

## **BAB 2**

### **TINJAUAN PUSTAKA**

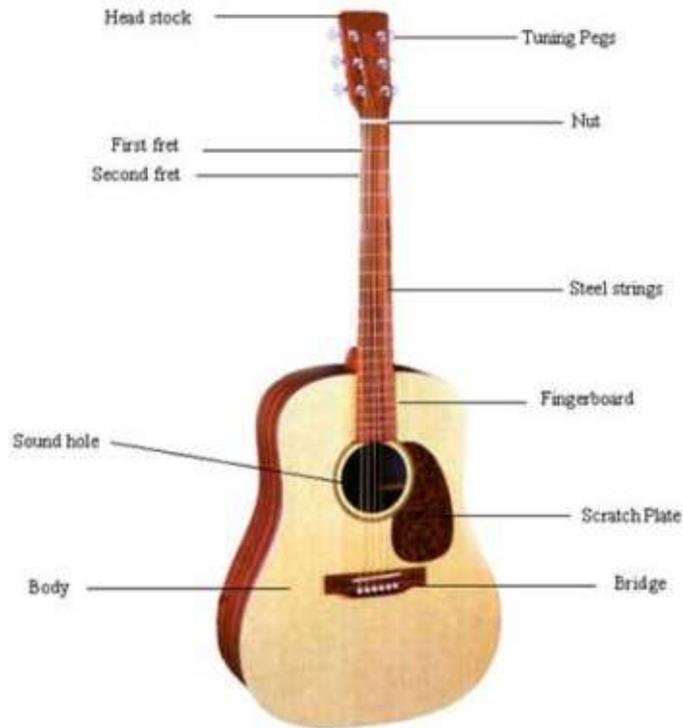
#### **2.1. Pengenalan Alat Musik Gitar**

Gitar merupakan instrumen musik yang dimainkan dengan cara dipetik. Awal mulanya diambil dari alat musik petik kuno di wilayah Persia pada tahun sekitar 1500 SM yang disebut dengan *Citar* atau *Sehtar* [1].

Pada alat musik petik yaitu gitar, senar merupakan suatu bagian sumber getar yang jenis dan ukurannya diproduksi secara tetap, sehingga perlu dikaji secara fisis maupun matematis mengenai besaran-besaran yang berhubungan dengan sumber bunyi tersebut. Selain itu terdapat *fret-fret* yang dipasang di sepanjang leher gitar yang digunakan untuk menghasilkan nada-nada tertentu, caranya dengan menekan senar diantara dua fret kemudian senar dipetik. Tinggi rendahnya nada bergantung dari panjang senar yang diperoleh saat senar ditekan diukur dari ganjal senar bawah. Semakin pendek ukuran senar maka nada yang diperoleh semakin tinggi, begitu pula sebaliknya [2].

##### **2.1.1. Bagian-Bagian Gitar**

Pada gitar akustik umumnya terdapat 6 senar. Namun ada beberapa gitar yang menggunakan lebih dari 6 senar. Ada juga yang kurang dari 6. Senar yang biasa digunakan yaitu senar yang terbuat dari baja (*string*) dan juga nilon. Gitar yang menggunakan senar nilon biasanya disebut dengan gitar klasik. Pada gitar klasik, bagian neck dan *fretboard*nya lebih lebar ketimbang gitar yang menggunakan senar jenis *string* [12].



*Gambar 1.1 Bagian-bagian Gitar*

Berikut penjelasan mengenai fungsi tiap bagian pada gitar akustik seperti pada gambar diatas :

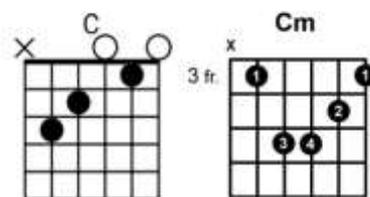
1. *Tuning Keys/ Tuning Machines* : berfungsi untuk mengencangkan/ mengendurkan senar gitar.
2. *Headstock* : berfungsi untuk mengatur penempatan senar agar tetap konsisten pada tempatnya.
3. *Nut* : berfungsi untuk mengatur penempatan senar agar tetap konsisten pada tempatnya.
4. *Fingerboard/ fretboard* : papan panjang tempat fret yang membagi wilayah nada.
5. *Fret* : logam melintang pada sepanjang *fingerboard* untuk membagi wilayah nada.
6. *Strap pin/ End pin/ Strap Button* : *pin* untuk memasukkan *strap* gitar.
7. *String* : senar gitar, biasa menggunakan nilon maupun baja atau nikel.
8. *Body* : badan gitar

