

BAB II LANDASAN TEORI

1.1. Tinjauan Perusahaan

Tahap tinjauan pada perusahaan ini merupakan peninjauan terhadap tempat penelitian studi kasus yang dilakukan di PO. Sumber Alam.

2.1.1. Sejarah Singkat Perusahaan

Perusahaan otobus Sumber Alam hingga saat ini sudah mengalami beberapa kali transformasi dalam perkembangannya. Pada kisaran tahun 1969 dibentuklah PO Tresno yang dirintis oleh Ibu Thung Tjie Hing yaitu nenek dari Bapak Judi Setijawan Hambali kemudian setelah Ibu Thung Tjie Hing mewariskan usaha tersebut kepada anak cucunya nama perusahaan tersebut berubah menjadi PO Hidup Baru.

Pada tahun 1975 didirikanlah PO Sumber Alam sebagai perusahaan keluarga, pada awal berjalannya PO Sumber Alam hanya menggunakan 6 unit bus. Dan pada 1984 adik dan kakak Bapak Judi Setijawan Hambali tidak ingin lagi ikut ambil bagian dalam mengembangkan usaha PO Sumber Alam ini. Dengan berbekal pengalaman yang ada dan keinginan keras untuk mengembangkan usaha ini akhirnya Bapak Judi Setijawan Hambali meneruskan usaha ini serta dibantu oleh sang istri Ibu Retno Harlani Judana.

2.1.2. Nilai – Nilai Perusahaan

Untuk mencapai tujuan perusahaan, setiap insan PO. SUMBER ALAM harus memiliki sikap bekerja keras, jujur, terbuka dan pantang menyerah dalam memberikan pelayanan kepada masyarakat. Mempertahankan kemandirian perusahaan dengan menciptakan suasana damai melalui kerjasama. Serta bersama-sama bekerja dalam suasana kebersamaan yang dapat memberikan rasa aman dengan menjaga lingkungan kerja yang menyenangkan dan perilaku yang ramah-tamah dilandasi kekuatan iman dan taqwa.

2.1.3. Visi dan Misi

Setiap organisasi harus memiliki falsafah dimana organisasi tersebut melakukan kebijakan dan tindakannya. PO. Sumber Alam memiliki Visi dan Misi sebagai berikut:

A. Visi Perusahaan

Menjadi armada angkutan pilihan utama di Indonesia.

B. Misi Perusahaan

1. Menyelenggarakan jasa angkutan darat yang mengutamakan keselamatan, ketepatan waktu, dan pelayanan yang prima dengan sentuhan keramah-tamahan.
2. Memaksimalkan pertumbuhan nilai perusahaan, efisiensi, dan menyejahterakan pegawai.
3. Menjadi mitra yang bisa dipercaya oleh masyarakat khususnya pengguna jasa transportasi angkutan darat PO. Sumber Alam.

2.1.4. Logo Perusahaan

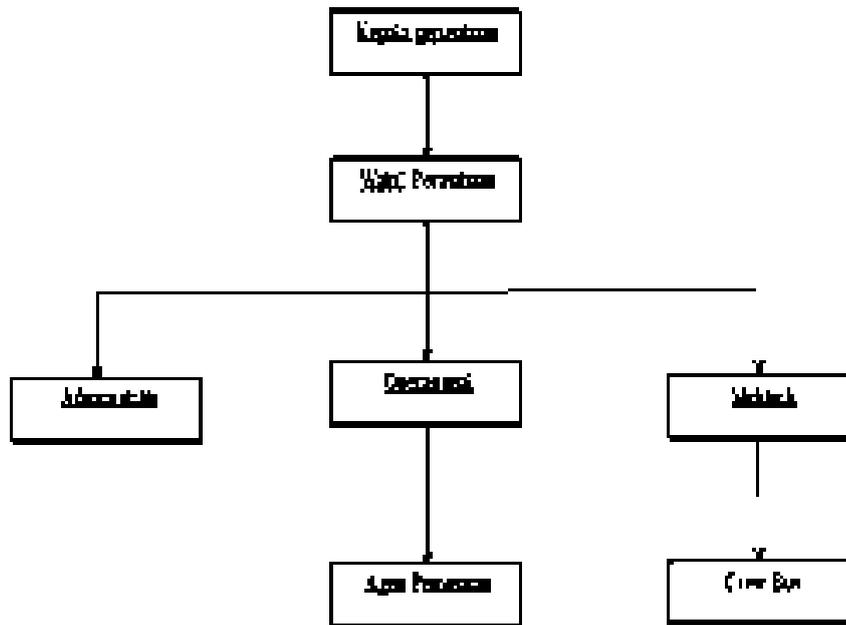
Setiap perusahaan memiliki logo yang menjadikan ciri khas suatu organisasi, sebagai perusahaan PO. Sumber Alam juga memiliki logo yang dapat dilihat pada gambar 2.1



