BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

Analisis sistem merupakan proses penguraian konsep ke dalam bagian-bagian yang lebih sederhana. Pada analisis sistem akan mencakup berbagai macam analisis yang berhubungan dengan aplikasi yang akan di bangun seperti analisis masalah, analisis kecerdasan buatan yang digunakan, analisis kebutuhan fungsional dan non-fungsional

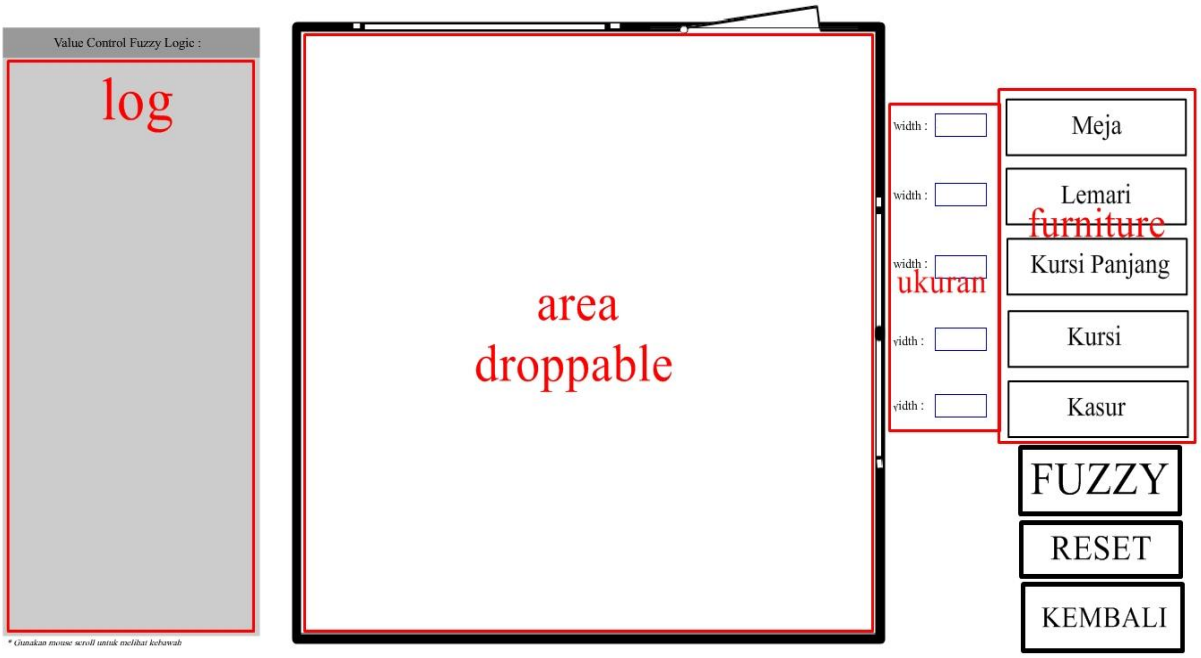
3.1.1. Analisis Masalah

Analisis masalah dalam penelitian ini adalah bagaimana menerapkan algoritma *fuzzy logic* sebagai penentuan tata letak *furniture* pada aplikasi tata letak denah ruangan. Analisis masalah sangat penting dalam suatu penelitian, karena pada tahapan ini merupakan salah satu proses mengidentifikasi masalah-masalah yang berkaitan dengan bagaimana suatu sistem akan melakukan proses peninjauan terhadap ruangan yang tersedia dan *furniture* berdasarkan algoritma *fuzzy logic*.

Adapun batasan yang terdapat dalam simulasi denah ini yaitu denah hanya berbentuk persegi empat yang dimiliki banyak ruangan, denahtersebut adalah ruangan kamar tidur yang berisi beberapa *furniture* seperti kasur, kursi, meja dan lemari baju yang menjadi standar penempatan disebuah ruang kamar tidur.

3.1.2. Analisis Simulasi

Dari aplikasi simulasi denah ini yaitu pengguna bertugas merencanakan dahulu tata letak *furniture* yang telah tersedia pada sebuah ruangan, pengguna *drag and drop furniture* pada area denah yang telah tersedia, pengguna dapat memindahkan serta memutar *furniture* sesuai keinginan. Atau pengguna dapat menggunakan fitur *'auto'* untuk mengaktifkan algoritma *fuzzy logic* yang ada.



Gambar 3.1 Aplikasi simulasi denah ruangan

1. Area *Droppable*

   Area *Droppable* area untuk menyimpan *furniture* yang diinginkan pengguna untuk di atur dan melihat hasil akhir.