9. *Sound Hole* : penghasil nada, berfungsi mengeluarkan suara getaran senar dan sebagai tempat sirkulasi udara didalam *body* gitar hingga daya akustiknya bagus.
10. *Position marker* : untuk menandai pembagian wilayah nada. Bisa berupa bulatan atau kadang ada yang di *custom* gambar , dll.
11. *Heel* : untuk menyangga/ menahan *neck* agar tidak bengkok/ melengkung.
12. *Bridge* : penahan senar ke badan gitar.
13. *Bridge pin* : untuk menahan senar/ *string* yang dipasang.
14. *Saddle* : seperti *nut*, tapi letaknya di *bridge* gitar.
15. *Rosette* : garis yang melingkar di tepi-tepi *sound hole*.

### 2.1.2. Akord Gitar

Kord adalah beberapa nadayang dibunyikan secara bersamaan. kord berfungsi sebagai pengiring dalam sebuah lagu atau sebagai penghias *fill -fill* (Penambahan nitasi-notasi) sebuah komposisi lagu. Setiap nada pembentuk akor dapat mengalami perubahan naik turun nada dengan tanda tertentu. Tanda yang digunakan untuk menaikkan nada seharga  $\frac{1}{2}$  laras, yaitu tanda kres ( $\sharp$ ) dan tanda yang digunakan untuk  $\frac{1}{2}$  laras, yaitu tanda mol( $\flat$ ) [3].

Secara mendasar, akord/ *chord* gitar terdiri dari dua macam yaitu Mayor dan Minor. *Chord* dasar terdiri dari 3 nada yang merupakan bagian dari tangga nada dasarnya. Untuk dapat memainkan *chord* mayor maka kita menyusunnya dari nada 1 – 3 – 5 dari tangga nada mayornya. Untuk menyusun *chord* minor maka kita akan menyusunnya dari nada 1 –  $3^{\flat}$  – 5 dari tangga nada [1].



Gambar 2.2 Contoh Akord Mayor dan Minor

## 2.2. Landasan Teori

Teori merupakan salah satu unsur terpenting dalam penelitian yang memiliki peran sangat besar dalam penelitian. Suatu landasan teori dari suatu penelitian disini bisa disimpulkan sebagai studi literatur atau tinjauan pustaka. Hasil dari landasan teori atau kajian teori ini diperoleh kesimpulan-kesimpulan atau pendapat-pendapat para ahli, kemudian dirumuskan pada pendapat baru.

### 2.2.1. Multimedia

Seperti namanya, multimedia adalah integrasi berbagai bentuk media seperti teks, grafik, audio, video, dll. Multimedia sering digunakan dalam dunia informatika ataupun selain dunia informatika seperti dalam dunia game dan juga dalam pembuatan website.

Multimedia dapat masuk dan menjadi alat bantu yang menyenangkan. Hal ini terjadi karena kekayaan elemen-elemen dan kemudahannya digunakan dalam banyak konten yang bervariasi. Beberapa bidang yang menggunakan multimedia diantaranya rumah, bisnis, sekolah dan tempat umum [13].

#### 2.2.1.1. Jenis-jenis multimedia

Berdasarkan teknik pengoprasianya multimedia dibagi menjadi beberapa jenis. Menurut buku berjudul *Multimedia Digital* (Iwan Binanto.2010:3) multimedia terbagi atas 3 jenis:

1. Multimedia Interaktif Pengguna dapat mengontrol apa dan kapan elemen-elemen multimedia akan dikirimkan atau ditampilkan.
2. Multimedia Hiperaktif Multimedia jenis ini mempunyai suatu struktur dari elemen-elemen terkait dengan pengguna yang dapat mengarahkannya. Dapat dikatakan bahwa multimedia jenis ini mempunyai banyak tautan atau link yang menghubungkan elemen-elemen multimedia yang ada.
3. Multimedia Linear Pengguna hanya menjadi penonton dan menikmati produk multimedia yang disajikan dari awal hingga akhir.

