

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Kesehatan**

Kesehatan adalah keadaan seimbang yang dinamis, dipengaruhi faktor genetik, lingkungan dan pola hidup sehari-hari seperti makan, minum, seks, kerja, istirahat, hingga pengelolaan kehidupan emosional. Status kesehatan tersebut menjadi rusak bila keadaan keseimbangan terganggu, tetapi kebanyakan kerusakan pada periode-periode awal bukanlah kerusakan yang serius jika orang mau menyadarinya[3].

Kesehatan merupakan salah satu faktor utama yang dapat mempengaruhi kebugaran dan penampilan tubuh, serta harta yang paling berharga yang tidak pernah bisa ditukar dengan apapun. Oleh karena itu setiap orang tentu mendambakan hidup sehat bahagia dan ingin selalu tampak sehat, bugar, penampilan yang bagus dan awet muda, tidak lekas keriput karena menua. Hal tersebut dapat dirasakan apabila kita pernah sakit. Olahraga dan kesehatan merupakan kebutuhan bagi setiap orang, karena semua orang pasti ingin sehat, tidak seorangpun yang ingin sakit atau terganggu kesehatannya.

#### **2.2 Penyakit Asma**

Asma adalah penyakit inflamasi kronik saluran napas yang disebabkan oleh reaksi hiperresponsif sel imun tubuh seperti mast sel, eosinophils, dan T-lymphocytes terhadap stimulus tertentu dan menimbulkan gejala dyspnea, *whizzing*, dan batuk akibat obstruksi jalan napas yang bersifat reversibel dan terjadi secara episodik berulang. Penyakit asma merupakan proses inflamasi kronik saluran pernapasan yang melibatkan banyak sel dan elemennya. Asma adalah suatu penyakit dengan adanya penyempitan saluran pernapasan yang berhubungan dengan tanggap reaksi yang meningkat dari trakea dan bronkus berupa hiperaktivitas otot polos dan inflamasi, hipersekresi mukus, edema dinding saluran pernapasan, deskuamasi epitel dan infiltrasi sel inflamasi yang disebabkan berbagai macam rangsangan.

### 2.3 Klasifikasi Asma

Asma diklasifikasikan atas asma saat tanpa serangan dan asma saat serangan (akut).[4]

#### 1. Asma saat tanpa serangan

Pada orang dewasa, asma saat tanpa atau diluar serangan, terdiri dari: 1)Intermitten, 2)Persisten ringan, 3)Persisten sedang, dan 4)Persisten berat

Tabel 2.1 Klasifikasi derajat asma berdasarkan gambaran klinis secara umum pada orang dewasa

Derajat Asma	Gejala	Gejala Malam	Faal Paru
Intermitten	Bulanan		APE $\geq$ 80%
	- Gejala <1x/minggu - Tanpa gejala diluar serangan - Serangan singkat	$\leq$ 2kali sebulan	- VEP $\geq$ 80% nilai prediksi APE $\geq$ 80%nilai terbaik. - Variabiliti APE <20%
Persisten ringan	Mingguan		APE>80%
	- Gejala >1x/minggu tetapi <1x/hari - Serangan dapat mengganggu aktifitas dan tidur	>2kali sebulan	- VEP $\geq$ 80% nilai prediksi APE $\geq$ 80%nilai terbaik. Variabiliti APE 20-30%
Persisten sedang	Harian		APE 60-80%

	- Gejala setiap hari - Serangan mengganggu aktifitas dan tidur - Membutuhkan bronkodilator setiap hari	>2kali sebulan	- VEP 60-80% nilai prediksi APE 60-80% nilai terbaik. Variabiliti APE >30%
Persisten berat	Kontinyu		APE ≤ 60%
	- Gejala terus-menerus - Sering kambuh - Aktifitas fisik terbatas	Sering	- VEP ≤ 60% nilai prediksi APE ≤ 60% nilai terbaik. Variabiliti APE >30%

## 2. Asma Saat Serangan

Klasifikasi derajat asma berdasarkan frekuensi serangan dan obat yang digunakan sehari-hari, asma juga dapat dinilai berdasarkan berat-ringannya serangan. *Global Initiative for Asthma* (GINA) membuat pembagian derajat serangan asma berdasarkan gejala dan tanda klinis, uji fungsi paru, dan pemeriksaan laboratorium. Derajat serangan menentukan terapi yang akan diterapkan. Klasifikasi tersebut meliputi asma serangan ringan, asma serangan sedang dan asma serangan berat.