Gambar 2.1.4.1 Logo Perusahaan

2.1.5. Struktur Organisasi Perusahaan

Struktur organisasi yang terdapat di PO. Sumber Alam terdapat pada gambar 2.2.



Gambar 2.1.5.1 Struktur Organisasi

2.1.6. Deskripsi Tugas

Berikut adalah deskripsi tugas dari masing-masing jabatan yang tertera dalam struktur organisasi :

1. Kepala Perusahaan
 - a. Bertanggung jawab atas aktivitas perusahaan dan kelancaran usaha perusahaan.
 - b. Memimpin aktivitas perusahaan sesuai dengan kebijakan yang telah ditetapkan.
 - c. Merencanakan dan membuat program di semua sistem yang ada dalam perusahaan dan mengkoordinasikannya.
 - d. Mengarahkan semua unit kerja pada tujuan perusahaan.
2. Wakil Perusahaan
 - a. Membantu tugas – tugas kepala perusahaan.
 - b. Mewakili kepala perusahaan apabila berhalangan.
3. Operasional
 - a. Bertanggung jawab atas operasional perusahaan secara

- keseluruhan.
- b. Mengadakan pengawasan jalannya operasional perusahaan.
 - c. Menyusun segala sesuatu yang berhubungan dengan operasional bus.
 - d. Menyusun pergantian tugas supir dan pembantu supir.
4. Administrasi
 - a. Bertugas melakukan pencatatan, meliputi administrasi kepegawaian, mengadakan korespondensi surat keluar masuk serta membuat laporan ke kantor perhubungan dan menginventaris peralatan perusahaan.
 5. Mekanik
 - a. Bertanggung jawab atas pelaksanaan tugas kegiatan untuk pemeliharaan bengkel.
 - b. Menyediakan armada bus yang siap digunakan untuk operasional.
 - c. Mengatasi kerusakan dan perbaikan.
 6. Agen Pemasaran
 - a. Bertanggung jawab dalam pelaksanaan tugas yang berkaitan dengan pemasaran. Tugas tersebut antara lain melaksanakan penjualan tiket kepada konsumen.
 7. Crew Bus
 - a. Mempersiapkan kendaraan yang dipercayakan dengan sebaik – baiknya.
 - b. Menjaga kendaraan dan keselamatan kerja.
 - c. Melaporkan jika ada kerusakan kendaraan.

1.2. Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan *chatbot* penyedia layanan informasi. Teori-teori yang terkait serta mendukung penelitian ini akan dibahas pada bab ini.

2.2.1. Multimedia

Multimedia berasal dari kata multi dan media yang memiliki makna multi berarti banyak dan media/medium berarti sarana untuk menyampaikan. Multimedia secara umum adalah kombinasi dari berbagai media seperti teks, gambar, suara, animasi, video dan lain-lain secara terpadu melalui komputer atau perangkat elektronik lain untuk mencapai tujuan tertentu. Elemen multimedia dibagi menjadi 2 bagian yaitu elemen multimedia yang diskret (tidak berbasis waktu) dan kontinyu (berbasis waktu).

Elemen multimedia diskret diantaranya adalah teks dan gambar. Berikut adalah penjelasan dari teks dan gambar :

- 1) Teks adalah suatu elemen multimedia yang paling dasar, teks terdiri dari gabungan kata yang digunakan untuk memberikan suatu pesan/informasi. Pemakaian teks dalam multimedia sangat bermanfaat bahkan dapat dikatakan hampir setiap produk multimedia mengandung teks didalamnya, hal tersebut dilakukan untuk dapat memaksimalkan penyampaian informasi.
- 2) Gambar adalah suatu citra dua dimensi yang dapat dimanipulasi oleh komputer. Beberapa contoh dari gambar diantaranya foto, grafik, ilustrasi, diagram, dan lain-lain. Gambar berfungsi sebagai visualisasi konsep verbal atau abstrak. Gambar digunakan agar memperjelas penyampaian suatu informasi verbal.

