

BAB 2

TINJAUAN PUSTAKA

2.1. Profil Dinas Perhubungan Kabupaten Majalengka

Dinas Perhubungan kabupaten Majalengka beralamat di Jl. Pangeran Muhammad Km. 5 Simpeureum, kecamatan Cigasong, Majalengka, Jawa Barat. Dinas Perhubungan kabupaten Majalengka memiliki tugas untuk menyelenggarakan urusan kebijakan perhubungan atau angkutan untuk daerah kabupaten Majalengka.

2.1.1. Sejarah Pendirian

Sebagai salah satu unsur pemerintah Tingkat I Jawa Barat Dinas Lalu Lintas dan Angkutan Jalan (DLLAJ) bertugas melaksanakan kebijakan yang ditetapkan oleh Dirjen Perhubungan Darat melalui Direktorat Lalu Lintas dan Angkutan Jalan Raya yang ditetapkan yang disesuaikan dengan kebijakan Pemerintah Daerah Tingkat I Jawa Barat khususnya dalam bidang Lalu Lintas dan Angkutan Darat. Tugas dan wewenang DLLAJ adalah mengawasi terselenggaranya urusan lalu lintas di daerah terutama dalam hal perijinan angkutan antar provinsi baik otobis maupun truk. Disamping itu juga sebagai delik-delik lalu lintas serta pengujian kendaraan bermotor.

Dari tahun 1942 sampai tahun 1945, Organisasi Pengatur Lalu Lintas Jalan tidak berfungsi, hal ini karena alasan politis serta kondisi pemerintahan pada saat itu dalam situasi perang kemerdekaan. Kemudian baru pada tahun 1950 organisasi Lalu Lintas Jalan diaktifkan kembali dibawah Departemen Lalu Lintas Jalan dan Pengairan Negara.

Sesuai dengan Peraturan Daerah Provinsi Jawa Barat Nomor 5 Tahun 2002 tentang Perubahan Atas Peraturan Daerah Provinsi Jawa Barat Nomor 15 Tahun 2002 tentang Dinas Daerah Provinsi Jawa Barat, maka Dinas Lalu Lintas dan Angkutan Jalan (DLLAJ) berubah menjadi Dinas Perhubungan.

Kedudukan Dinas Perhubungan Komunikasi dan Informatika berdasarkan dasar Hukum Peraturan Daerah Nomor. 10 tahun 2009 tanggal 1 Desember 2009, tentang Kedudukan, tugas pokok dan Fungsi susunan Organisasi Perangkat Daerah [4].

2.1.2. Visi dan Misi Dinas Perhubungan Kabupaten Majalengka

Visi misi adalah konsep perencanaan yang disertai dengan tindakan sesuai dengan apa yang direncanakan untuk mencapai suatu tujuan.

Visi dari Dinas Perhubungan kabupaten Majalengka adalah:

“Terwujudnya transportasi, komunikasi dan informatika yang handal, berbasis masyarakat religius, maju dan sejahtera.

Misi dari Dinas Perhubungan kabupaten majalengka adalah:

1. Penyediaan sarana fasilitas aparatur.
2. Menciptakan kawasan lalu lintas yang aman, tertib dan nyaman.
3. Meningkatkan sarana dan prasarana angkutan yang representatif.
4. Meningkatkan akses informasi secara merata bagi masyarakat.
5. Membangun kredibilitas lembaga sebagai pengelola komunikasi dan informatika.
6. Meningkatkan kualitas SDM dan pelayanan jasa perhubungan, komunikasi dan informatika.

2.1.3. Tugas Pokok

Dinas Perhubungan mempunyai tugas pokok merumuskan, menyelenggarakan, membina, dan mengevaluasi penyusunan dan pelaksanaan kebijakan daerah dalam bidang perhubungan yang meliputi Bidang Teknik Keselamatan dan Angkutan Jalan, Bidang Lalu Lintas, dan Bidang Prasarana Perhubungan [2].

