**BAB I**

**PENDAHULUAN**

# Latar Belakang Masalah

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

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

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

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

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

Berdasarkan permasalah yang ada sebagai solusi dalam penelitian ini diusulkan membuat aplikasi yang dapat membantu siswa SMA/K agar tidak kebingungan dalam mencari informasi Perguruan Tinggi yang ada di kota Bandung dengan menggunakan metode webhook yang dapat terintegrasi dengan LINE Chatbot . Aplikasi ini dibangun agar siswa dapat mencari tahu tentang informasi kampus yang diinginkannya dengan cara tanya jawab kepada chatbot 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/K mengalami kesulitan dalam mencari informasi seputar Perguruan Tinggi diminatinya yang ada di Kota Bandung.
2. Siswa sekolah SMA/K mengalami kebingungan dalam memilih Perguruan Tinggi diminatinya yang ada di Kota Bandung.
3. Belum adanya aplikasi media informasi Perguruaan Tinggi pada media sosial LINE.
4. Bagaimana menghasilkan media informasi Perguruan Tinggi di Kota Bandung dengan LINE Bot

# Maksud dan Tujuan

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

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

# Batasan Masalah

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

1. Informasi data Perguruan Tinggi masih di sekitar Kota Bandung yang berstatus Negeri dan Swasta. Seperti ITB, UNPAD, UNISBA, UNIKOM, ITHB, ITENAS, Tel-U dst.
2. Informasi jurusan atau prodi yang ditampilkan hanya S1
3. Informasi biaya jurusan atau prodi yang ditampilkan tiap semester sebagian dihitung berdasarkan rata-rata dari total biaya 8 semester .
4. Informasi biaya jurusan atau prodi 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/K yang akan melanjutkan ke Perguruan Tinggi di kota Bandung.
7. Bahasa pemograman yang digunakan dalam pembangunan aplikasi adalah PHP, CSS, dan Javascript.
8. Konten pada aplikasi yang akan dibuat berbahasa Indonesia.
9. Metode yang digunakan pada penelitian ini adalah metode forward chaining.
10. Penelitian ini tidak terlalu mendalami mengenai AI
11. API Line yang digunakan versi tidak berbayar
12. Layanan pada aplikasi yang akan dibangun yakni: pencarian kampus, pencarian lokasi kampus, pencarian biaya semester, dan pencarian lokasi.
13. Masukan (input) dapat diberikan melalui teks, location dan rich menu.
14. Keluaran (output) yang diberikan oleh bot berupa teks dan gambar dalam bentuk menu dan konten
15. Protokol yang digunakan untuk mengirimkan dan menerima pesan antara bot dengan line adalah protokol HTTPS
16. Format data yang digunakan saat mengirim dan menerima pesan antara bot dengan line adalah JSON.
17. Metode pengembangan perangkat lunak yang digunakan adalah metode Waterfall.
18. Informasi yang dimiliki oleh chatbot sudah disimpan didalam di database
19. Database yang digunakan adalah database MySQL (MariaDB) dengan engine InnoDB.
20. 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 yang digunakan dalam penulisan proposal ini adalah sebagai berikut :

## Metode Pengumpulan Data

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

1. Studi Literatur

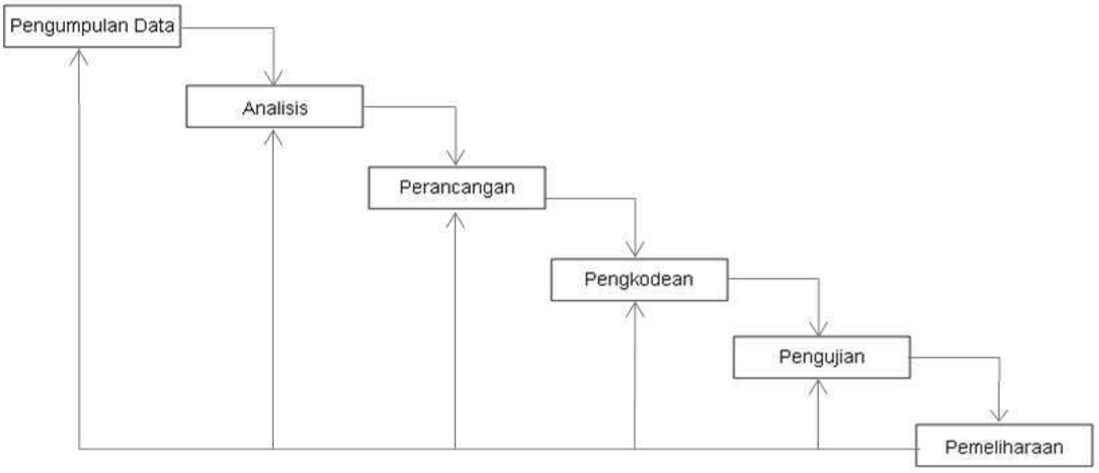
Pengumupulan data dengan cara mengumpulkan literatur – literatur yang berkaitan dengan judul penelitian

1. Kuisoner.

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

## Metode Pembangunan Perangkat Lunak

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



Gambar I.3 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)**

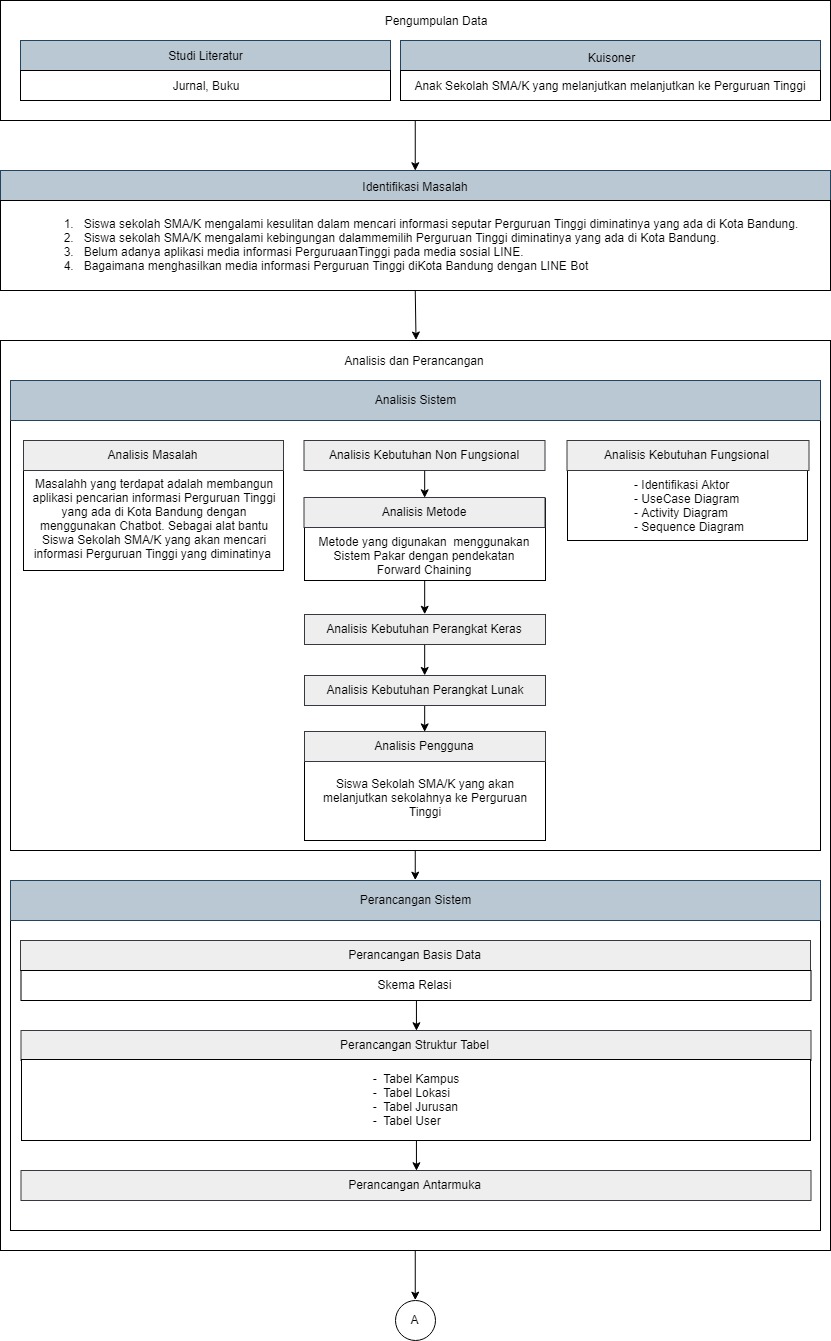
Setelah tahap perancangan sistem, selanjutnya dilakukan penterjemahan kode program yang dimengerti oleh mesin. Pada tahap ini dilakukan pembuatan komponen-komponen sistem yang meliputi implementasi modul-modul program, antarmuka dan basis data.

1. **Pengujian (Testing)**

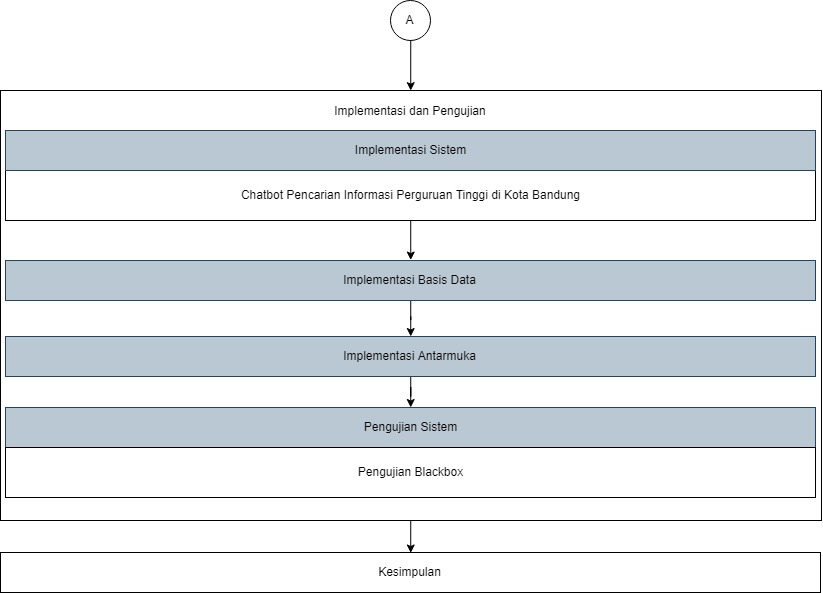
Melakukan pengujian secara formal terhadap kode program yang telah dibuat untuk memastikan semua kebutuhan atau persyaratan telah terpenuhi.

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.



Gambar I.1 Alur Penelitian



Gambar I.2 Alur Penelitian lanjutan 1

# Sistematika Penulisan

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

**BAB I PENDAHULUAN**

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

**BAB II LANDASAN TEORI**

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

**BAB III ANALISIS DAN PERANCANGAN SISTEM**

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

**BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM**

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

**BAB V KESIMPULAN DAN SARAN**

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

**BAB II**

**LANDASAN TEORI**

**2.1 Landasan Teori**

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

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

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

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

Menurut beberapa ahli kecerdasan buatan didefinisikan sebagai berikut :

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

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

1. Sudut pandang Penelitian

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

**2.2.1 Konsep Dasar AI atau Kecerdasan Buatan**

Secara Garis besar Kecerdasan Buatan dibagi menjadi 4 kategori diantaranya :

1. Thinking humanly

Pendekatan ini dilakukan dengan dua cara :

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

Terdapat dua masalah dalam pendekatan ini diantaranya :

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

Selain dari keempat kategori Kecerdasan Buatan diatas, ada beberapa konsep yang harus dipahami dalam Kecerdasan Buatan diantaranya [8]:

1. Turing Test – Metode Pengujian Kecerdasan

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

1. Pemrosesan Simbolik

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

1. Heuristic

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

1. Penarikan Kesimpulan (Inferencing)

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

1. Pencocokan Pola (Pattern Matching)

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

**2.2.2 Lingkup Utama Dalam Kecerdasan Buatan**

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

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

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

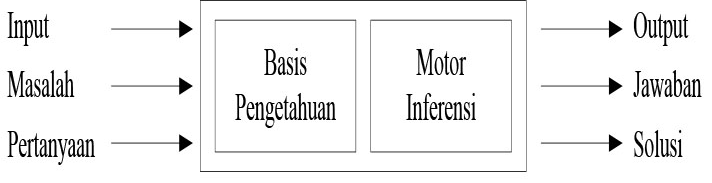
**2.2.3 Tujuan Kecerdasan Buatan**

Tujuan dari kecerdasan buatan menurut Winston dan Prendergast :

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

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

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



Gambar II.1 Proses Motor Inferensi

**2.3 Sistem Pakar**

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

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

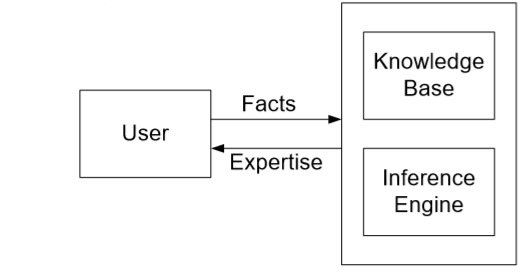
Hal yang unik dari sistem pakar adalah kemampuan untuk menjelaskan dimana keahlian tersimpan dalam basis pengetahuan. Kemampuan komputer untuk mengambil kesimpulan dilakukan oleh komponen yang dikenal dengan mesin inferensi yaitu meliputi prosedur tentang pemecahan masalah.