### 2.2.2. *Augmented Reality*

*Augmented reality* merupakan teknologi yang menggabungkan benda-benda nyata dan maya pada lingkungan nyata dan dapat berjalan secara interaktif dalam waktu nyata [4]. AR merupakan salah satu jenis *mixed reality* (MR) yang

mencakup *virtual reality* (VR) yaitu istilah yang digunakan pada teknik grafis komputer untuk lingkungan yang dibuat 3D, kemudian *telepresence* sebagai sistem yang membuat pengguna memiliki informasi mengenai lingkungan *remote site* dan diberikan tampilan yang sesuai, sehingga menjadikan pengguna seolah-olah berada dalam *remote site*, dan yang terakhir adalah AR sendiri merupakan teknologi yang berada diantara VR dan *telepresence* yang membuat pengguna bisa melihat dunia nyata dan *virtual* secara bersamaan [5].

AR memiliki tiga karakteristik yang menjadi dasar diantaranya adalah kombinasi pada dunia nyata dan *virtual*, kemudian interaksi yang berjalan secara *real-time* dan yang terakhir adalah bentuk objek yang berupa model 3 dimensi atau yang biasa disebut 3D dimana semuanya mencakup data lokasi, audio, video, ataupun dalam bentuk 3D [6].

#### **2.2.2.1. Metode Augmented Reality**

Ada beberapa metode yang digunakan pada *Augmented Reality* berdasarkan ada atau tidak adanya penanda (*Marker*). Berikut penjelasan metode yang digunakan dalam menangkap atau mengenali tanda yang dijadikan acuan tersebut :

##### *1. Marker Based Tracking*

*Marker* biasanya merupakan ilustrasi hitam dan putih persegi dengan batas hitam tebal dan latar belakang putih. Komputer akan mengenali posisi dan orientasi *marker* dan menciptakan dunia *virtual* 3D yaitu titik (0,0,0) dan 3 sumbu yaitu X,Y,dan Z. *Marker Based Tracking* ini sudah lama dikembangkan sejak 1980-an dan pada awal 1990-an mulai dikembangkan untuk penggunaan *Augmented Reality*. [7]

*Marker based AR* adalah memberikan *tracking* yang cepat dan menghemat komputasi. Pada metode ini mengharuskan adanya *marker* yang dicetak baik berupa foto sebuah objek nyata maupun gambar buatan serta memiliki pola yang unik [5]. Sebuah *marker* yang baik adalah *marker* yang mudah dikenali dalam kondisi apapun, misalnya ketika kondisi cahaya kurang dan posisi kamera yang berpindah-pindah, tetapi tidak mempengaruhi tingkat keakuratan dan sensitivitas dari *marker*, sehingga akan tetap terdeteksi oleh sistem AR [8].

##### *2. Markerless Augmented Reality*

Salah satu metode *Augmented Reality* yang saat ini sedang berkembang adalah metode "*Markerless Augmented Reality*", dengan metode ini pengguna tidak perlu lagi menggunakan sebuah *marker* untuk menampilkan elemen-elemen digital. Sekalipun dinamakan dengan *markerless* namun aplikasi tetap berjalan dengan melakukan pemindaian terhadap object, namun ruang lingkup yang dipindai lebih luas dibanding dengan *marker AR*

### 2.2.3. Unity

Unity 3D merupakan *game engine*, yaitu *software* yang digunakan untuk memudahkan dalam membuat *game* [9]. Kelebihan *Unity* dibandingkan dengan *game engine* lainnya adalah kemampuan membuat *game cross platform*. Dengan *Unity 3D*, *game* yang dibuat dapat dimainkan di berbagai perangkat seperti *smartphone* dan *game console*. *Unity* sendiri dapat membuat berbagai *game*, seperti *RPG (Role Playing Game)*, *shooter*, *racing* dan lain sebagainya.

Untuk membangun sebuah karya pada *Unity 3D*, disediakan berbagai pilihan bahasa pemrograman untuk mengembangkan *game*, antara lain *Jaca Script*, *C#*, dan *Boo* [9].

