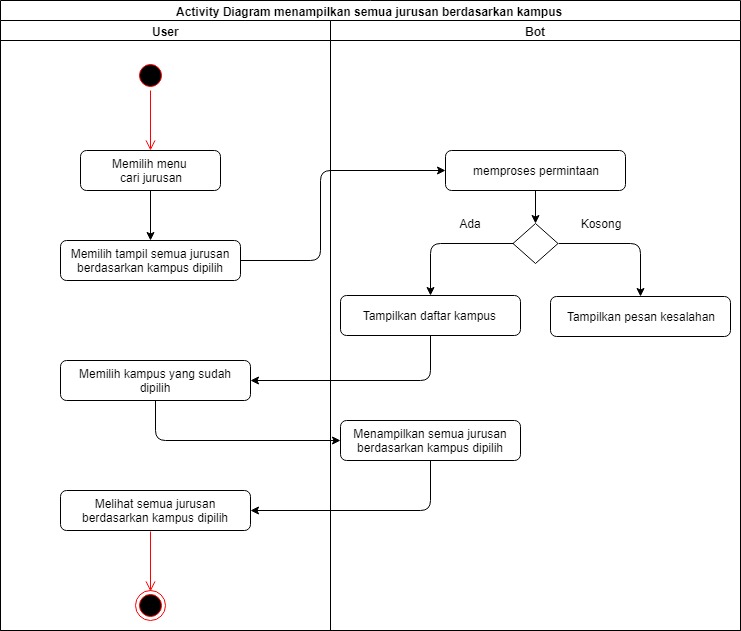
Activity diagram Mencari informasi jurusan berdasarkan akreditasi yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi jurusan berdasarkan akreditasi. Gambar III.25 menunjukkan activity diagram mencari informasi jurusan berdasarkan akreditasi :



Gambar III.25 Activity Diagram Mencari Jurusan Berdasarkan Akreditasi

* 1. Activity Diagram menampilkan semua jurusan berdasarkan kampus

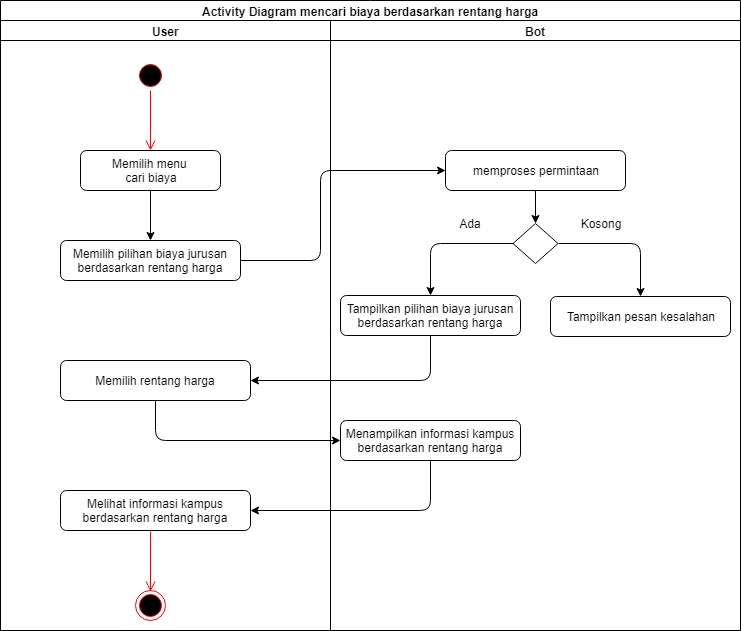
Activity diagram menampilkan semua informasi jurusan berdasarkan kampus yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi semua jurusan berdasarkan kampus. Gambar III.26 menunjukkan activity diagram menampilkan semua informasi jurusan berdasarkan kampus :



Gambar III.26 Activity Diagram Menampilkan Semua Jurusan Berdasarkan Kampus

* 1. Activity Diagram menampilkan biaya berdasarkan rentang harga

Activity diagram menampilkan biaya berdasarkan rentang harga yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi biaya berdasarkan rentang harga. Gambar III.27 menunjukkan activity diagram menampilkan biaya berdasarkan rentang harga :



Gambar III.27 Activity Diagram Mencari Biaya Berdasarkan Rentang Harga

1. Class Diagram

Class diagram keseluruhan digunakan untuk menggambarkan struktur dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem serta hubungannya antara kelas. Class diagram keseluruhan dapat dilihat pada gambar iii.18.

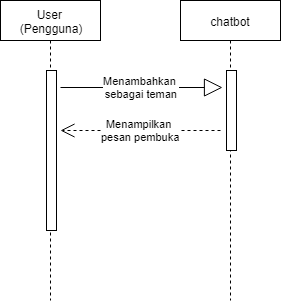


Gambar III.28 Class Diagram

* 1. **Sequence Diagram**

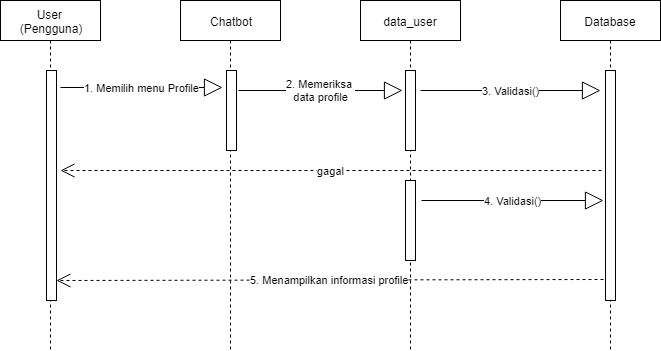
Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram jenis ini memberikan kejelasan sejumlah objek dan pesan-pesan yang diletakkan di antaranya di dalam sebuah use case. Berikut Sequence Diagram yang terdapat pada pembangunan chatbot pengatur keuangan pribadi :

* + 1. Sequence Diagram Mengikuti Bot



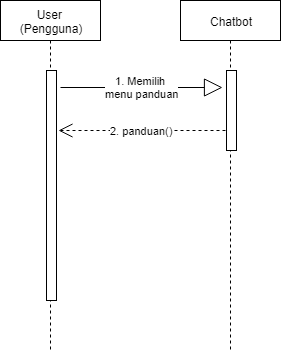
Gambar III.29 Sequence Diagram Mengikuti Bot

* + 1. Sequence Diagram Melihat Informasi Profile



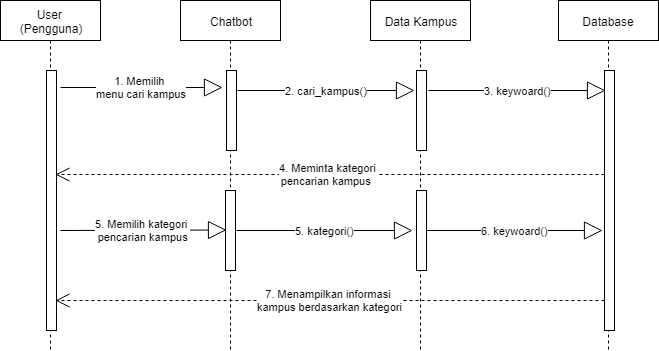
Gambar III.30 Sequence Diagram Melihat Informasi Profile

3. Sequence Diagram Melihat Panduan



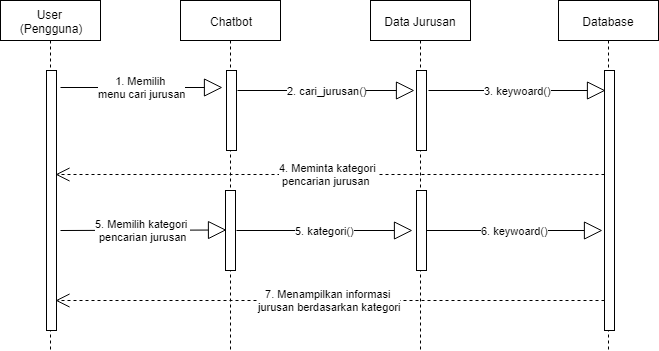
Gambar III.29 Sequence Diagram Melihat Panduan

4. Sequence Diagram Mencari Kampus



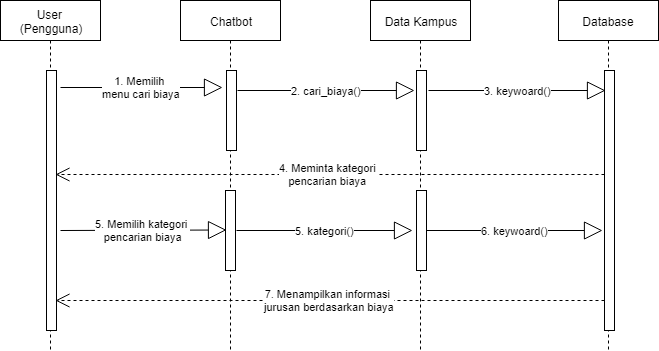
Gambar III.31 Sequence Diagram Mencari Kampus