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

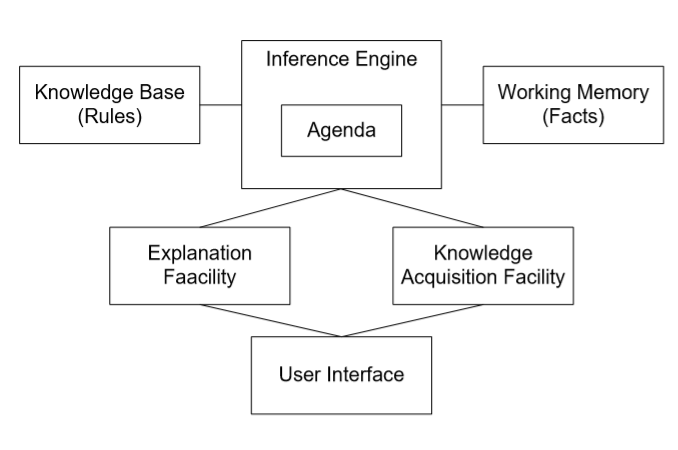
Keunikan lain dari sistem pakar adalah kemampuan dalam menjelaskan atau memberi saran rekomendasi serta juga menjelaskan mengapa beberapa tindakan saran tidak direkomendasikan.



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 pengetahuanpengetahuan yang dibutuhkan untuk memahami, memformulasikan dan menyelesaikan masalah.

**2.3.3 Basis Aturan Sistem Pakar**

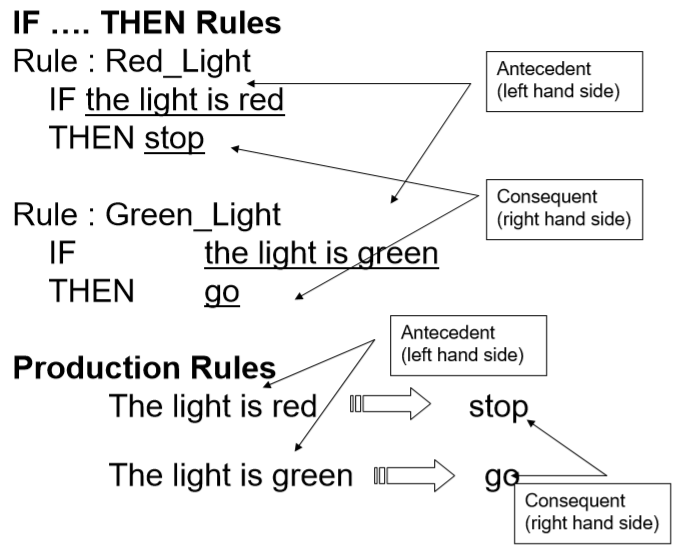
Adapun basis aturan dalam Sistem Pakar adalah sebagai berikut :

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

- Aktivasi aturan akan membangkitkan fakta baru di sisi kanan

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

Contoh Basis Aturan (Rule Based) Sistem Pakar



Gambar II.4

**2.3.4 Keuntungan Sistem Pakar**

Keuntungan Sistem Pakar adalah sebagai berikut :

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

**2.3.5 Kelemahan Sistem Pakar**

Adapun kelemahan dalam Sistem Pakar adalah sebagai berikut :

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

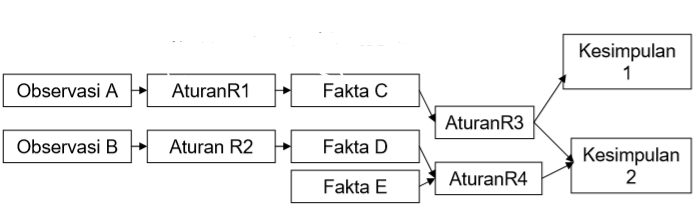
**2.4 Penarikan Kesimpulan (Inferencing)**

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

**2.4.1 Metode Algoritma Forward Chaining**

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

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



Gambar II.5 Aturan Forward Chaining

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

Contoh :

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

R1 : if A and B then C

R2 : if C then D

R3 : if A and E then F

R4 : if A then G

R5 : if F and G then D

R6 : if G and E then H

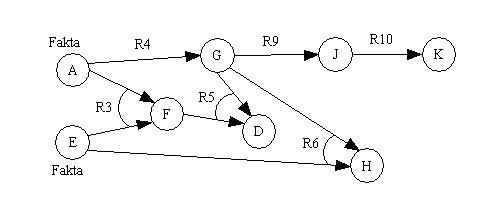
R7 : if C and H then I

R8 : if I and A then J

R9 : if G then J

R10 : if J then K

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



Gambar II.6 Forward Chaining

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

**2.5 Chatbot**

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

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

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

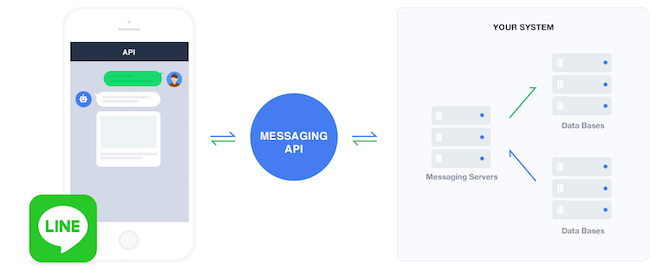
Bot biasa digunakan dalam IRC (Internet Chat Groups) di mana bot tersebut melakukan berbagai fungsi seperti menyapa partisipan baru dan memonitor penggunaan kata-kata yang tidak pantas. Sekarang ini, bot dapat mengumpulkan informasi, mengolah informasi tersebut, dan memberikan respon terhadap pengguna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

**2.5.1 Cara Kerja Chatbot**

Pada dasarnya 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 “Toko kami besok buka jam 9 pagi dan tutup jam 5 sore”. Disini anda hanya perlu memberikan perintah atau melakukan sebuah percakapan baik itu dalam bentuk teks maupun audio, dan bot akan mencari data yang anda butuhkan sesuai dengan kata kunci yang anda berikan.

**2.5.2 LINE Messaging API**

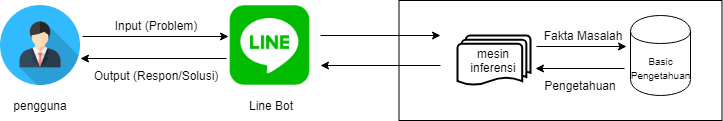
Api (*Application Programming Interface*) merupakan kumpulan dari fungsi-fungsi yang digunakan oleh *programmer* untuk membangun sebuah aplikasi. Pada line *chatbot* ini, api yang digunakan yaitu *messaging api* pada line yang berfungsi untuk saling mengirimkan data antar aplikasi line dengan sistem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

Mudah dipelajari

Mudah digunakan

Menghemat waktu dan uang

Telah melalui uji coba

Menawarkan software auto installer

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. Webhook merupakan satu cara yang efisien bagi provider maupun bagi konsumen. Satu-satunya kelemahan webhook adalah kesulitan di awal saat mengaturnya.

**2.9 Pengujian Black box**

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

**BAB III**

**ANALISIS DAN PERANCANGAN**

## 3.1 Analisis Sistem

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

**3.1.1 Analisis Masalah**

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

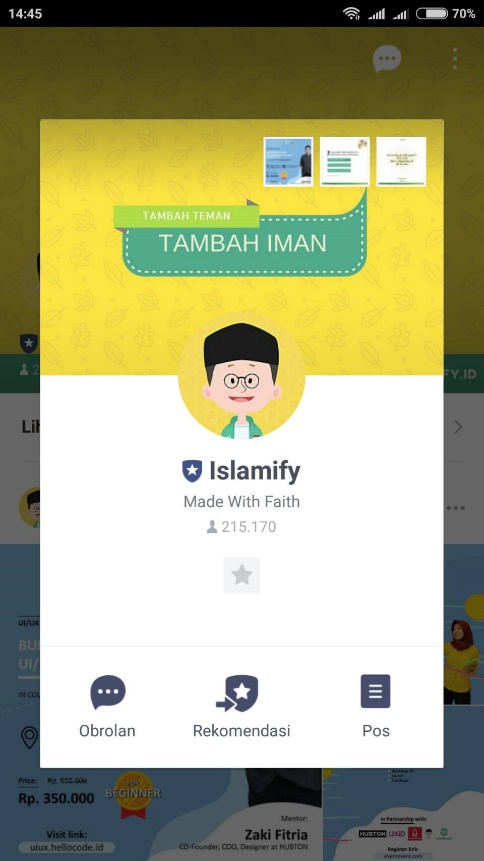
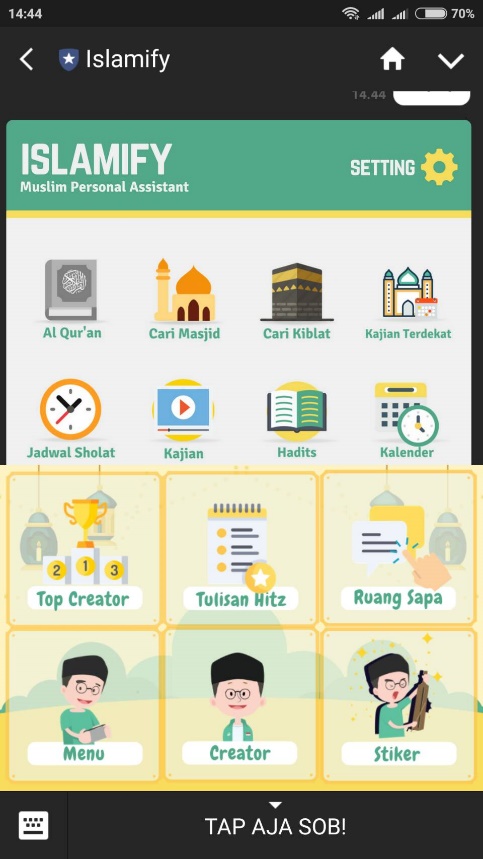
**Untuk mendapatkan Informasi Perguruan Tinggi yang diminatinya, saat ini siswa SMA/K dalam mencari informasi Perguruan Tinggi menggunakan cara mengunjungi setiap website Perguruan Tinggi satu persatu yang diinginkan untuk mendapatkan informasi mengenai Perguruan Tinggi. Berdasarkan data kuisoner terhadap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