Elemen multimedia kontinyu diantaranya suara, animasi dan video. Berikut adalah penjelasan suara, animasi dan video :

- 1) Suara adalah gelombang yang dibangkitkan oleh suatu benda yang bergetar dalam media seperti udara. Suara biasa digunakan untuk memperjelas suatu teks maupun gambar. Suara memiliki sebuah frekuensi dan frekuensi yang dapat diterima oleh pendengaran manusia adalah 20Hz sampai 20.000Hz.
- 2) Animasi adalah kumpulan gambar yang bergerak secara urut guna menyajikan suatu proses tertentu. Animasi berguna untuk menyajikan suatu peristiwa yang sulit digambarkan, dalam multimedia animasi banyak digunakan dalam menyajikan materi pembelajaran yang sulit.
- 3) Video adalah suatu rekaman peristiwa yang berisikan urutan gambar

bergerak yang disertai dengan suara. Video digital kini menjadi suatu elemen multimedia yang cukup populer dikarenakan mudah dalam pengolahannya, akan tetapi video membutuhkan tempat penyimpanan yang besar.

2.2.2. Chatbot

Chatterbot (disebut juga *chatbot* atau *bots*) adalah sebuah program *software* komputer yang dirancang untuk mensimulasikan percakapan intelektual dengan satu atau lebih manusia baik secara audio maupun teks. Pada mulanya, program komputer (*bots*) ini diuji melalui *Turing Test*, yaitu dengan merahasiakan identitasnya sebagai mesin sehingga dapat mengelabui orang yang bercakap-cakap dengannya. Jika pengguna tidak dapat mengidentifikasi *bots* sebagai suatu program komputer, maka *chatbot* tersebut dikategorikan sebagai kecerdasan buatan atau *artificial intelligence*.

Chatbot telah dimanfaatkan untuk tujuan praktis seperti bantuan online, layanan personal, atau akuisisi informasi, dalam hal ini dapat dilihat fungsi program sebagai suatu jenis agen percakapan atau *conversational agent*. Yang membedakan chatbot dengan sistem pemrosesan bahasa alami atau *Natural Language Processing System* adalah kesederhanaan algoritma yang digunakan. Meskipun banyak *bots* yang tampaknya dapat mengartikan dan menanggapi *input* manusia, sebenarnya *bots* tersebut hanya memindai kata kunci dalam input dan membalasnya dengan kata kunci yang paling cocok, atau pola kata-kata yang paling mirip dari basis data tekstual [3].

Chatbot terdiri dari tiga kombinasi, di mana ketiga kombinasi inilah yang membentuk sebuah *chatbot*, di antaranya adalah:

- a. *User interface* dalam *chatbot* ini sendiri adalah jembatan antara chatbot dan user saling berinteraksi melalui aplikasi pesan berbasis *text*. *User Interface* haruslah dapat memberikan pengalaman yang lebih baik kepada *user* ketika berinteraksi dengan *Chatbot*.
- b. *Artificial Intelligence* (Kecerdasan Buatan) akan membuat *chatbot* mengerti dan memahami setiap interaksi yang terjadi dengan user. *Chatbot* menangani pemecahan masalah melalui aturan yang telah ditentukan

sebelumnya di pohon keputusan.

- c. Integrasi dengan sistem lainnya akan menambah kekayaan fitur yang terdapat di dalam suatu *chatbot*. Dengan mengintegrasikan *chatbot* ke sistem yang lain dapat menyediakan informasi tambahan. Dengan cara ini *chatbot* mampu memberikan informasi yang lebih kaya kepada *user*.