### 2.2.4. Vuforia SDK

*Vuforia* merupakan *software development kit* yang dikembangkan oleh Qualcomm untuk membantu pengembang dalam menciptakan aplikasi atau *game* yang menggunakan teknologi *Augmented Reality*, sehingga membuat aplikasi atau *game* yang dibuat dengan teknologi ini akan terlihat lebih interaktif dan hidup. Pengembang dapat membuat objek *virtual 3D* yang dapat berinteraksi dengan pengguna aplikasi baik *game*, aplikasi pembelajaran, video, aplikasi dongeng, dan masih banyak lagi. Terdapat berbagai fitur menarik pada *Vuforia SDK* ini, seperti memindai objek, memindai teks, mengenali bingkai penanda, tombol *virtual*, mengidentifikasi permukaan objek secara pintar, memindai dengan berbasis *cloud*, mengenali target gambar, mengenali target benda silinder, dan mengenali objek target yang telah ditetapkan [10].

Dengan adanya *Vuforia SDK* ini akan memudahkan dan mempercepat pengembangnya dalam membuat aplikasi yang mempunyai teknologi *Augmented Reality* karena *library* dan fungsi-fungsi intinya sudah dibuat oleh Qualcomm

sehingga pengembang tinggal berimajinasi dan mengembangkan aplikasi menarik menggunakan SDK ini [10].

#### **2.2.5. Blender**

Blender merupakan *software* pembuat objek 3D yang mampu untuk membuat model dan animasi. *Blender* dapat diperoleh secara gratis di situs resminya. *Blender* juga tersedia untuk sistem operasi 32 bit dan 64 bit baik untuk Windows, Linux, Free BSD, dan Mac OSX. Akan tetapi, saat ini *Blender* sudah tidak mendukung untuk sistem operasi Windows XP. *Blender* sendiri merupakan *software* 3 dimensi yang ringan dengan ukuran file yang kurang dari 100 MB. Oleh karena itu, untuk menggunakan Blender tidak harus menggunakan spesifikasi komputer yang terlalu tinggi. *Blender* dapat digunakan dengan baik bahkan hanya dengan komputer dengan *prosesor* dual core [9].

#### **2.2.6. Adobe Photoshop CS 6**

*Photoshop* adalah sebuah program penyunting gambar standar industri yang ditujukan untuk para profesional *raster* grafik. *Photoshop* memiliki kemampuan untuk membaca dan menulis gambar berformat *raster* dan *vector* seperti .png, .gif, .jpeg, dan lain-lain. *Photoshop* juga memiliki beberapa format file khusus seperti .psd, .psb dan .pdd [11].

#### **2.2.7. Android**

*Android* adalah sistem operasi berbasis *linux* yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer *tablet*. *Android* awalnya dikembangkan oleh android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya tahun 2005.

*Android* tidak membedakan antara aplikasi inti ponsel dan aplikasi pihak ketiga. Mereka semua dapat dibangun untuk memiliki akses yang sama ke kemampuan telepon yang menyediakan pengguna dengan spektrum aplikasi dan layanan yang luas. Dengan perangkat yang dibangun di *Platform* Android, pengguna dapat sepenuhnya menyesuaikan telepon dengan minat mereka. Mereka dapat menukar homescreen ponsel, gaya dialer, atau aplikasi apa pun. Mereka bahkan dapat menginstruksikan ponsel mereka untuk menggunakan aplikasi melihat foto favorit mereka untuk menangani tampilan semua foto [14].

### 2.2.7.1. Arsitektur Android

Arsitektur pada sistem *Android* memiliki 5 tingkat lapisan *layer*. Setiap lapisan *layer* tersebut menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Pemisahan *layer* tersebut bertujuan untuk memberikan abstraksi sehingga memudahkan pengembang aplikasi.

Secara garis besar Arsitektur Android dapat dijelaskan sebagai berikut:

1. *Application* dan *Widgets*. *Application* dan *Widgets* ini adalah *layer* dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut.
2. *Application Framework Android* adalah “*Open Development Platform*” yaitu *Android* menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. *Application Frameworks* ini adalah *layer* dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi *Android*, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content providers* yang berupa sms dan panggilan telepon.
3. *Libraries, Libraries* ini adalah *layer* dimana fitur-fitur *Android* berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya.
4. *Android Run Time Layer* yang membuat aplikasi *Android* dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi *Android*.
5. *Linux Kernel*. *Linux Kernel* adalah *layer* dimana inti dari sistem operasi *Android* itu berada. Berisi file-file sistem yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi *Android* lainnya. *Linux Kernel* yang digunakan *Android* adalah *Linux Kernel release 2.6*.

