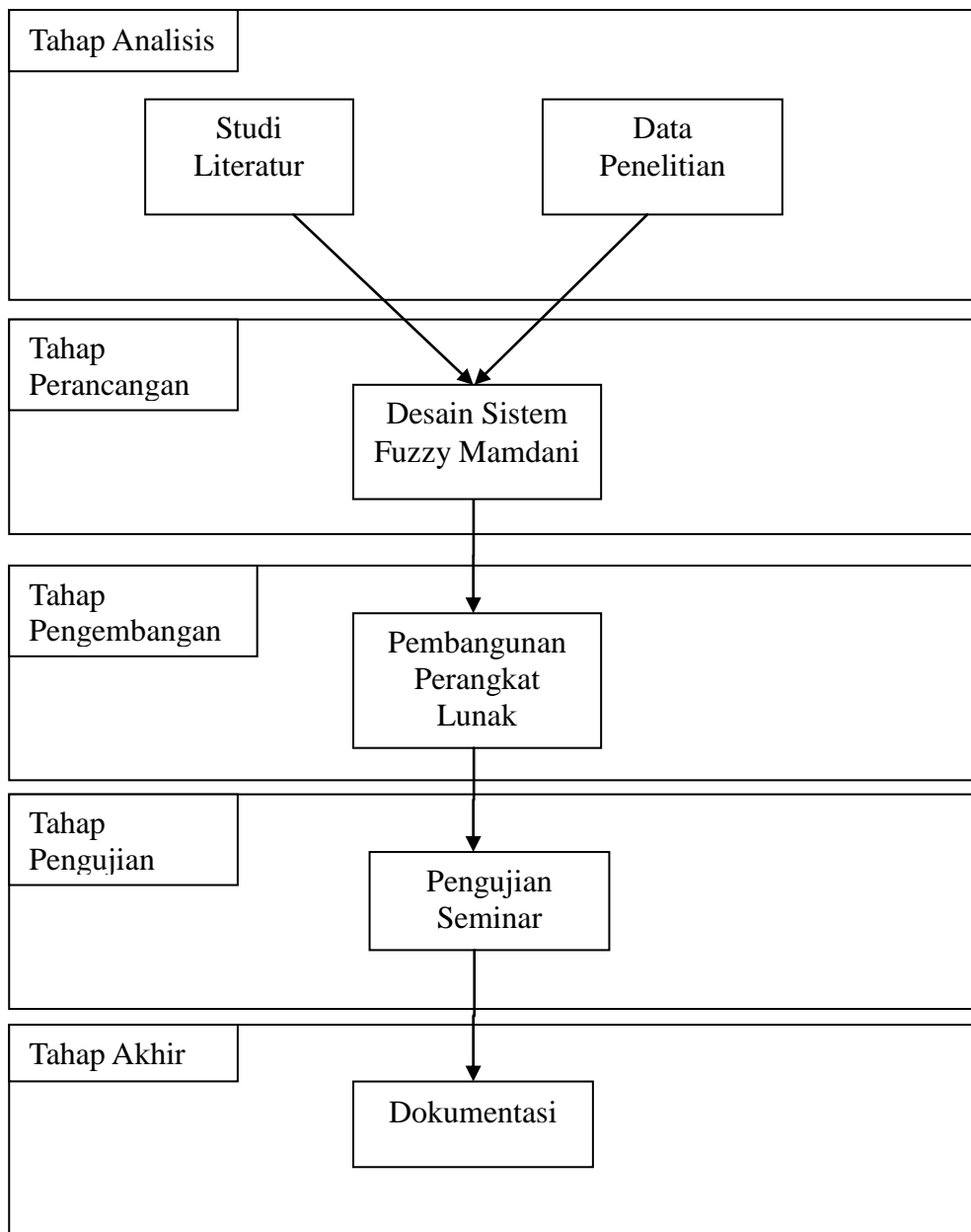


BAB III

METODE PENELITIAN

3.1 Desain penelitian



Gambar 3.1 *Skema Desain Penelitian*

Berdasarkan skema desain penelitian pada gambar 3.1 diatas, Tahapan penelitian yang akan dilakukan meliputi langkah – langkah berikut:

1. Studi literatur di Tahap Analisis, mencari teori pendukung penelitian dan ilmu – ilmu dasar yang akan digunakan untuk memahami dan mengembangkan *game* dan sistem *fuzzy logic* metode mamdani
2. Data penelitian di Tahap Analisis, data yang didapat dari studi literatur akan dipertimbangkan untuk melakukan perancangan dan pembangunan perangkat lunak, penentuan *fuzzy set*, penentuan *fuzzy rule*, dan penentuan kriteria dan spesifikasi *game*. Selain mendapatkan data dari 19ip roses19, didapat juga data tentang kemampuan sebuah unit dalam berbagai permainan *real time strategy* dari website yang mengeluarkan *walkthrough* untuk *game* tersebut
3. Desain Sistem Fuzzy Mamdani di Tahap Perancangan, setelah mendapatkan data penelitian yang sudah dirangkum dan spesifikasi dari sistem, dirancanglah sistem *fuzzy logic* metode mamdani yang akan diimplementasikan kedalam *game real time strategy* yang dibangun. Metode yang dipilih untuk komposisi aturan adalah metode *Max*. Sedangkan untuk metode penegasan (*defuzzyfication*) menggunakan metode *centroid of gravity*
4. Pembangunan perangkat lunak di Tahap Pengembangan, pada tahapan ini sistem akan dibangun dengan menggunakan metode sekuensial linear dan memakai tahap dasar pembangunan perangkat lunak seperti analisis, desain, *coding*, dan pengujian
5. Pengujian Seminar di Tahap Pengujian, pada tahapan ini perangkat lunak dan dokumentasi awal akan diuji, kemudian akan mendapatkan revisi dari penguji untuk pembuatan dokumentasi tahap akhir
6. Dokumentasi di Tahap Akhir, hasil penelitian akan didokumentasikan dalam skripsi dan dokumen teknis perangkat lunak

3.2 Alat dan Bahan

Pada penelitian ini, peneliti akan menggunakan perangkat keras dan perangkat lunak sebagai berikut:

Perangkat Keras:

- a) *Processor* Intel Core i3 Quad Core
- b) SRAM 2 GB
- c) *Harddisk* kosong 10 GB
- d) Monitor beresolusi
- e) Mouse dan keyboard

Perangkat Lunak:

- a) Geany Text Editor
- b) Bahasa Pemrograman Python
- c) *Game Framework* PyGame
- d) Inkscape dan GIMP
- e) Spritelib
- f) StarUML

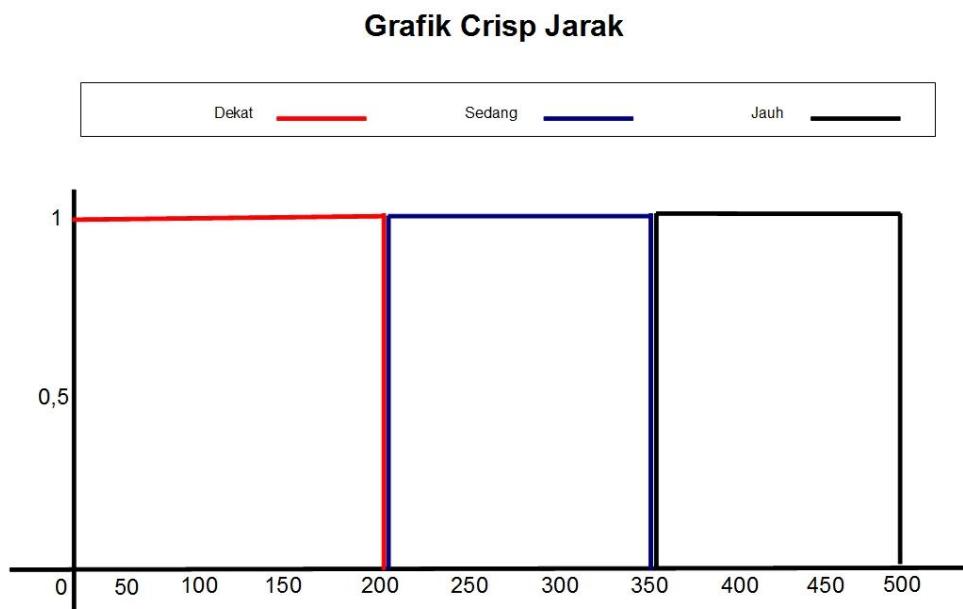
Bahan penelitian yang digunakan adalah *paper*, *textbook*, *tutorial*, artikel majalah, dan dokumentasi lainnya yang didapat melalui observasi di perpustakaan dan *World Wide Web*.