5. Sequence Diagram Mencari Jurusan



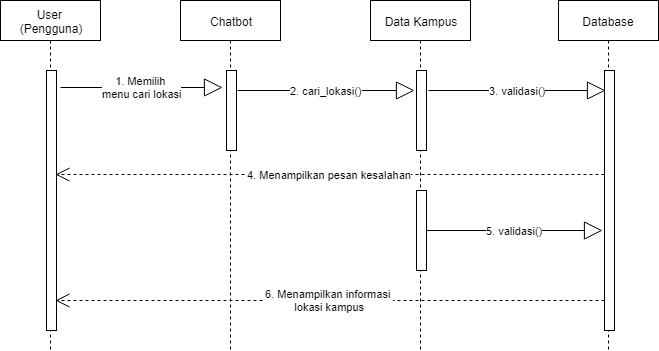
Gambar III.31 Sequence Diagram Mencari Jurusan

6. Sequence Diagram Mencari Biaya



Gambar III.31 Sequence Diagram Mencari Biaya

7. Sequence Diagram Mencari Lokasi



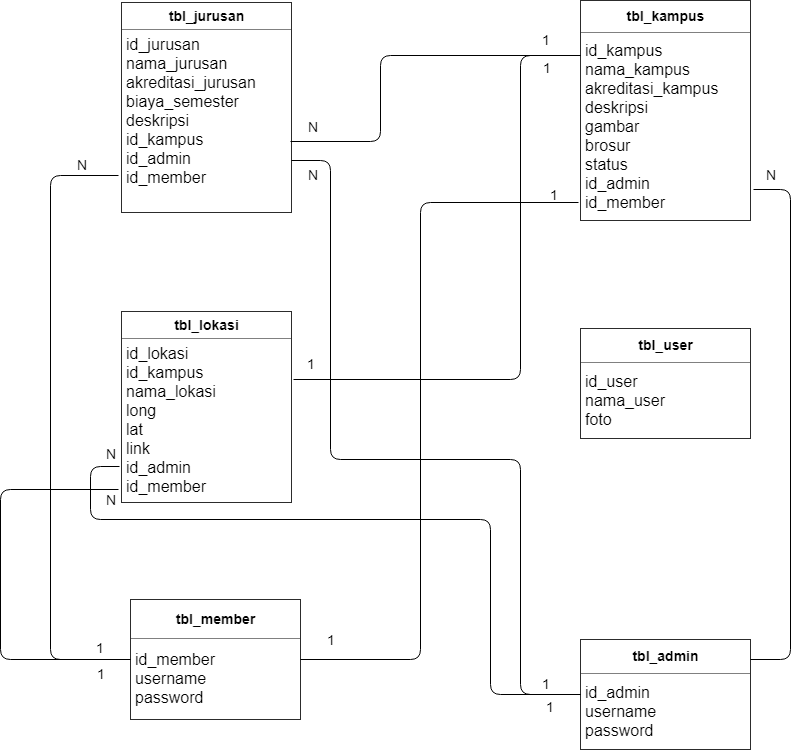
Gambar III.31 Sequence Diagram Mencari Lokasi

* 1. **Perancangan Basis Data**

Perancangan basis data yaitu perancangan yang bisa menciptakan atau merancang kumpulan data yang terhubung dan tersimpan secara bersamasama.

**3.10.1 Skema Relasi**

Pada tahap ini akan digambarkan struktur database yang digunakan. Struktur relasi database ini dapat dilihat pada gambar berikut.



Gambar III.30 Skema Relasi

**3.10.2 Struktur Tabel**

Struktur tabel merupakan penjelasan mengenai masing-masing tabel pada database yang dibangun untuk sistem di mana akan dirancang karakteristik setiap atribut dan tabel yang akan dibangun. Adapun perancangan dari struktur tabel adalah sebagai berikut.

1. Tabel Jurusan

Tabel III.16 Struktur Tabel Jurusan

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Ukuran** | **Keterangan** |
| id\_jurusan | Int | 10 | *Primary Key* |
| nama\_jurusan | Varchar | 25 |  |
| akreditasi\_jurusan | Varchar | 5 |  |
| biaya\_semester | Int | 100 |  |
| deskripsi | Varchar | 255 |  |
| id\_admin |  |  | *Foreign Key* |
| id\_member |  |  | *Foreign Key* |
| id\_kampus | Varchar | 10 | *Foreign Key* |

1. Tabel Kampus

Tabel III.17 Struktur Tabel Kampus

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Ukuran** | **Keterangan** |
| id\_kampus | Int | 10 | *Primary Key* |
| nama\_kampus | Varchar | 25 |  |
| akreditasi\_kampus | Varchar | 3 |  |
| deskripsi | Varchar | 100 |  |
| gambar | Varchar | 100 |  |
| brosur | Varchar | 255 |  |
| Status | Varchar | 10 |  |
| id\_admin | Varchar | 10 | *Foreign Key* |
| id\_member | Varchar | 10 | *Foreign Key* |

1. Tabel Lokasi

Tabel III.18 Struktur Tabel Lokasi

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Ukuran** | **Keterangan** |
| id\_lokasi | Int | 10 | *Primary Key* |
| id\_kampus | Int | 10 | *Foreign Key* |
| nama\_lokasi | Varchar | 255 |  |
| long | Varchar | 100 |  |
| Lat | Varchar | 100 |  |
| link | Varchar | 255 |  |
| id\_admin | Varchar | 10 | *Foreign Key* |
| id\_member | Varchar | 10 | *Foreign Key* |

1. Tabel User

Tabel III.19 Struktur Tabel User

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Ukuran** | **Keterangan** |
| id\_user | Int | 100 | *Primary Key* |
| nama\_user | Varchar | 25 |  |
| foto | Varchar | 255 |  |

1. Tabel Member

Tabel III.20 Struktur Tabel Member

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Ukuran** | **Keterangan** |
| id\_member | Int | 10 | *Primary Key* |
| username | Varchar | 25 |  |
| password | Varchar | 50 |  |

1. Tabel Admin

Tabel III.21 Struktur Tabel Admin

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Ukuran** | **Keterangan** |
| id\_admin | Int | 10 | *Primary Key* |
| username | Varchar | 25 |  |
| password | Varchar | 50 |  |

1. **Perancangan Antarmuka Aplikasi**

Perancangan antarmuka dilakukan untuk menggambarkan antarmuka aplikasi yang akan dibangun. Tabel III.22 s.d. Tabel III.55 menunjukkan perancangan antarmuka yang dilakukan.

**7.1 Perancangan Antarmuka FrontEnd**

Tabel III.22 Perancangan Antarmuka Menu

|  |  |
| --- | --- |
| No : A1 (Menu) | 1. Tekan Menu **Cari Kampus** untuk masuk ke A2 2. Tekan Menu **Cari Jurusan** untuk masuk ke A3 3. Tekan Menu **Cari Biaya** untuk masuk ke A4 4. Tekan Menu **Profile** untuk masuk ke A5 5. Tekan Menu **Cari Lokasi** untuk masuk ke A6 6. Tekan Menu **Panduan** untuk masuk ke A7 |
|  |
| Ukuran : 2500x1686px |

Tabel III.23 Perancangan Antarmuka Cari Kampus

|  |  |
| --- | --- |
| No : A2 (Menu Cari Kampus) | 1. Tekan **Akreditasi Kampus** untuk masuk ke A2.1 2. Tekan **Status Masuk** untuk masuk ke A2.2 |
|  |
| Font : Arial |

Tabel III.24 Perancangan Antarmuka Akreditasi Kampus

|  |  |
| --- | --- |
| No : A2.1 (Akreditasi Kampus) | 1. Tekan pilihan A,B atau C untuk masuk ke A2.1.1 |
|  |
| Font : Arial |

Tabel III.25 Perancangan antarmuka pilihan B

|  |  |
| --- | --- |
| No : A2.1.1 (Pilihan B) |  |
|  |
| Font : Arial |

Tabel III.26 Perancangan Antarmuka Status Kampus

|  |  |
| --- | --- |
| No : A2.2 (Status Kampus) | 1. Tekan **Negeri** atau **Swasta** untuk masuk ke A2.2.1 |
|  |
| Font : Arial |

Tabel III.27 Perancangan Antarmuka Swasta

|  |  |
| --- | --- |
| No : A2.2.1 (Swasta) |  |
|  |
| Font : Arial |

Tabel III.28 Perancangan Antarmuka Cari Jurusan

|  |  |
| --- | --- |
| No : A3 (Menu Cari Jurusan) | 1. Tekan **Akreditasi Jurusan** untuk masuk ke A3.1 2. Tekan **Tampil Jurusan** untuk masuk ke A3.2 |
|  |
| Font : Arial |