Tabel 2. 2 Klasifikasi asma menurut derajat serangan.

Gejala dan tanda	Berat Serangan Akut			Keadaan Mengancam Jiwa
	Ringan	Sedang	Berat	
Sesak Nafas	Berjalan	Berbicara	Istirahat	

Posisi	Dapat Tidur Terlentang	Lebih suka duduk	Duduk Membungkuk	
Cara Berbicara	Satu Kalimat	Beberapa kata	Kata demi kata	
Kesadaran	Mungkin Gelisah	Gelisah	Gelisah	Mengantuk, Gelisah, Kesadaran Menurun
Frekuensi Nafas	<20/menit	20-30/menit	>30/menit	
Nadi	<100	100-120	>120	Bradikardia
Pulsus Paradoksus	-10mmHg	+/-10-20mmHg	+25 mmHg	- Kelemahan otot
Otot Bantu Nafas dan Retraksi Suprasternal	-	+	+	Torakoabdominal paradoksial
Mengi	Akhir ekspirasi paksa	Akhir ekspirasi	Inspirasi dan ekspirasi	Silent Chest
APE	>80%	60-80%	<60%	-
PaO2	>8 mmHg	60-80 mmHg	<60 mmHg	-
PaCO2	<45 mmHg	<45 mmHg	>45 mmHg	-
SaO2	>95%	91-95%	<90%	-

Klasifikasi asma yang berdasarkan pada Pedoman *Global Initiative For Asthma* (GINA) yang membagi klasifikasai menjadi dua. Pertama, berdasarkan tingkat keparahan (intermitten, persisten ringan, persisten sedang dan persisten berat) dan yang kedua, klasifikasi terbaru, berdasarkan kontrol asma (terkontrol, terkontrol sebagian, dan tidak terkontrol).

Tabel 2. 3 Derajat Kontrol Asma

Karakteristik	Terkontrol	Terkontrol Sebagian	Tidak Terkontrol
Gejala Harian	Tidak ada ( $\leq 2x/mgg$ )	$> 2x/mgg$	$\geq 3/mgg$ Gambaran Asma Terkontrol sebagian adad ala setiapminggu
Keterbatasan Aktivitas	Tidak ada	Ada (tidak sering)	
Gejala noktrunal / terbangun karena asma	Tidak ada	Ada (tidak sering)	
Kebutuhan Pelega	Tidak ada ( $\leq 2x/mgg$ )	$> 2x/mgg$	
Fungsi Paru	Normal	$< 80\%$ prediksi/ nilai terbaik	
Eksaserbasi	Tidak ada	$\geq 1/tahun$	

Berdasarkan gejala siang, aktivitas, gejala malam, pemakaian obat pelega dan eksaserbasi, GINA membagi asma menjadi asma terkontrol, terkontrol sebagian, dan tidak terkontrol. Klasifikasi tersebut dikenal dengan istilah kontrol asma, yang berarti pengendalian terhadap perkembangan penyakit asma.

Beberapa faktor berikut dapat mempengaruhi terjadinya serangan asma :

a. Alergen

Merupakan suatu bahan penyebab alergi. Hal ini dibedakan menjadi tiga yaitu, Inhalan (yang masuk melalui saluran pernapasan), misalnya; debu, bulu binatang, serbuk bunga, bakteri, polusi. Ingestan (yang masuk melalui mulut), misalnya; makanan dan obat-obatan. Kontak (yang masuk melalui kontak dengan kulit), misalnya; perhiasan, logam, jam tangan.

b. Perubahan cuaca

Cuaca yang lembab dan hawa dingin sering mempengaruhi asma, perubahan cuaca juga dapat menjadi pemicu terjadinya serangan asma. Perubahan cuaca yang dimaksud seperti, musim hujan, musim bunga, musim kemarau. Hal ini erat hubungannya dengan angin, serbuk bunga dan debu.

c. Lingkungan.