2. *Log*

   Area untuk melihat detail *input* dan *output*an *furniture* pada algoritma fuzzy yang telah diterapkan di aplikasi

3. *Furniture*

   Area untuk memilih *furniture* yang ingin digunakan pada area *droppable*

4. Ukuran

   Inputan ukuran untuk setiap *furniture* yang ingin di pilih

Ada beberapa aturan menurut ahli yang diterapkan di simulasi denah ruangan ini yaitu pada area *Droppable* adalah sebagai berikut :

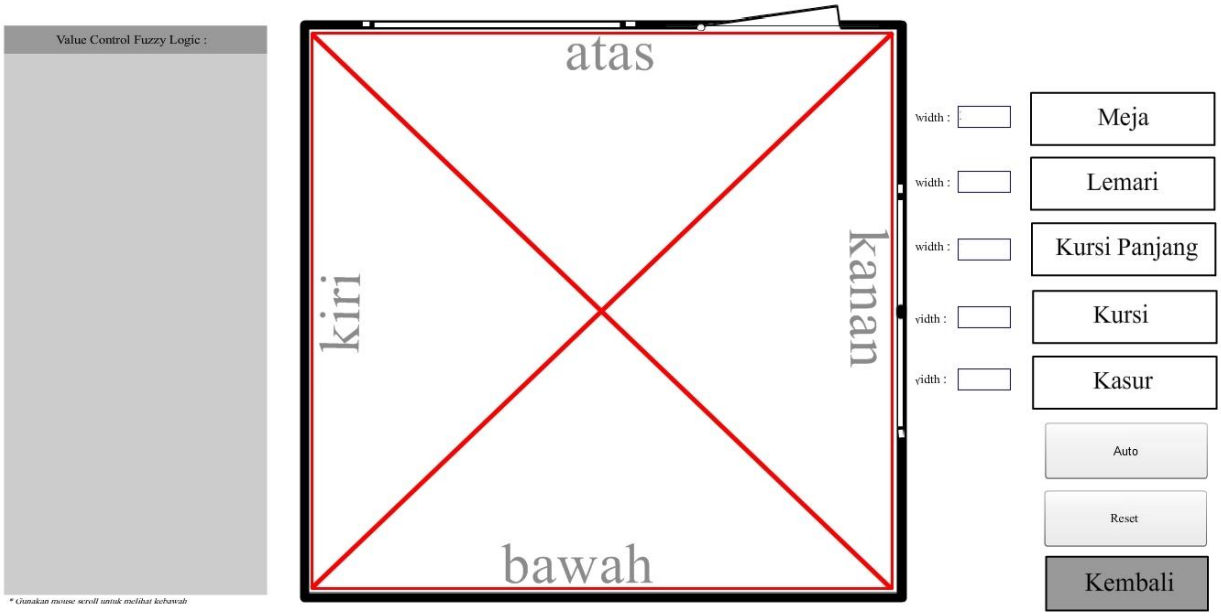
1. *Furniture* tidak boleh menempel pada tembok atau batas area yang terdapat jendela, *furniture* diberi jarak agar tidak menempel pada jendela yang bertujuan untuk memberikan ruang untuk udara dari luar
2. *Furniture* tidak boleh menumpuk satu sama lain, *furniture* tidak diperbolehkan menumpuk karena pada pada kenyataanya *furniture* tidak mungkin untuk di tumpuk keatas

3. *Furniture* diberi jarak satu sama lain, furniture diberi jarak agar karena ada ruang untuk penghuni tersebut untuk berpindah-pindah atau melangkah dan bergerak

3.1.3. Analisis Alur Simulasi

Aplikasi simulasi denah adalah aplikasi simulasi yang menggunakan grafik 2 dimensi. Tujuan aplikasi simulasi ini adalah menempatkan *furniture* yang diinginkan pada sebuah ruangan. Pengguna merancang terlebih dahulu tata letak sebelum mengimplementasikannya pada keadaan yang nyata.

Aplikasi simulasi denah ini mengimplementasikan algoritma *fuzzy logic* untuk memberikan keputusan atau rekomendasi tata letak *furniture* pada denah ruangan. Berikut pembagian area denah yang dibagi menjadi empat bagian, atas, kanan, bawah, kiri. :



Gambar 3.2 Area denah pada aplikasi

3.2. Analisis Masukan

Analis masukan yang dilakukan dalam penelitian ini adalah menganalisis variable dalam algoritma *fuzzy logic* pada aplikasi simulasi denah, variable tersebut antara lain Jumlah *Furniture*, Relasi *Furniture* dan Ukuran *Furniture*.

