

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Masalah Mental**

Masalah mental merupakan sebuah permasalahan yang sering terjadi di berbagai Negara, tentunya Indonesia pun tidak luput dari ini. Masalah mental adalah sebuah keadaan dimana seseorang mengalami sebuah guncangan dalam kehidupannya yang menyebabkan terjadinya kecemasan, merasa tidak percaya diri, gangguan kesehatan, dan paling parah yaitu mengalami depresi yang mengancam nyawa. Prevalensi penyakit mental ini menjadi sebuah kecacatan yang cukup besar, dengan kombinasi gangguan dan kecemasan menjadi yang paling berpengaruh dalam gejala ini. Gejala penyakit mental dan stigma yang menyertainya berdampak negatif pada harga diri orang, mengganggu hubungan, dan membatasi kemampuan untuk memperoleh tempat tinggal, pekerjaan, dan pendidikan, Menurut Survei Nasional Direktur Pusat Konseling di 274 lembaga, 85% direktur pusat melaporkan peningkatan masalah psikologis yang parah selama 5 tahun terakhir, termasuk ketidakmampuan belajar (71%), melukai diri sendiri (51%), gangguan makan (38%), penggunaan alkohol (45%), penggunaan obat-obatan terlarang lainnya (49%), masalah kekerasan seksual di kampus (33%), dan masalah yang terkait dengan pelecehan seksual sebelumnya (34%) [6] [7]. Di Indonesia sendiri berbagai cara telah dilakukan mulai dari penyuluhan terhadap masalah mental mulai dari mendirikan tempat bagi para penderita dan tersedia nya psikiater untuk menjadi penuntun dalam mengatasi masalah tersebut, tetapi jumlah psikiater itu sendiri tidak mencukupi untuk mengatasi jumlah orang yang mengalami masalah mental, akibatnya presentase kenaikan nya terus bertambah.

#### **2.2. *Chatting***

*Chatting* merupakan sebuah komunikasi yang dilakukan secara online melalui platform media social yang tersambung dengan jaringan, komunikasi

tersebut dilakukan dengan cara mengetik dan mengirim kan pesan tersebut kepada lawan bicara yang berada di tempat lain [8]. Dengan adanya aplikasi chatting maka penyampaian sebuah pesan akan menjadi lebih efisien karna pesan akan terkirim dalam waktu tertentu dan tidak memakan biaya yang banyak. Sudah banyak aplikasi berbasis chatting yang memiliki peran besar tentunya di Indonesia [9].

### **2.3. Blackbox**

Black box adalah sebuah metode pengujian dalam merancang sebuah aplikasi yang dimana metode ini hanya berfokus kepada fungsionalitas aplikasi seperti fungsi fungsi yang berjalan, kesesuaian alur kerja aplikasi yang diinginkan. Metode ini juga mengabaikan mekanisme sistem dan komponen sehingga output yang dihasilkan akan merespon pada input yang dipilih oleh pengguna dan kondisi yang di eksekusi pada program aplikasi [10].

### **2.4. Android**

Android merupakan salah satu sistem operasi atau *operating system* berbasis mobile yang sangat banyak di gunakan sekarang ini [11]. Android diperkenalkan pada tahun 2007 oleh Google dengan basis kernel Linux untuk mendukung kinerja perangkat elektronik layar sentuh, android bersifat open source atau bebas digunakan, dimodifikasi, diperbaiki dan didistribusikan oleh para pembuat ataupun pengembang perangkat lunak [12].

### **2.5. Android Studio**

Android studio merupakan software untuk mengembangkan sebuah aplikasi berbasis android yang memiliki sistem 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 pada tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android [13]. Android studio sendiri dikembangkan berdasarkan IntelliJ

IDEA yang mirip dengan *Eclipse* disertai dengan ADT *plugin* (*Android Development Tools*) [14].