**3.1.2 Analisis Aplikasi Sejenis**

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

1. **Islamify**

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

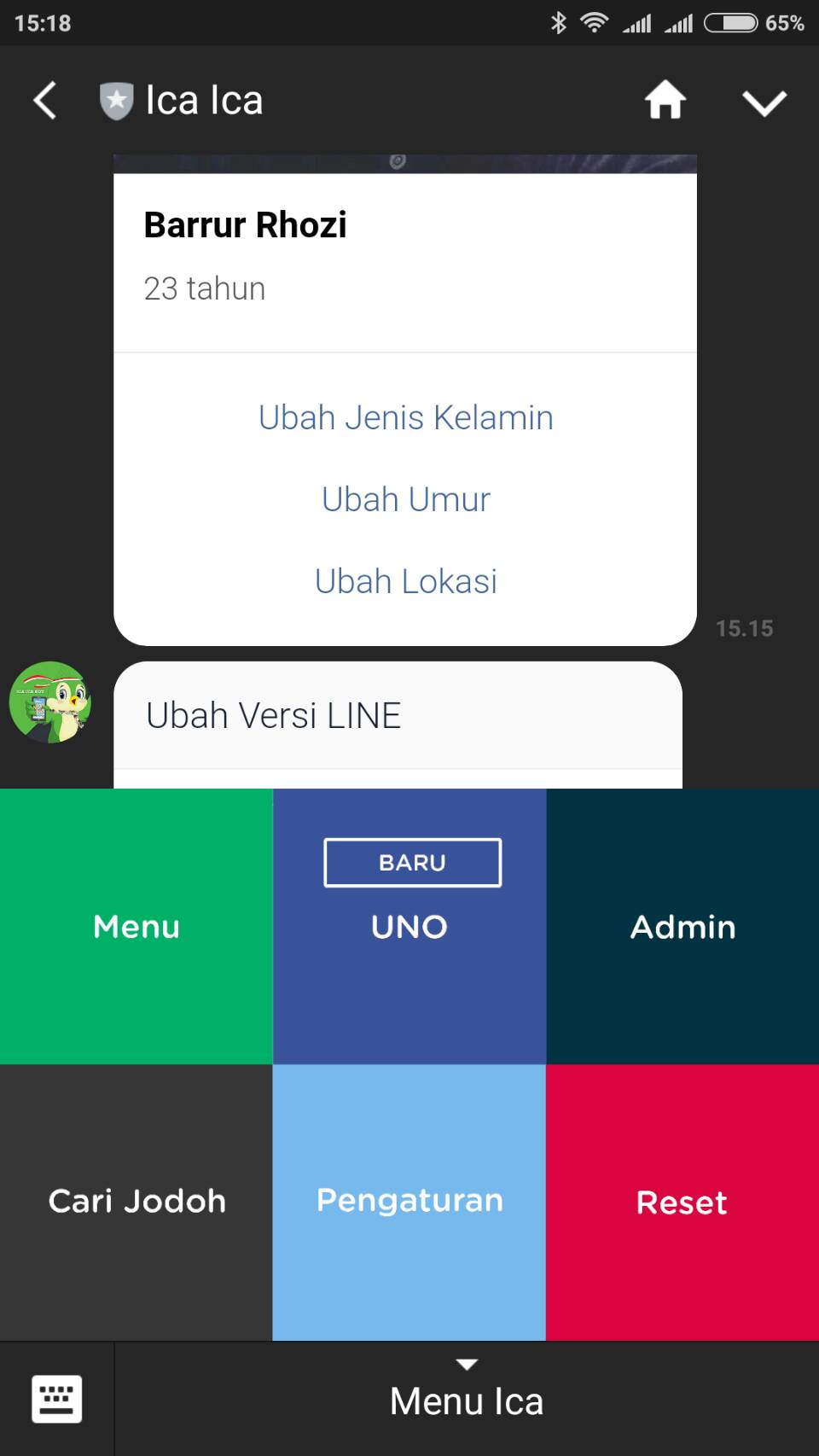
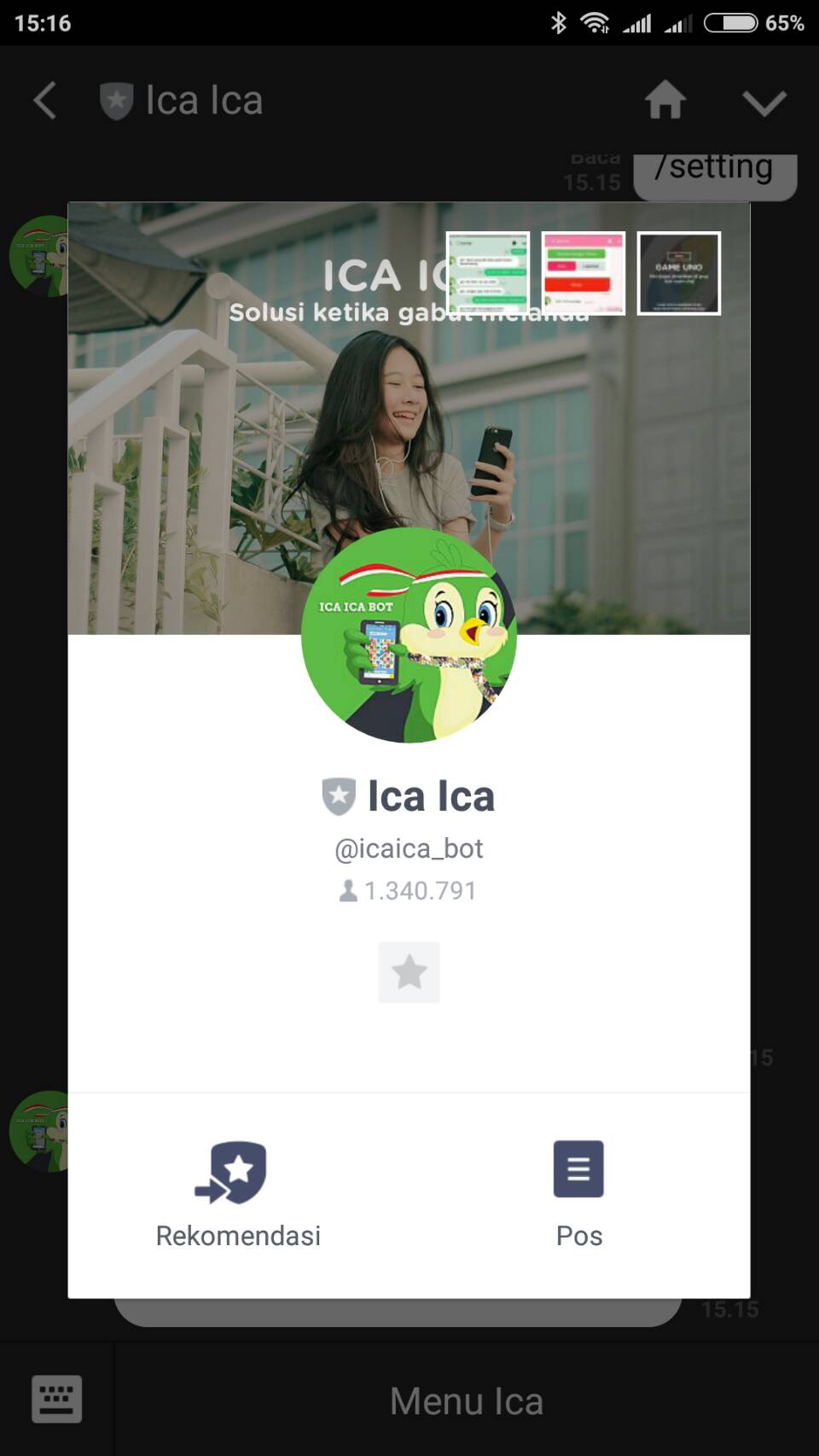
****