Erat kaitannya dengan sebab terjadinya asma, ini berhubungan dengan dimana seseorang tinggal.

d. Aktivitas fisik

Melakukan aktivitas fisik yang berat, terutama yang berlangsung secara kompetitif, dan orang dengan HRB (Hiperaktivitas Bronchi) lebih mudah mendapat serangan asma hal ini dikenal dengan istilah EIA (Exercise Induced Asthma).

e. Stress.

Gangguan emosi seseorang dapat menjadi pencetus terjadinya serangan asma, selain itu juga dapat memperberat serangan yang sudah ada.

## 2.4 Jenis Obat Asma

Obat asma terdiri dari obat pelega dan pengontrol. Obat pelega diberikan kepada saat serangan sedangkan obat pengontrol ditujukan untuk pencegahan serangan asma dan di berikan dalam jangka Panjang dan terus menerus. Untuk mengontrol asma

digunakan anti inflamasi (Kontikosteroid inhalasi). [5] Dibawah ini adalah tabel jenis obat.

Tabel 2.4 Derajat Kontrol Asma

Jenis Obat	Golongan	Nama Generik	Bentuk/Kemasan Obat
Pengontrol (Antiinflamasi)	Steroid Inhalasi	Flutikason propionate Budesonide	IDT IDT, turbuhaler
	Antileukokotrin	Zafirlukast	Oral (Tablet)
	Kortikostreoid Sistemik	Metilprednisolon Prednison	Oral (Injeksi) Oral
	Agonis beta-2 kerjalama	Prokaterol Formoterol Salmeterol	Oral Turbuhaler IDT
	Kombinasi steroid dan agonis beta-2 kerjalama	Flutikason+Salmaterol Budesonide+formoterol	IDT Turbuhaler
Pelega (Bronkodilator)	Agonis beta-2 kerja cepat	Salbutamol  Terbutalin  Prokaterol	Oral, IDT, totacap solution  Oral, IDT, turbuhaler, solution, ampul(injeksi)  IDT

	Antikolinergik	Fenoterol	IDT, solution
		Ipratropium bromide	IDT,solution
	Metilsantin	Teofilin	Oral
		Aminofilin	Oral, injeksi
		Teofilin lepas lambat	Oral
	Kortikosteroid Sistemik	Metilprednisolon	Oral, inhaler
		Prednison	Oral

Keterangan :