## 2.6. **Firebase**

*Firebase* adalah BaaS (*Backend as a Service*) yang saat ini dimiliki oleh Google. *Firebase* ini merupakan solusi yang ditawarkan oleh Google untuk mempermudah pekerjaan *Mobile Apps Developer*. Dengan adanya *Firebase*, *apps developer* bisa mengembangkan aplikasi tanpa harus memberikan *effort* yang besar untuk urusan backend. Salah satu fitur yang dimiliki oleh *Firebase* yaitu *Firebase Real Time Database*, fitur ini yang memberikan sebuah NoSQL database yang bisa diakses secara Real Time oleh pengguna aplikasi [15].

## 2.7. **Bahasa Java**

Bahasa java merupakan bahasa pemrograman yang biasa di pakai untuk membuat sebuah software yang bersifat *simple*, dan *object oriented*. *Simple* karena java dikembangkan dari bahasa C++ dengan menghilangkan fungsi yang jarang digunakan dan kerumitan yang ada didalamnya serta memiliki sintaks yang akan memberikan kemudahan dalam penggunaannya. *Object Oriented* karena menggunakan mekanisme obyek dalam desain dan pemrogramannya. Java memiliki dukungan luas terhadap pemrograman jaringan dan internet sehingga sangat banyak digunakan dalam aplikasi sistem terdistribusi dan jaringan internet. Source Bahasa Pemrograman Java akan di *compile* menjadi 100 bytecode. Platform kartu Java dikhususkan untuk mengembangkan aplikasi dengan spesifikasi yang memiliki RAM 512 byte, ROM 16 Kbytes, dan CPU delapan bit [16] [17].

## 2.8. **RUP ( *Rational Unified Process* )**

RUP atau Rational Unified Process adalah sebuah metode pengembangan perangkat lunak yang dilakukan secara berulang – ulang dan hanya berfokus kepada arsitektur aplikasi. RUP menggunakan use-case untuk membangun sebuah

perangkat lunak, dan juga memiliki beberapa fase proses, berikut adalah fase – fase yang terjadi pada metode RUP :

1. *Inception*

Inception adalah tahap dimana proses model bisnis dibuat dan di sesuaikan dengan kebutuhan system aplikasi berdasarkan *requirement*

2. *Elaboration*

Elaboration adalah tahap yang berfokus pada perencanaan system arsitektur. Tahap elaboration juga merupakan sebuah tahap dimana analisis dan desain system implementasi dibuat sesuai dengan dengan purwarupa system.

3. *Construction*

Construction merupakan tahap dimana interface dan desain pada sebuah aplikasi dibuat sesuai dengan perencanaan.

4. *Transition*

Transition adalah tahap terakhir dari metode RUP dimana tahap ini merupakan sebuah tahap pengujian dimana aplikasi akan di uji sesuai dengan fungsinya, untuk mendeteksi kesalahan system dan fungsionalitas nya. Teknik pengjian ini menggunakan Blackbox Testing

RUP adalah proses yang dapat di konfigurasi dan metode yang paling cocok untuk pengembangan perangkat lunak. Unified Process ini terdiri atas arsitektur proses yang sangat sederhana dan sangat jelas, dan bisa bervariasi dengan jenis proses yang berbeda [18].

## **2.9. *Fuzzy Logic***

Logika *Fuzzy* pertama kali di kembangkan oleh Prof . Lofti Zadeh yang berasal dari Universitas California, USA. Profesor Zadeh memodifikasi sebuah himpunan data yang dimana setiap anggota data tersebut memiliki anggota 0 sampai dengan 1 yang disebut dengan himpunan kabur. Pada umumnya sistem logika fuzzy sering diterapkan untuk mengontrol sebuah perangkat yang non liner dan sistem kontrol yang dinamikanya diketahui secara pasti, seperti kontrol posisi motor