### 2.2.8. Pemrograman Berbasis Objek

Pemrograman berorientasi objek adalah sebuah konsep pemrograman untuk membuat kode program yang lebih terstruktur, terkelompok berdasarkan objek-

objek yang terlibat sehingga bagian-bagiannya dapat digunakan untuk membuat aplikasi lain. Pemrograman berorientasi objek membagi-bagi kode program aplikasi menjadi kumpulan bungkusan benda/ objek dipandang dari sudut pandang aplikasi komputer dan proses yang ada didalam aplikasi [15].

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. Konsep pemrograman 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.

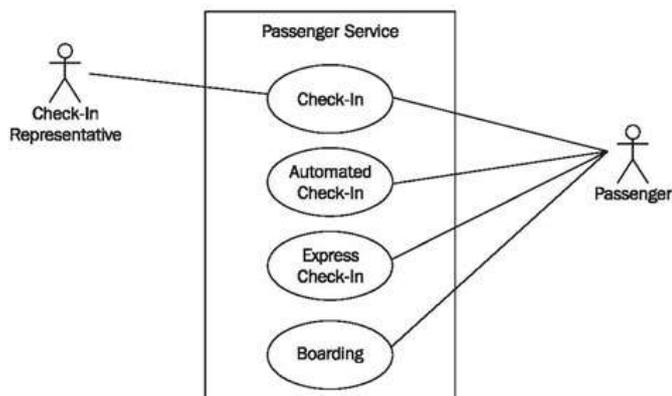
### **2.2.9. Unified Modelling Language (UML)**

UML adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena developer harus melakukan penelusuran dan mempelajari kode program.

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 standarisasi teknologi objek agar UML dijadikan bahasa dan notasi pemodelan dilakukan pada tahun 1997. OMG menerima UML, UML telah menjadi standar de-facto karena penciptapenciptanya sangat populer. Banyak pengembang perangkat lunak yang mengadopsi UML [16].

#### **2.2.9.1. Use Case Diagram**

*Use case diagram* digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case* diagram terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case*.

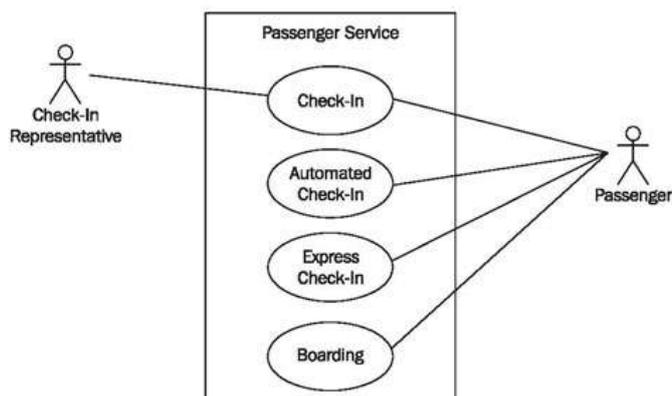


Gambar 2.3 contoh Use Case Diagram

### 2.2.9.2. Activity Diagram

Activity diagram menggambarkan aliran fungsionalitas sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis (*business work flow*). Dapat juga digunakan untuk menggambarkan aliran kejadian (*flow of event*) dalam .

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.

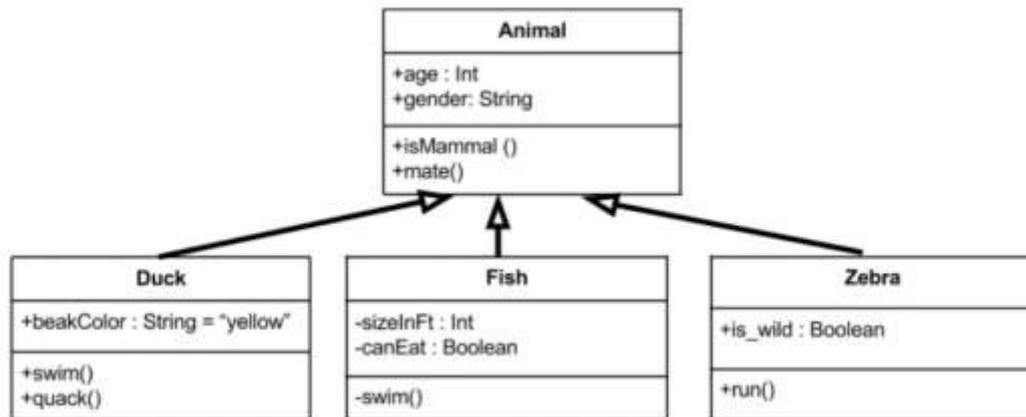


Gambar 2.4 contoh Activity Diagram

### 2.2.9.3. Class Diagram

Class diagram menggambarkan struktur statis *class* di dalam sistem. *class* merepresentasikan sesuatu yang ditangani oleh sistem. *class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain),