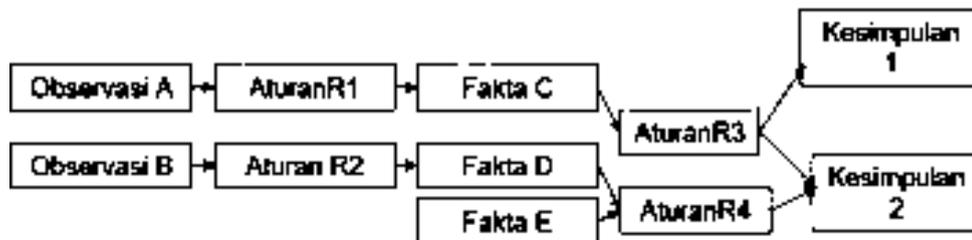
Chatbot biasa digunakan dalam IRC (*Internet Chat Groups*) di mana robot tersebut melakukan berbagai fungsi seperti menyapa partisipan baru dan memonitor penggunaan kata-kata yang tidak pantas. Sekarang ini, robot dapat mengumpulkan informasi, mengolah informasi tersebut, dan memberikan respon terhadap pengguna dan event yang terjadi di internet. Beberapa jenis robot bekerja secara silent di internet tanpa sepengetahuan kita namun ada pula robot yang bekerja secara interaktif yang dapat kita ajak berbicara yaitu *chatbot*.

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 antara 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.

2.2.3. Metode 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 [4].



Gambar 2.2.3.1 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.

2.2.4. Visual Studio Code

Visual Studio Code adalah aplikasi *editor* teks gratis dikembangkan oleh *Microsoft* yang dapat digunakan di semua bahasa pemrograman yang ada tanpa perlu berganti aplikasi *editor*, serta dapat dijalankan di berbagai *platform Operating System (OS)* seperti *windows*, *linux*, dan *mac OS*. *Visual Studio Code* memudahkan para *Programmer* saat berganti bahasa pemrograman tanpa perlu berganti aplikasi *editor* serta memahami dan konfigurasi *tools* kembali di aplikasi *editor* barunya. *Visual Studio Code* juga memberikan kebebasan kepada penggunanya dalam tema, *debugger*, *extension*, dan lainnya.

Kelebihan *Visual Studio Code* :

a) *Cross Platform*

Visual Studio Code dapat berjalan di berbagai *OS*, seperti *windows*, *linux* bahkan *mac OS*. Sehingga memudahkan *Programmer* yang memiliki *multi device* untuk mengerjakan proyek mereka.

b) *Multi Language*

Visual Studio Code dapat digunakan untuk berbagai bahasa pemrograman, Seperti *javascript*, *python*, *java*, *C++* dan lain lain. Agar mempermudah melakukan *coding* dengan bahasa pemrograman tersebut, terdapat ekstensi sesuai dengan bahasa

pemrograman yang membantu *programmer*. contohnya seperti ekstensi *Open in Browser* ekstensi ini memberikan antarmuka dari *file HTML* dengan mengunggah *file* tersebut langsung ke *default browser*.

c) *Extension*

Visual Studio Code memberikan kebebasan kepada pengguna fitur-fitur tambahan pilihan yang dapat memperluas kemampuan *editor* sesuai dengan kebutuhan pengguna. Fitur tambahan ini diberi nama *extensions*, yang dapat di *download* dan *install* secara langsung di *editor Visual Studio Code* tanpa harus mencari di *browser* dan *install* secara manual.

d) *Integrated Terminal*

Visual Studio Code terdapat *terminal* tersendiri yang telah disediakan, sehingga *programmer* tidak perlu membuka *terminal* lain dan memudahkan pekerjaan serta menghemat waktu pekerjaan. Terdapat juga pilihan menambah atau berganti terminal untuk menjalankan program yang di tulis di *Visual Studio Code*.

e) *Command Palette*

Command palette adalah seperti *Command line* dari *Visual Studio Code*, di sini pengguna dapat memberikan perintah kepada *editor* dengan mengetik perintah seperti menjalankan program dengan *terminal*, mencari *file*, *disable extension* dan lain lain.

Kekurangan *Visual Studio Code* :

Kekurangan *Visual Studio Code* adalah terletak pada performa jika dibandingkan dengan *text editor* lain.