2.1.4. Fungsi

Dalam melaksanakan tugas pokok Dinas Perhubungan mempunyai fungsi:

1. Perumusan kebijakan teknis bidang perhubungan yang meliputi Bidang Teknik Keselamatan dan Angkutan Jalan, Bidang Lalu Lintas, dan Bidang Prasarana Perhubungan.
2. Penyelenggaraan urusan yang meliputi Bidang Teknik Keselamatan dan Angkutan Jalan, Bidang Lalu Lintas, dan Bidang Prasarana Perhubungan.
3. Pembinaan, pelaksanaan dan evaluasi urusan bidang perhubungan yang meliputi Bidang Teknik Keselamatan dan Angkutan Jalan, Bidang Lalu Lintas, dan Bidang Prasarana Perhubungan.

2.2. Landasan Teori

Teori memiliki peran sangat besar dalam penelitian serta merupakan unsur terpenting dalam penelitian. Landasan teori dari penelitian disini bisa disebut sebagai studi literatur atau tinjauan pustaka. Hasil dari landasan teori diperoleh dari simpulan-simpulan atau pendapat para ahli yang kemudian diciptakan pada pendapat yang baru.

2.2.1. Multimedia

Multimedia, berasal dari pada kata multi dan media. Multi berarti banyak atau lebih dari satu dan media berasal dari kata medium yang berarti tempat, alat atau sarana. Jadi berdasarkan kata multimedia dapat dirumuskan sebagai wadah atau penggabungan beberapa media yang kemudian didefinisikan sebagai elemen-elemen pembentuk multimedia. Elemen-elemen tersebut berupa teks, gambar, suara, animasi dan video.

2.2.2. Modeling

Modeling merupakan istilah untuk menunjukan terjadinya proses belajar melalui observasi, bukan sekedar meniru tetapi meliputi penambahan atau pengurangan meliputi proses kognitif.

2.2.3. Chatbot

Chatbot adalah sebuah program otomatisasi yang membantu proses interaksi dengan pengguna secara otomatis berdasarkan serangkaian kondisi atau peristiwa yang telah ditentukan sebelumnya.

Dengan teknologi otomatisasi ini, memungkinkan *costumer service* tanpa terjun langsung untuk melayani pengguna. Bisa dibidang *chatbot* sebuah *costumer service online* yang bersifat otomatis.

2.2.3.1. Perkembangan Chatbot

Pada tahun 1950, Alan Turing merumuskan *Turing Test*, sebuah pendekatan berbasis percakapan yang digunakan untuk menentukan apakah mesin dapat meniru manusia. Tes ini melibatkan seorang interogator (manusia) dihadapkan dengan dua subjek, yaitu manusia dan mesin. Seorang *interogator* memberikan pertanyaan melalui ketikan kata-kata, dan dituntut untuk menentukan subjek mana yang merupakan manusia dan mana yang sebuah mesin. Sebuah mesin dinyatakan lolos jika seorang *interogator* tidak dapat membedakan antara manusia dan mesin [5].

Eliza, induk dari semua *chatbot* yang diciptakan oleh Joseph Weizenbaum tahun 1966. Sistem operasi Eliza didesain agar dapat mengenali kata atau kalimat kunci dari input, sehingga dapat memproduksi respon sesuai dengan yang sudah di program.

Parry, *chatbot* yang diciptakan oleh Kenneth Colby tahun 1972. Dengan alasan rasa ketidakpuasan psikoanalisis yang tidak mampu untuk menghasilkan data yang akurat untuk memajukan ilmu pengetahuan. Parry dapat digambarkan sebagai *chatbot* dengan tingkat keseriusan yang lebih jauh dalam menciptakan sebuah kecerdasan buatan.

Pada tahun 1995, ALICE atau *Artificial Linguistic Internet Computer Entity* diciptakan oleh Richard Wallace. ALICE dapat menggunakan pemrosesan Bahasa alami atau *Natural Language Processing* (NLP). ALICE memungkinkan untuk memberikan respon pada level yang lebih rumit. ALICE akan menyimpan percakapan yang di terimanya dan akan menambahkan kedalam basis datanya.