**Gambar III.1 Bot Islamify**

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

1. **Ica Ica**

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

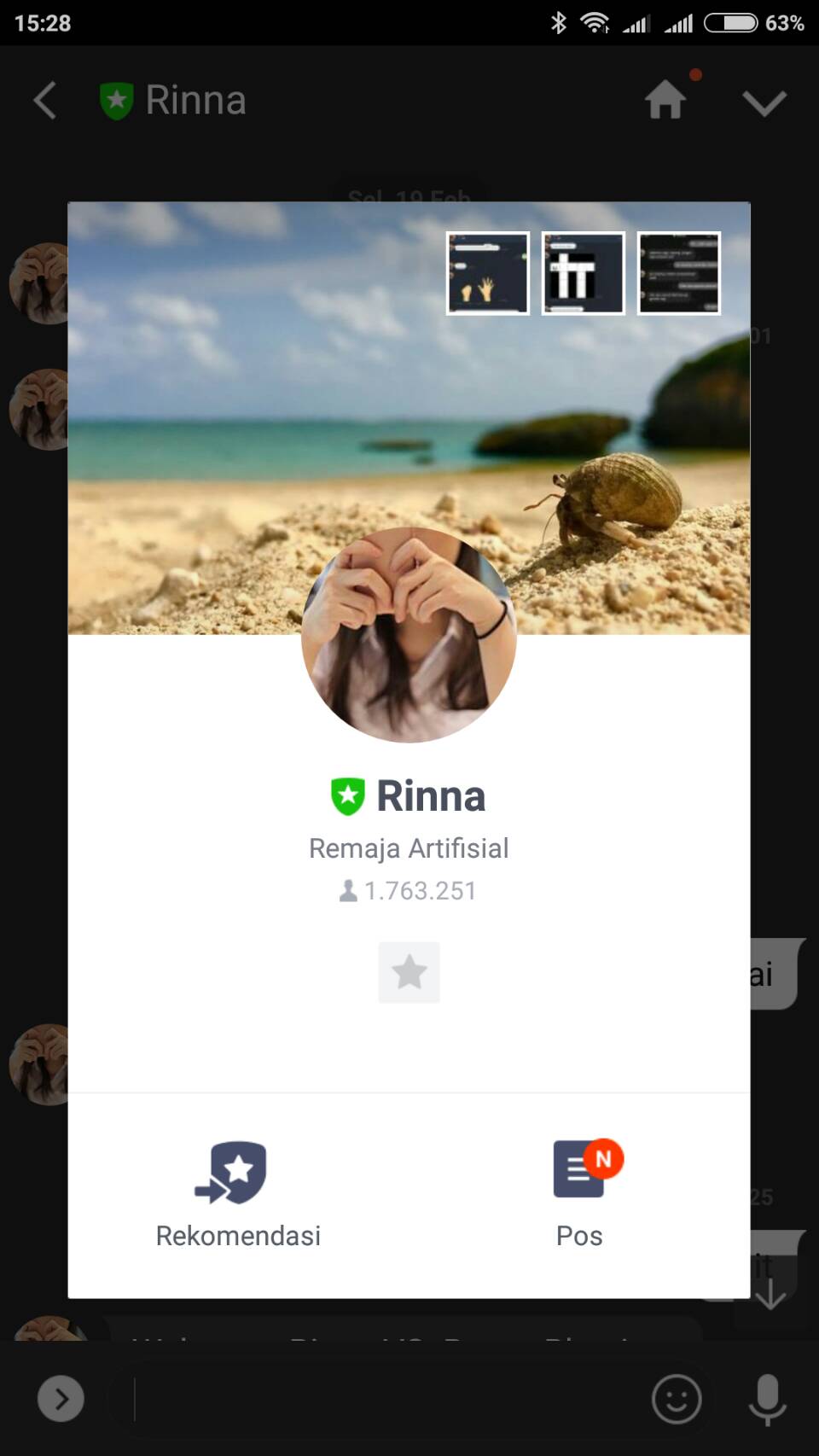


Gambar III.2 Ica Ica Bot

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

1. Rinna

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

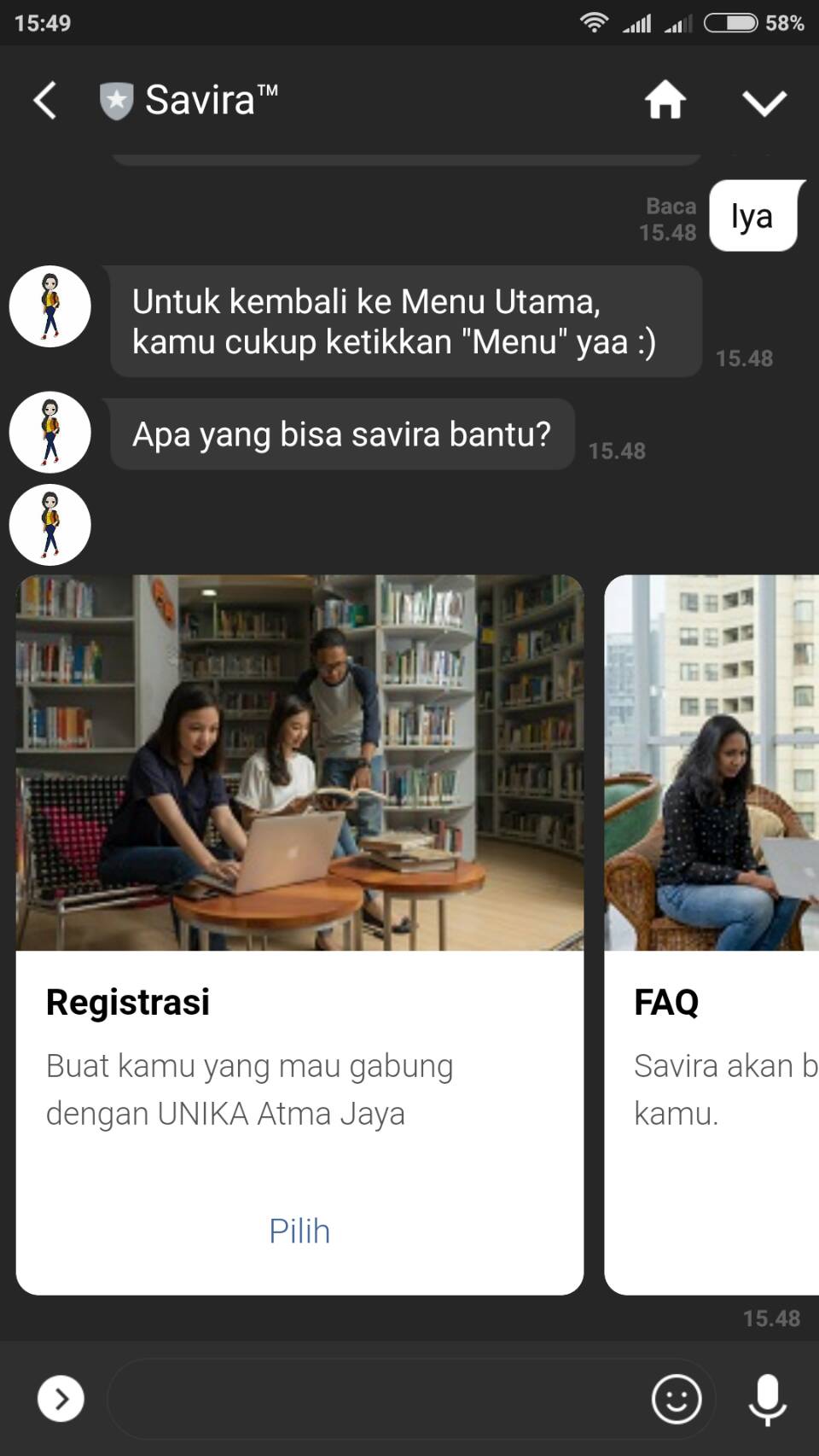
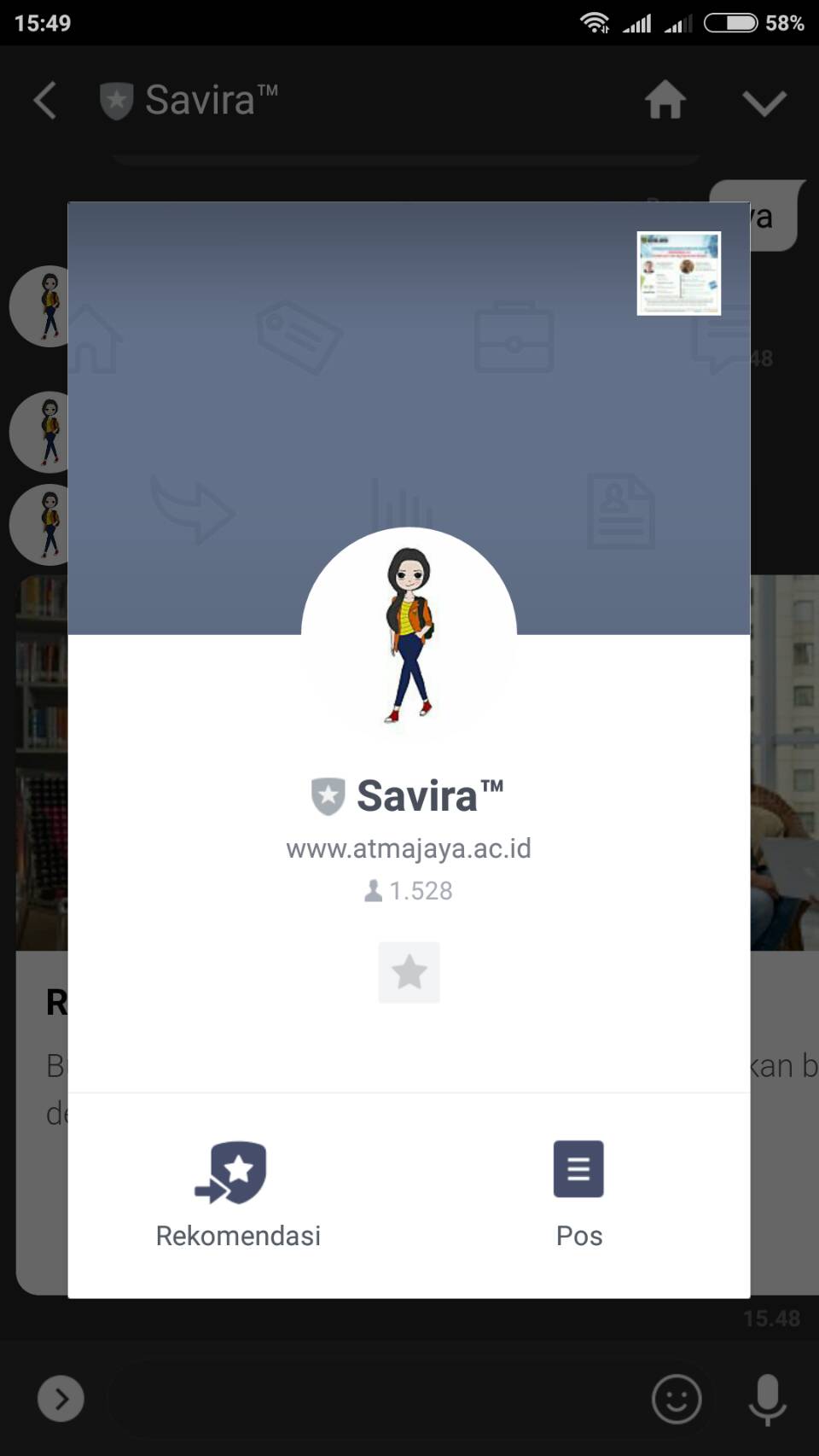


Gambar III.3 Bot Rinna

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

1. SaviraTM

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



Gambar III.4 Bot Savira

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

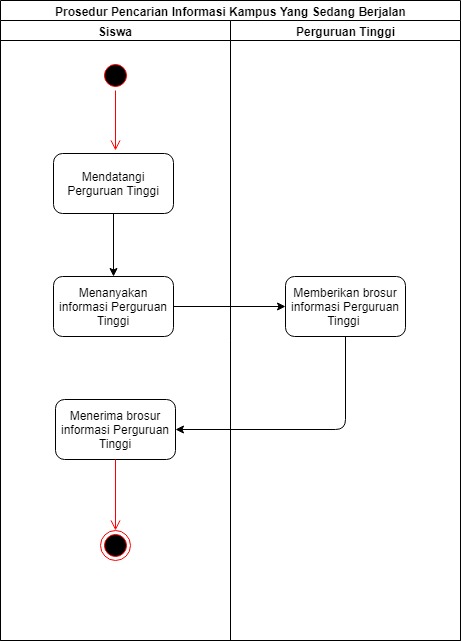
Berdasarkan berbagai aplikasi sejenis mengenai chatbot pada gambar diatas, maka dapat hasil analisis aplikasi bot dapat dilihat dibawah ini

**Tabel III.1 Analisis Aplikasi Sejensi**

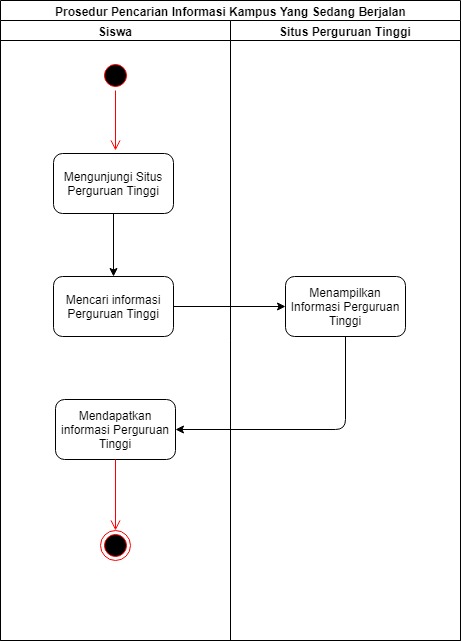
|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Konten/Fitur | Menyimpan Informasi Pengguna | Menggunakan Rich Menu |
| Islamify | 1. Al-Qur’an 2. Lokasi Masjid 3. Cari Kiblat 4. Jadwal Sholat 5. Kajian 6. Hadits 7. Kalender 8. Stiker | 1. Nama Pengguna 2. Jenis Kelamin | Ya |
| Icaica | 1. Permainan (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 |
| SkyBot | 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

u

Gambar III.6 Prosedur yang berjalan 1

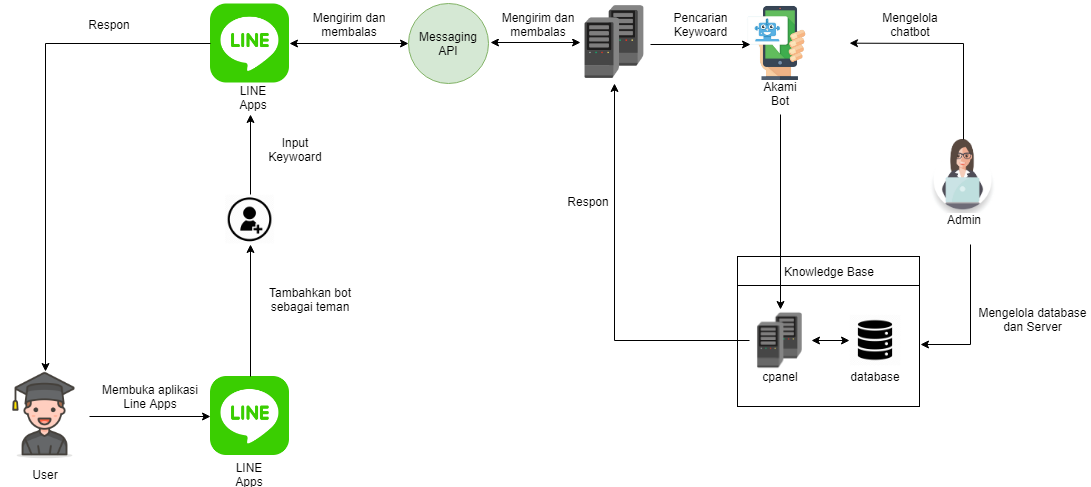
**3.3 Analisis Solusi Yang Ditawarkan**

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

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

**3.4 Analisis Arsitektur Sistem**

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 Arsitektur Sistem

Berikut adalah penjelasan dari arsitektur sistem pada chatbot yang dibangun.

**Pengguna (User) :**

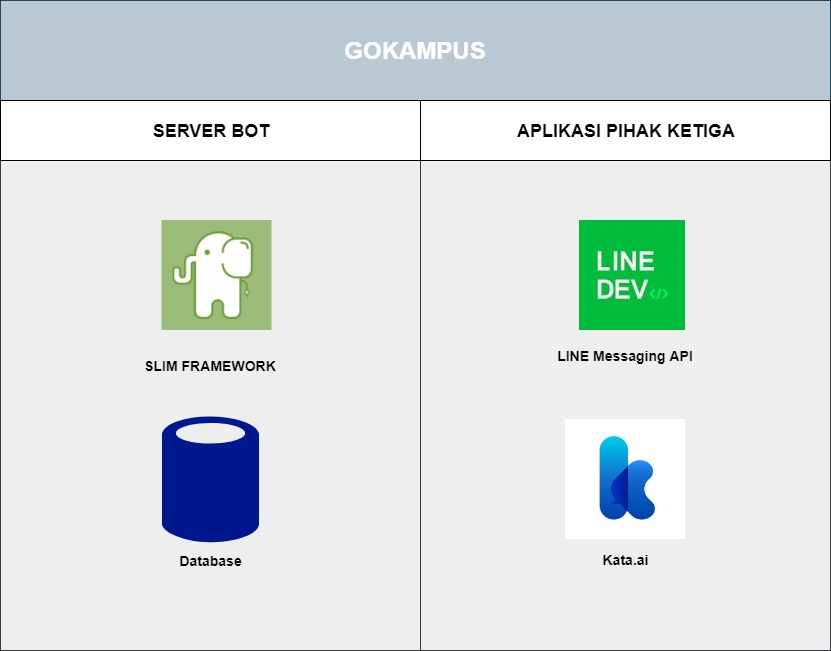
1. Pengguna atau user membuka aplikasi LINE.
2. Pengguna atau user menambahkan bot sebagai teman, bisa melalui id line atau barcode.
3. Pengguna menginputkan keyword.
4. LINE akan mengirim keyword melalui Messaging API, lalu diteruskan ke server LINE.
5. Server LINE akan mengirim keyword pada sistem bot, kemudian sistem bot akan mengolah keyword menggunakan metode *Text Mining.*
6. Keyword yang dihasilan akan dilakukan pencocokan dengan data yang terdapat pada database.
7. Jika sudah ditemukan keyword yang cocok, maka sistem akan mecari respon dari keyword tersebut.
8. Jika jawaban sudah ditemukan makan respon akan diteruskan ke server LINE.
9. Server LINE akan meneruskan respon melalui Messaging API ke aplikasi LINE.
10. Respon akan ditampilkan pada layar sebagai output.

Admin (pengembang) :

1. Admin mengelola sistem bot.
2. Admin mengelola database diantaranya menginput data, menghapus data, dan mengubah data pada database.