IDT : Inhalasi Dosis terukur = dapat digunakan Bersama dengan spacer

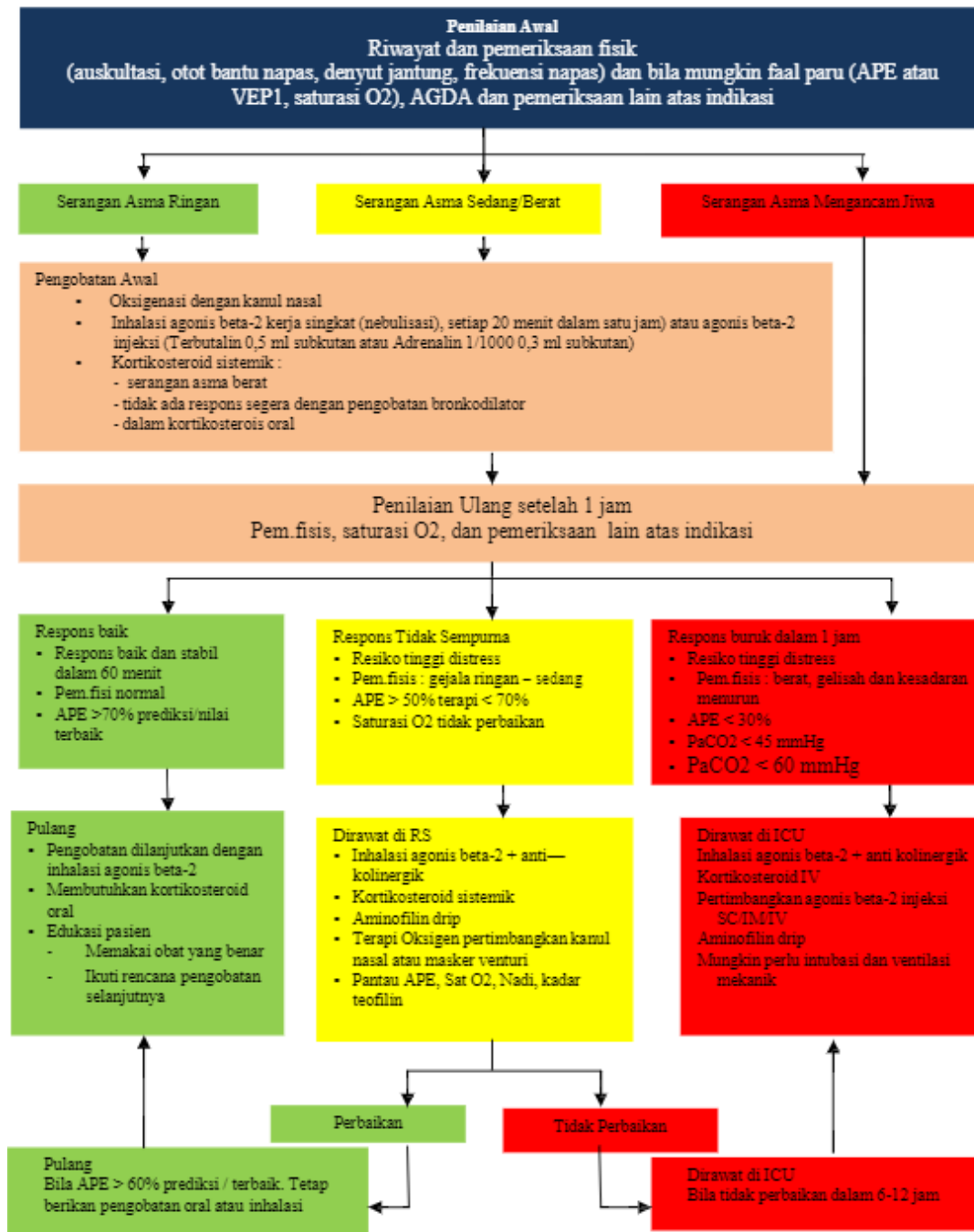
Solution : Larutan untuk penggunaan nebulisasi dengan nebulizer

Oral : Dapat berbentuk sirup atau tablet

Injeksi : Dapat digunakan untuk subkutan, im dan iv

Dibawah ini adalah penataleksanaan penyakit asma [6]





Gambar 2.1 Penataaksanaan Penyakit Asma

Selain edukasi dan Obat-obatan diperlukan juga menjaga kebugaran antara lain dengan melakukan senam asam. Pada orang dewasa, dengan senam asma Indonesia teratur,

senam terkontrol akan tetap terjaga, sedangkan pada anak dapat menggunakan olahraga lain yang menunjang kebugaraan.

## 2.5 *Forward chaining*

*Forward chaining* adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari rules IF-THEN. Bila ada fakta yang cocok dengan bagian IF, maka rule tersebut dieksekusi. Bila sebuah rule dieksekusi, maka sebuah fakta baru (bagian THEN) ditambahkan kedalam database. Setiap kali pencocokan, dimulai dari rule teratas. Setiap rule hanya boleh dieksekusi satu saja. Proses pencocokan berhenti bila tidak ada lagi rule yang dieksekusi. Metode pencarian yang digunakan adalah *Depth-First Search* (DFS), *Breadth-First Search* (BFS) atau Best First Search [7].

Alasan mengapa menggunakan *Forward chaining* karena terdapat banyak cara atau aturan yang berbeda untuk mendapatkan kesimpulan yang sedikit, dan ingin mendapatkan kesimpulan dari fakta-fakta yang sudah ada sebelumnya. Pelacakan *Forward chaining* dapat digunakan dalam tipe sistem seperti[8]:

1. Ada satu atau beberapa kondisi yang menggambarkan suatu sistem.
2. Sistem mencari aturan yang sesuai dengan bagian if pada setiap kondisi.
3. Kondisi baru dapat terbentuk dari kesimpulan yang diperoleh berdasarkan aturan yang ada.
4. Kondisi yang baru akan diproses oleh sistem, dan akan dicari aturan yang sesuai dengan kondisi tersebut (kembali ke langkah 2). Jika tidak ada aturan yang sesuai dengan kondisi tersebut, maka tahapan ini berakhir.