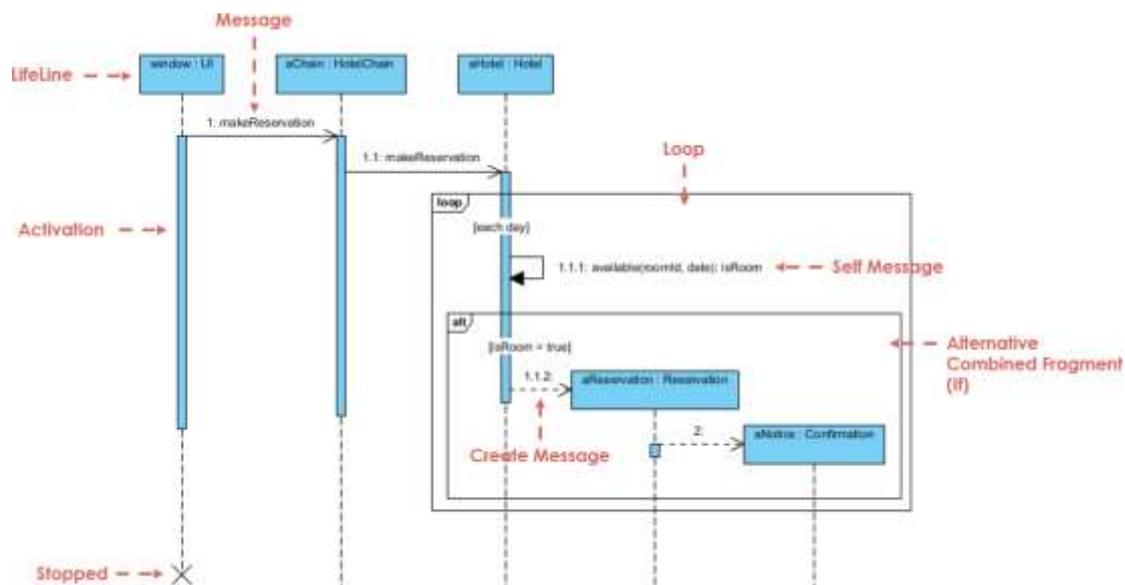
*dependent* (satu class tergantung/menggunakan class yang lain), *specialized* (satu class merupakan spesialisasi dari class lainnya), atau package (group bersama sebagai satu unit). sebuah sistem biasanya mempunyai beberapa class diagram.



Gambar 2.5 contoh Class Diagram

#### 2.2.9.4. Sequence Diagram

*Sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. *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 2.6 contoh Sequence Diagram

### 2.2.10. Data dan Informasi

Hubungan antara data dan informasi sangatlah erat sebagaimana hubungan antara sebab dan akibat. Bahwa data merupakan bentuk dasar dari sebuah informasi, sedangkan informasi merupakan elemen yang dihasilkan dari suatu bentuk pengolahan data.

#### 2.2.10.1. Data

Secara konseptual, data adalah deskripsi tentang benda, kejadian, aktivitas dan transaksi, yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai. Data sering kali disebut sebagai bahan mentah informasi. Berikut adalah kutipan pengertian data dari sudut pandang yang berbeda :

1. Menurut kamus bahasa Inggris-Indonesia, data diterjemahkan sebagai istilah yang berasal dari kata “*datum*” yang berarti fakta atau bahan-bahan keterangan.
2. Dari sudut pandang bisnis, terdapat pengertian data bisnis sebagai berikut: “*Bussines data is an organization’s description of things (resources) and events (transactions) that it faces*”. Jadi data, dalam hal ini disebut sebagai data bisnis, merupakan deskripsi organisasi tentang sesuatu (*resource*) dan kejadian (*transaction*) yang terjadi.