Tabel III.29 Perancangan Antarmuka Akreditasi Jurusan

|  |  |
| --- | --- |
| No : A3.1 (Akreditasi Jurusan) | 1. Tekan pilihan A,B, atau Cuntuk masuk ke A3.1.1 |
|  |
| Font : Arial |

Tabel III.30 Perancangan Antarmuka Akreditasi C

|  |  |
| --- | --- |
| No : A3.1.1 (Akreditasi C) |  |
|  |
| Font : Arial |

Tabel III.31 Perancangan Antarmuka Jurusan

|  |  |
| --- | --- |
| No : A3.2 (Tampil Jurusan) | 1. Tekan **UNIKOM, UNPAS, TEL-U** atau **ITENAS** untuk masuk ke A.3.2.1 |
|  |
| Font : Arial |

Tabel III.32 Perancangan Antarmuka Unikom

|  |  |
| --- | --- |
| No : A3.2.1 (UNIKOM) |  |
|  |
| Font : Arial |

Tabel III.33 Perancangan Antarmuka Cari Biaya

|  |  |
| --- | --- |
| No : A4 (Menu Cari Biaya) | 1. Tekan **Rentang Biaya** untuk masuk ke A4.1 2. Tekan **Tampil Biaya** untuk masuk ke A4.2 |
|  |
| Font : Arial |

Tabel III.34 Perancangan Antarmuka Rentang Biaya

|  |  |
| --- | --- |
| No : A4.1 (Rentang Biaya) | 1. Tekan **rentang biayanya** untuk masuk ke A4.1.1 |
|  |
| Font : Arial |

Tabel III.35 Perancangan Antarmuka Rentang Biaya

|  |  |
| --- | --- |
| No : A4.1.1 (rentang 9jt – 10jt) |  |
|  |
| Font : Arial |

Tabel III.36 Perancangan Antarmuka Tampil Biaya

|  |  |
| --- | --- |
| No : A4.2 (Tampil Biaya) | 1. Tekan **UNIKOM, UNPAS, TEL-U** atau **ITENAS** untuk masuk ke A4.2.1 |
|  |
| Font : Arial |

Tabel III.37 Perancangan Antarmuka Tampil Biaya

|  |  |
| --- | --- |
| No : A4.2.1 (UNIKOM) | 1. Tekan **Download Brosur** untuk mengunduhnya |
|  |
| Font : Arial |

Tabel III.38 Perancangan Antarmuka Menu Profile

|  |  |
| --- | --- |
| No : A5 (Menu Profile) |  |
|  |
| Font : Arial |

Tabel III.39 Perancangan Antarmuka Menu Lokasi

|  |  |
| --- | --- |
| No : A6 (Menu Lokasi) | 1. Tekan **UNIKOM, UNPAS, TEL-U** atau **ITENAS** untuk masuk lokasi gmaps |
|  |
| Font : Arial |

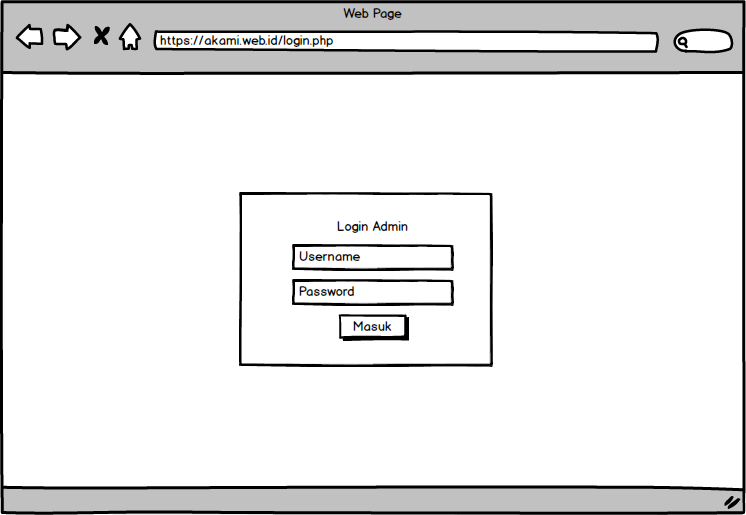
Tabel III.40 Perancangan Antarmuka Menu Panduan

|  |  |
| --- | --- |
| No : A7 (Menu Panduan) |  |
|  |
| Font : Arial |

**7.2 Perancangan Antarmuka BackEnd**

1. Perancangan antarmuka login admin

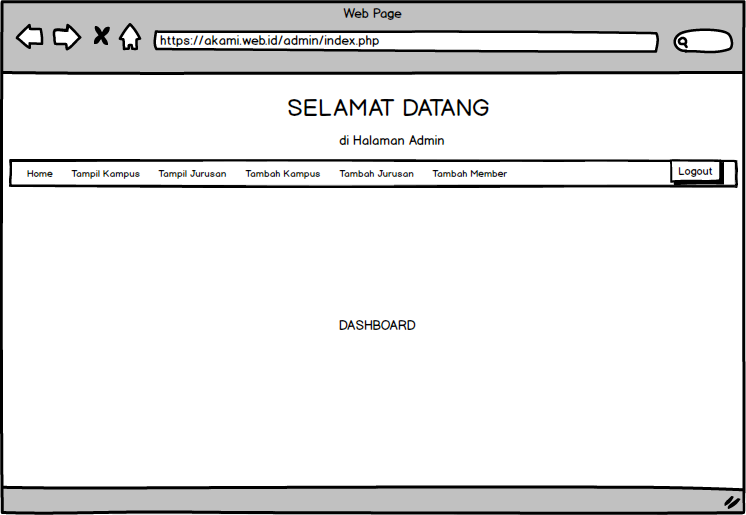
Berikut adalah tampilan antar muka ketika login



Gambar III.31 Perancangan antarmuka login admin

1. Perancangan antarmuka dashboard

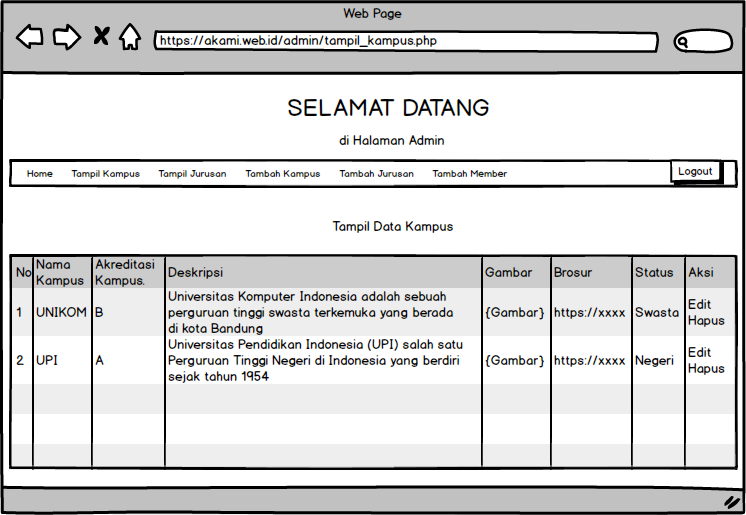
Berikut adalah tampilan antar muka ketika berhasil login



Gambar III.32 Perancangan antarmuka dashboard admin

1. Perancangan antarmuka Tampil Kampus (Admin)

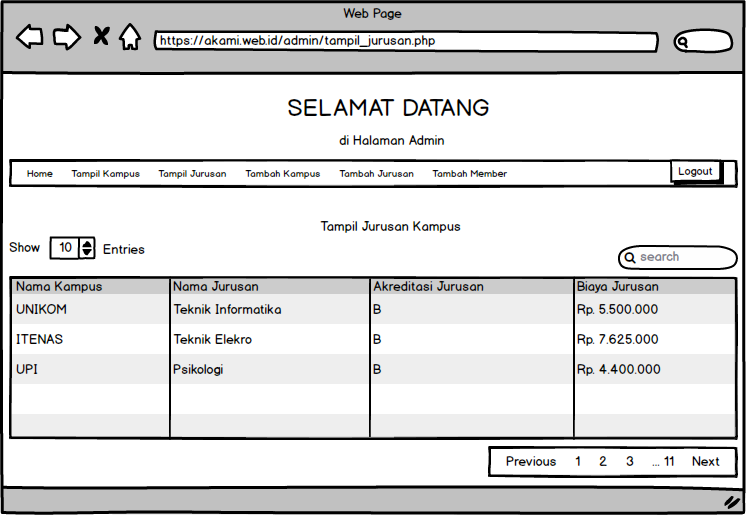
Berikut adalah antar muka menampilkan semua kampus