2.2.5. TypeScript

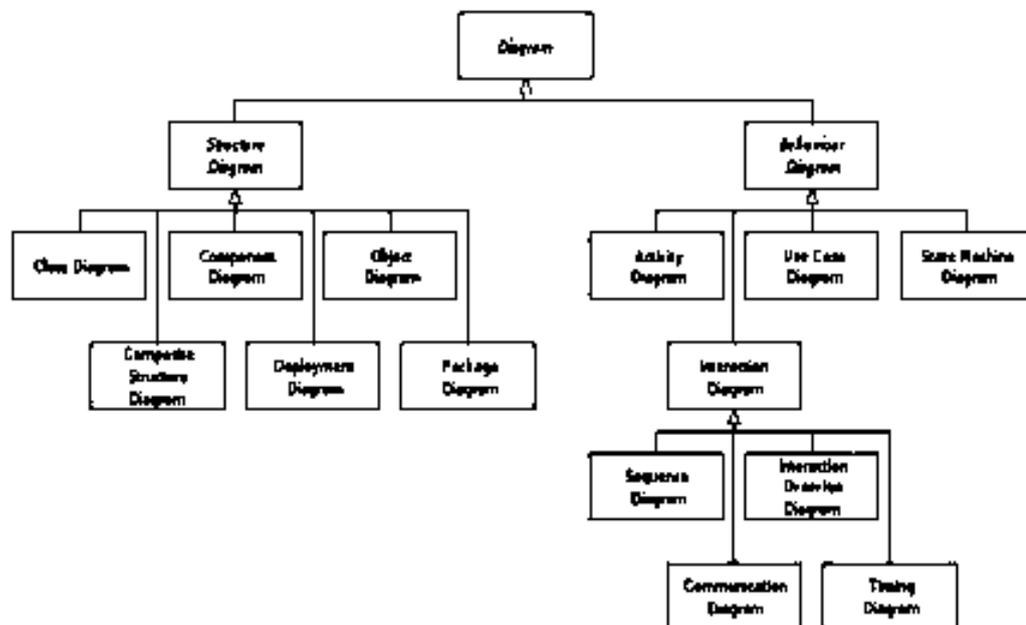
Typescript adalah bahasa pemrograman berbasis *JavaScript* yang menambahkan fitur strong-typing & konsep pemrograman OOP klasik (*class*, *interface*). Di dalam dokumentasinya, *TypeScript* disebut sebagai super-set dari *JavaScript*, artinya semua kode *JavaScript* adalah kode *TypeScript* juga. Bahasa

pemrograman ini menawarkan *class*, *module*, dan *interface* yang membuat *developer* bisa mengembangkan aplikasi kompleks dengan lebih mudah. Hal inilah yang membedakannya dengan *javascript* [5].

2.2.6. Unified Modeling Language (UML)

Unified modeling language (UML) adalah suatu alat bantu dalam pengembangan sistem berorientasi objek. UML menyediakan bahasa pemodelan visual yang berguna bagi pengembang dalam membuat suatu cetak biru atas visi mereka dalam bentuk baku, dan dapat mengkomunikasikan rancangan pengembang dengan pengembang yang lain. UML telah dijadikan suatu standar baku dalam Object Oriented Analysis & Design (OOAD). Karakter penting yang melekat dalam UML adalah sketsa, cetak biru, dan bahasa pemrograman [6].

Diagram UML dikelompokkan menjadi 2 yaitu Diagram Struktur atau Statis Diagram dan Diagram Perilaku Sistem atau Behaviour Diagram.



Gambar 2.2.6.1 Diagram UML

UML terdiri dari bermacam-macam diagram yang digunakan untuk pemodelan dalam pengembangan sistem mulai dari tahap analisi sampai implementasi. Pada saat melakukan desain sistem tidak harus semua diagram pada UML diimplementasikan akan tetapi perlu adanya kekonsistenan rancangan diagram

yang satu dengan lainnya.

1. Diagram Use Case (Use Case Diagram)

Use case diagram merupakan suatu pemodelan yang menitik beratkan pada fungsionalitas utama dalam suatu konteks sistem. Use case diagram juga menekankan tingkah laku fungsional utama dalam sistem berinteraksi dengan objek diluar sistem tersebut. Selain itu, use case diagram juga telah menitik beratkan jenis hubungan diantara fungsi utama.

2. *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 suatu sistem dapat berinteraksi baik antar sistem maupun diluar sistem.