Terdapat dua cara untuk melakukan pencarian dalam metode *Forward chaining* :

1. Memberikan semua data sekaligus ke dalam sistem pakar dalam satu sesi konsultasi. Hal ini dapat digunakan dalam sistem pakar yang terautomatisasi.
2. Memberikan hanya data dengan elemen tertentu kepada sistem pakar selama proses konsultasi.

Langkah-langkah yang harus dilakukan dalam membuat sistem *Forward chaining* berbasis aturan, yaitu:

1. Pendefinisian Masalah
2. Pendefinisian Data Input Sistem *Forward chaining* memerlukan data awal untuk memulai inferensi.
3. Pendefinisian Struktur Pengendalian Data.
4. Penulisan Kode Awal.
5. Pengujian Sistem.
6. Perancangan Antarmuka
7. Evaluasi Sistem.

Berikut ini adalah algoritma dari metode forward-chaining:

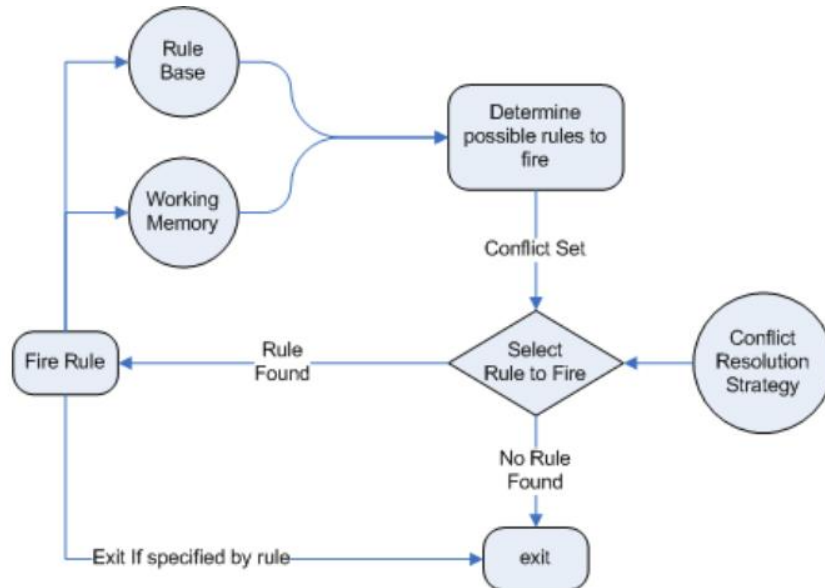
```

function PL-FC-ENTAILS? (KB, Q) returns True
or False
inputs: KB, the knowledge base, a set of
          propositional definite clauses q, the
          query, a proposition symbol
          count ← a table, where count[ e ] is the
          number of symbols in c's premise
          inferred ← a table, where inferred[s] is
          initially false for all symbols
          agenda ← a queue of symbols, initially
          symbols known to be true in KB

while agenda is not empty do
  p ← PoP(agenda)
  if p = q then return true
  if inferred[p]= false then
    inferred[p] ← true
    for each clause c in KB where p
    is in c.PREMISE do
      decrement count[c]
      if count[c] = 0 then
        add
        c.CONCLUSION to agenda
  return false

```

Jika algoritma diatas ditransformasikan dalam bentuk flow chart adalah seperti gambar dibawah ini :



Gambar 2.2 Flowchart Forward Chaining

## 2.6 Chatbot

*Chatbot* atau percakapan dengan bot merupakan suatu aplikasi kecerdasan buatan yang mampu mensimulasikan percakapan yang cerdas, sesuai dengan ilmu pengetahuan yang diberikan. *Chatbot* merupakan agen cerdas yang dapat meniru kemampuan manusia untuk dapat melakukan percakapan dengan pengguna yaitu manusia. Pembangunan chatbot dapat dilakukan dengan menggunakan pendekatan dari bidang *Question and Answering*. *Chatbot* dapat diimplementasikan untuk bidang komersial, pendidikan, hiburan, dan sektor pelayanan publik[9].