1. Jumlah *Furniture*

   Variable ini terdapat pada perhitungan internal berapa banyak *furniture* sejenis yang berada dalam area ruangan, variable jumlah berawal dari nilai 1, dan akan bertambah setiap kali *furniture* yang sejenis ditambahkan dalam ruangan.

   Dengan range nilai variable jumlah diantara 1 - 10 dengan penjelasan sebagai berikut:

   a. Sedikit: memiliki nilai 1 - 3

   b. Sedang: memiliki nilai 2 - 4

   c. banyak: memiliki nilai 4 – 10

2. Relasi Furniture

   Variable ini terdapat pada perhitungan berapa banyak *furniture* sejenis yang berada dalam area ruangan yang merupakan relasi dari *furniture* tersebut, relasi disini adalah kebiasaan suatu *furniture* berada didekat *furniture* lain, misalnya meja dengan kursi, variable relasi berawal dari nilai 0, dan akan bertambah setiap kali furniture yang sejenis ditambahkan dalam ruangan.

   Dengan range nilai variable Relasi diantara 0 - 10 dengan penjelasan sebagai berikut:

   a. Sedikit: memiliki nilai 0 - 3

   b. Sedang: memiliki nilai 1 - 4

   c. banyak: memiliki nilai 3 – 10

3. Ukuran Furniture

   Variable ini terdapat pada perhitungan internal berapa ukuran *furniture* ada dalam area ruangan.

   Dengan range nilai variable ukuran diantara 50 - 400 dengan penjelasan sebagai berikut:

   a. Kecil : memiliki nilai 50 - 200

   b. Sedang : memiliki nilai 150 -300

   c. Besar : memiliki nilai 250 - 400

3.3. Analisis Algoritma Logika Fuzzy

Analisis algoritma yang dilakukan dalam penelitian ini adalah menganalisis bagaimana cara kerja algoritma logika fuzzy pada penataan *furniture* dalam aplikasi simulasi denah ini. Algoritma logika *fuzzy* ini memiliki 3 variabel yaitu Jumlah *Furniture*, Relasi *Furniture* dan Ukuran *Furniture*. Algoritma logika *fuzzy* ini menerapkan 4 proses tahapan yaitu fuzzyfikasi, pembuatan aturan dasar, penalaran dan defuzzifikasi. Berikut adalah ilustrasi logika *fuzzy* yang diperoleh dari 3 variable masukan yang telah ditentukan pada tabel input dan output algoritma logika fuzzy berikut.

Tabel 3.1 Input dan Output Algoritma Logika Fuzzy

NO	Jumlah Furniture	Relasi Furniture	Ukuran Furniture	Hasil
1	Banyak	Banyak	Besar	Atas
2	Banyak	Sedang	Sedang	Bawah
3	Banyak	Sedikit	Kecil	Atas
4	Sedang	Banyak	Besar	Kiri
5	Sedang	Sedang	Sedang	Kanan
6	Sedang	Sedikit	Kecil	Bawah
7	Sedikit	Sedang	Sedang	Kiri
8	Sedikit	Sedikit	Kecil	Bawah
9	Sedikit	Banyak	Besar	Kanan

Pada tabel input dan output digunakan untuk menentukan keputusan yang dilakukan oleh algoritma logika *fuzzy*. Berikut penjelasan dari masing – masing *input* dan *output* tersebut adalah:

Rule Logika *Fuzzy* pada aplikasi simulasi denah ini adalah :

1. **IF** Jumlah IS Banyak **AND** Relasi IS Banyak **AND** Ukuran IS Besar **THEN** Lokasi IS Atas
2. **IF** Jumlah IS Banyak **AND** Relasi IS Sedang **AND** Ukuran IS Sedang **THEN** Lokasi IS Bawah

3. **IF** Jumlah IS Banyak **AND** Relasi IS Sedikit **AND** Ukuran IS Kecil **THEN** Lokasi IS Atas
4. **IF** Jumlah IS Sedang **AND** Relasi IS Banyak **AND** Ukuran IS Besar **THEN** Lokasi IS Kiri
5. **IF** Jumlah IS Sedang **AND** Relasi IS Sedang **AND** Ukuran IS Sedang **THEN** Lokasi IS Kanan
6. **IF** Jumlah IS Sedang **AND** Relasi IS Sedikit **AND** Ukuran IS Kecil **THEN** Lokasi IS Bawah
7. **IF** Jumlah IS Sedikit **AND** Relasi IS Sedang **AND** Ukuran IS Sedang **THEN** Lokasi IS Kiri
8. **IF** Jumlah IS Sedikit **AND** Relasi IS Sedikit **AND** Ukuran IS Kecil **THEN** Lokasi IS Bawah
9. **IF** Jumlah IS Sedikit **AND** Relasi IS Banyak **AND** Ukuran IS Besar **THEN** Lokasi IS Kanan