**3.5 Analisis Arsitektur Aplikasi Yang Dibangun**

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

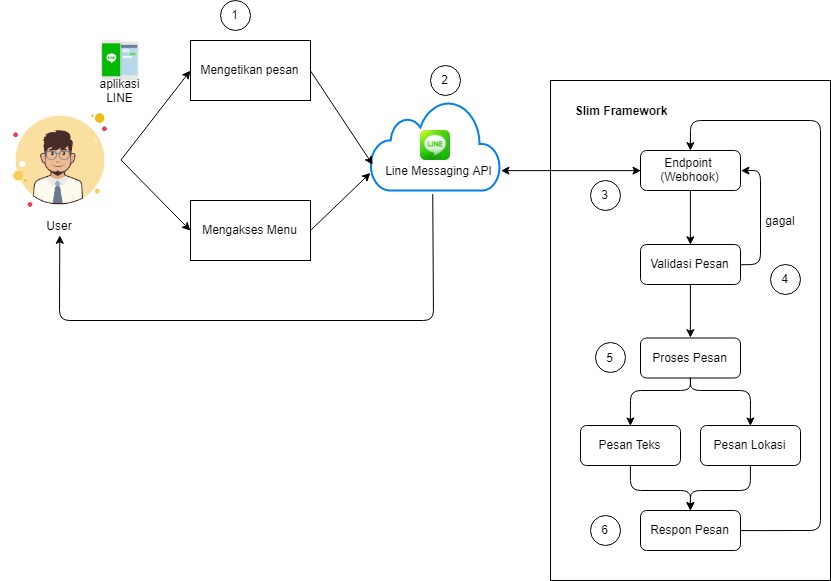


Gambar III.8 Analisis Arsitektur Aplikasi Yang Dibangun

Penjelasan untuk setiap bagian aplikasi adalah sebagai berikut:

1. Server Bot
2. Slim Framework

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



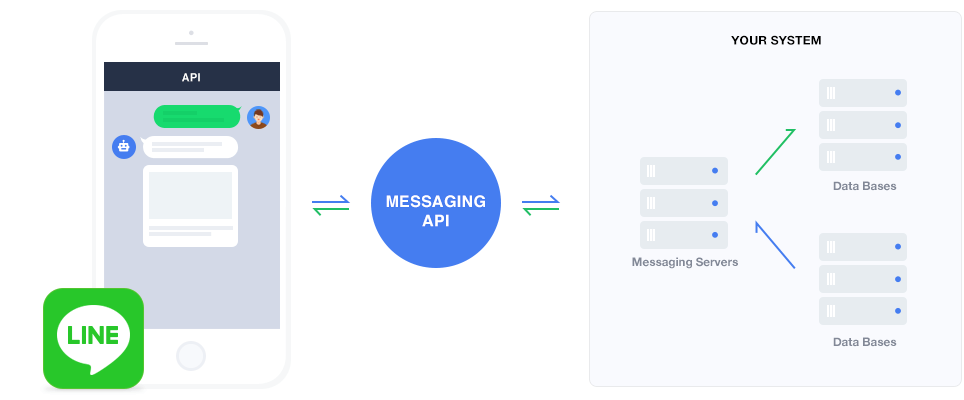
Gambar III.9 Alur Slim Framework

1. Database

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

1. Aplikasi Pihak Ketiga
2. Line Messaging API

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



Gambar III.10 Line Messaging API

1. Kata.ai

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

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

Dalam pembangunan aplikasi chatbot ini, penulis memanfaatkan teknologi Slim Framework, Deploy Heroku, aplikasi pihak ketiga yaitu kata.ai dan Line Messaging API.

1. Slim Framework

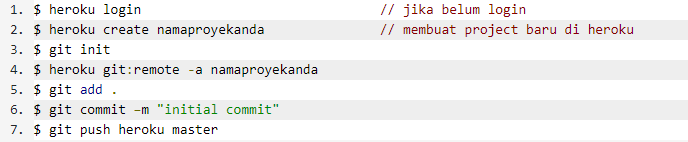
Slim framework digunakan untuk menangani request dan response yang dilakukan oleh user melalui Line Messaging API. Line messaging api akan mengirimkan request kepada aplikasi slim melalui route (webhook) yang sebelumnya sudah diatur.



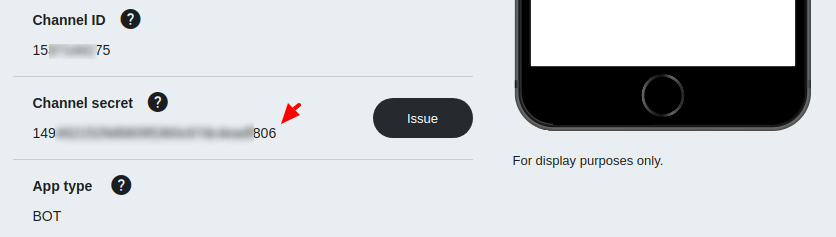
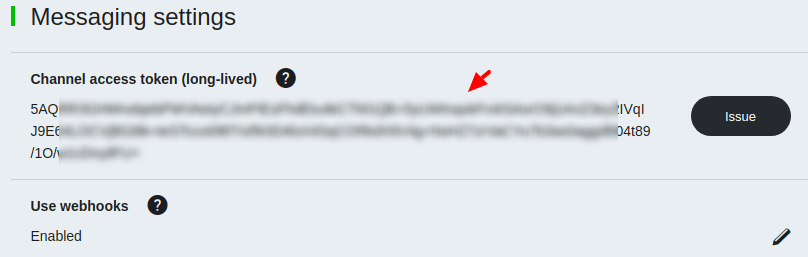
III.11 Alur Penggunaan Slim Framework

1. Deploy Heroku

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

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

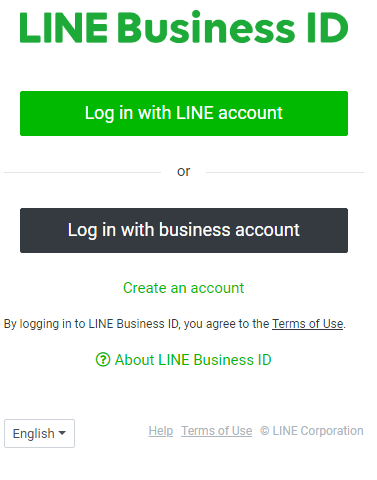
Setelah proses push berhasil, cek aplikasi yang dibuat di browser dengan alamat URL  [*https://namaproyekanda.herokuapp.com*](https://namaproyekanda.herokuapp.com)*.* Langkah selanjutnya menintegrasikan Line Channel terhadap webhook yang sudah dibuat sebelumnya.



Gambar III.12 Integrasi LINE Channel API

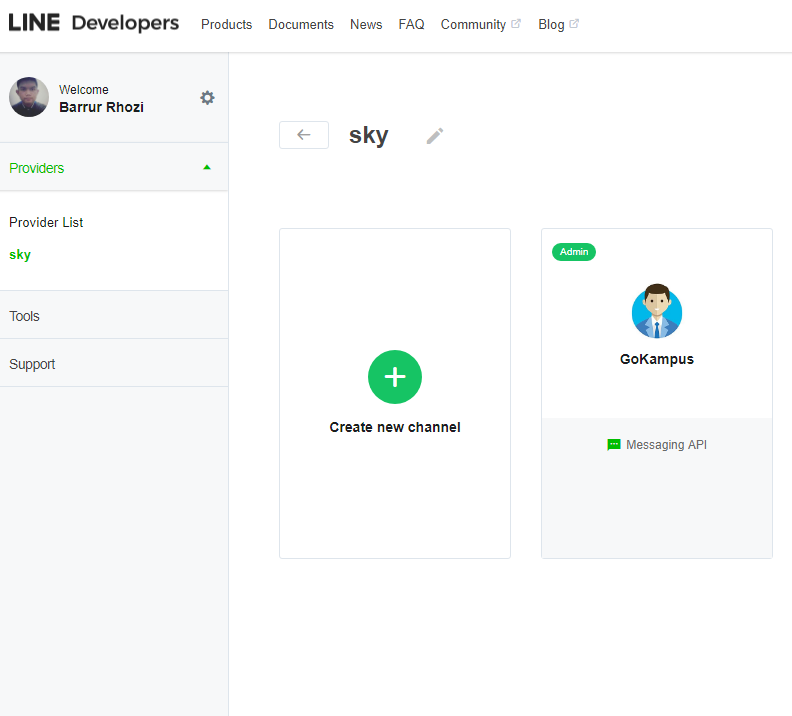
1. Line Messaging API

Untuk dapat memanfaatkan fitur line messaging api, developer harus membuat channel messaging api terlebih dahulu. Channel ini merupakan representasi bot yang akan dibuat pada aplikasi Line. Untuk membuat channel ini, developer harus mempunyai akun Line Developer terlebih dahulu dengan cara mendaftar pada situs line developer. Gambar III.4 merupakan halaman masuk line developer.



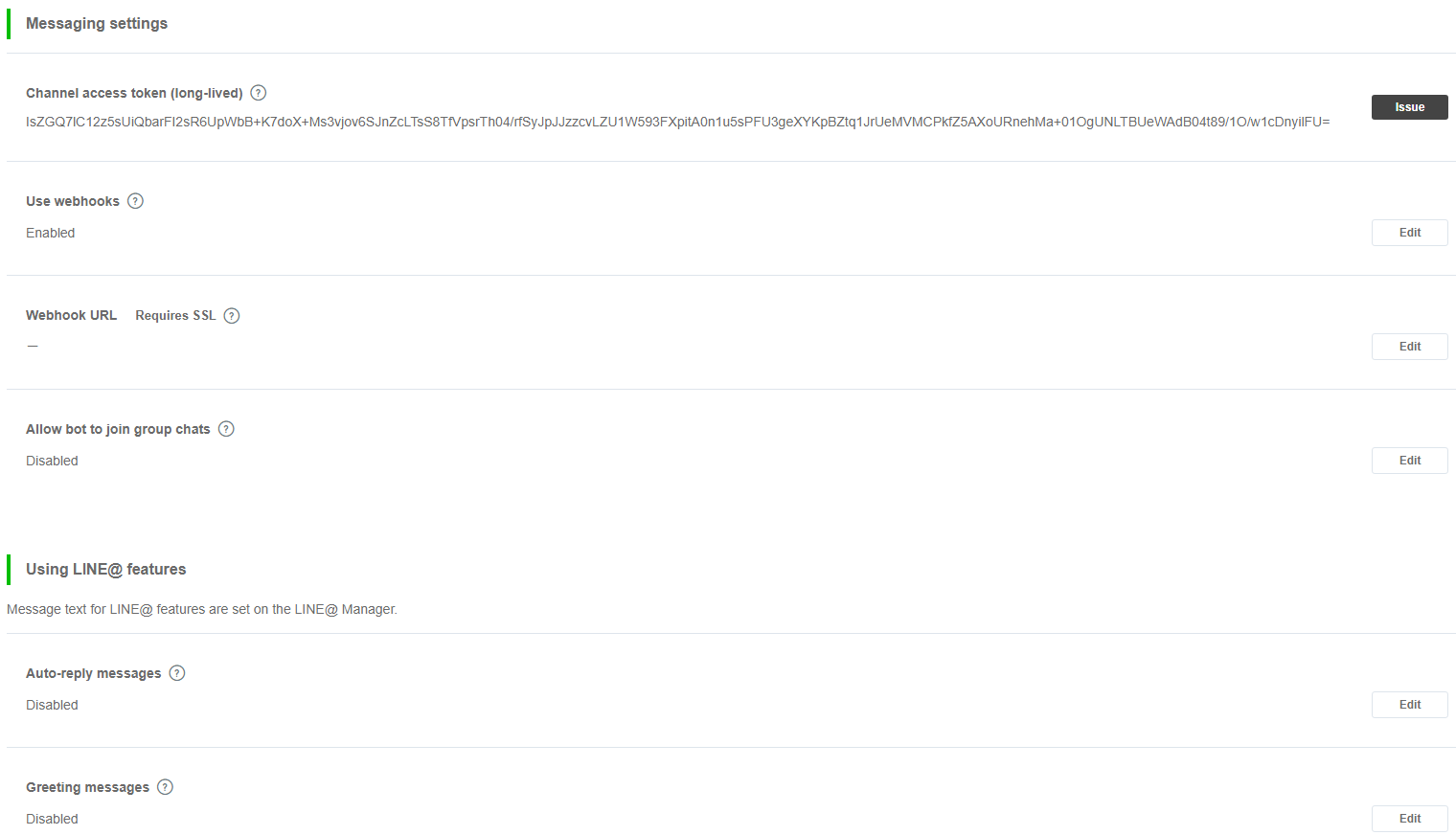
Gambar III.13 Halaman masuk line developer

Setelah masuk, developer dapat membuat channel messaging api untuk membuat bot. Gambar III.14 merupakan channel line messaging api.