2.2.3.2. Jenis Chatbot

Saat ini pengembangan *chatbot* yang banyak digunakan ada dua jenis linguistik dan *Machine Learning* berikut jenis dari *chatbot* [6].

a. *Chatbot* berbasis Linguistik

Chatbot linguistik disebut sebagai *chatbot* berbasis aturan, yaitu *chatbot* yang memberikan kontrol menggunakan logika 'jika' atau 'kemudian' untuk membuat arus percakapan. Penggunaan Bahasa dapat diatur untuk menyesuaikan kata-kata, urusan, sinonim, cara umum untuk membuat frase pertanyaan dan banyak lagi. Hal tersebut dilakukan untuk memastikan bahwa pertanyaan dengan arti yang sama akan menerima jawaban yang sama. Jika ada pertanyaan yang tidak dapat dijawab oleh *chatbot*, maka disitulah peran manusia untuk menyesuaikan kondisi dan menjawab pesan itu.

b. Machine Learning (AI *Chatbot*)

Chatbot ini menggunakan teknologi yang didukung oleh *Artificial Intelegent*, teknologi ini lebih kompleks dari pada *chatbot* berbasis aturan dan cenderung lebih komunikatif, berdasarkan data dan prediktif. Sistem percakapan berdasarkan *Machine Learning* bisa menjadi mengesankan jika pertanyaan atau permintaan sudah dapat teratasi. Berdasarkan sifatnya, *chatbot* ini belajar dari pola dan pengalaman sebelumnya.

c. Hybrid

Chatbot model ini menggunakan pendekatan yang menggabungkan teknologi linguistik dan *artificial intelegent*, namun tetap mempertahankan sifat maupun unsur asli dari kedua teknologi tersebut, dan menawarkan kemampuan untuk memberikan solusi percakapan *chatbot artificial intelegent* yang lebih rinci.

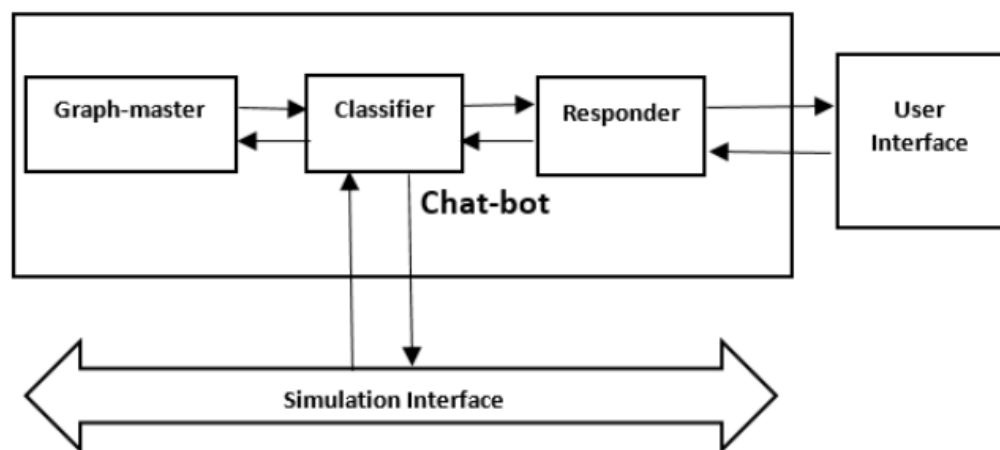
d. Voice Command

Chatbot ini memungkinkan pengguna untuk melakukan percakapan menggunakan perintah suara. Dimana hal ini akan jauh lebih praktis untuk digunakan oleh *customer* maupun *client*.

2.2.3.3. **Komponen *Chatbot***

Paket software *chatbot* terdiri dari tiga komponen penting: Responder, *Classifier* dan Graph master. Antarmuka antara rutin utama dan

pengguna adalah Responder. Responden mentransfer data dari pengguna ke Classifier. Responder mengontrol input dan outputnya. Fungsi Classifier adalah menormalkan dan menyaring data masukan. Masukan pengguna diganti dan dibagi menjadi komponen logis oleh Classifier. Classifier memindahkan kalimat yang dinormalisasi ke dalam komponen Graph-master dari chatbot. Classifier memproses output dari komponen master Graph dari chatbot. Classifier juga menangani instruksi sintaks database (misalnya AIML). Komponen graph-master chatbot menangani pola pencocokan. Graph-master bertanggung jawab untuk mengatur penyimpanan konten otak chatbot. Komponen Graph-master menyimpan isi otak sebagai grafik. Graph-master juga menangani proses pencocokan pola dan algoritma pencocokan pola. Diagram komponen chatbot ditunjukkan pada gambar 2.1.