3.3.1. Tahap Fuzzyfikasi

1. Variabel Jumlah *Furniture*

Variabel jumlah *furniture* yang ada terdiri dari 3 himpunan *fuzzy* yaitu Sedikit dengan *range* (1 – 3), Sedang dengan *range* (2 – 4), Banyak dengan *range* (4 – 10).

Penjelasan jumlah *furniture* yang ada adalah dari banyaknya *furniture* yang ada pada denah ruangan yang ada.

Contoh kasus

Misal didapatkan jumlah *furniture* yang ada pada ruangan adalah 4 buah *furniture*, maka nilai 4 akan di konversi kedalam nilai *fuzzy*, dimana nilai 4 berada pada nilai linguistik sedang dan banyak kemudian dihitung menggunakan rumus fungsi Linear Naik.

Fungsi Keanggotaan :

$$\mu\text{Sedikit}[x] = \begin{cases} 0 & ;x \leq 1 \\ \frac{x-1}{3-1} & ;1 < x \leq 3 \\ 1 & ;x \geq 3 \end{cases} \quad (3.1)$$

$$\mu\text{Sedang}[x] = \begin{cases} 0 & ;x \leq 2 \\ \frac{x-2}{4-2} & ;2 < x \leq 4 \\ 1 & ;x \geq 4 \end{cases} \quad (3.2)$$

$$\mu\text{Banyak}[x] = \begin{cases} 0 & ;x \leq 4 \\ \frac{x-4}{10-4} & ;4 < x \leq 10 \\ 1 & ;x \geq 10 \end{cases} \quad (3.3)$$

derajat keanggotaan dari jumah *furniture* 4 adalah :

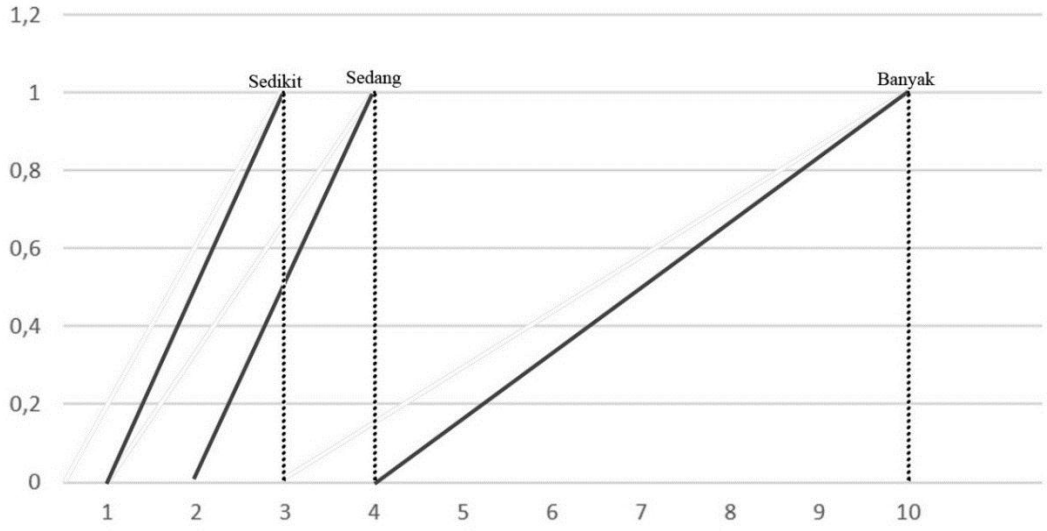
$\mu Sedikit[4] = 1 \; ; 4 \geq 3$

$\mu Sedang[4] = \frac{4-2}{4-2} ; 1 < 4 \leq 4$

$= \frac{2}{2} = 1$

$\mu Banyak[4] = \frac{4-4}{10-4} ; 4 < 4 \leq 10$

$= \frac{1}{6} = 0{,}1667$



Gambar 3.3 Grafik Kasus Fungsi Keanggotaan Variabel Jumlah *Furniture*

2. Variabel Relasi *Furniture*

Variabel Relasi *Furniture* yang ada terdiri dari 3 himpunan *fuzzy* yaitu Sedikit dengan *range* (0 - 3), Sedang dengan *range* (1 – 4), Banyak dengan *range* (3 – 10).