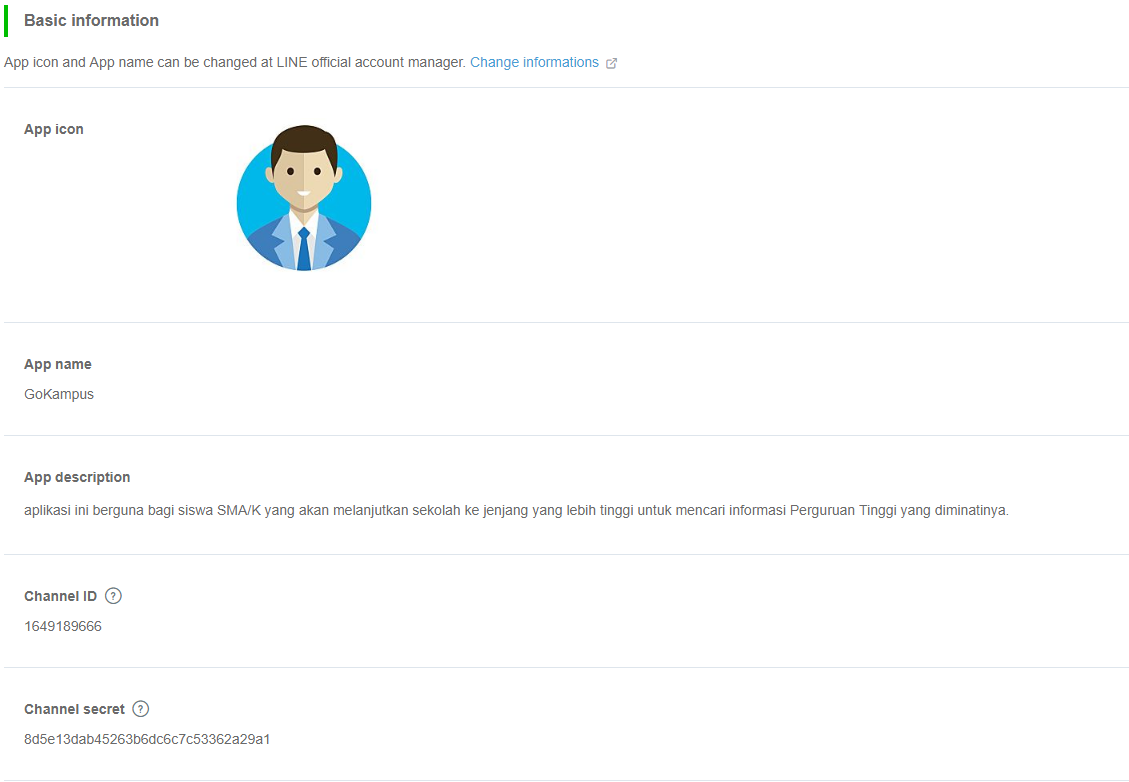
Gambar III.14 Channel Line Messaging API

Untuk dapat terhubung ke server bot, developer harus mengatur webhook (endpoint) pada line developer. Setiap request yang dikirimkan melalui bot akan dikirimkan ke alamat tersebut. Gambar III.7 menunjukkan pengaturan webhook pada situs line developer.



Gambar III.15 Pengaturan Webhook pada Line Developer

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

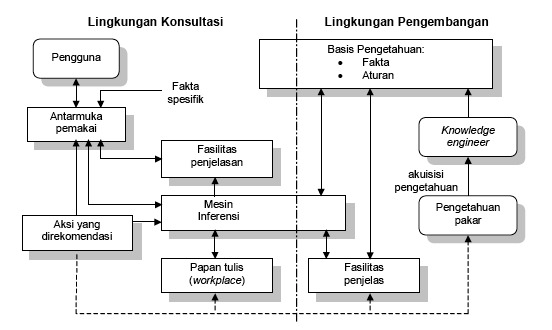


Gambar III.16 Pengaturan umum bot pada line developer

## 3.7 Analisis Metode

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

**3.7.1 Diagram Struktur Sistem Pakar**

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

**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

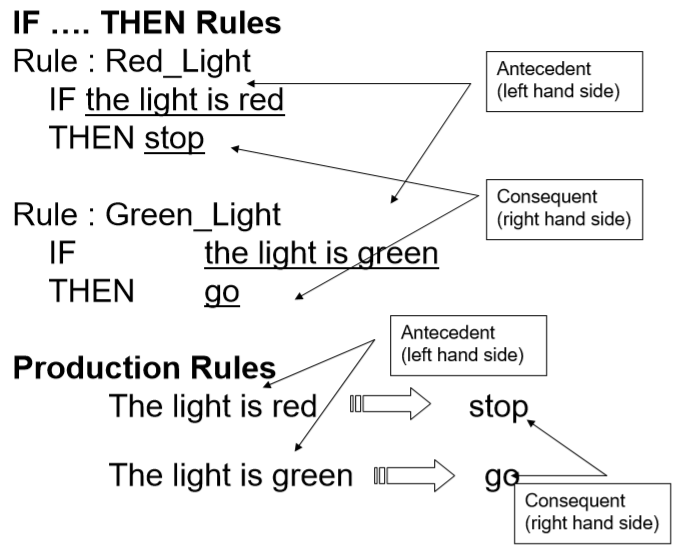
Mengembangkan Sistem Pakar dapat dilakukan dengan 2 cara:

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.

**3.7.2 Forward Chaining**

Forward chaining disebut juga penalaran dari bawah ke atas karena penalaran dari fakta pada level bawah menuju konklusi pada level atas didasarkan pada fakta. Penalaran dari bawah ke atas dalam suatu sistem pakar dapat disamakan untuk pemgrograman konvensional dari bawah ke atas. Fakta merupakan satuan dasar dari paradigma berbasis pengetahuan karena mereka tidak dapat diuraikan ke dalam satuan paling kecil yang mempunyai makna.



**3.7.3 Algoritma Forward Chaining**

Untuk mempresentasikan aplikasi berikut kedalam algoritma , dapat dituliskan menggunakan metode JIKA MAKA (IF THEN), berikut role algoritmanya :

**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.

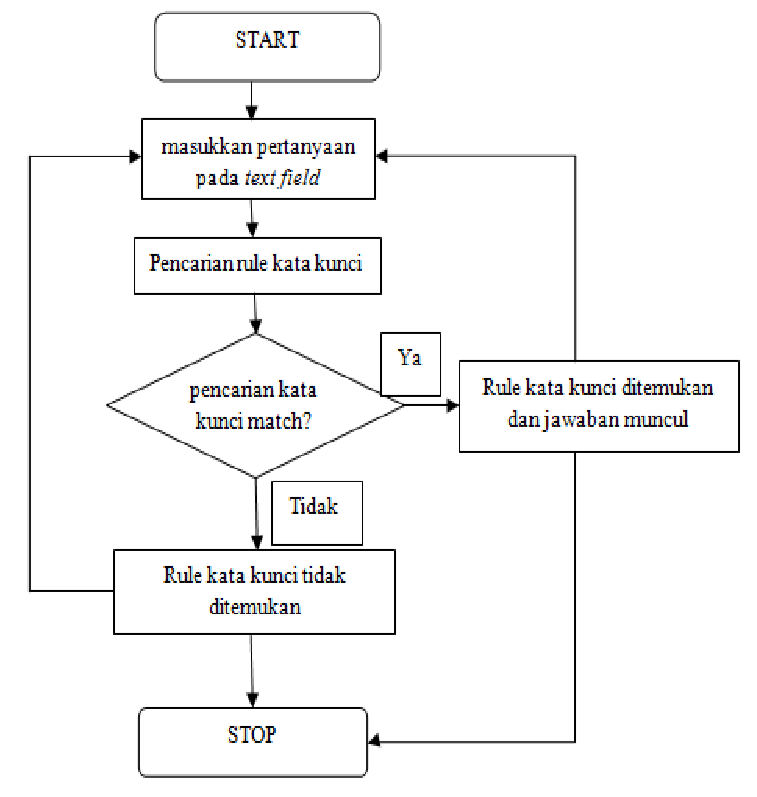
* Rule 4

**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.

**3.7.4 Proses Forward Chaining**



Gambar III.17 Proses Forward Chaining

Proses forward chaining dimulai dengan user memasukkan informasi atau pertanyaan ke dalam text field yang sudah disediakan oleh aplikasi. Kemudian pertanyaan tersebut dipisah menjadi beberapa kata dan dicari kata kunci yang sesuai dengan yang ada pada aplikasi setelah kata kunci ditemukan maka aplikasi menjawab pertanyaan user dengan jawaban yang benar sedangkan jika aplikasi tidak dapat menemukan kata kunci maka jawaban muncul tetapi tidak berhubungan dengan pertanyaan dari user.

Berikut adalah adalah tabel hasil implementasi algoritma sistem pakar pada aplikasi chatbot.

Tabel. III.1 Implementasi implementasi algoritma sistem pakar

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Masukan Pertanyaan** | **Kata Kunci (Keywoard)** | **Hasil** |
| 1. | Tolong carikan kampus negeri di Bandung | Cari kampus Negeri | Menampilkan list kampus negeri |
| 2. | Tolong carikan alamat kampus di Bandung | Cari lokasi kampus | Menampilkan list lokasi kampus |
| 3. | Carikan akreditasi kampus | Cari akreditasi kampus | Menampilkan akreditasi kampus |
| 4. | Carikan akreditasi jurusan | Cari akreditasi jurusan | Menampilkan akreditasi jurusan |
| 5. | Cari biaya jurusan teknik informatika | Biaya jurusan teknik informatika | Menampilkan kampus berdasarkan akreditasi jurusan |

**3.7.5 Fungsi STRPOS**

## 3.8 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.

**3.8.1 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.2 Analisis Perangkat Keras

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

1. Analisis Perangkat Lunak

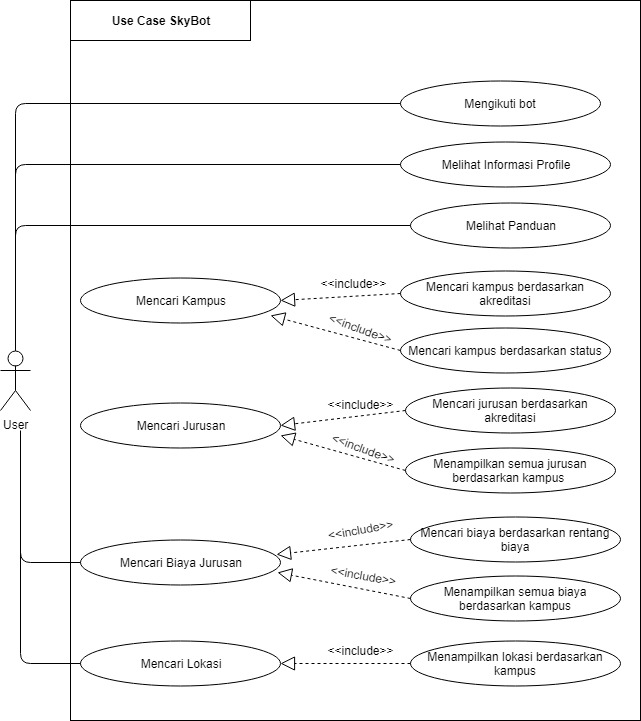
Berikut adalah spesifikasi perangkat lunak yang digunakan dalam membangun dan mengimplementasikan aplikasi chatbot sebagai berikut :

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

**3.8.2 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.17 Use Case Diagram

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

Tabel III.3 Definisi Actor

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

* 1. **Definisi Use Case**

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

Tabel III.4 Definisi Use Case

|  |  |  |
| --- | --- | --- |
| **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.5 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.6 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.7 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.8 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.9 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.10 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.11 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.12 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.13 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.14 Use Case Description menampilkan Info lokasi Berdasarkan kampus

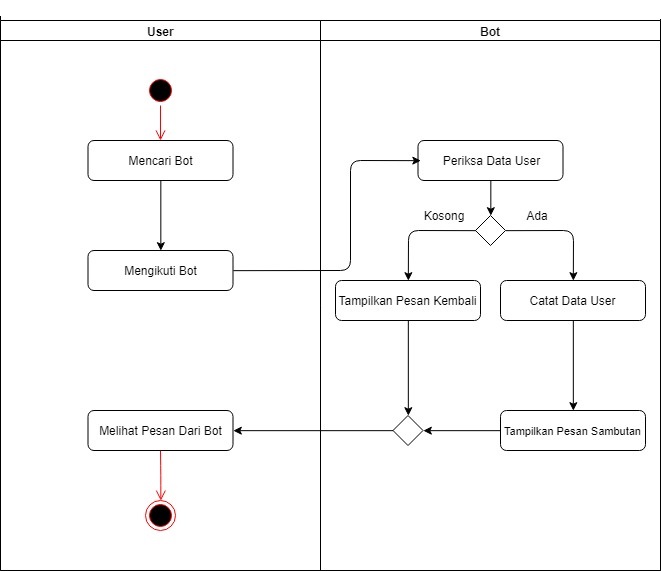
|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Menampilkan info lokasi berdasarkan kampus | |
| **Related Requirements** |  | |
| **Goal in Context** | Menampilkan informasi lokasi berdasarkan kampus | |
| **Preconditions** | User sudah mengikuti bot | |
| **Successful end Condition** | User berhasil 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.10 menunjukkan activity diagram mengikuti bot :