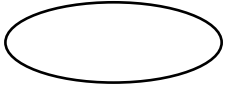



## 2.7 UML

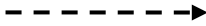

*Unified Modeling Language (UML)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem [10]. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

### 1. Use Case Diagram

*Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* dapat dilihat pada tabel 2.5

Tabel 2.5 Use Case Diagram




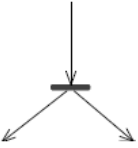
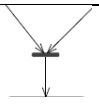
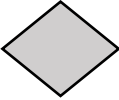
Gambar	Keterangan
	<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja</p>
	<p><i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i>, tetapi tidak memiliki kontrol terhadap <i>use case</i></p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem</p>


	<i>Include</i> , merupakan di dalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi

## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity Diagram* dapat dilihat pada tabel 2.6

Tabel 2.6 Simbol Activity Diagram





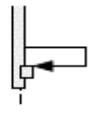

Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	<i>End Point</i> , akhir aktivitas
	<i>Activities</i> , menggambar kan suatu proses/kegiatan bisnis
	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Points</i> , menggambar kan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>


	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa
---	---

### 3. Diagram Urutan (*Sequence Diagram*)

*Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* dapat dilihat pada tabel 2.7

Tabel 2.7 Simbol Sequence Diagram

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek
	<i>Message</i> , simbol mengirim pesan antar <i>class</i>
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi

	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>
---	---

#### 4. Diagram Kelas (*Class Diagram*)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

*Class Diagram* secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalization* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*

### 2.8 *Smartphone*

*Smartphone* adalah *handphone* cerdas yang memiliki kelebihan dibanding alat telekomunikasi lainnya. Kelebihannya terlihat dari proses pembuatannya dan proses penggunaannya. *Smartphone* merupakan ponsel multimedia yang menggabungkan fungsionalitas PC dan handser sehingga menghasilkan gadget yang mewah, dimana terdapat pesan teks, kamera, pemutar musik, video, *game*, akses email, tv digital, *search engine*, pengelola informasi pribadi, fitur GPS, jasa telepon internet dan bahkan terdapat telepon yang juga berfungsi sebagai kartu kredit[11].

### 2.9 *Android*

Android adalah *operating system* (OS) berlisensi platform terbuka berbasis linux yang diperuntukan untuk mobile device seperti *smartphone* maupun tablet PC. *Android Open Source Project* (AOSP) yang dibuat oleh Google. Beberapa produsen mobile menanamkannya untuk menjadikan android sebagai sistem operasi. Berbagai keunggulan sistem operasi android mampu memenuhi kebutuhan informasi manusia secara cepat dan menarik. Android disematkan berbagai fitur sebagai fungsi utama[12].



1. Kerangka aplikasi, memungkinkan penggunaan dan penghapusan komponen yang tersedia.
2. Dalvik mesin virtual, mesin virtual dioptimalkan untuk perangkat telepon seluler.
3. Grafik: grafik di 2D dan grafis 3D berdasarkan pustaka OpenGL.
4. SQLite: sebagai penyimpanan data.
5. Dukungan pembacaan terhadap banyak format media.
6. GSM, *Bluetooth*, EDGE, 3G, 4G dan WiFi (tergantung piranti keras).
7. Kamera, *Global Positioning System* (GPS), kompas, NFC dan *accelerometer*.

## 2.10 Kotlin

Kotlin adalah bahasa pemrograman yang diketik secara statis yang menargetkan *Java virtual machine* (JVM), Android, JavaScript, dan Native. Kotlin dikembangkan oleh JetBrains, proyek Kotlin dimulai pada tahun 2010 dan merupakan Bahasa pemrograman open source. Kotlin versi 1.0 secara resmi dirilis pada bulan Februari 2016. Dalam mengembangkan aplikasi Android, Kotlin menawarkan beberapa fitur yaitu: [13]