Di dalam penelitian ini akan digunakan variable *fuzzy* yang mengadopsi dari Muhammad, Irsyad Arif, Imam Kuwardayan, Rully Soelaiman. (2011) yang akan digunakan oleh unit musuh. Variabel *fuzzy* tersebut antara lain :

1. Variabel jarak (*range*), merupakan variabel *fuzzy* yang meninjau seberapa jauh unit musuh bisa dijangkau oleh unit *player*. Terdiri dari tiga himpunan *fuzzy* untuk variabel jarak, yaitu : DEKAT, SEDANG, dan JAUH
2. Variabel darah (*health*), merupakan variabel *fuzzy* yang meninjau keadaan kesehatan unit musuh untuk menentukan suatu aksi. Terdiri dari lima himpunan *fuzzy* untuk variabel darah, yaitu : SANGAT_RENDAH, RENDAH, SEDANG, TINGGI, dan SANGAT_TINGGI
3. Variabel aksi (*action*), merupakan variabel *fuzzy* yang menentukan tindakan unit musuh terhadap unit *player* yang dihadapinya selama permainan. Terdiri dari tiga himpunan *fuzzy* untuk variabel aksi, yaitu : KEMBALI_KE_MARKAS, JALAN, dan SERANG

Untuk mendapatkan grafik keanggotaan *fuzzy*, kita harus mengetahui grafik logika Boolean yang akan diubah ke grafik keanggotaan *fuzzy*. Dalam menentukan grafik keanggotaan *fuzzy* dapat digunakan metode intuisi yang dijelaskan Ross, Timothy J. (2010). Metode intuisi adalah kemampuan seseorang untuk mengembangkan grafik keanggotaan *fuzzy* berdasarkan pemahaman dan kecerdasan mereka.

Kita awali dengan menentukan grafik keanggotaan *fuzzy* untuk variabel *fuzzy* jarak. Diketahui grafik logika Boolean atau *crisp* jarak berikut:



Gambar 3.2 Grafik Crisp Jarak

Setelah mengetahui grafik logika Boolean, dapat ditentukan sejumlah titik secara acak yang terdapat pada rentang terendah sampai maksimum. Rentang yang terdapat pada grafik Boolean jarak adalah 0 – 500. Ditentukan titik – titik berikut : 0, 50, 100, 150, 200, 230, 250, 270, 300, 330, 350, 380, 400, 430, 450, dan 500. Titik – titik tersebut akan menentukan pembentukan grafik keanggotaan *fuzzy*.

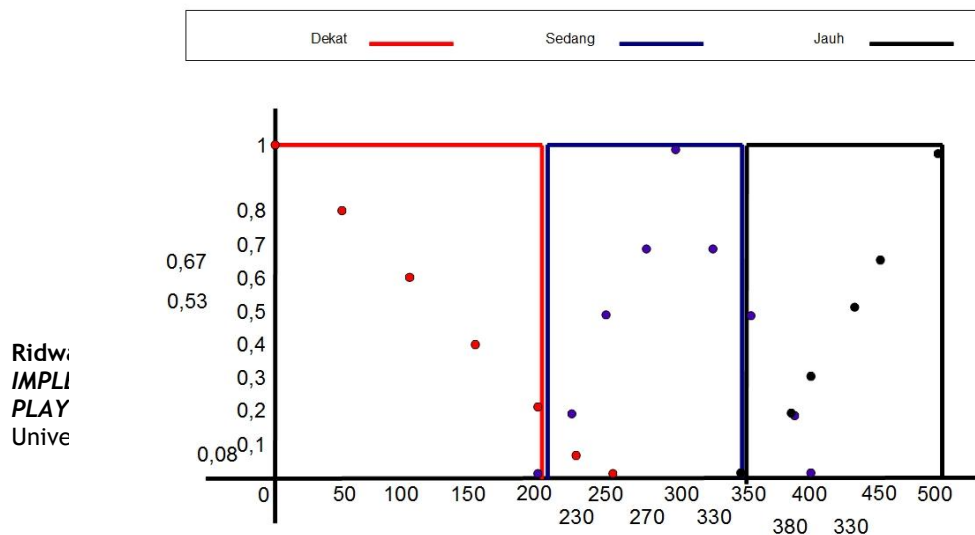
Titik tersebut akan ditentukan nilai keanggotaannya di masing – masing himpunan grafik Boolean. Nilai keanggotaannya terdiri dari rentang 0 hingga 1. Hasil penentuan nilai keanggotaan titik – titik diatas pada himpunan grafik Boolean jarak terdapat pada tabel berikut ini:

Tabel 3.1 Tabel Penentuan Keanggotaan Titik di Grafik Crisp Jarak

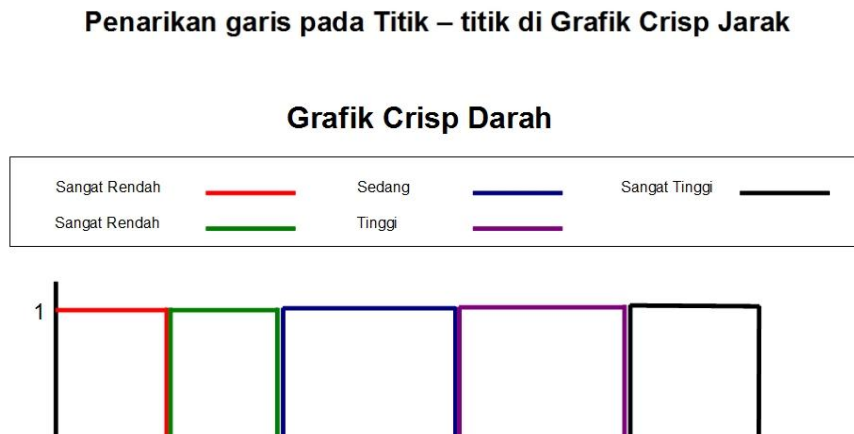
| No | Poin | Intuisi | | |
|----|------|---------|--------|------|
| | | Dekat | Sedang | Jauh |
| 1 | 0 | 1 | 0 | 0 |
| 2 | 50 | 0,8 | 0 | 0 |
| 3 | 100 | 0,6 | 0 | 0 |
| 4 | 150 | 0,4 | 0 | 0 |
| 5 | 200 | 0,2 | 0 | 0 |
| 6 | 230 | 0,08 | 0,3 | 0 |
| 7 | 250 | 0 | 0,5 | 0 |
| 8 | 270 | 0 | 0,7 | 0 |
| 9 | 300 | 0 | 0,1 | 0 |
| 10 | 330 | 0 | 0,7 | 0 |
| 11 | 350 | 0 | 0,5 | 0 |
| 12 | 380 | 0 | 0,2 | 0,2 |
| 13 | 400 | 0 | 0 | 0,3 |
| 14 | 430 | 0 | 0 | 0,53 |
| 15 | 450 | 0 | 0 | 0,67 |
| 16 | 500 | 0 | 0 | 1 |

Kemudian titik – titik diatas beserta nilai hasil intuisinya, dipetakan kedalam grafik Boolean jarak. Hasil pemetaannya dapat dilihat pada grafik berikut:

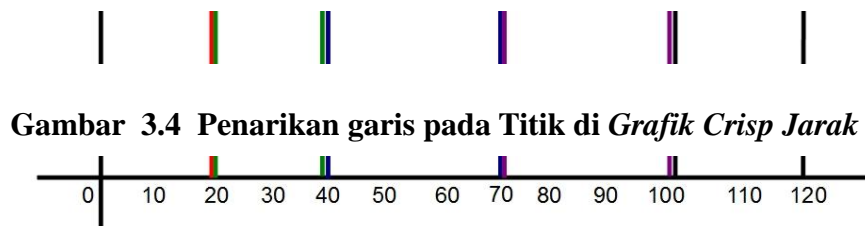
Penentuan Keanggotaan Titik di Grafik Crisp Jarak



Setelah memetakan titik – titik yang ditentukan ke grafik Boolean jarak, sekarang tarik garis yang melewati titik – titik tersebut:



Gambar 3.3 Penentuan Keanggotaan Titik di *Grafik Crisp Jarak*



Berikutnya akan ditentukan grafik keanggotaan *fuzzy* untuk variabel *fuzzy* darah. Diketahui grafik logika Boolean atau *crisp* dari darah berikut:

Setelah mengetahui grafik logika Boolean dari darah, dapat ditentukan sejumlah titik secara acak yang terdapat pada rentang terendah sampai maksimum. Rentang yang terdapat pada grafik Boolean darah adalah 0 – 120. Ditentukan titik – titik berikut : 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, dan 120. Titik – titik tersebut akan menentukan pembentukan grafik keanggotaan *fuzzy*.

Titik tersebut akan ditentukan nilai keanggotaannya di masing – masing himpunan grafik Boolean darah. Nilai keanggotaannya terdiri dari rentang 0 hingga 1. Hasil penentuan nilai keanggotaan titik – titik diatas pada himpunan