Gambar III.33 Perancangan antarmuka tampil kampus admin

1. Perancangan antarmuka Tampil Jurusan (Admin)

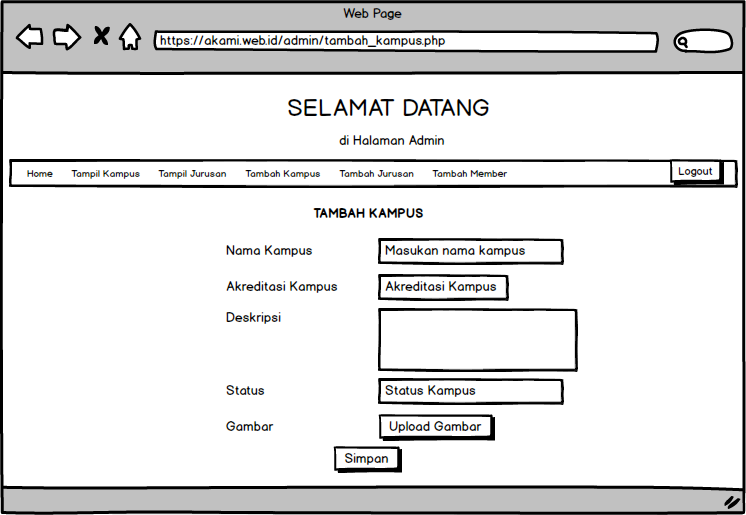
Berikut adalah antar muka menampilkan semua jurusan kampus



Gambar III.34 Perancangan antarmuka tampil jurusan admin

1. Perancangan antarmuka Tambah Kampus (Admin)

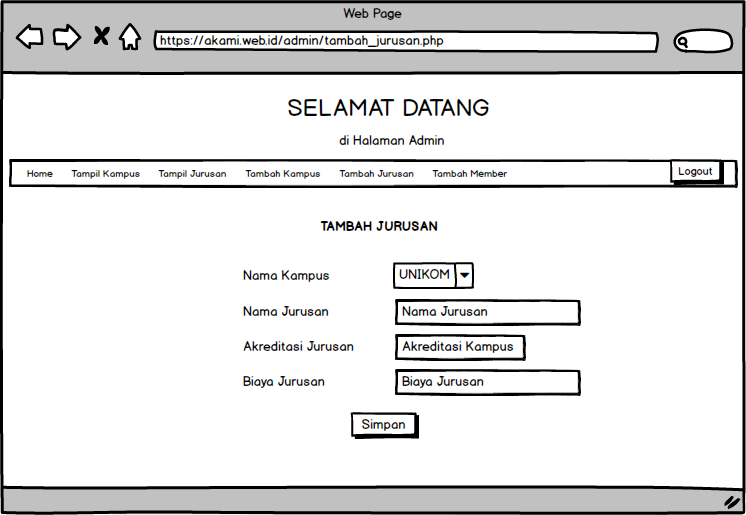
Berikut adalah antar muka ketika menambahkan kampus



Gambar III.35 Perancangan antarmuka tambah kampus admin

1. Perancangan antarmuka Tambah Jurusan (Admin)

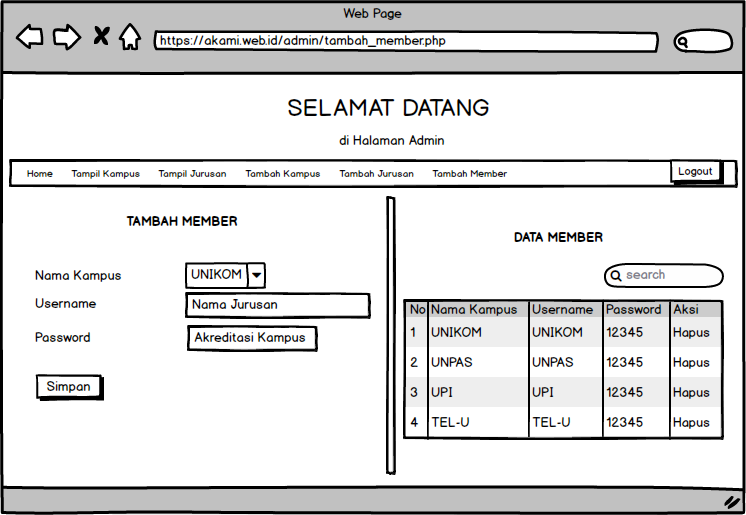
Berikut adalah perancangan antar muka ketika menambahkan jurusan



Gambar III.36 Perancangan antarmuka tambah jurusan admin

1. Perancangan antarmuka Tambah Member (Admin)

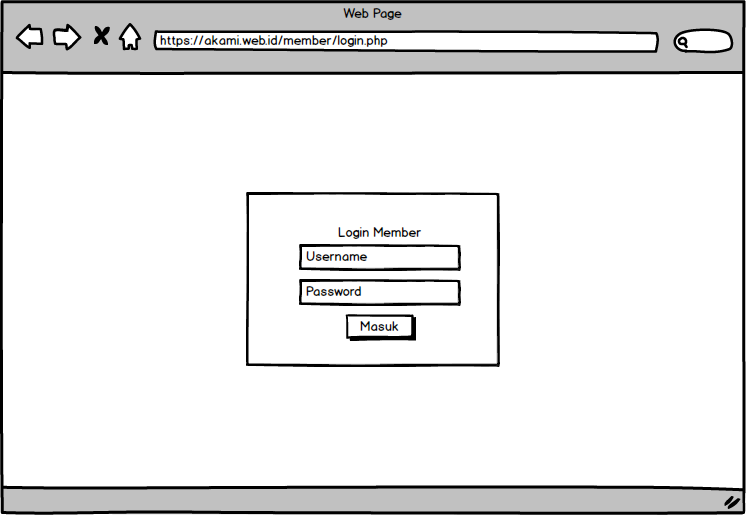
Berikut adalah perancangan antar muka ketika menambahkan member



Gambar III.37 Perancangan antarmuka tambah member admin

1. Perancangan antarmuka login (Member)

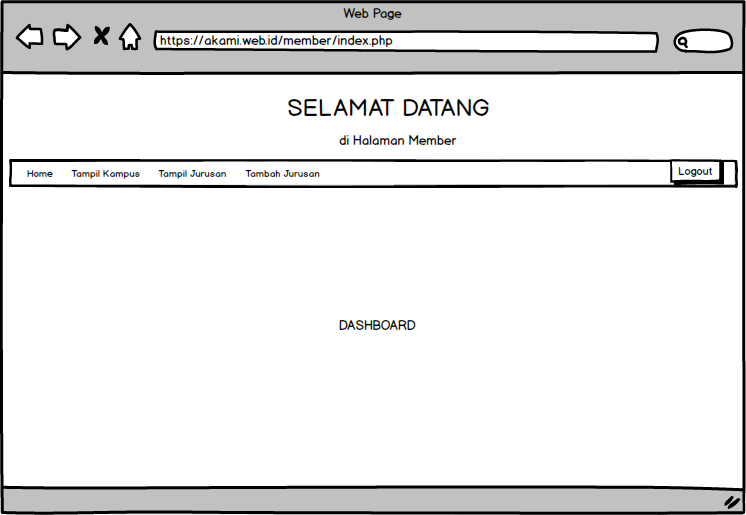
Berikut adalah antar muka login



Gambar III.38 Perancangan antarmuka login member

1. Perancangan antarmuka dashboard (Member)

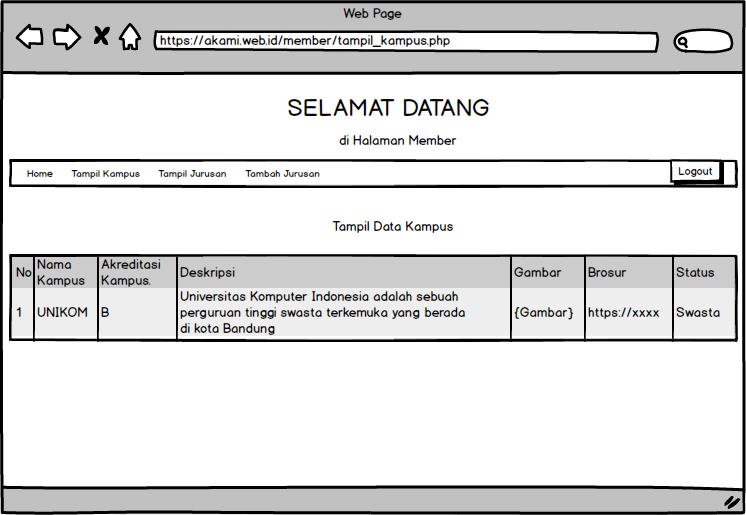
Berikut adalah antarmuka dashboard ketika login berhasil