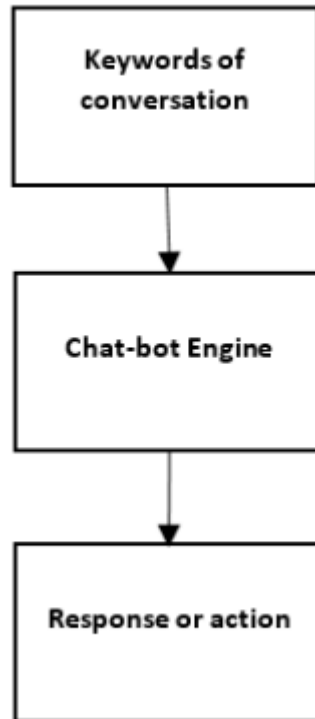


Gambar 2. 1 *Komponen Chatbot*

2.2.3.4. Text Processing, Response dan Action

Input teks pengguna dipecah menjadi beberapa kata terpisah untuk menandai label part-of-speech sehubungan dengan posisi dan tetangga mereka dalam teks masukan. Pada tahap berikutnya, dengan bantuan berbagai jenis tata bahasa, kata-kata yang diberi label secara individual digabungkan untuk mengembangkan ungkapan. Pada tahap operasi chinking, kata kunci penting diambil dari frasa dengan menghapus kata-kata yang tidak diinginkan. Kata kunci diperiksa dan

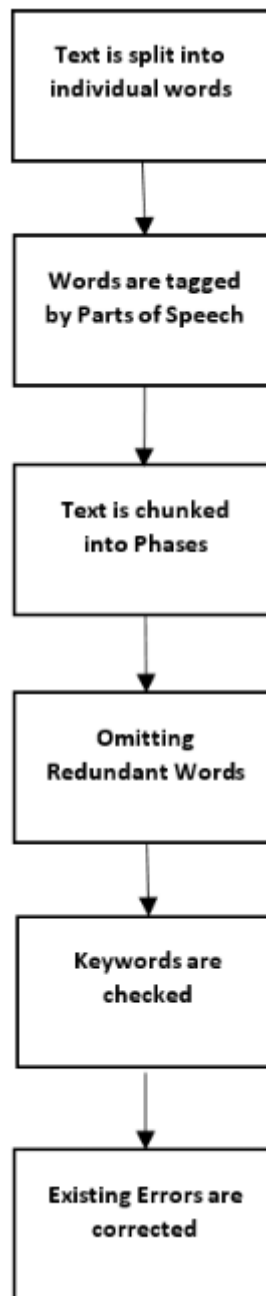
dikoreksi jika tidak benar. Tahap fase pemrosesan teks yang berbeda digambarkan pada gambar



Gambar 2. 2 Fase Respond dan Tindakan

2.2.3.5. Text Processing

Chatbot dapat dirancang untuk memberikan respons yang diharapkan terhadap percakapan teks bahasa manusia. Mesin *Chatbot* dilengkapi dengan kata kunci yang diambil dari pemrosesan teks bahasa alami. Outputnya adalah respon, yang akan ditunjukkan kepada pengguna. Diagram yang menggambarkan respons dan fase pengambilan tindakan ditunjukkan pada gambar 2.3.



Gambar 2. 3 Fase dari *Text Processing*

2.2.4. Basis Data

Database atau Basis Data adalah sekumpulan data yang saling terhubung satu dengan yang lainnya atau sekumpulan tabel yang saling terhubung satu dengan yang lainnya. dan fungsi dari database adalah menyimpan suatu data pada tabel-tabel dan dikumpulkan menjadi satu dengan database. Database juga bisa di umpamakan sebagai sebuah rumah dengan beberapa kamar-kamar dan sebuah

property seperti almari meja belajar tempat tidur itu bisa di sebut dengan data *querynya*.

2.2.4. Python

Python sendiri merupakan bahasa Pemrograman yang telah populer serta banyak digunakan di Data Analysis, para Software Engineers, dan Data Scientists dalam mengoperasikan pembangunan di salah satu aplikasi serta menggali secara lebih mendalam pada machine learning.