Penjelasan Relasi *Furniture* yang ada adalah dari banyaknya *furniture* yang kebiasaan suatu *furniture* berada didekat furniture lain

Contoh kasus

Misal didapatkan Relasi *Furniture* yang ada pada ruangan adalah 3 buah *furniture*, maka nilai 3 akan di konversi kedalam nilai *fuzzy*, dimana nilai 3 berada pada nilai linguistik sedikit, sedang dan banyak kemudian dihitung menggunakan rumus fungsi Linear Naik.

Fungsi Keanggotaan :

$$\mu\text{Sedikit}[x] = \begin{cases} 0 & ;x \leq 0 \\ \frac{x-0}{3-0} & ;0 < x \leq 3 \\ 1 & ;x \geq 3 \end{cases} \quad (3.4)$$

$$\mu\text{Sedang}[x] = \begin{cases} 0 & ;x \leq 1 \\ \frac{x-1}{4-1} & ;1 < x \leq 4 \\ 1 & ;x \geq 4 \end{cases} \quad (3.5)$$

$$\mu\text{Banyak}[x] = \begin{cases} 0 & ;x \leq 3 \\ \frac{x-3}{10-3} & ;3 < x \leq 10 \\ 1 & ;x \geq 10 \end{cases} \quad (3.6)$$

derajat keanggotaan dari jumah furniture 3 adalah :

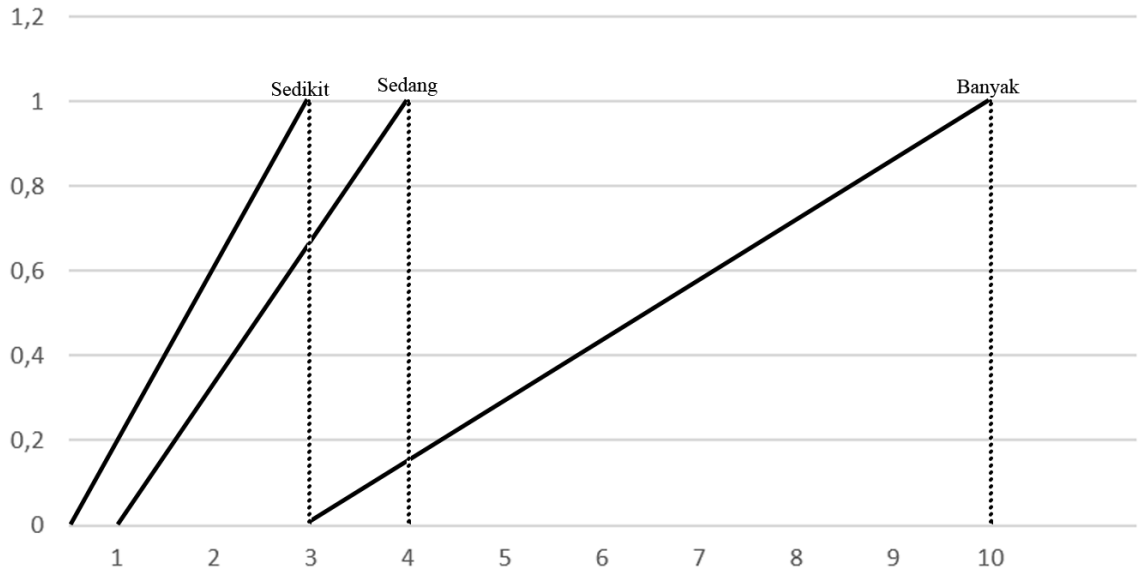
$$\mu Sedikit[3] = 1 \;;3 \geq 3$$

$$\mu Sedang[3] = \frac{3-1}{4-1};\; 1 < 3 \leq 4$$

$$= \frac{2}{3} = 0{,}667$$

$$\mu Banyak[3] = \frac{3-3}{10-3};\; 3 < 3 \leq 10$$

$$= \frac{0}{7} = 0$$



Gambar 3.4 Grafik Kasus Fungsi Keanggotaan Variabel Relasi Furniture

3. Variabel Ukuran *Furniture*

Variabel Relasi *Furniture* yang ada terdiri dari 3 himpunan *fuzzy* yaitu Kecil dengan *range* (50 - 200), Sedang dengan *range* (150 – 300), Besar dengan *range* (250 - 400).