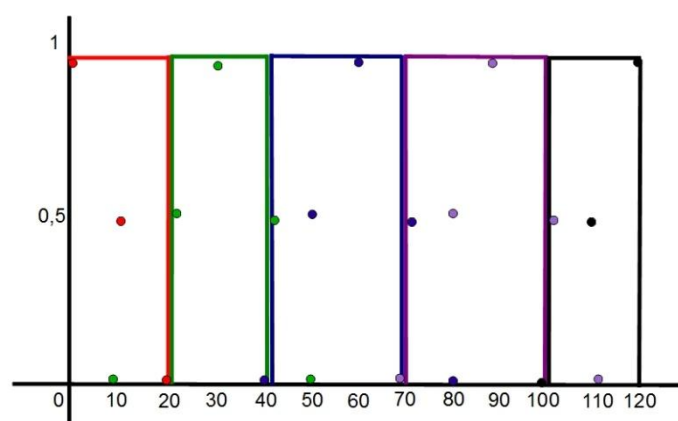
grafik Boolean darah terdapat pada tabel berikut ini:

Tabel 3.2 Tabel Penentuan Keanggotaan Titik di Grafik Crisp Darah

| No | Poin | Intuisi | | | | |
|----|------|---------------|--------|--------|--------|---------------|
| | | Sangat Rendah | Rendah | Sedang | Tinggi | Sangat Tinggi |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 2 | 10 | 0,5 | 0 | 0 | 0 | 0 |
| 3 | 20 | 0 | 0,5 | 0 | 0 | 0 |
| 4 | 30 | 0 | 1 | 0 | 0 | 0 |
| 5 | 40 | 0 | 0,5 | 0 | 0 | 0 |
| 6 | 50 | 0 | 0 | 0,5 | 0 | 0 |
| 7 | 60 | 0 | 0 | 1 | 0 | 0 |
| 8 | 70 | 0 | 0 | 0,5 | 0 | 0 |
| 9 | 80 | 0 | 0 | 0 | 0,5 | 0 |
| 10 | 90 | 0 | 0 | 0 | 0,1 | 0 |
| 11 | 100 | 0 | 0 | 0 | 0,5 | 0 |
| 12 | 110 | 0 | 0 | 0 | 0 | 0,5 |
| 13 | 120 | 0 | 0 | 0 | 0 | 1 |

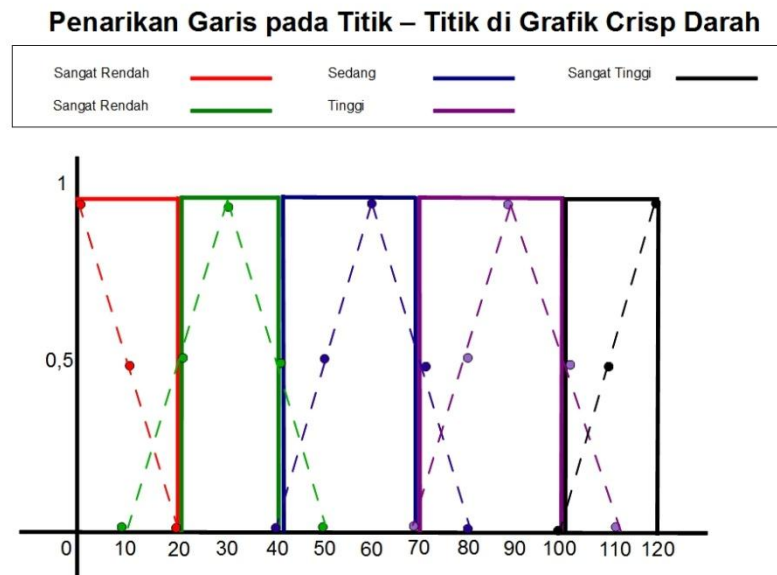
Kemudian titik – titik diatas beserta nilai hasil intuisinya, dipetakan kedalam grafik Boolean darah. Hasil pemetaannya dapat dilihat pada grafik

Penentuan Keanggotaan Titik di Grafik Crisp Darah



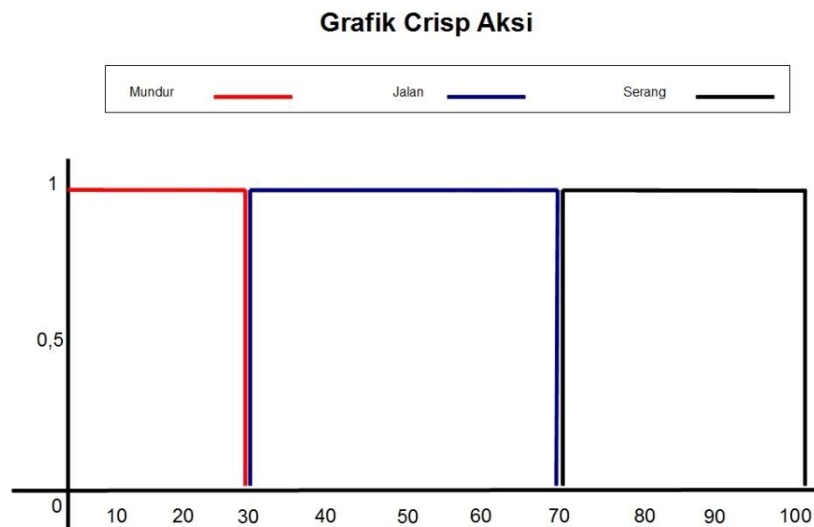
berikut:

Setelah memetakan titik – titik yang ditentukan ke grafik Boolean darah, sekarang tarik garis yang melewati titik – titik tersebut:



Gambar 3.7 Penarikan Garis pada Titik di *Grafik Crisp Darah*

Berikutnya akan ditentukan grafik keanggotaan *fuzzy* untuk variabel *fuzzy* aksi. Diketahui grafik logika Boolean atau *crisp* dari aksi berikut:



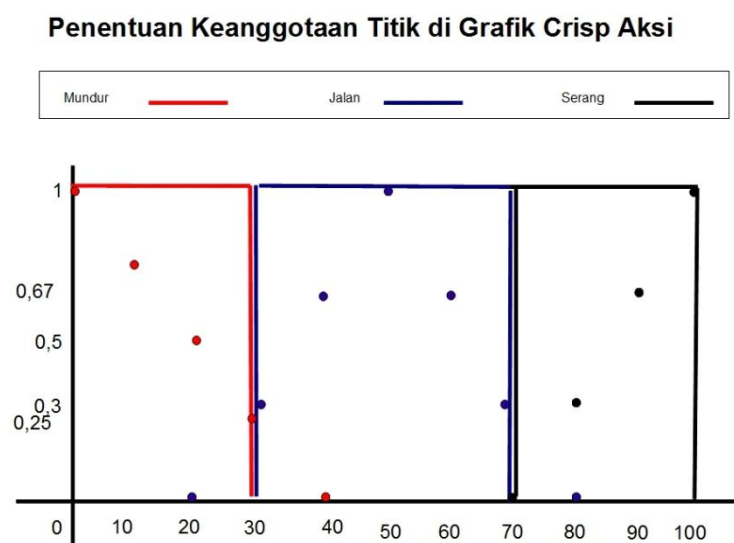
Setelah mengetahui **Gambar 3.8 Grafik Crisp Aksi** aksi, dapat ditentukan sejumlah titik secara acak yang terdapat pada rentang terendah sampai maksimum. Rentang yang terdapat pada grafik Boolean aksi adalah 0 – 100. Ditentukan titik – titik berikut : 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, dan 100. Titik – titik tersebut akan menentukan pembentukan grafik keanggotaan *fuzzy*.

Titik tersebut akan ditentukan nilai keanggotaannya di masing – masing himpunan grafik Boolean aksi. Nilai keanggotaannya terdiri dari rentang 0 hingga 1. Hasil penentuan nilai keanggotaan titik – titik diatas pada himpunan grafik Boolean aksi terdapat pada tabel berikut ini:

Tabel 3.3 Tabel Penentuan Keanggotaan Titik di Grafik Crisp Aksi

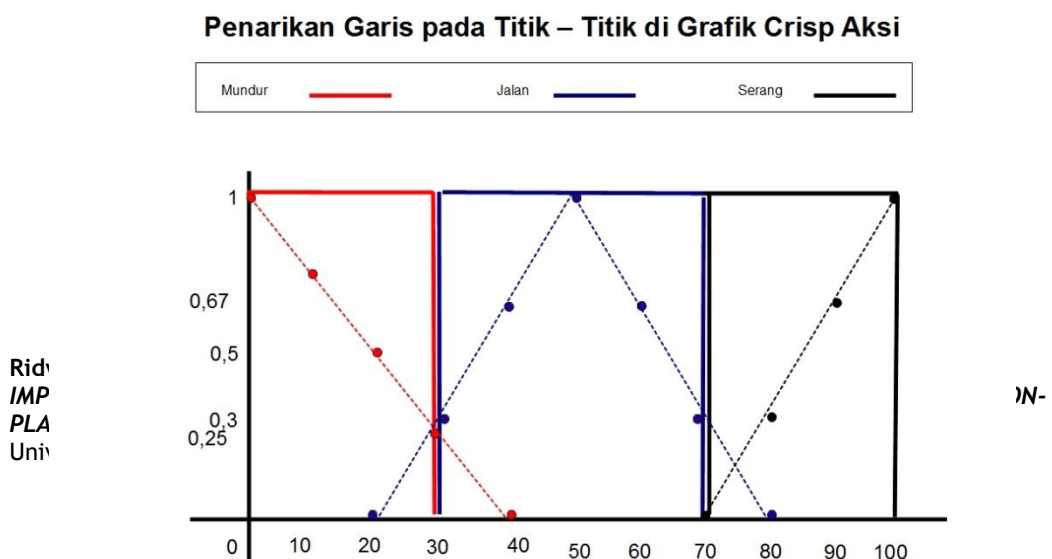
| No | Poin | Intuisi | | |
|----|------|---------|-------|--------|
| | | Mundur | Jalan | Serang |
| 1 | 0 | 1 | 0 | 0 |
| 2 | 10 | 0,75 | 0 | 0 |
| 3 | 20 | 0,5 | 0 | 0 |
| 4 | 30 | 0,25 | 0,3 | 0 |
| 5 | 40 | 0,2 | 0,67 | 0 |
| 6 | 50 | 0 | 1 | 0 |
| 7 | 60 | 0 | 0,67 | 0 |
| 8 | 70 | 0 | 0,3 | 0 |
| 9 | 80 | 0 | 0 | 0,3 |
| 10 | 90 | 0 | 0 | 0,67 |
| 11 | 100 | 0 | 0 | 1 |