2.2.5. Flask

Flask adalah kerangka kerja aplikasi web (*web framework*) yang bersifat *micro*, istilah ini dipakai karena *flask* menyediakan pustaka atau alat bantu inti yang digunakan untuk proses membangun sebuah aplikasi web, jadi istilah *micro* ini bukan berarti bahwa mengembangkan aplikasi web dengan *flask* hanya dilakukan dalam skala kecil. Apabila ada kebutuhan lain yang dibutuhkan, *flask* mengizinkan untuk menambah fitur lain yang spesifik sebagai ekstensi.

2.2.6. Chatterbot

Chatterbot adalah *library* Python yang memudahkan untuk menghasilkan respon otomatis atas masukan pengguna. *Chatterbot* menggunakan pilihan algoritme pembelajaran mesin untuk menghasilkan berbagai jenis respon. Ini memudahkan pengembang untuk membuat *chatbot* dan percakapan otomatis dengan pengguna.

2.2.5. Pemrograman Berorientasi Objek

Pemrograman berorientasi objek (*Object Oriented Programming* atau disingkat OOP) merupakan konsep pemrograman yang difokuskan pada penciptaan kelas yang merupakan abstraksi/*prototype* dari suatu objek. Sedangkan sisten berorientasi objek adalah sistem yang dibuat berdasarkan metode orientasi objek. Bahasa pemrograman yang dapat menggunakan konsep OOP diantaranya Java, C++, C#, PHP, Eiffel, dan Smalltalk. Bagian -bagian dari struktur OOP yaitu:

1. Kelas (*Class*)

Kelas merupakan penggambaran dari sebuah objek atau benda, sifat objek, dan juga apa yang bisa dilakukan oleh objek tersebut.

2. Objek (Obyek)

Objek merupakan instans (perwujudan) dari suatu kelas. Objek merupakan suatu entitas yang mampu menyimpan informasi dan mempunyai operasi yang dapat diterapkan atau dapat berpengaruh pada status objeknya.

3. Metode (Method)

Metode adalah prosedur atau fungsi yang dimiliki oleh sebuah objek. Sebuah kelas boleh memiliki lebih dari satu metode yang berfungsi untuk memanipulasi objek itu sendiri. Metode juga merupakan cara objek untuk berkomunikasi antar objek.

4. Atribut (Attribute)

Atribut adalah sifat karakteristik atau kondisi yang dimiliki oleh suatu objek. Atribut dapat berupa nilai atau elemen – elemen data yang dimiliki oleh objek dalam kelas objek.

5. Enkapsulasi (Encapsulation)

Enkapsulasi ialah lapisan pelindung yang mencegah kode dan data yang secara acak diakses oleh kode lain yang didefinisikan di luar kelas.

6. Pewarisan (Inheritance)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya. Konep pemograman dimana sebuah *class* dapat menurunkan property dan metode yang dimilikinya kepada *class* lain.

7. Antarmuka (Interface)

Antarmuka atau interface sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah interface dapat diimplementasikan oleh kelas lain.

8. Generalisasi dan Spesialisasi

Generalisasi dan Spesialisasi menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus.

9. Polimorfism (Polymorfism)

Polimorfisme yaitu suatu objek bisa bertindak lain terhadap message/method yang sama.

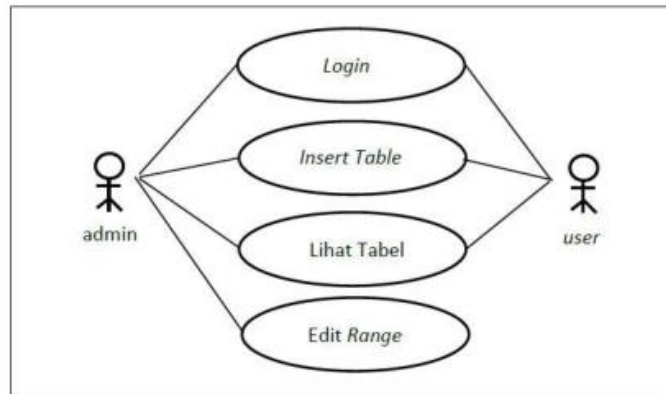
2.2.6. Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa penulisan spesifikasi standar cetak biru perangkat lunak. UML digunakan untuk menspesifikasikan, mendokumentasikan, dan membangun sistem perangkat lunak.