3. Diagram Kelas (*Class Diagram*)

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. Class diagram menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Kelas pada kelas diagram terdiri dari 3 bagian utama yaitu nama kelas, isi properti dari kelas beserta metode yang ada pada kelas tersebut. Kelas juga memiliki jenis-jenis hubungan seperti asosiatif, dependensi, agregasi, komposisi, spesifikasi dan generalisasi. Hubungan ini digunakan untuk menggambarkan bagaimana hubungan dan interaksi yang terjadi antar kelas. Masing-masing komponen penyusun kelas memiliki hak akses seperti public, private, dan protected.

4. Diagram Skuensial (*Diagram Sequence*)

Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case: interaksi yang terjadi antar class, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

5. Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas adalah diagram flowchart yang diperluas yang menunjukkan aliran kendali satu aktivitas ke aktivitas lain di sistem. Diagram aktivitas ini digunakan untuk memodelkan aspek dinamis sistem. Diagram aktivitas mendeskripsikan aksi-aksi dan hasilnya [6].

2.2.7. Use Case Diagram

Use case merupakan deskripsi fungsionalitas sistem dari perspektif pengguna. *Use case* mendeskripsikan interaksi antara pengguna sistem dengan sistem itu sendiri. Pengguna dalam *use case* biasa disebut dengan aktor, *actor* merupakan peran yang dapat digunakan oleh pengguna dalam suatu sistem.

Menurut Jacobson tahun 1992 model use case merupakan bagian dari model requirement. Tujuan dari use case diagram diantaranya :

1. Digunakan dalam mengumpulkan suatu kebutuhan dalam sebuah sistem.
2. Digunakan untuk pandangan dari luar sistem.
3. Digunakan dalam mengidentifikasi faktor baik didalam maupun diluarnya yang mempengaruhi sistem.
4. Digunakan untuk menunjukkan interaksi sistem dan pengguna

Berikut use case diagram dapat dilihat pada Gambar 2.5 Use Case Diagram [6].



Gambar 2.2.7.1 Use Case

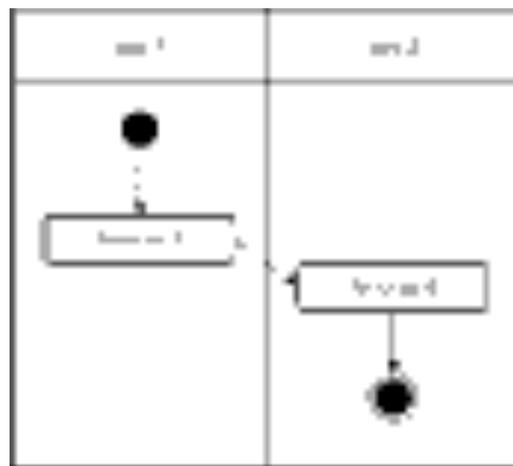
2.2.8. Use Case Scenario

Use case scenario merupakan urutan dari langkah-langkah yang menerangkan pengguna dengan *System*. Setiap scenario akan mendeskripsikan urutan dari kejadian/peristiwa. *Use case scenario* dapat dikatakan sebuah rincian informasi dari sebuah *use case* [6].

2.2.9. Activity Diagram

Activity diagram merupakan penggambaran aspek dinamis dalam sistem,

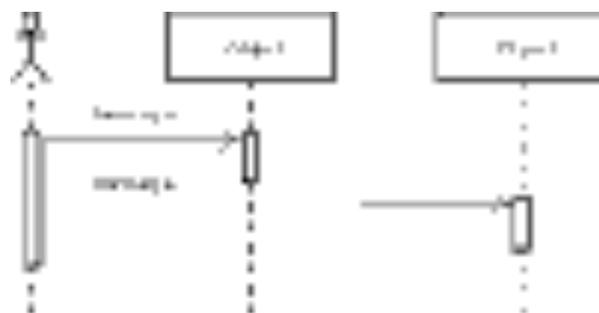
activity diagram dapat dengan mudah mendeskripsikan aliran kerja dan proses bisnis dalam suatu bisnis. Tujuan pemakaian *activity diagram* adalah penggambaran aliran aktivitas dari suatu sistem, penggambaran urutan aktifitas, dan penggambaran paralelisme, percabangan serta aliran konkuren dari suatu sistem. Berikut adalah contoh dari *activity diagram* dapat dilihat pada Gambar 2.6 *Activity Diagram* [6].