Penjelasan Ukuran *Furniture* yang ada adalah dari besarnya *furniture* yang ada pada suatu ruangan tertentu.

Contoh kasus

Misal didapatkan Ukuran *Furniture* yang ada pada ruangan adalah 250, maka nilai 150 akan di konversi kedalam nilai *fuzzy*, dimana nilai 250 berada pada nilai linguistik sedang dan besar kemudian dihitung menggunakan rumus fungsi Linear Naik.

Fungsi Keanggotaan :

$$\mu\text{Kecil}[x] = \begin{cases} 0 & ;x \leq 10 \\ \frac{x-50}{200-50} & ;50 < x \leq 200 \\ 1 & ;x \geq 200 \end{cases} \quad (3.4)$$

$$\mu\text{Sedang}[x] = \begin{cases} 0 & ;x \leq 150 \\ \frac{x-150}{300-150} & ;150 < x \leq 300 \\ 1 & ;x \geq 300 \end{cases} \quad (3.5)$$

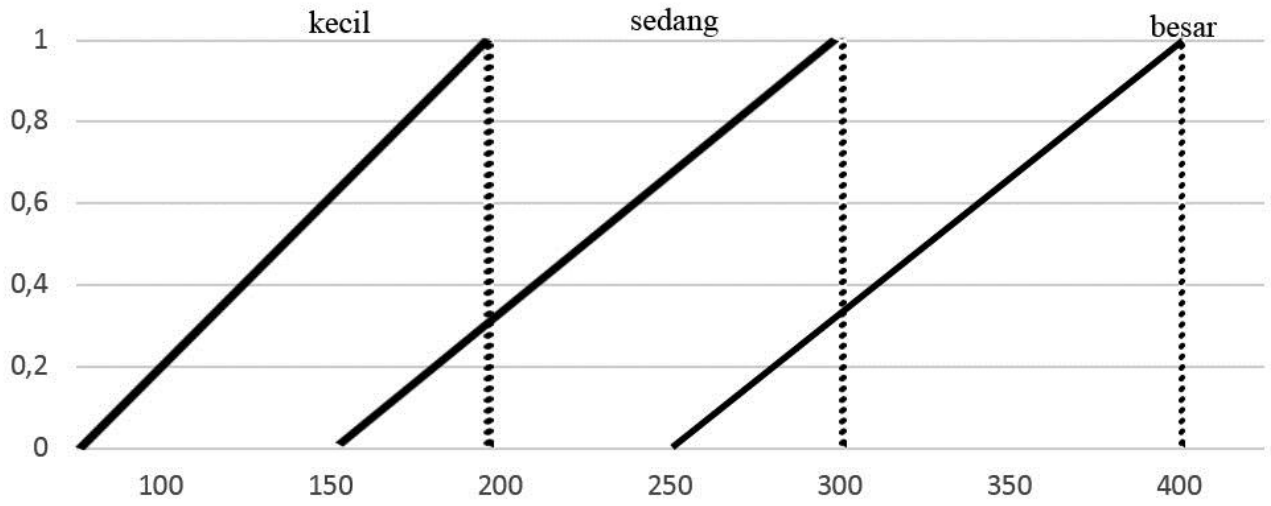
$$\mu\text{Besar}[x] = \begin{cases} 0 & ;x \leq 250 \\ \frac{x-250}{400-250} & ;250 < x \leq 400 \\ 1 & ;x \geq 400 \end{cases} \quad (3.6)$$

derajat keanggotaan dari jumah *furniture* 150 adalah :

$\mu Kecil[250] = 1\ ;250 \geq 150$

$\mu Sedang[250] = 1\ ;250 \geq 250$

$\mu Besar\ [250] = 0\ ;150 \leq 250$



Gambar 3.5 Grafik Kasus Fungsi Keanggotaan Variabel Ukuran Furniture

3.3.2. Tahap Pembentukan *Rule Base*

Dalam hal ini, aturan dasar atau *rule base* dibentuk sesuai dengan yang memenuhi kondisi soal.