Pengembangan UML dimulai dari kerja sama Grady Booch dan James Rumbaugh pada 1994 untuk menggabungkan dua metodologi terkenal yaitu Booch dan OMT. Kemudian Ivan Jacobson, pencipta metode OOSE (Object Oriented Software Engineering) bergabung. Usulan UML diberikan kepada OMG (Object Management Group) sebagai konsorsium standardisasi teknologi objek agar UML dijadikan bahasa dan notasi pemodelan dilakukan pada tahun 1997. OMG menerima UML, UML telah menjadi standar de-facto karena pencipta-penciptanya sangat populer. Banyak pengembang perangkat lunak yang mengadopsi UML.

2.2.6.1. Use case Diagram

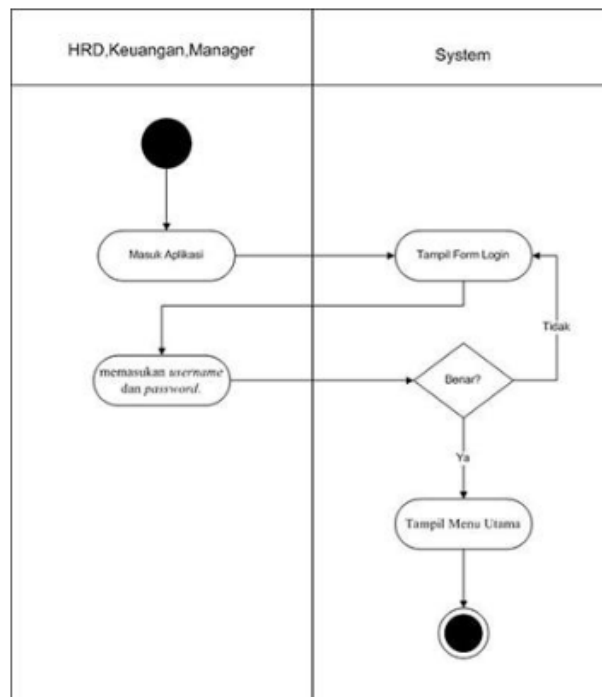
Use case Diagram menjabarkan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (Aktor). Diagram ini menunjukkan fungsionalitas suatu sistem yang berinteraksi dengan dunia luar. *Use case* Diagram dapat digunakan selama proses analisis untuk menangkap requirement sistem dan untuk memahami bagaimana sistem bekerja.



Gambar 2. 4 Contoh Use case Diagram

2.2.6.2. Activity Diagram

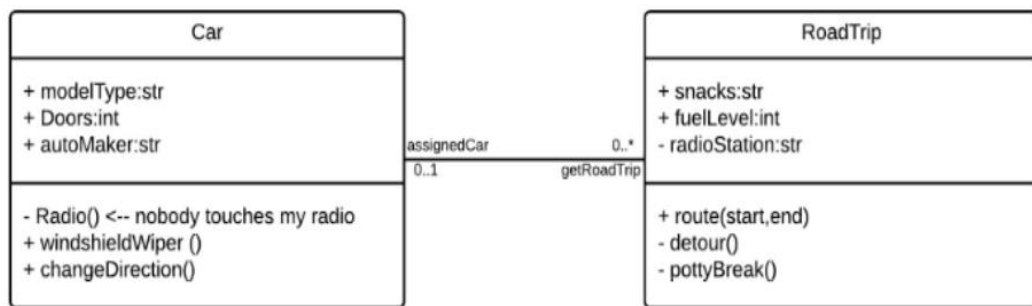
Activity diagram menggambarkan perilaku dinamis dari suatu sistem melalui aliran control diantara aksi-aksi yang dilakukan sistem. Ini mirip dengan diagram alir kecuali bahwa diagram aktivitas dapat menunjukkan aliran-aliran konkuren. Komponen utama suatu diagram aktivitas adalah node aksi (*action node*), direpresentasikan dengan suatu persegi panjang dengan sudut bulat, yang berhubungan dengan suatu tugas yang dilakukan oleh sistem perangkat lunak.