Kemudian titik – titik diatas beserta nilai hasil intuisinya, dipetakan kedalam grafik Boolean aksi. Hasil pemetaannya dapat dilihat pada grafik berikut:



Gambar 3.9 Penentuan Keanggotaan Titik di *Grafik Crisp Aksi*

Setelah memetakan titik – titik yang ditentukan ke grafik Boolean aksi, sekarang tarik garis yang melewati titik – titik tersebut:



Dengan demikian berdasarkan gambar x, y, dan z, domain himpunan *fuzzy* dari variabel diatas ditentukan seperti berikut ini :

1. Variabel jarak : DEKAT = [0, 250], SEDANG = [200, 400], JAUH = [350, 500]
2. Variabel darah : SANGAT_RENDAH = [0, 20], RENDAH = [10, 50], SEDANG = [40, 80], TINGGI = [70, 110], SANGAT_TINGGI = [100, 120]
3. Variabel aksi : KEMBALI_KE_MARKAS = [0, 40], JALAN = [20, 80], SERANG = [70, 100]

yang akan di **Gambar 3.10 Penarikan Garis Pada Titik di Grafik Crisp Aksi** ingksi segitiga dipilih karena kemudahannya dalam menghitung keanggotaan sebuah nilai pada himpunan *fuzzy*. Untuk proses defuzzyfikasi, digunakan metode *centroid (Composite Moment)* atau nama lainnya adalah metode *center of gravity*.

Sebagai pembanding terhadap *fuzzy logic* yang akan dibangun. Dibawah ini terdapat *crisp logic* yang akan menjadi tolak ukur untuk menguji kinerja *fuzzy logic* yang akan dibangun. Berikut adalah kode program dari *crisp logic* yang akan dibangun:

Listing 3.1 Crisp Logic untuk Dibandingkan dengan Fuzzy Logic yang akan dibangun

```
def rulebasedrts(jarak=0, darah=0):
    print "jarak : ", jarak
    print "darah : ", darah
```

```
if jarak >=0 and jarak <=200:
    if darah >=0 and darah <= 20:
        print "fallback"
    elif darah > 20 and darah <= 40:
        print "fallback"
    elif darah > 40 and darah <= 70:
        print "walk"
    elif darah > 70 and darah <= 100:
        print "walk"
    elif darah > 100 and darah <= 120:
        print "attack"

elif jarak > 200 and jarak <=350:
    if darah >=0 and darah <= 20:
        print "fallback"
    elif darah > 20 and darah <= 40:
        print "walk"
    elif darah > 40 and darah <= 70:
        print "attack"
    elif darah > 70 and darah <= 100:
        print "attack"
    elif darah > 100 and darah <= 120:
        print "attack"

elif jarak > 350 and jarak <=500:
    if darah >=0 and darah <= 20:
        print "fallback"
    elif darah > 20 and darah <= 40:
        print "walk"
    elif darah > 40 and darah <= 70:
        print "attack"
    elif darah > 70 and darah <= 100:
        print "attack"
    elif darah > 100 and darah <= 120:
        print "attack"
```