Gambar 2.2.9.1 Activity Diagram

2.2.10. Sequence Diagram

Sequence diagram digunakan dalam menggambarkan suatu perilaku pada sebuah scenario. Tujuan digunakannya *sequence diagram* adalah menunjukkan urutan waktu aliran pesan antar objek. *Sequence diagram* dapat diartikan model inteksi tingkat tinggi antar objek aktif dalam System atau model interaksi antara instance objek dalam kolaborasi yang merealisasikan suatu *use case*. Berikut contoh dari *sequence diagram* dapat dilihat pada Gambar 2.7 *Sequence Diagram* [6].



Gambar 2.2.10.1 Sequence Diagram

2.2.11. Class Diagram

Class diagram merupakan diagram statis yang mewakili pandangan statis dari suatu aplikasi. Class diagram dapat digunakan dalam memvisualisasikan, menggambarkan, mendokumentasikan berbagai aspek sistem dan juga untuk membangun kode eksekusi dalam suatu aplikasi yang akan dibangun. Class diagram dapat menggambarkan atribut, operation, dan constraint dalam suatu sistem, class diagram merupakan diagram UML yang dapat langsung dipetakan dengan bahasa perorientasi objek. Penggunaan class diagram memiliki tujuan analisis dan desain pandangan statis suatu aplikasi, menerangkan tanggung jawab sistem, forward and reverse engineering [6].

2.2.12. Entity Relationship Model

Entity Relationship Model merupakan sebuah konsep didasarkan pada persepsi dunia nyata yang terdiri dari kumpulan objek dasar yang disebut entitas dan hubungannya antar objek yang disebut dengan relationship. Pada *Entity Relationship Model*, data yang ada dapat ditransformasikan kedalam perangkat konseptual dengan bantuan diagram yaitu ERD (*Entity Relationship Diagram*).

ERD menggambarkan arti dari aspek data seperti bagaimana entity-entity, atribut-atribut, dan relationship tersebut disajikan. Sebelum membuat ERD, tentunya ada beberapa aspek yang harus dipahami yaitu data yang diperlukan dan juga ruang lingkungannya. Dalam membuat ERD ada yang perlu diperhatikan dalam menentukan sesuatu konsep apakah merupakan suatu entity, atribut atau relationship [7].

Terdapat beberapa notasi yang harus diperhatikan dalam membuat sebuah ERD. Berikut ini merupakan beberapa notasi yang ada pada saat pembuatan ERD:

1. Entitas (*Entity*)

Entitas merupakan sebuah objek yang dapat dibedakan dengan yang lain dalam dunia nyata. Entity dapat berupa objek secara fisik seperti orang, rumah, atau kendaraan. Entity dapat juga berupa objek secara konsep, seperti pekerjaan, perusahaan, dan sebagainya. Entitas digambarkan dengan bentuk persegi panjang.

2. Atribut

Atribut merupakan karakteristik yang terdapat dalam entitas atau relationship, yang menyediakan penjelasan yang mendetail mengenai entitas atau relationship tersebut. Atribut digambarkan dengan bentuk oval.

3. Relationship

Relationship merupakan hubungan yang terjadi antara satu atau lebih entitas. *Relationship* digambarkan dengan bentuk belah ketupat.

4. Cardinality (Kardinalitas)

Kardinalitas menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas lainnya. Terdapat tiga jenis kardinalitas relasi yaitu satu ke satu (1:1), satu ke banyak (1:N / N:1) dan banyak ke banyak (N:N).

a) Satu ke Satu (1:1)

Misalkan terdapat entitas A dan B. Setiap entitas pada himpunan entitas A dapat berelasi dengan paling banyak satu entitas pada himpunan entitas B, demikian juga sebaliknya.

b) Satu ke Banyak atau Banyak ke Satu (1:N / N:1)

Misalkan terdapat entitas A dan B. Setiap entitas pada himpunan A dapat berelasi dengan banyak entitas pada himpunan B, tetapi tidak sebaliknya atau banyak entitas pada himpunan A dapat berelasi dengan paling banyak satu entitas pada entitas B, tetapi tidak sebaliknya.

c) Banyak ke Banyak (N:N)

Setiap entitas pada himpunan entitas A dapat berelasi dengan banyak entitas pada himpunan entitas B, demikian juga sebaliknya.