3. Menurut (Jogiyanto, HM, 1989 : 8) data merupakan bentuk jamak dari datum (kenyataan) yang berupa fakta-fakta, angka-angka, gambar-gambar yang dapat ditarik kesimpulannya.
4. Gordon B. Davis dalam bukunya *Management Informations System : Conceptual Foundations, Structure, and Development* menyebut data sebagai bahan mentah dari informasi, yang dirumuskan sebagai sekelompok lambang-lambang tidak acak yang menunjukkan jumlah atau tindakan atau hal-hal lain.

Dari keempat pengertian diatas, dapat diambil kesimpulan bahwa data adalah bahan baku informasi, didefinisikan sebagai kelompok teratur simbol-simbol yang mewakili kuantitas, tindakan, benda, dan sebagainya. Data terbentuk dari karakter, dapat berupa alphabet, angka, maupun simbol khusus seperti \*, \$, dan /. Data disusun untuk diolah dalam bentuk struktur data, struktur file, dan basis data.[17]

#### **2.2.10.2. Skala Likert**

Skala data yang digunakan untuk pengukuran variabel independen adalah skala likert. Skala likert adalah suatu skala *psikometrik* yang digunakan dalam kuesioner dan merupakan salah satu teknik yang dapat digunakan dalam evaluasi suatu program atau kebijakan perencanaan. Skala Likert ini merupakan skala yang dapat dipergunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang mengenai suatu gejala atau fenomena. Dan pada evaluasi, skala likert digunakan untuk menilai keberhasilan suatu kebijakan atau program, menilai manfaat pelaksanaan suatu kebijakan atau program, dan mengetahui kepuasan stakeholder terhadap pelaksanaan suatu kebijakan atau program.

Rensis Likert telah mengembangkan sebuah skala untuk mengukur sikap masyarakat di tahun 1932 yang sekarang terkenal dengan nama skala Likert. Skala likert hanya menggunakan item yang secara pasti baik dan secara pasti buruk. Skala likert menggunakan ukuran ordinal, karenanya hanya dapat membuat rangking, tetapi tidak dapat diketahui berapa kali satu responden lebih baik atau lebih buruk dari responden lainnya dalam skala [18].

### 2.2.10.3. Prosedur Pembuatan Skala Likert

Prosedur dalam pembuatan skala likert adalah sebagai berikut.

1. Peneliti mengumpulkan item-item yang relevan dengan masalah yang sedang diteliti, dan terdiri atas item yang cukup jelas disukai dan tidak disukai.
2. Kemudian item-item tersebut dicoba kepada sekelompok responden yang cukup *representative* dari populasi yang diteliti.
3. Responden di atas diminta untuk mengecek tiap item, apakah ia menyenangkan atau tidak menyukainya. Responsi tersebut dikumpulkan dan jawaban yang memberikan indikasi menyenangkan diberikan skor tertinggi. Tidak ada masalah untuk memberikan angka 5 untuk yang tertinggi dan skor 1 untuk yang terendah atau sebaliknya. Yang penting adalah konsistensi dari arah sikap yang diperlihatkan. Demikian juga, apakah jawaban “setuju” atau “tidak setuju” disebut yang disenangi, tergantung dari isi pertanyaan dan isi dari item-item yang disusun.
4. Total skor dari masing-masing individu adalah penjumlahan dari skor masing-masing item dari individu tersebut.
5. Respons dianalisis untuk mengetahui item-item mana yang sangat nyata batasan antara skor tinggi dan skor rendah dalam skala total. Misalnya, respons responden pada upper 25% dan lower 25% dianalisis untuk melihat sampai berapa jauh tiap item dalam kelompok ini berbeda. Item-item yang tidak menunjukkan beda yang nyata, apakah masuk ke dalam skor tinggi atau rendah juga dibuang untuk mempertahankan konsistensi internal dari pernyataan.

### 2.2.11. Metode Pengujian Sistem

Metode pengujian sistem untuk mengetahui efektifitas dari perangkat lunak (*software*) yang digunakan selain memberikan kesempatan kepada pengguna untuk mengoperasikan dan melakukan pengecekan terhadap laporan yang dihasilkan melalui *software*. Metode pengujian sistem terdiri dari pengujian *black-box*.

#### **2.2.11.1. Pengujian *Black-Box***

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 [19].