1. **Kompatibilitas:** Kotlin sepenuhnya kompatibel dengan JDK 6, sehingga Kotlin dapat berjalan pada perangkat Android versi lama. Kotlin juga sepenuhnya didukung di Android Studio dan kompatibel dengan Android build system.
2. **Performa:** Sebuah aplikasi Kotlin dapat berjalan secepat Java, karena memiliki struktur bytecode yang sangat mirip. Dengan dukungan Kotlin untuk fungsi inline, kode menggunakan lambda sering berjalan lebih cepat daripada kode yang sama yang ditulis di Java.
3. **Interoperabilitas:** Kotlin 100% dapat beroperasi dengan Java, memungkinkan programmer untuk menggunakan semua Android *Library* dalam aplikasi, termasuk pengolahan anotasi, sehingga databinding dan Dagger dapat bekerja.
4. **Footprint:** Kotlin memiliki runtime library yang sangat padat/tersusun rapat, yang ke depannya dapat dikurangi melalui penggunaan ProGuard. Dalam aplikasi nyata, runtime pada Kotlin hanya menambahkan beberapa ratus metode dan kurang dari 100K untuk ukuran dari file .apk.

5. Waktu kompilasi: Kotlin mendukung kompilasi Inkremental yang efisien, sehingga sementara ada beberapa tambahan overhead untuk clean build, pembangunan bertahap ini biasanya sama cepat atau lebih cepat daripada dengan Java.
6. *Learning Curve*: untuk pengembang Java, memulai menggunakan Kotlin sangat mudah. *Automated Java to Kotlin converter* yang terdapat dalam plugin Kotlin membantu pengembang Java pada langkah pertama. Terdapat juga Kotlin Koans yang menawarkan panduan dalam penggunaan Bahasa pemrograman ini.

### 2.11 Android Studio

Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada event Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android. Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan *Eclipse* disertai dengan ADT *plugin (Android Development Tools)*. Android studio memiliki fitur[14]:

1. Projek berbasis pada *Gradle Build* .
2. *Refactory* dan pembenahan bug yang cepat .
3. *Tools* baru yang bernama “Lint” dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
4. Mendukung *Proguard And App-signing* untuk keamanan.
5. Memiliki GUI aplikasi android lebih mudah .
6. Didukung oleh *Google Cloud Platfrom* untuk setiap aplikasi yang dikembangkan.

### 2.12 Basis Data

Data adalah fakta atau angka yang disimpan atau dicatat. Informasi adalah data yang diolah dan disajikan sehingga memiliki suatu arti. Basis data selain untuk mencatat atau menyimpan data-data, juga dapat digunakan untuk membentuk suatu informasi. Basis data terdiri dari 2 tabel atau lebih dimana tabel-tabel tersebut saling

berhubungan digunakan untuk memenuhi kebutuhan para pemakai dalam suatu organisasi. Setiap tabel pada basis data biasanya berisi tentang suatu hal. Sistem manajemen basis data adalah perangkat lunak yang menciptakan, proses, dan mengelola basis data[15].

Data model adalah sekumpulan konsep untuk menggambarkan data, hubungan antar data, dan batasan-batasan pada data dalam organisasi. Tujuan pemodelan data adalah untuk menyajikan data dan membuat data mudah dimengerti sehingga jika pemodelan ini dilakukan kita dapat dengan mudah merancang basis data. Salah satu Model Basis Data yaitu Model hierarki. Model ini adalah suatu jenis terbatas dari model jaringan. Pada model ini data disajikan sebagai kumpulan record dan relasi disajikan sebagai set-set tetapi model ini memungkinkan sebuah node (anak) untuk hanya memiliki satu orang tua tetapi satu orang tua bisa memiliki beberapa anak. Model hierarki disajikan seperti model graf pohon (tree) dimana record-record ditampilkan sebagai node (anak) yang juga disebut sebagai segment dan set disebut edge.

### **2.13 Firebase**

*Firestore* adalah BaaS (*Backend as a Service*) yang saat ini dimiliki oleh Google. *Firestore* ini merupakan solusi yang ditawarkan oleh Google untuk mempermudah pekerjaan *Mobile Apps Developer*. Dengan adanya *Firestore*, *apps developer* bisa fokus mengembangkan aplikasi tanpa harus memberikan *effort* yang besar untuk urusan backend. Salah satu fitur yang dimiliki oleh *Firestore* yaitu *Firestore Real Time Database*, fitur ini yang memberikan sebuah NoSQL database yang bisa diakses secara Real Time oleh pengguna aplikasi. Dan hebatnya adalah aplikasi bisa menyimpan data secara lokal ketika tidak ada akses internet, kemudian melakukan sync data segera setelah mendapatkan akses internet[16].