Gambar III.39 Perancangan antarmuka dashboard member

1. Perancangan Antarmuka Tampil Kampus (member)

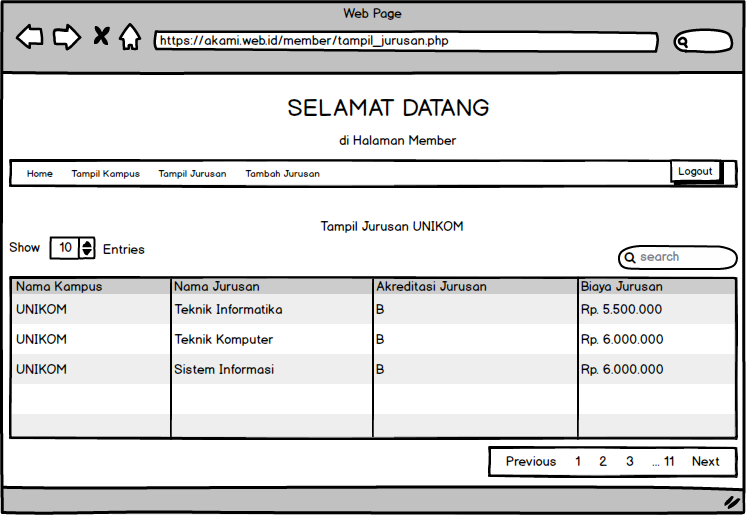
Berikut adalah antarmuka menampilkan kampus



Gambar III.40 Perancangan antarmuka tampil kampus member

1. Perancangan Antarmuka Tampil Jurusan (Member)

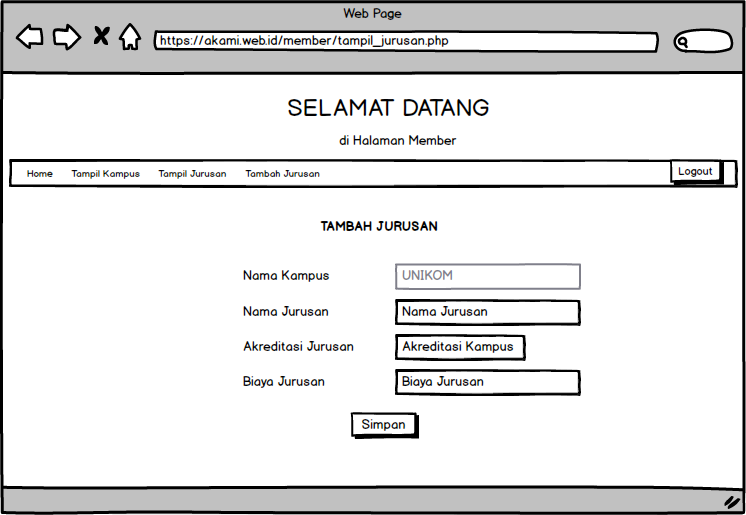
Berikut adalah antar muka menampilkan jurusan kampus



Gambar III.41 Perancangan antarmuka tampil jurusan member

1. Perancngan Antarmuka Tambah Jurusan (Member)

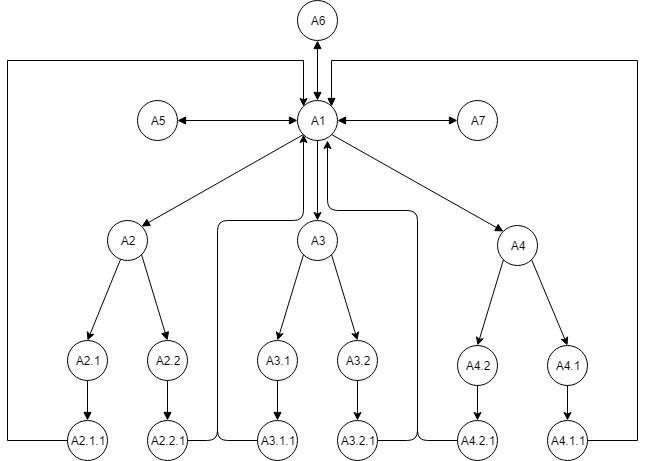
Berikut adalah antar muka ketika menambahkan jurusan kampus



Gambar III.42 Perancangan antarmuka tampil jurusan member

**3.12 Perancangan Jaringan Semantik Antarmuka**

Jaringan semantik merupakan gambaran mengenai hubungan navigasi menu dari satu halaman ke halaman lainnya. Perancangan jaringan semantik pada aplikasi ditunjukkan pada Gambar 3.41.



Gambar 3.41 Perancangan Jaringan Semantik Antarmuka

**BAB IV**

**IMPLEMENTASI DAN PENGUJIAN SISTEM**

* 1. **Implementasi Sistem**

Tahap implementasi merupakan tahap penciptaan perangkat lunak, tahap kelanjutan dari kegiatan perancangan sistem. Tahap ini merupakan tahap dimana sistem siap untuk dioperasikan, yang terdiri dari penjelasan mengenai lingkungan implementasi, dan implementasi program.

**4.1.1 Implementasi Perangkat Lunak**

Spesifikasi perangkat lunak yang digunakan dalam membangun aplikasi virtual asisten pencarian informasi Perguruan Tinggi di Kota Bandung dengan menggunakan metode forward chaining dapat dilihat pada tabel 4.1.

**Tabel 4.1 Implementasi Spesifikasi Perangkat Lunak**

|  |  |  |
| --- | --- | --- |
| No | Perangkat Lunak | Keterangan |
| 1 | Sistem Operasi | Android v5 |
| 2 | Bahasa Pemograman | PHP, Javascript |
| 3 | DBMS | MySQL |
| 4 | Web Browser | Google Chrome |
| 5 | Code Editor | Sublime Text |
| 6 | UML Modeler | Draw.io |

**4.1.2 Implementasi Perangkat Keras**

Spesifikasi perangkat keras yang digunakan dalam membangun aplikasi virtual asisten pencarian informasi kampus di Kota Bandung dengan menggunakan forward chaining.

**Tabel 4.2 Implementasi Spesifikasi Perangkat Keras**

|  |  |
| --- | --- |
| Perangkat | Spesifikasi |
| Processor | Mediatex |
| Sistem Operasi | Android v5 |
| RAM | 3 GB |
| ROM | 32 GB |
| Layar | 5.5 Inch |

**4.1.3 Implementasi Web Hosting**

Kebutuhan web hosting pada penelitian ini digunakan untuk keperluan aplikasi virtual asisten pencarian informasi kampus di Kota Bandung agar sistem dapat berjalan maka perlu di upload di web hosting. Adapun spesifikasi web hosting dan nama domain (situs) yang dipakai adalah :

Nama Domain / Situs : <https://akami.web.id>

**Tabel 4.3 Implementasi Web Hosting**

|  |  |
| --- | --- |
| *Disk Space* | 400 MB |
| *Bandwith* | Unlimitid |
| *Development* | 1. PHP version 5.0.X with Zend Optimizer 2. PERL version 5.8.3 3. MySQL 4. SSL 5. CPanel 6. phpMyAdmin |

**4.2 Implementasi Basis Data**

Database yang digunakan dalam pembuatan aplikasi ini yaitu menggunakan phpMyAdmin yang merupakan webbase control panel untuk MySQL yang telah terinstall didalam komputer kita, dari sini kita dapat membuat memodifikasi menghapus database dan table data yang ada pada MySQL. Implementasi database dalam bahasa SQL adalah sebagai berikut :

1. *Tabel Admin*

CREATE TABLE `tbl\_admin` (

`id\_admin` int(10) NOT NULL,

`nama\_admin` varchar(25) NOT NULL,

`username` varchar(25) NOT NULL,

`password` varchar(25) NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=utf8;

1. *Tabel User*

CREATE TABLE `tbl\_user` (

`id\_user` varchar(100) NOT NULL,

`nama\_user` varchar(25) NOT NULL,

`foto` varchar(255) NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

1. *Tabel Member*

CREATE TABLE `tbl\_member` (

`id\_member` int(10) NOT NULL,

`nama\_member` varchar(25) NOT NULL,

`username` varchar(25) NOT NULL,

`password` varchar(25) NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=utf8;

1. **Tabel Jurusan**

CREATE TABLE `tbl\_jurusan` (

`id\_jurusan` int(10) NOT NULL,

`nama\_jurusan` varchar(25) NOT NULL,

`akreditasi\_jurusan` varchar(5) NOT NULL,

`biaya\_semester` int(100) NOT NULL,

`deskripsi` varchar(1000) NOT NULL,

`id\_kampus` int(10) NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

1. **Tabel Kampus**

CREATE TABLE `tbl\_kampus` (

`id\_kampus` int(10) NOT NULL,

`nama\_kampus` varchar(25) NOT NULL,

`akreditasi\_kampus` varchar(3) NOT NULL,

`deskripsi` varchar(1000) NOT NULL,

`gambar` varchar(100) NOT NULL,

`brosur` varchar(255) NOT NULL,

`status` varchar(10) NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

1. **Tabel Lokasi**

CREATE TABLE `tbl\_lokasi` (

`id\_lokasi` int(10) NOT NULL,

`id\_kampus` int(10) NOT NULL,

`nama\_lokasi` varchar(255) NOT NULL,

`long` varchar(100) NOT NULL,

`lat` varchar(100) NOT NULL,

`link` varchar(255) NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

**4.3 Implementasi Antarmuka**

Implementasi antarmuka merupakan pemaparan tentang tampilan antarmuka sistem yang dibangun berdasarkan kepada rancangan antarmuka yang telah dilakukan pada analisis sebelumnya. Implementasi dalam penelitian ini terbagi menjadi 2 bagian yakni implementasi antarmuka backend dan implementasi antarmuka LINE Bot. Implementasi antarmuka yang dipaparkan diwakili dengan nama menu, deskripsi menu serta nama file.