5. Entitas Lemah (*Weak Entity*)

Entitas lemah adalah suatu entitas dimana keberadaannya tergantung dari keberadaan entitas lainnya. Entitas Lemah digambarkan dengan bentuk Persegi panjang yang didalamnya terdapat persegi panjang lainnya.

2.2.13. Model Pengujian Sistem

Metode pengujian sistem merupakan sebuah proses untuk mengetahui efektifitas dari aplikasi yang dipakai selain memberikan kesempatan kepada pengguna untuk menggunakan dan juga melakukan pengecekan terhadap laporan yang dihasilkan melalui software.

2.2.13.1. Pengujian Black Box

Pengujian Black Box merupakan pengujian yang berfokus pada apakah unit program yang dibuat sudah memenuhi kebutuhan yang sudah disebutkan dalam spesifikasi. Pada pengujian black box, cara pengujian dilakukan dengan mengeksekusi unit atau modul, setelah itu diamati untuk mengetahui apakah unit itu sesuai keluarannya. Proses yang terdapat pada pengujian black box adalah analisis batasan nilai yang berlaku pada setiap data [8].

2.2.14. Business Process Model and Notation (BPMN)

BPMN adalah Bahasa simbol yang digunakan sebagai standar pemodelan proses bisnis. BPMN merupakan bahasa yang diciptakan agar mudah dipahami untuk menggambarkan proses bisnis. Pada BPMN terdapat definisi dan karakterisasi tertentu yang membuatnya mudah untuk memahami proses yang dibuat tanpa dijelaskan secara rinci kepada seseorang baik yang ada di dalam maupun luar perusahaan. BPMN terdiri dari berbagai simbol dan elemen yang merupakan gambaran urutan dari suatu kegiatan proses bisnis [9]. Dalam BPMN terdapat elemen-elemen yang dikategorikan kedalam lima kategori dasar yaitu:

1. *Flow Object*

Flow object adalah elemen utama yang menggambarkan perilaku dari proses. Pada flow object terdapat tiga elemen yaitu Event, Activity, dan Gateway.

2. *Connecting Object*

Connecting Object merupakan garis yang menghubungkan flow objek ke flow objek lainnya. Terdapat tiga elemen yang terdapat pada connecting object yaitu *Sequence Flow*, *Message Flow*, dan *Association*.

3. *Swimlanes*

Swimlanes merupakan komponen yang menggambarkan siapa saja aktor yang terlibat dalam suatu aktivitas dalam proses bisnis. *Swimlanes* terdiri dari dua elemen yaitu *Pool* dan *Lane*.

4. *Artifacts*

Pada *artifacts* terdapat tiga elemen yaitu *Data Object*, *Group*, *Text Annotation*.

2.2.15. Skala Likert

Skala *Likert* adalah merupakan sebuah skala psikometrik (Pengukuran Psikologis) yang sering digunakan dalam angket dan merupakan skala yang paling banyak digunakan untuk keperluan riset yang berupa survei. Istilah skala ini diambil dari nama Rensis Likert, yaitu seseorang yang menerbitkan suatu laporan yang menjelaskan penggunaannya. Ketika menanggapi pernyataan dalam skala *Likert*, responden menentukan tingkat persetujuan terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang ada. Biasanya tingkat persetujuan disediakan sebanyak lima pilihan dengan format sebagai berikut [10].

1. Sangat Setuju
2. Setuju
3. Ragu-Ragu
4. Tidak Setuju
5. Sangat Tidak Setuju

Untuk keperluan analisis kuantitatif, maka untuk setiap jawaban dari responden dapat diberi skor, misalnya:

- | | |
|------------------------|---|
| a. Sangat Setuju | 5 |
| b. Setuju | 4 |
| c. Ragu-Ragu | 3 |
| d. Tidak Setuju | 2 |
| e. Sangat Tidak Setuju | 1 |

Instrumen penelitian dengan menggunakan skala *likert* ini dapat dibuat dalam bentuk *checklist* atau juga bisa dalam bentuk pilihan ganda.