servo dan kontrol komponen lengan pada robot, dan untuk mengelola pengambilan keputusan atau sistem diagnosis yang kompleks, sistem fuzzy memiliki proses defuzzifikasi bawaan yang memperhitungkan ketidakpastian pada beberapa siklus data yang memiliki gangguan seperti jaringan saraf untuk memperhitungkan ketidakpastian [19] [20]. Nilai pada logika fuzzy bisa menjadi benar atau salah pada waktu yang bersamaan tetapi besar nilai kebenaran dan kesalahan tergantung pada bobot pada anggotanya. Terdapat 3 metode dalam *fuzzy inference system* yaitu metode Tsukamoto, metode Mamdani, dan Metode Sugeno. Selain itu terdapat juga *fuzzy clustering* yang berisi metode tentang *fuzzy c - means* dan *subtractive clustering*. Berikut beberapa penjelasan tentang 3 metode dalam fuzzy inference:

#### 1. Metode Tsukamoto

Metode Tsukamoto memiliki aturan jika setiap konsekuensinya itu berbentuk IF-THEN yang harus di representasikan dengan himpunan fuzzy dan fungsi anggota, sebagai hasilnya, keluaran dari inferensi di setiap aturan akan di tegaskan berdasarkan  $\alpha$ -predikat. Hasil akhirnya dapat di peroleh dengan bobot rata rata

#### 2. Metode Mamdani

Metode Mamdani atau yang dikenal dengan metode Min-Max ini di perkenalkan oleh Ebrahim Mamdani pada tahun 1975, untuk mendapatkan output dari metode mamdani ini memerlukan 4 proses yaitu :

##### a) Pembentukan sebuah himpunan

Proses ini membagi variable output menjadi 1 atau lebih dari himpunan fuzzy

##### b) Fungsi implikasi

Fungsi implikasi pada metode mamdani adalah fungsi Min

##### c) Komposisi pada aturan

Jika komposisi system terdiri dari beberapa aturan, maka kumpulan korelasi antar aturan tersebut akan menghasilkan inferensi

d) Penegasan

Proses defuzzy menghasilkan input dari sebuah himpunan yang di peroleh melalui aturan fuzzy, sedangkan output nya dihasilkan merupakan sebuah bilangan yang ada pada domain fuzzy.

3. Metode Fuzzy Sugeno

Metode sugeno ini hampir mirip dengan metode fuzzy mamdani, yang membedakan adalah output dari fuzzy sugeno bukan berupa himpunan fuzzy, melainkan konstanta atau persamaan linear. Metode fuzzy sugeno ini di perkenalkan oleh Takagi Sugeno Kang pada tahun 1985. Terdapat 2 model pada metode fuzzy sugeno :

a. Model Fuzzy Sugeno Orde-Nol

Bentuk model dari fuzzy adalah

IF  $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_n \text{ is } A_n)$

THEN  $z = k$

$A_1$  sebagai himpunan ke -i dan k adalah konstanta

b. Model fuzzy Sugeno Orde-Satu

IF  $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_n \text{ is } A_n)$

THEN  $z = p_1 * x_1 + \dots + x_n + q$

**2.9.1 Variable Fuzzy**

Variable Fuzzy adalah sebuah variable yang digunakan dalam system yang berfungsi sebagai indikator atau penamaan yang akan di hitung, contohnya seperti umur, berat badan, tinggi, dan temperatur.

**2.9.2. Himpunan Fuzzy**

Himpunan fuzzy merupakan sebuah himpunan yang digunakan pada sebuah variable pada system fuzzy yang biasa nya mempunyai 2 atribut yaitu:

1. Numeris yang berupa sebuah angka atau nilai yang bertujuan untuk menunjukan ukuran dari suatu variable fuzzy, contohnya 10, 15, 30, 45