**Tabel 4.4 Implementasi Antarmuka Admin**

|  |  |  |
| --- | --- | --- |
| **Menu** | **Deskripsi** | **Nama File** |
| Login Admin | Menampilkan form isian username dan password untuk mendapatkan hak akses untuk menggunakan fitur – fitur aplikasi. | Admin/login.php |
| Home / Dasbor | Digunakan untuk melihat home / dasbor admin | Admin/Index.php |
| Tampil Kampus | Digunakan untuk menampilkan informasi kampus yang ada di Kota Bandung | Admin/Tampil\_kampus.php |
| Tampil Jurusan | Digunakan untuk menampilkan jurusan tiap kampus yang ada di Kota Bandung | Admin/Tampil\_jurusan.php |
| Tambah Kampus | Digunakan untuk menambah data kampus yang ada di Kota Bandung | Admin/Tambah\_kampus.php |
| Tambah Jurusan | Digunakan untuk menambah data jurusan tiap kampus yang ada di Kota Bandung | Admin/Tambah\_jurusan.php |
| Tambah Member | Digunakan untuk menambah member | Admin/Tambah\_Member.php |

**Tabel 4.5 Implementasi Antarmuka Member**

|  |  |  |
| --- | --- | --- |
| **Menu** | **Deskripsi** | **Nama File** |
| Login Member | Menampilkan form isian username dan password untuk mendapatkan hak akses untuk menggunakan fitur – fitur aplikasi. | Member/login.php |
| Dasbor | Digunakan untuk melihat home / dasbor admin | Member/Index.php |
| Tampil Kampus | Digunakan untuk menampilkan informasi kampus yang ada di Kota Bandung | Member/Tampil\_kampus.php |
| Tampil Jurusan | Digunakan untuk menampilkan jurusan tiap kampus yang ada di Kota Bandung | Member/Tampil\_jurusan.php |
| Tambah Jurusan | Digunakan untuk menambah data jurusan tiap kampus yang ada di Kota Bandung | Member/Tambah\_jurusan.php |

**4.6 Implementasi Antarmuka LINE Bot Akami**

|  |  |  |
| --- | --- | --- |
| **Menu** | **Deskripsi** | **Nama File** |
| Webhook | Sebagai akses LINE Bot ke dalam server aplikasi | bot.php |
| Cari Kampus | Menu yang menampilkan daftar pencarian kampus | Cari\_kampus.php |
| Cari Jurusan | Menu yang menampilkan daftar jurusan tiap kampus | Cari\_jurusan.php |
| Cari Biaya | Menu yang menampilkan daftar biaya tiap kampus | Cari\_biaya.php |
| Cari Lokasi | Menu yang menampilkan lokasi kampus | Cari\_lokasi.php |
| Profile | Menu yang menampilkan data profile Line pengguna | profile.php |
| Bantuan | Menu yang menampilkan informasi bantuan mengenai Line Bot Akami | bantuan.php |

**4.2 Pengujian Sistem**

Merupakan tahap untuk menemukan kesalahan ataupun kekurangan pada sistem perangkat lunak yang dibangun. Hal ini bertujuan untuk mengetahui apakah perangkat lunak yang dibangun telah memenuhi kriteria sesuai dengan tujuan perancangan sebelumnya. Pengujian yang dilakukan terhadap sistem yang dibangun terbagi menjadi dua langkah diantaranya pengujian alpha dan pengujian beta.

* + 1. **Pengujian Alpha**

Pengujian alpha yang dilakukan menggunakan metode black box dengan tujuan untuk mengetahui fungsionalitas sistem apakah telah bekerja sesuai dengan yang hasil diharapkan. Skenario pengujian yang akan dilakukan terbagi ke dalam dua sistem yakni skenario pengujian sistem website HelloUMKM dan skenario pengujian sistem LINE Bot

**4.2.1.1 Skenario Pengujian Sistem Admin**

Skenario pengujian yang akan dilakukan pada sistem website backend akami.web.id dapat dilihat pada tabel 4.7.

**Tabel 4.7 Skenario Pengujian Sistem Admin**

|  |  |  |
| --- | --- | --- |
| **Komponen Uji** | **Skenario Pengujian** | **Jenis Pengujian** |
| Login | Melakukan login dengan akun admin | Black box |
| Tampil Kampus | Menghapus data kampus | Black box |
| Merubah data kampus |
| Tampil Jurusan | Menghapus data jurusan | Black box |
| Merubah data jurusan |
| Tambah Kampus | Menambah data kampus | Black box |
| Tambah Jurusan | Menambah data jurusan | Black box |
| Tambah Member | Menambah member | Black box |

**4.2.1.2 Skenario Pengujian Sistem Member**

**Tabel 4.8 Skenario Pengujian Sistem Member**

|  |  |  |
| --- | --- | --- |
| **Komponen Uji** | **Skenario Pengujian** | **Jenis Pengujian** |
| Login | Melakukan login dengan akun member | Black box |
| Tampil Kampus | Menghapus data kampus | Black box |
| Merubah data kampus |
| Tampil Jurusan | Menghapus data jurusan | Black box |
| Merubah data jurusan |
| Tambah Jurusan | Menambah data jurusan | Black box |

**4.2.1.2 Skenario Pengujian Sistem LINE Bot**

**Tabel 4.9 Skenario Pengujian Sistem Member**

|  |  |  |
| --- | --- | --- |
| **Komponen Uji** | **Skenario Pengujian** | **Jenis Pengujian** |
| Cari Kampus | Mencari Kampus berdasarkan akreditasi | Akurasi |
| Mencari Kampus berdasarkan status |
| Cari Jurusan | Mencari jurusan berdasarkan akreditasi | Akurasi |
| Mencari jurusan berdasarkan kampus yang dipilih |
| Cari Biaya | Mencari biaya berdasarkan rentang biaya | Akurasi |
| Mencari biaya berdasarkan kampus yang dipilih |
| Cari Lokasi | Mencari kampus berdasarkan nama kampus yang dipilih | Akurasi |
| Profile | Menampilkan profile pengguna Line | Akurasi |
| Bantuan | Menampilan bantuan | Akurasi |

**4.2.1.3 Kasus dan Hasil Pengujian**

1. Kasus dan Hasil Pengujian Sistem Website Chatbot

1) Login

a) Melakukan login dengan akun admin

**Tabel 4.10. Pengujian Login Admin**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kasus dan Hasil Uji (Data Benar)** | | | |
| **Data Masukan** | **Yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| Username : admin  Password : admin | Sistem akan menerima akses login dan dialihkan ke halaman dashboard | Dialihkan ke halaman dashboard | [√] Diterima  [ ] Ditolak |
| **Kasus dan Hasil Uji (Data Salah)** | | | |
| Data Masukan | Yang Diharapkan | Hasil Pengujian | Kesimpulan |
| Username : admin  Password : 12345 | Sistem akan menolak akses login dan menampilkan pesan “Login gagal” | Login gagal dan tampl pesan “Login gagal” | [√] Diterima  [ ] Ditolak |
| Username : (kosong)  Password : (kosong) | Sistem akan menolak akses login dan menampilkan pesan “harap isi bidang ini” | Login gagal dan tampl pesan “harap isi bidang ini” | [√] Diterima  [ ] Ditolak |