3.3 Metode Penelitian

3.3.1 Proses Pengumpulan Data

Pada penelitian ini, penulis mengumpulkan data dan informasi yang akurat dan terkini yang dapat mendukung proses penelitian. Berikut adalah metode pengumpulan data yang penulis lakukan:

a. Studi Literatur

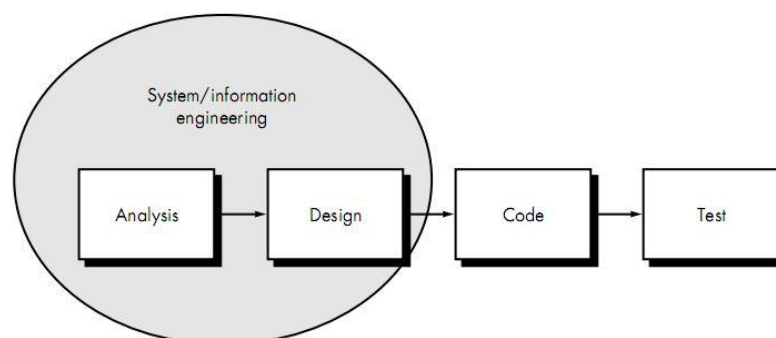
Studi literatur dilakukan untuk mempelajari metode *fuzzy logic*, *game development*, dan *game design* melalui literatur – literatur seperti jurnal, *textbook*, *tutorial*, karya ilmiah, *paper*, dan sumber ilmiah lainnya yang didapat dari internet dan perpustakaan.

b. Observasi

Pada tahap ini, pengumpulan data dilakukan dengan cara meneliti langsung terhadap topik permasalahan yang diambil guna melengkapi data – data yang diperlukan selama penelitian berlangsung.

3.3.2 Proses Pembangunan Perangkat Lunak

Di dalam proses pembangunan perangkat lunak digunakan model sekuensial linier (Pressman, Roger, 2002). Berikut adalah tahapan – tahapan dari rekayasa sistem dengan model sekuensial linier:



Ridwan F.
IMPLEMENTASI
PLAYER C

Universitas Pendidikan Indonesia | repositorv.uni.edu | perpustakaan.uni.edu

NT NON-

Gambar 3.11 Proses Rekayasa Model Sekuensial Linier (Pressman, Roger, 2002)

a. Analysis

Seluruh kebutuhan *software* harus sudah terkumpul di fase ini. Selain itu kegunaan dan batasan *software* pun harus didapatkan pada fase ini. Informasi yang digunakan untuk menentukan kebutuhan *software* ini biasanya didapat dari wawancara, survey, atau diskusi. Informasi yang didapat akan diubah menjadi sebuah *software requirement specification*. Pada tahap ini, untuk memodelkan sistem digunakan *Unified Modeling Language* (UML).

b. Design

Tahap ini dilakukan sebelum melakukan *coding*. Tahap ini bertujuan untuk memberikan gambaran apa yang harus dikerjakan pada saat *coding* nanti. Selain itu di fase ini kita menentukan *interface* dari perangkat lunak yang telah dirancang, kebutuhan *hardware* yang akan digunakan untuk jalannya sistem, serta kebutuhan data yang direpresentasikan dalam *file* dan *database*.

c. Coding

Dalam tahap ini mulai dilakukan tahap *coding*. Pembuatan *software* dipecah menjadi modul – modul khusus dan spesifik sesuai dengan kebutuhan pada fase sebelumnya. Kemudian setelah modul – modul tersebut beres akan diperiksa apakah sudah memenuhi kebutuhan dan fungsional yang diinginkan atau belum

d. Testing

Pada tahap ini modul – modul yang dibuat sudah digabung dan dilakukan pengujian untuk mengetahui apakah program yang dibuat telah sesuai desain atau masih terdapat ketidaksesuaian dengan spesifikasi yang telah ditentukan. Metode *blackbox* akan digunakan untuk pengujian sistem.

3.3.3 Pemodelan Sistem

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi *standard* dalam 33ip rose untuk visualisasi, merancang dan mendokumentasikan perangkat lunak. UML menawarkan standar untuk merancang sebuah model sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi perangkat lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun yang mendukung pemrograman berorientasi objek.

Untuk membuat suatu model, UML memiliki diagram grafis sebagai berikut:

1. *Use Case Diagram*
2. *Class Diagram*
3. *Statechart Diagram*
4. *Activity Diagram*
5. *Sequence Diagram*
6. *Collaboration Diagram*
7. *Component Diagram*
8. *Deployment Diagram*

UML memiliki beberapa notasi yang akan digunakan ketika melakukan pemodelan sistem. Berikut adalah tabel yang berisi notasi, deskripsi dan simbiol notasi yang digunakan dalamUML.

Selain itu akan disusun juga *Game Design Description* (GDD) yang bertujuan

untuk menambah gambaran dari *gameplay* yang akan diciptakan pada perangkat lunak yang akan dibangun.