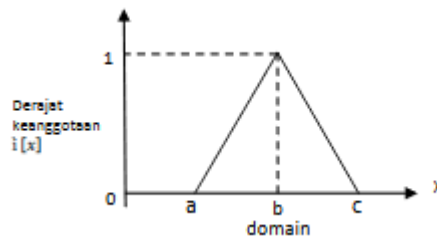
- Linguistik yang berupa persamaan atau penamaan yang menggunakan bahasa atau istilah untuk kondisi tertentu yang biasanya mewakili keadaan, contohnya muda, tua, remaja, anak-anak. Variable linguistic adalah Variable yang mempunyai nilai kata atau kalimat dalam natural atau bahasa yang sering digunakan untuk mengklasifikasikan beberapa hal tertentu

### 2.9.3. Fungsi Keanggotaan

Fungsi keanggotaan atau *membership functions* merupakan suatu kurva yang berfungsi sebagai pemetaan titik input data kedalam sebuah nilai dalam keanggotaan yang memiliki interval dari 1 sampai dengan 1, beberapa contoh dalam fungsi keanggotaan.

#### 1. Fungsi Segitiga

Fungsi segitiga ini biasanya memiliki satu nilai  $x$  yang memiliki derajat keanggotaan yang sama dengan 1, contohnya ketika  $x = b$  tetapi nilai di sekitar  $b$  memiliki keanggotaan yang cukup jauh dari 1. Berikut fungsi grafik dan notasi dari fungsi segitiga



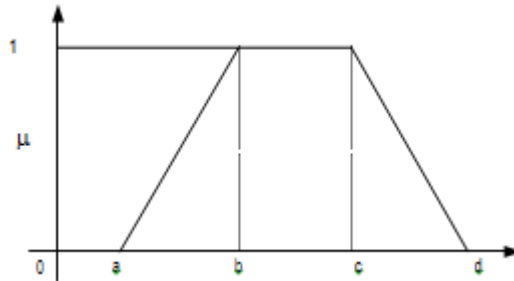
Fungsi keanggotaan :

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ \frac{(c-x)}{(c-b)}; & b \leq x \leq c \end{cases}$$

Dimana variable  $x$  adalah variable yang di cari, variable  $a$  adalah batas bawah, variable  $b$  adalah batas tengah, dan variable  $c$  adalah batas atas

## 2. Fungsi Trapesium

Fungsi trapesium memiliki beberapa nilai  $x$  yang memiliki keanggotaan derajat yang sama dengan 1, ketika  $b \leq x \leq c$ , untuk keanggotaan  $a < x < b$  dan  $c < x < d$  memiliki karakteristik yang sama dengan fungsi segitiga. Berikut grafiknya



Fungsi keanggotaan :

$$\mu(x) = 1; \begin{cases} 0; x \leq a \text{ atau } x \geq d \\ \frac{(x-a)}{(b-a)}; a < x < b \\ \frac{(d-x)}{(d-c)}; c < x \leq d \end{cases}$$

Dimana  $x$  adalah variable yang di cari,  $a$  adalah batas bawah,  $b$  dan  $c$  adalah batas tengah, dan  $d$  batas atas

### 2.10. UML ( Unified Modeling Language)

*Unified Modeling Language (UML)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun

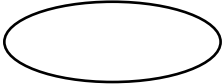
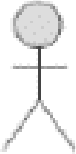






perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

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 berikut

Tabel 2.1 Use Case Diagram



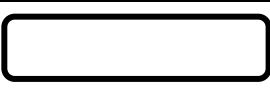
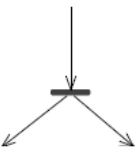
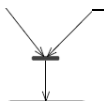
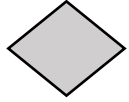
Gambar	Keterangan
	<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktir, yang dinyatakan dengan menggunakan kata kerja</p>
	<p><i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktir, 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</p>


	panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
	<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 table berikut

Tabel 2.2 Simbol *Activity Diagram*





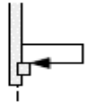


Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	<i>End Point</i> , akhir aktivitas
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis
	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Points</i> , menggambarkan 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 2.3 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*, *Generalisation* 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*