1. Melakukan login dengan akun member

**Tabel 4.11. Pengujian Login Member**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kasus dan Hasil Uji (Data Benar)** | | | |
| **Data Masukan** | **Yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| Username : Unikom  Password : 12345 | Sistem akan menerima akses login dan dialihkan ke halaman dashboard | Dialihkan ke halaman dashboard | [√] Diterima  [ ] Ditolak |
| **Kasus dan Hasil Uji (Data Salah)** | | | |
| Data Masukan | Yang Diharapkan | Hasil Pengujian | Kesimpulan |
| Username : Unikom  Password : Unikom | Sistem akan menolak akses login dan menampilkan pesan “Login gagal” | Login gagal dan tampl pesan “Login gagal” | [√] Diterima  [ ] Ditolak |
| Username : (kosong)  Password : (kosong) | Sistem akan menolak akses login dan menampilkan pesan “Username dan Password harus diisi” | Login gagal dan tampl pesan “Username dan Password harus diisi” | [√] Diterima  [ ] Ditolak |

1. Manajemen Admin
2. Tampil Kampus

Tabel 4.12. Melihat Daftar Kampus

|  |  |  |  |
| --- | --- | --- | --- |
| **Kasus dan Hasil Uji (Data Benar)** | | | |
| **Data Masukan** | **Yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| Pilih menu tampil kampus yang akan dilihat detail datanya, lalu klik menu tampil kampus | Sistem akan mengalihkan ke halaman detail data kampus | Dialihkan ke halaman tampil kampus | [√] Diterima  [ ] Ditolak |

1. Hapus Data Kampus

Tabel 4.13. Melihat Daftar Kampus

|  |  |  |  |
| --- | --- | --- | --- |
| **Kasus dan Hasil Uji (Data Benar)** | | | |
| **Data Masukan** | **Yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| Pilih data kampus yang akan dilakukan penghapusan, lalu klik tombol hapus di kolom aksi | Sistem akan melakukan penghapusan data kampus dan menampilkan pesan“Yakin ingin menghapus data ?” | Data kampus berhasil dihapus | [√] Diterima  [ ] Ditolak |

1. Ubah Data Kampus

Tabel 4.14. Melihat Daftar Kampus

|  |  |  |  |
| --- | --- | --- | --- |
| **Kasus dan Hasil Uji (Data Benar)** | | | |
| **Data Masukan** | **Yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
|  |  |  | [√] Diterima  [ ] Ditolak |

1. Tampil Jurusan

Tabel 4.15. Melihat Data Jurusan

|  |  |  |  |
| --- | --- | --- | --- |
| **Kasus dan Hasil Uji (Data Benar)** | | | |
| **Data Masukan** | **Yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| Pilih menu tampil jurusan yang akan dilihat detail datanya, lalu klik menu tampil jurusan | Sistem akan mengalihkan ke halaman detail data jurusan | Dialihkan ke halaman tampil jurusan | [√] Diterima  [ ] Ditolak |

1. Tambah Kampus

Tabel 4.16 Menambah Data Kampus

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Kasus dan Hasil Uji (Data Benar)** | | | | | | |
| **Data Masukan** | | **Yang Diharapkan** | | **Hasil Pengujian** | | **Kesimpulan** |
| Nama Kampus : **UNPAD**  Akreditasi Kampus : **A**  Gambar : [**https://example/unpad.jpg**](https://example/unpad.jpg)  Brosur : **https:example/unpad.zip**  Status : **Negeri**  Deskripsi :  **Unpad sebuah perguruan tinggi negeri di Bandung, Jawa Barat, Indonesia.** | | Sistem akan melakukan penambahan data kampus dan menampilkan pesan“Yakin ingin menambah data ?” | | Data kampus berhasil ditambah | | [√] Diterima  [ ] Ditolak |
| **Kasus dan Hasil Uji (Data Salah)** | | | | | | |
| Data Masukan | Yang Diharapkan | | Hasil Pengujian | | Kesimpulan | |
| Nama Kampus : {Kosong}  Akreditasi Kampus :  Gambar : [**https://example/unpad.jpg**](https://example/unpad.jpg)  Brosur : **https:example/unpad.zip**  Status : **Negeri**  Deskripsi :  **Unpad sebuah perguruan tinggi negeri di Bandung, Jawa Barat, Indonesia.** | Sistem akan menolak penambahan data dan menampilkan pesan “data harus diisi lengkap” | | Data kampus gagal ditambah | | [√] Diterima  [ ] Ditolak | |
| Username : (kosong)  Password : (kosong) | Sistem akan menolak akses login dan menampilkan pesan “Username dan Password harus diisi” | | Login gagal dan tampl pesan “Username dan Password harus diisi” | | [√] Diterima  [ ] Ditolak | |

1. Tambah Jurusan

Tabel 4.17 Menambah Data Jurusan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Kasus dan Hasil Uji (Data Benar)** | | | | | | |
| **Data Masukan** | | **Yang Diharapkan** | | **Hasil Pengujian** | | **Kesimpulan** |
| Nama Kampus : **UNPAD**  Nama Jurusan : **Teknik Informatika**  Akreditasi Jurusan : **A**  Biaya Jurusan : 4500000 | | Sistem akan melakukan penambahan data jurusan dan menampilkan pesan“Yakin ingin menambah data ?” | | Data jurusan berhasil ditambah | | [√] Diterima  [ ] Ditolak |
| **Kasus dan Hasil Uji (Data Salah)** | | | | | | |
| Data Masukan | Yang Diharapkan | | Hasil Pengujian | | Kesimpulan | |
| Nama Kampus : **UNPAD**  Nama Jurusan : {kosong}  Akreditasi Jurusan : **{kosong}**  Biaya Jurusan : {kosong} | Sistem akan menolak penambahan data dan menampilkan pesan “data harus diisi lengkap” | | Data jurusan gagal ditambah | | [√] Diterima  [ ] Ditolak | |

1. Tambah Member

Tabel 4.18 Menambah Data Member

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Kasus dan Hasil Uji (Data Benar)** | | | | | | |
| **Data Masukan** | | **Yang Diharapkan** | | **Hasil Pengujian** | | **Kesimpulan** |
| Nama Kampus : **UNIKOM**  Username : **Unikom**  Password : **12345** | | Sistem akan melakukan penambahan data member dan menampilkan pesan“Yakin ingin menambah data ?” | | Data member berhasil ditambah | | [√] Diterima  [ ] Ditolak |
| **Kasus dan Hasil Uji (Data Salah)** | | | | | | |
| Data Masukan | Yang Diharapkan | | Hasil Pengujian | | Kesimpulan | |
| Nama Kampus : **UNIKOM**  Username : **{kosong}**  Password : **{kosong}** | Sistem akan menolak penambahan data dan menampilkan pesan “data harus diisi lengkap” | | Data member gagal ditambah | | [√] Diterima  [ ] Ditolak | |

1. Manajemen Member
2. Tampil Kampus

Tabel 4.19 Tampil Data Kampus

|  |  |  |  |
| --- | --- | --- | --- |
| **Kasus dan Hasil Uji (Data Benar)** | | | |
| **Data Masukan** | **Yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| Pilih menu tampil kampus yang akan dilihat detail datanya, lalu klik menu tampil kampus | Sistem akan mengalihkan ke halaman detail data kampus | Dialihkan ke halaman tampil kampus | [√] Diterima  [ ] Ditolak |

1. Ubah Data Kampus

Tabel 4.20. Ubah Data Kampus

|  |  |  |  |
| --- | --- | --- | --- |
| **Kasus dan Hasil Uji (Data Benar)** | | | |
| **Data Masukan** | **Yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
|  |  |  | [√] Diterima  [ ] Ditolak |

1. Tampil Jurusan

Tabel 4.21. Melihat Data Jurusan

|  |  |  |  |
| --- | --- | --- | --- |
| **Kasus dan Hasil Uji (Data Benar)** | | | |
| **Data Masukan** | **Yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| Pilih menu tampil jurusan yang akan dilihat detail datanya, lalu klik menu tampil jurusan | Sistem akan mengalihkan ke halaman detail data jurusan | Dialihkan ke halaman tampil jurusan | [√] Diterima  [ ] Ditolak |

1. Hapus Data Jurusan

Tabel 4.22. Menghapus data jurusan

|  |  |  |  |
| --- | --- | --- | --- |
| **Kasus dan Hasil Uji (Data Benar)** | | | |
| **Data Masukan** | **Yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| Pilih data jurusan yang akan dilakukan penghapusan, lalu klik tombol hapus di kolom aksi | Sistem akan melakukan penghapusan data kampus dan menampilkan pesan“Yakin ingin menghapus data ?” | Data kampus berhasil dihapus | [√] Diterima  [ ] Ditolak |

1. Tambah Jurusan

Tabel 4.23 Menambah Data Jurusan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Kasus dan Hasil Uji (Data Benar)** | | | | | | |
| **Data Masukan** | | **Yang Diharapkan** | | **Hasil Pengujian** | | **Kesimpulan** |
| Nama Kampus : **UNIKOM**  Nama Jurusan : **Teknik Informatika**  Akreditasi Jurusan : **B**  Biaya Jurusan : 5500000 | | Sistem akan melakukan penambahan data jurusan dan menampilkan pesan“Yakin ingin menambah data ?” | | Data jurusan berhasil ditambah | | [√] Diterima  [ ] Ditolak |
| **Kasus dan Hasil Uji (Data Salah)** | | | | | | |
| Data Masukan | Yang Diharapkan | | Hasil Pengujian | | Kesimpulan | |
| Nama Kampus : **UNIKOM**  Nama Jurusan : **{kosong}**  Akreditasi Jurusan : **{kosong}**  Biaya Jurusan : **{kosong}** | Sistem akan menolak penambahan data dan menampilkan pesan “data harus diisi lengkap” | | Data jurusan gagal ditambah | | [√] Diterima  [ ] Ditolak | |

1. Kasus dan Hasil Pengujian Sistem LINE Bot

Pertanyaan – pertanyaan yang akan dijadikan masukkan pada LINE Bot terlihat pada tabel 4.24.