Gambar 2. 5 Contoh Activity Diagram

2.2.6.3. Class Diagram

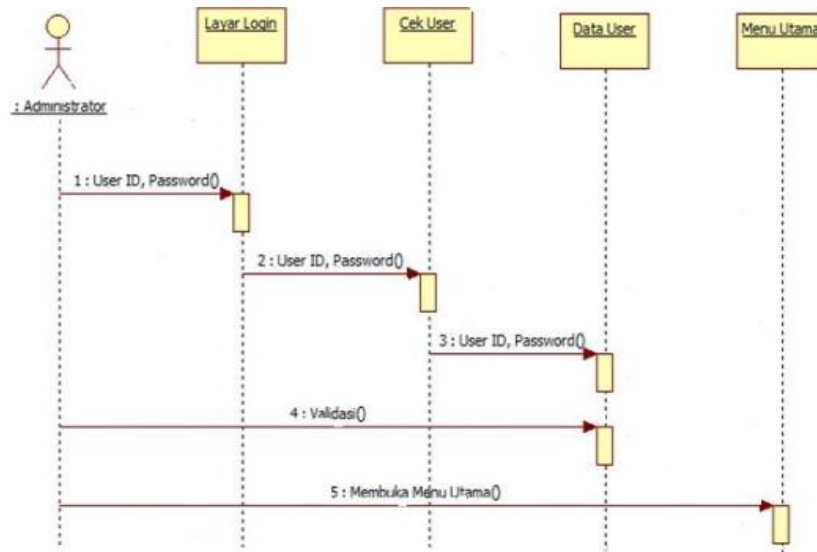
Class Diagram menjelaskan dalam struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class* Diagram memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas dalam model desain dari suatu sistem. Selama proses analisis, *class* diagram memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap desain, *class* diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.



Gambar 2. 6 Contoh Class Diagram

2.2.6.4. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.



Gambar 2. 7 Contoh Sequence Diagram

2.2.7. Metode Pengujian Sistem (*Blackbox*)

Metode pengujian sistem dilakukan untuk mengetahui kesesuaian dari perangkat lunak yang digunakan selain memberikan kesempatan kepada pengguna untuk mengoperasikan dan melakukan pengecekan terhadap laporan yang dihasilkan melalui perangkat lunak. Metode pengujian sistem yang dilakukan yaitu pengujian *blackbox*.

Pengujian yang dilakukan untuk antarmuka perangkat lunak, pengujian ini dilakukan untuk memperlihatkan bahwa fungsi-fungsi bekerja dengan baik dalam arti masukan yang diterima dengan benar dan keluaran yang dihasilkan benar-benar tepat, pengintegrasian dari eksternal data berjalan dengan baik. Metode pengujian *black-box* memfokuskan pada requirement fungsi dari perangkat lunak, pengujian ini merupakan komplemetari dari pengujian *white-box*. Pengujian *white-box* dilakukan terlebih dahulu pada proses pengujian, sedangkan pengujian *black-box* dilakukan pada tahap akhir dari pengujian perangkat lunak. Proses yang terdapat dalam proses pengujian *black-box* yaitu analisis batasan nilai yang berlaku untuk setiap data.

2.2.8. Perangkat Lunak Pendukung

Pada pembangunan perangkat lunak ini dibutuhkan perangkat lunak pendukung, diantaranya:

2.2.8.1. Sublime Text

Pada pembangunan program ini untuk membangun kode menggunakan *Sublime Text*. *Sublime Text* adalah *source-code editor* yang dirancang oleh Jon Skinner dan dikembangkan oleh Sublime Hq Pty Ltd untuk Windows, Linux, dan OS X.

2.2.8.2. Draw.io

Draw.io adalah perangkat lunak pemodelan atau perancangan yang mendukung UML (*Unified Modeling Language*). Dalam penelitian yang dibuat, peranan Draw.io sangat penting yaitu sebagai pembuatan model-model, rancangan, dan skema untuk memenuhi analisis kebutuhan yang akan dibuat. Seperti *usecase diagram*, *activity diagram*, *class diagram*, dan sebagainya.