Tabel 3.2 Tahap Pembentukan *Rule Base*

NO	IF	Jumlah Furniture	AND	Relasi Furniture	AND	Ukuran Furniture	THEN	Hasil
1	IF	Sedikit	AND	Sedang	AND	Sedang	THEN	Kiri
2	IF	Sedang	AND	Sedikit	AND	Kecil	THEN	Bawah
3	IF	Banyak	AND	Sedang	AND	Sedang	THEN	Bawah

3.3.3. Tahap Mesin Inferensi

[R1] IF JumlahFurniture Sedikit and RelasiFurniture Sedang and UkuranFurniture Sedang Then Hasil Kiri

α- $pred_1$ = µSedikit n Sedang n Sedang

= MIN (µSedikit[4]; µSedang[3] µSedang[250])

= MIN (1 ; 0,667 ; 1)

= 0,667

[R2] IF JumlahFurniture Sedang and RelasiFurniture Sedikit and UkuranFurniture Kecil Then Hasil Bawah

α- $pred_1$ = µSedang n Sedikit n Kecil

= MIN (µSedang[4]; µSedikit[3] µKecil[250])

= MIN (1 ; 1 ; 1)

= 1

[R3] IF JumlahFurniture Banyak and RelasiFurniture Sedang and UkuranFurniture Sedang Then Hasil Bawah

α- $pred_1$ = µBanyak n Sedang n Sedang

= MIN (µSedikit[4]; µSedang[3] µSedang[250])

= MIN (0,1667 ; 0,667 ; 1)

= 0,1667

3.3.4. Tahap Defuzzyfikasi

$$\boldsymbol{z = \frac{\sum aiZi}{ai}} \qquad (3.7)$$

αi = Nilai minimal dari derajat keanggotaan

Zi = Nilai domain dari variabel linguistik

z= Nilai defuzzyfikasi

Nilai Deffuzyfikasi

$$z = \frac{0{,}667 * 4 + 0{,}667 * 3 + 1 * 250}{0{,}667 + 0{,}667 + 0.667}$$

$$= \frac{2{,}668 + 2{,}001 + 250}{2{,}001}$$

= 127,271

Kesimpulan

Nilai yang di dapat dari proses *defuzzyfikasi* tersebut selanjutnya digunakan untuk penentuan keputusan dengan aturan sebagai berikut :

Tabel 3.3 Aturan nilai keputusan

No	Hasil	Nilai
1	Bawah	0 – 190
2	Atas	191-299
3	Kanan	300-309
4	Kiri	310-319

Maka hasil nilai dari hasil 127,271 adalah di bawah

3.4. Analisis Kebutuhan Non Fungsional

Analisis non-fungsional merupakan analisis yang dibutuhka untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan sistem yang akan berjalan sampai dengan algoritma tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran sehingga menghasilkan suatu keluaran yang diinginkan. Pada analisis kebutuhan sistem non fungsional ini dijelaskan analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras dan analisis pengguna.

3.4.1. Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk implementasi algoritma pada aplikasi simulasi denah ini adalah :

1. Sistem Operasi *Microsoft Windows 7*
2. Adobe Flash CS5

3.4.2. Analisis Kebutuhan Perangkat Keras

Komputer terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi-instruksi kepada perangkat keras untuk melakukan tugas tertentu, sehingga dapat menjalankan suatu sistem di dalamnya.

Perangkat keras minimum bagi pengguna untuk dapat menggunakan aplikasi simulasi denah ini dengan baik adalah sebagai berikut :

1. *Processor Pentium V 1,5* Ghz
2. RAM 1GB
3. Keyboard standard
4. Mouse standard
5. Memori 1GB

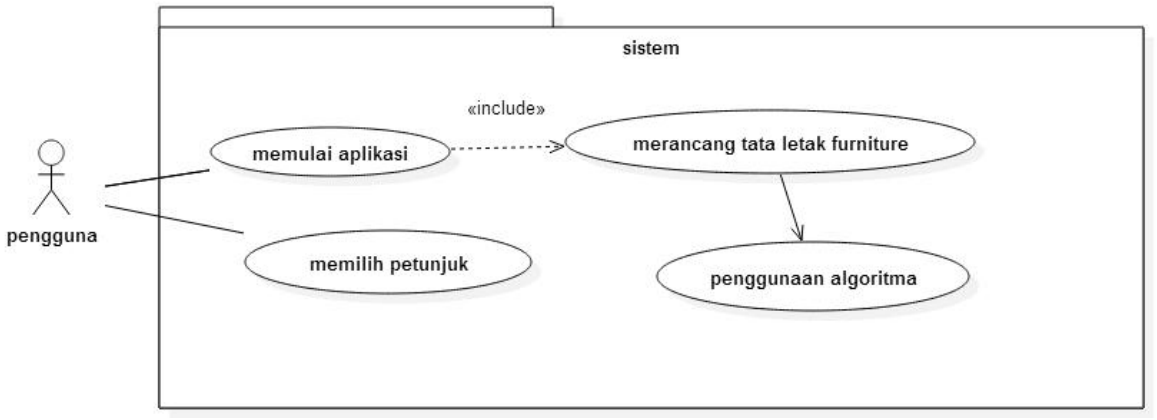
3.5. Analisis Kebutuhan Fungsional

Kebutuhan fungsional meliputi beberapa diagram UML, diantaranya: diagram *use case*, diagram *activity*, diagram *class* dan diagram *sequence*. Diagram tersebut dipergunakan sebagai desain rancangan untuk mempermudah proses pengembangan aplikasi khususnya untuk aplikasi yang dikembangkan menggunakan pemrograman berorientasi objek.