Tabel 4.24. Data Uji Pertanyaan LINE Bot

|  |  |
| --- | --- |
| No | Pertanyaan |
| 1 | Min di Bandung ada Kampus apa saja ? |
| 2 | Tolong cari biaya yang ada tiap kampus ? |
| 3 | Carikan akreditasi kampus nya A? |
| 4 | Carikan saya status kampus negeri di kota Bandung ? |
| 5 | Cari jurusan cari |
| 6 | Cari akreditasi jurusan |
| 7 | Jurusan apa saja sih yang ada di unikom? |
| 8 | Cari kamps dong |
| 9 | tknik informatika |
| 10 | Cek Biaya unikom |
| 11 | Cek biaya unkom |
| 12 | Tampilkan biaya kampus |
| 13 | Saya mau cari rentang biaya kampus |
| 14 | Ada kampus apa saja ini |
| 15 | Saya mau cari lokasi kampus |

* + - 1. Cari Kampus

|  |  |  |  |
| --- | --- | --- | --- |
| Pertanyaan | Perhitungan Jaro Winkler | Kata Kunci | Kesimpulan |
| Min di Bandung ada kampus apa saja ? | Min = 0  di = 0  Bandung = 0  ada = 0  kampus = 1  apa = 0,5  saja = 0 | kampus | Sesuai |
| Carikan akreditasi kampus nya A | Carikan = 0  akreditasi = 1  kampus = 1  nya = 0  A = 0 | Akreditasi | Sesuai |
| Carikan saya status kampus negeri di bandung | Carikan = 0  Saya = 0  Status = 1  kampus = 1 | status | Sesuai |
| Cari kmpus di Bandung | Cari = 0  kmpus = 0,950  di = 0  Bandung = 0 | kampus | Sesuai |
| Ada kampus apa saja ini | Ada = 0  Kampus = 1  Apa = 0,5  Saja = 0  Ini = 0 | kampus | Sesuai |

* + - 1. Cari Jurusan

|  |  |  |  |
| --- | --- | --- | --- |
| Pertanyaan | Perhitungan Jaro Winkler | Kata Kunci | Kesimpulan |
| Cari jurusan | cari = 0  jurusan = 1 | jurusan | Sesuai |
| cari jrusan | cari = 0  jrusan = 0.952 | jrusan | Sesuai |
| cari akreditasi jurusan | Cari = 0  Akreditasi = 1  Jurusan = 1 | akreditasi | Tidak sesuai |
| Jurusan apa saja yang ada di unikom | jurusan = 1  apa = 0,5  saja = 0  yang = 0  ada = 0  di = 0  unikom = 1 | unikom | sesuai |
| Teknik informatika | teknik informatika = 1 | Teknik informatika | Tidak sesuai |

* + - 1. Cari Biaya

|  |  |  |  |
| --- | --- | --- | --- |
| Pertanyaan | Perhitungan Jaro Winkler | Kata Kunci | Kesimpulan |
| Cek biaya unikom | Cek = 0  biaya = 1  unikom = 1 | biaya unikom | Sesuai |
| rentang biaya | rentang = 1  biaya = 1 | rentang | Sesuai |

4) Cari Lokasi

|  |  |  |  |
| --- | --- | --- | --- |
| Pertanyaa | Perhitungan Jaro Winkler | Kata Kunci | Kesimpulan |
| Cek lokasi | Cek = 0  lokasi = 1 | lokasi | Sesuai |
| Tampilkan profile | Cek = 0  lokasi = 1 | lokasi | Sesuai |
| Lokasi unikom | Cek = 0  lokasi = 1  kampu = 0,952  dong | lokasi | Sesuai |
| Cari lokasi kampus terdekat | cari = 0  lokasi = 1  kampus = 1 | lokasi | Tidak sesuai |

4.2.1.4 Kesimpulan Hasil Pengujian Alpha

**1. Kesimpulan Hasil Pengujian Website Akami Bot**

Berdasarkan dari hasil pengujian alpha dengan menggunakan metode black box pada keseluruhan fungsionalitas perangkat lunak yang dibangun dengan masukkan menggunakan sample data uji, maka dapat diperoleh kesimpulan bahwa fungsionalitas perangkat lunak yang dibangun telah sesuai dengan keluaran yang diharapkan

* 1. **Kesimpulan Hasil Pengujian LINE Bot**

Pada LINE Bot ditetapkan nilai minimum kesalahan kata atau lower-bound pada perhitungan Jaro-Winkler Distance sebesar 0.86, hal ini untuk mengurangi tingkat ambiguitas kata yang diberikan pengguna dengan kata kunci yang telah ditetapkan. Sehingga pada pengujian alpha yang dilakukan pada LINE Bot dapat diperoleh akurasi dengan perhitungan sebagai berikut :

* Total Pertanyaan : 15
* Total Jawaban sesuai : 13
* Akurasi = (Total Jawaban Sesuai / Total Pertanyaan) \* 100%

= (13 / 15) \* 100 %

= 0.86%

* + 1. **Pengujian Beta**

Pengujian beta berfokus pada kualitas perangkat lunak yang telah dibangun, hal ini bertujuan untuk mengetahui sejauh mana kualitas perangkat lunak yang dibangun jika dilihat dari sisi pengguna. Pada pengujian beta dilakukan pengujian secara langsung kepada anak sekolah SMA/K sederajat sebagai pengguna aplikasi.

**4.2.2.1 Skenario Pengujian Beta Dengan Pengguna**

Pengujian beta yang dilakukan pengguna menggunakan metode kuesioner dengan perhitungan skala likert. Skala likert terdiri dari pernyataan – pernyataan yang dilengkapi dengan 5 pilihan tingkat persetujuan responden. Pernyataan yang akan diberikan kepada responden pada pengujian ini dapat dilihat pada tabel 4.25

**Tabel 4.25 Daftar Pernyataan Kuisoner**

|  |  |
| --- | --- |
| **No** | **Pertanyaan** |
| 1 | Apakah aplikasi Akami Bot membantu dalam pencarian kampus ? |
| 2 | Saya merasa aplikasi ini mudah digunakan |
| 3 | Line Bot mendukung aktivitas mengenai pencarian informasi kampus yang ada di Kota Bandung |
| 4 | Saya merasa Akami Bot dapat melayani pencarian kampus yang diminati secara cepat |

Sedangkan nilai skala jawaban dan rentang nilai indeks yang disediakan dalam pengisian kuesioner dapat dilihat pada tabel 4.26

|  |  |  |
| --- | --- | --- |
| **Nilai** | **Rentang Nilai Indeks** | **Keterangan** |
| 1 | 0% - 19% | Sangat Tidak Setuju |
| 2 | 20% - 39% | Kurang Setuju |
| 3 | 40% - 59% | Cukup Setuju |
| 4 | 60% - 79% | Setuju |
| 5 | 80% - 100% | Sangat Setuju |

4.2.2.2 Kesimpulan Hasil Pengujian Beta

Berdasarkan data hasil kuisoner

**BAB V**

**KESIMPULAN DAN SARAN**

**5.1 Kesimpulan**

Kesimpulan yang dapat diambil setelah melalui tahap – tahap pembangunan sistem dengan menerapkan metode forward chaining dan LINE Bot sebagai media informasi pencarian Perguruan Tinggi di Kota Bandung. Maka pada penelitan yang telah dilakukan diperoleh kesimpulan:

1. Membantu siswa sekolah SMA sederajat dalam mencari informasi seputar Perguruan Tinggi yang diminatinya yang ada di Kota Bandung.
2. Menghasilkan media informasi Perguruan Tinggi pada media sosal Line

**5.2 Saran**

Sistem interaksi pada LINE Bot yang dibangun masih memiliki beberapa kekurangan. Pengembangan dan penyempurnaan sistem diperlukan untuk dapat memperbaiki dan meningkatkan fungsionalitas sistem sehingga mampu memberikan solusi yang baik untuk permasalahan yang ada.