Gambar III.18 Mengikuti Bot

* 1. Activity Diagram Melihat Informasi Profile

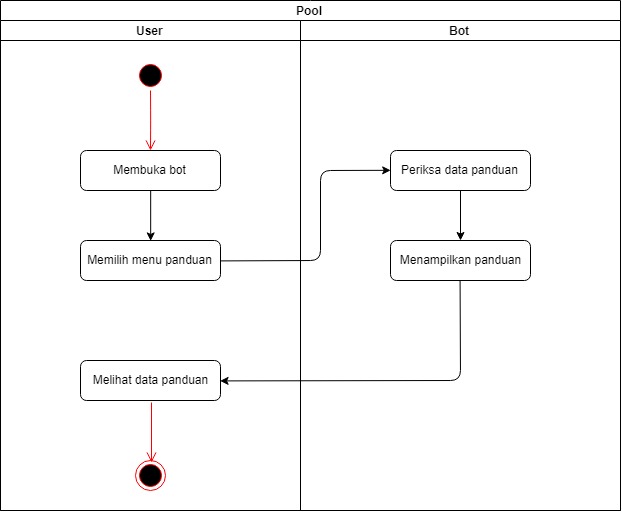
Activity diagram Melihat Informasi Profil menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna melihat informasi profil. Gambar III.11 menunjukkan activity diagram melihat informasi profil:



Gambar III.19 Activity Diagram Melihat informasi profil

* 1. Activity Diagram Melihat Panduan

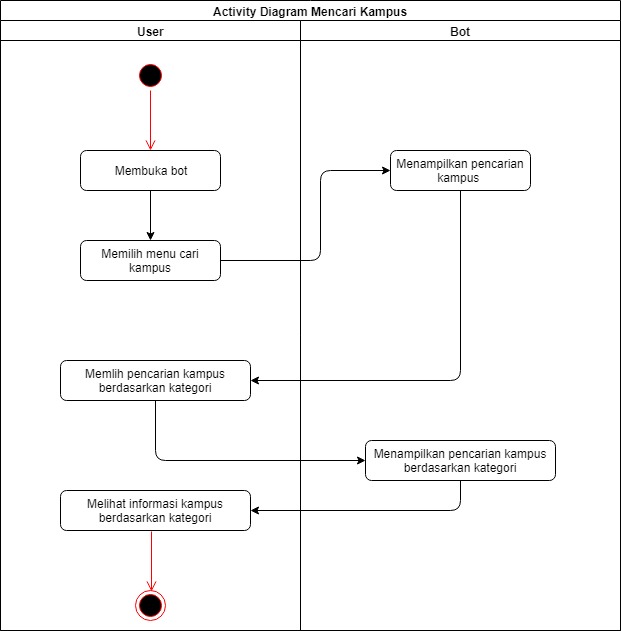
Activity diagram Melihat Stiker menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna melihat panduan. Gambar III.12 menunjukkan activity diagram melihat informasi panduan :



Gambar III.20 Activity Diagram Menu Melihat Panduan

* 1. Activity Diagram Menu Mencari Kampus

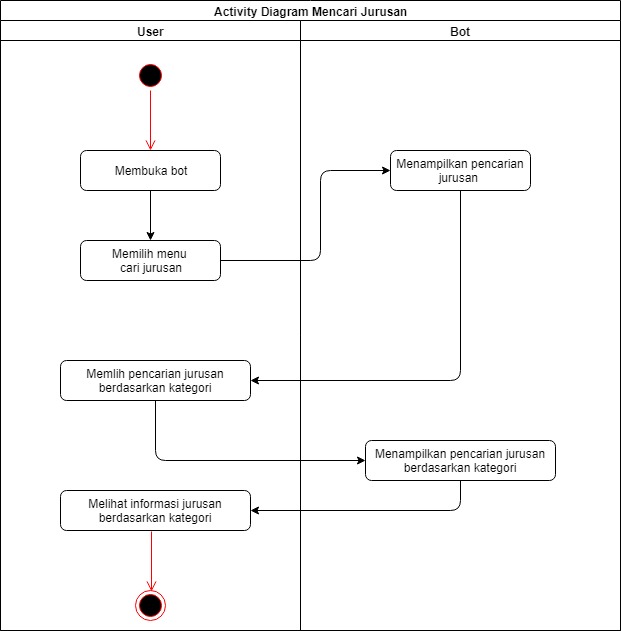
Activity diagram mencari kampus di Kota Bandung menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari kampus di Kota Bandung. Gambar III.13 menunjukkan activity diagram mencari kampus di Kota Bandung:



Gambar III.21 Activity diagram menu cari kampus

* 1. Activity Diagram menu cari jurusan

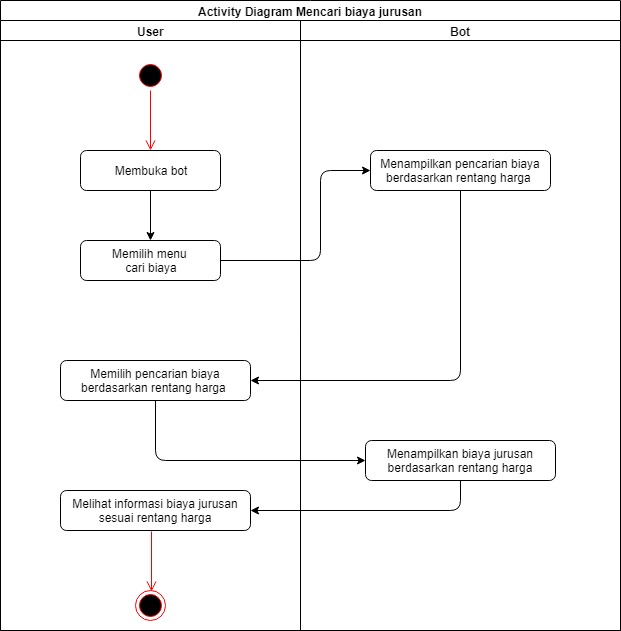
Activity diagram cari jurusan di Kota Bandung menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari jurusan di Kota Bandung. Gambar III.13 menunjukkan activity diagram mencari jurusan di Kota Bandung:



Gambar III.22 Activity diagram menu cari jurusan

* 1. Activity Diagram menu cari biaya

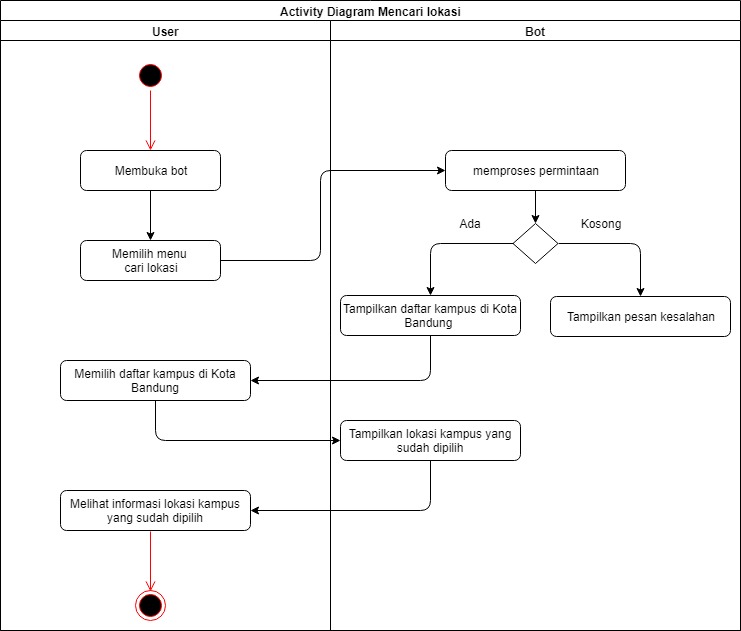
Activity diagram Mencari informasi pencarian biaya jurusan yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi cari biaya jurusan berdasarkan rentang harga. Gambar III.14 menunjukkan activity diagram mencari informasi biaya jurusan berdasarkan rentang harga:



Gambar III.23 activity diagram menu cari biaya

* 1. Activity Diagram menu cari lokasi

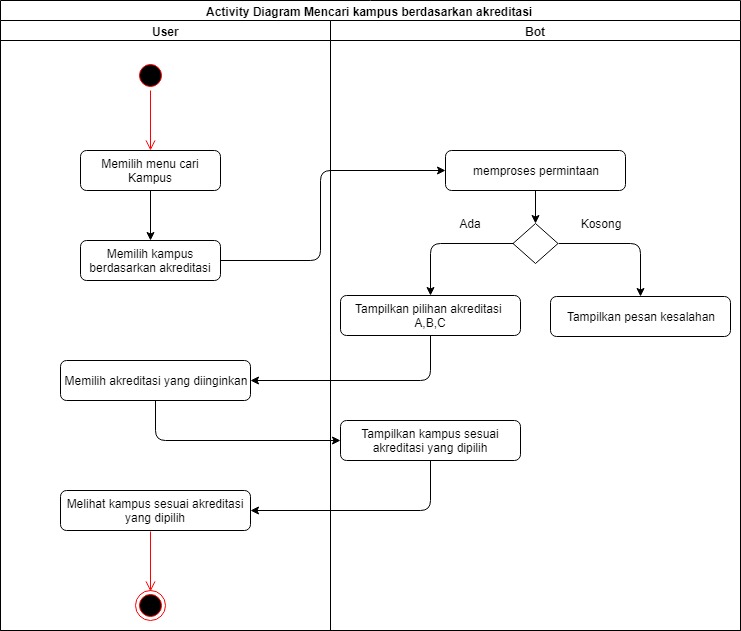
Activity diagram Mencari informasi pencari lokasi kampus yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi lokasi kampus. Gambar III.15 menunjukkan activity diagram mencari informasi lokasi kampus :



Gambar III.24 activity diagram menu cari lokasi

* 1. Activity Diagram Mencari kampus berdasarkan akreditasi

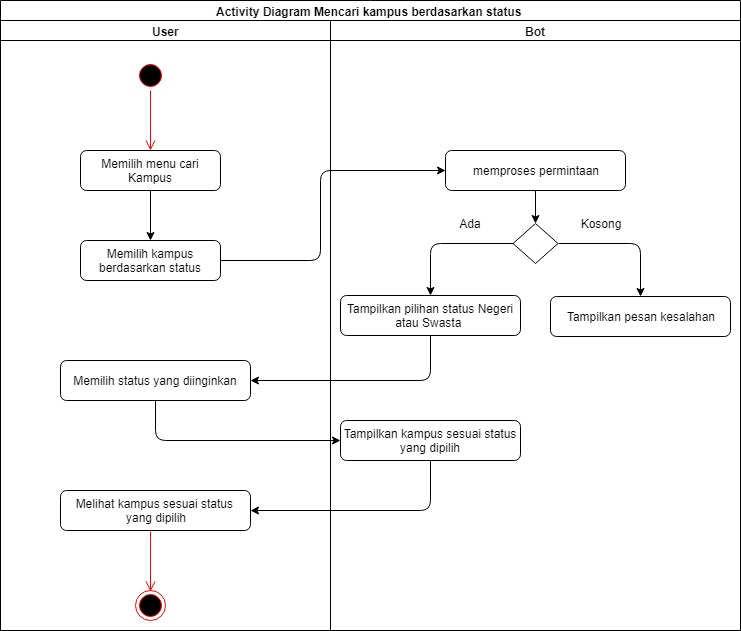
Activity diagram Mencari informasi kampus berdasarkan akreditasi yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi kampus berdasarkan akreditasi. Gambar III.16 menunjukkan activity diagram mencari informasi kampus berdasarkan akreditasi :



Gambar III.25 activity diagram mencari kampus berdasarkan akreditasi

* 1. Activity Diagram Mencari kampus berdasarkan status

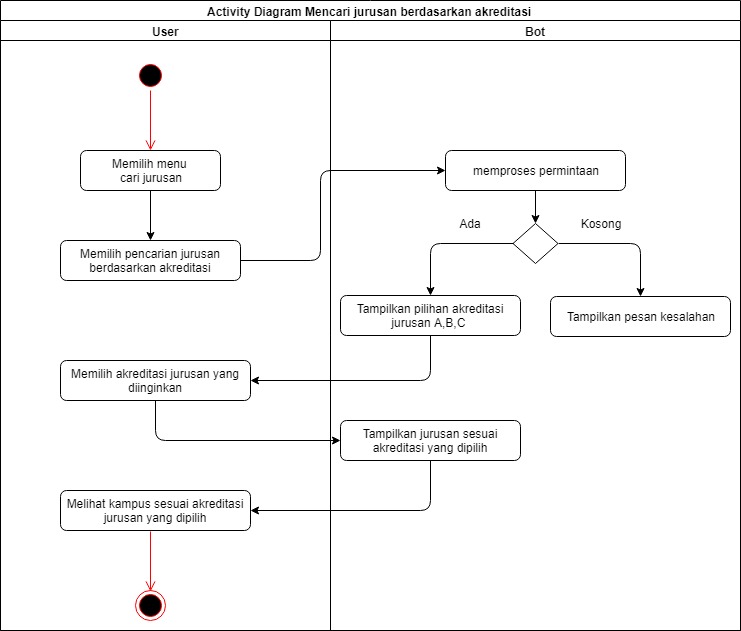
Activity diagram Mencari informasi kampus berdasarkan status yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi kampus berdasarkan status. Gambar III.17 menunjukkan activity diagram mencari informasi kampus berdasarkan status :



Gambar III.26 activity diagram mencari kampus berdasarkan status

* 1. Activity Diagram Mencari jurusan berdasarkan akreditasi

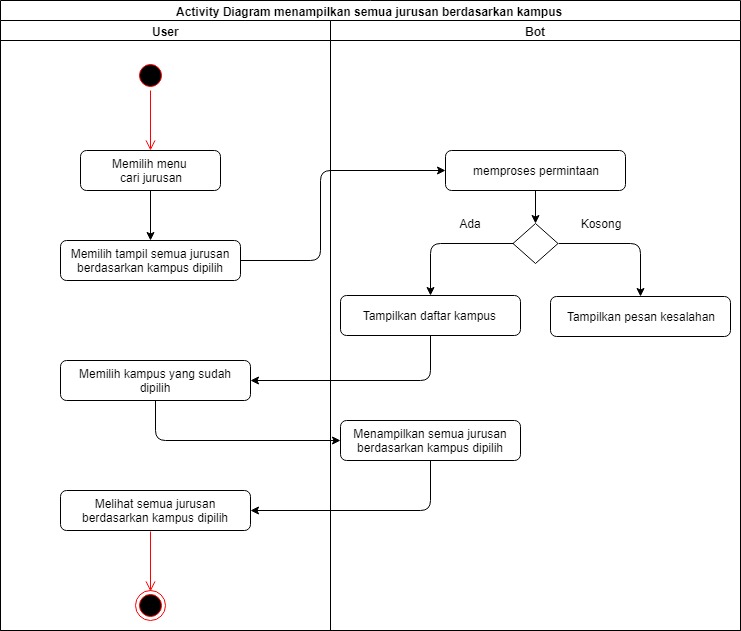
Activity diagram Mencari informasi jurusan berdasarkan akreditasi yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi jurusan berdasarkan akreditasi. Gambar III.17 menunjukkan activity diagram mencari informasi jurusan berdasarkan akreditasi :



Gambar III.27 activity diagram mencari jurusan berdasarkan akreditasi

* 1. Activity Diagram menampilkan semua jurusan berdasarkan kampus

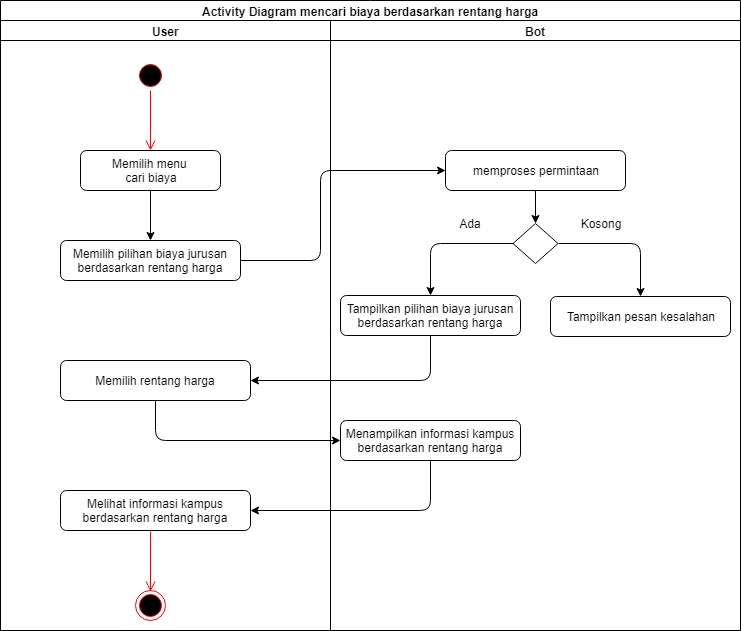
Activity diagram menampilkan semua informasi jurusan berdasarkan kampus yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi semua jurusan berdasarkan kampus. Gambar III.17 menunjukkan activity diagram menampilkan semua informasi jurusan berdasarkan kampus :



Gambar III.28 activity diagram menampilkan semua jurusan berdasarkan kampus

* 1. Activity Diagram menampilkan biaya berdasarkan rentang harga

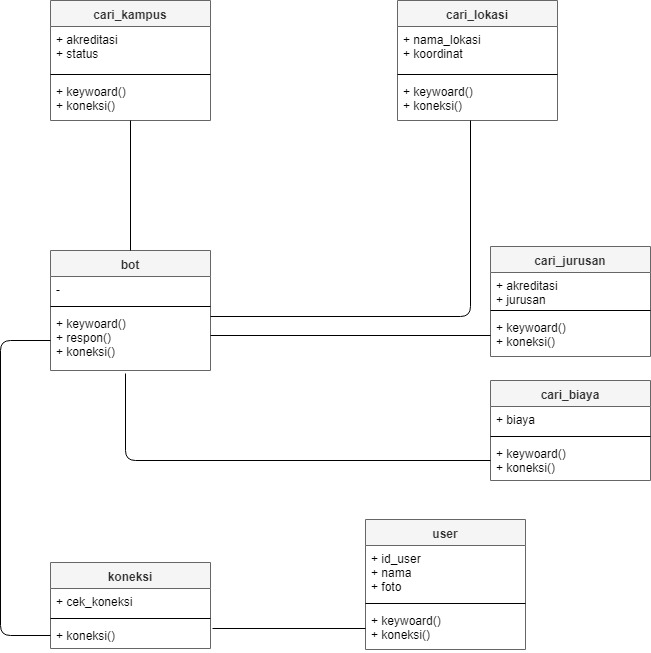
Activity diagram menampilkan biaya berdasarkan rentang harga yang menggambarkan kegiatan-kegiatan yang terjadi pada saat pengguna mencari informasi biaya berdasarkan rentang harga. Gambar III.17 menunjukkan activity diagram menampilkan biaya berdasarkan rentang harga :



Gambar III.29 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.

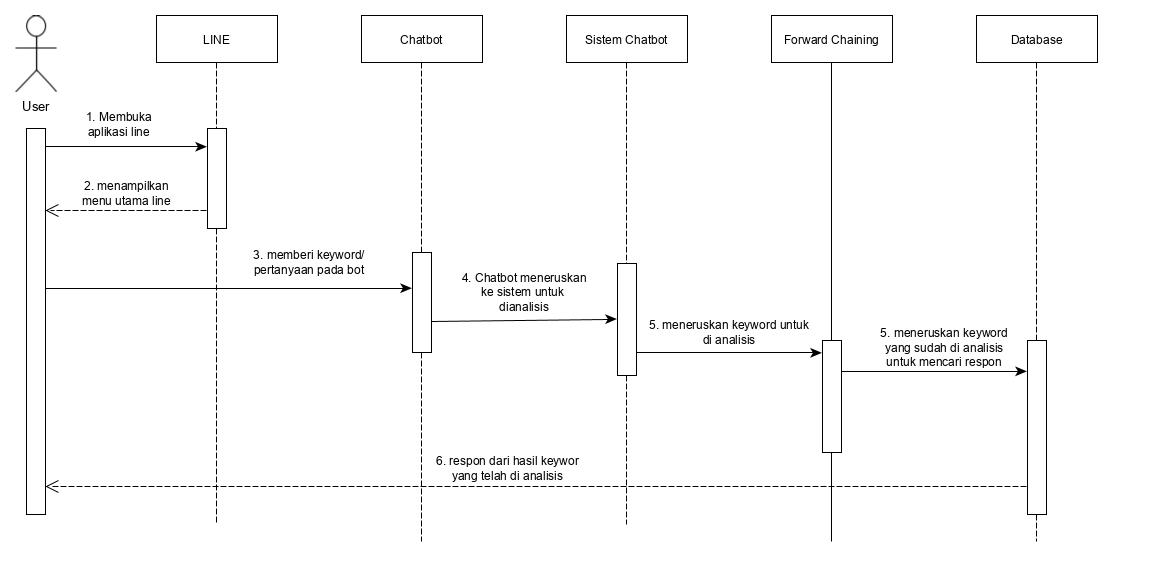


Gambar III.30 Class Diagram

1. **Sequence Diagram**

Proses ini menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antara objek yang terjadi pada titik tertentu dalam eksekusi sistem. Sequence diagram system klasifikasi ini yaitu Sequence Diagram Pertanyaan. Sequance Diagram Pertanyaan merupakan diagram yang menyajikan proses menu pertanyaan dan interaksi dengan sistem chatbot dapat dilihat pada gambar 3.15. Penjelasan langkah-langkahnya adalah sebagai berikut :

1. User (siswa SMA/K) membuka aplikasi LINE.
2. LINE akan menampilkan menu utama LINE.
3. User memberika keyword/pertanyaan pada chatbot.
4. Chatbot akan meneruskannya ke sistem untuk dianalisa.
5. Sistem chatbot menganalisa dengan menggunakan metode forward chaining
6. Setelah dianalisis maka diteruskan ke database untuk mencari respon
7. User menerima respon jawaban dari hasil keyword yang sudah diolah



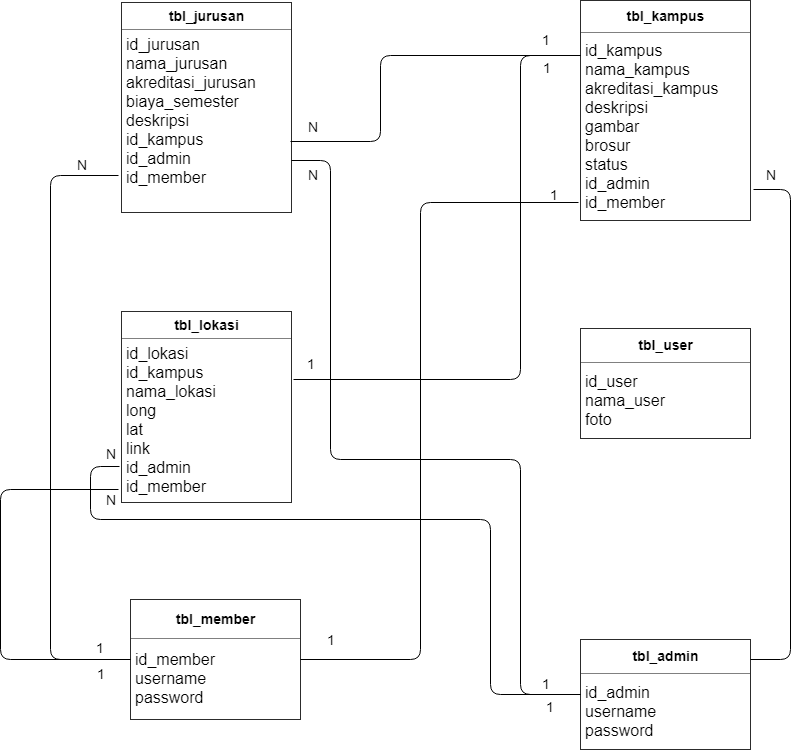
Gambar 1.6.31 Sequence Diagram Chatbot

1. **Perancangan Basis Data**

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

**5.1 Skema Relasi**

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



**5.2 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.15 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.16 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.17 Struktur Tabel Kampus**

|  |  |  |  |
| --- | --- | --- | --- |
| **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.18 Struktur Tabel Kampus**

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

* + 1. Tabel Member

**Tabel III.19 Struktur Tabel Kampus**

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

* + 1. Tabel Admin

**Tabel III.20 Struktur Tabel Kampus**

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

1. **Perancangan Antarmuka**

Perancangan antarmuka dilakukan untuk menggambarkan antarmuka aplikasi yang akan dibangun. Tabel III.21 s.d. Tabel III.55 menunjukkan perancangan antarmuka yang dilakukan.

Tabel III.21 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.22 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.23 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.x Perancangan antarmuka pilihan B

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

Tabel III.24 Perancangan antarmuka status kampus

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

Tabel III.25 Perancangan antarmuka swasta

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

Tabel III.26 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.27 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.28 Perancangan antarmuka Akreditasi C

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

Tabel III.29 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.30 Perancangan antarmuka unikom

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

Tabel III.31 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.32 Perancangan antarmuka rentang biaya

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

Tabel III.33 Perancangan antarmuka rentang biaya

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

Tabel III.34 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.35 Perancangan antarmuka tampil biaya

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

Tabel III.36 Perancangan antarmuka menu profile

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

Tabel III.37 Perancangan antarmuka menu lokasi

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

Tabel III.38 Perancangan antarmuka menu panduan

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

1. Perancangan Jaringan Semantik Antarmuka