3.5.1. Diagram Use case

Use Case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan di buat [8].

Berikut ini adalah perancangan proses-proses yang terddapat pada aplikasi simulasi denah, yang digambarkan dengan *Use Case Diagram* seperti berikut :



Gambar 3.6 Use Case Diagram

3.5.1.1. Definisi Actor

Definisi *Actor* berfungsi untuk menjelaskan *Actor* yang terdapat pada *Use Case Diagram*. Definisi *Actor* diterangkan pada table berikut

Table 3.4 Definisi *Actor*

No	*Actor*	Deskripsi
1	Pengguna	Orang yang menggunakan aplikasi

3.5.1.2. Definisi Use Case

Definisi *Use Case* berfungsi untuk mejelaskan fungsi *Use Case* yang terdapat pada *Use Case Diagram*. Definisi *Use Case* diterangkan pada table berikut

Table 3.5 Definisi *Use Case*

No	*Use Case*	Deskripsi
1	Memulai Aplikasi	Proses untuk memulai aplikasi simulasi
2	Merancang Tata Letak	Proses pengguna untuk merancang *furniture*
3	Output	Proses menampilkan hasil perancangan menurut algoritma

3.5.2. Skenario *Use Case*

Skenario setiap bagian pada *use case* menunjukkan proses apa yang terjadi pada setiap bagian didalam *use case* tersebut, dimana pengguna memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh sistem kepada

pengguna setelah pengguna memberikan perintah pada setiap bagian - bagian *use case*.

Tabel 3.6 Skenario *Use Case* Memulai Aplikasi

Identifikasi	
No	1
Nama	Memulai aplikasi
Tujuan	Untuk memulai aplikasi
Deskripsi	Proses untuk memulai aplikasi
Aktor	pengguna
Skenario Utama	
Kondisi awal	Aktor berada di dalam permainan
Aksi Aktor	**Reaksi Sistem**
1. Memulai permainan	
	2. Menampilkan layar utama
3. Memilih menu utama	
	4. Menampilkan layar permainan
Kondisi akhir	Aplikasi simulasi di mulai

Tabel 3.7 Skenario *Use Case* Merancang Tata Letak

Identifikasi	
No	2
Nama	Merancang Tata Letak
Tujuan	Untuk Merancang Tata Letak pada Aplikasi
Deskripsi	Proses untuk merancang tata letak furniture pada denah
Aktor	pengguna
Skenario Utama	
Kondisi awal	Aktor berada di dalam permainan

Aksi Aktor	Reaksi Sistem
1. Klik *furniture* dalam aplikasi	
2. *Drag and Drop furniture* pada denah yang tersedia di aplikasi	
	3. Menampilkan denah yang telah di rancang
Kondisi akhir	Denah terancang sesuai keinginan pengguna

Tabel 3.8 Skenario *Use Case* Penggunaan Algoritma

Identifikasi	
No	3
Nama	Penggunaan Algoritma
Tujuan	Pengimplementasian Algoritma
Deskripsi	Proses untuk mengimplementasikan algoritma fuzzy yang telah di terapkan
Aktor	pengguna
Skenario Utama	
Kondisi awal	Aktor berada di dalam permainan
Aksi Aktor	**Reaksi Sistem**
1. Klik *furniture* dalam aplikasi	
2. *Drag and Drop furniture* pada ruangan yang tersedia di aplikasi	
3. Klik tombol *fuzzy* dalam aplikasi	4. Sistem memproses algoritma *fuzzy pada aplikasi*
	5. Sistem menghitung jumlah *furniture*
	6. Sistem menghitung relasi *furniture*
	7. Sistem menghitung ukuran *furniture*

8. Melihat hasil dari algoritma	9. Sistem menampilkan hasil algoritma *fuzzy logic* pada area *droppable*
Kondisi akhir	Denah terancang sesuai keinginan pengguna

Tabel 3.9 Skenario *Use Case* Memilih Petunjuk

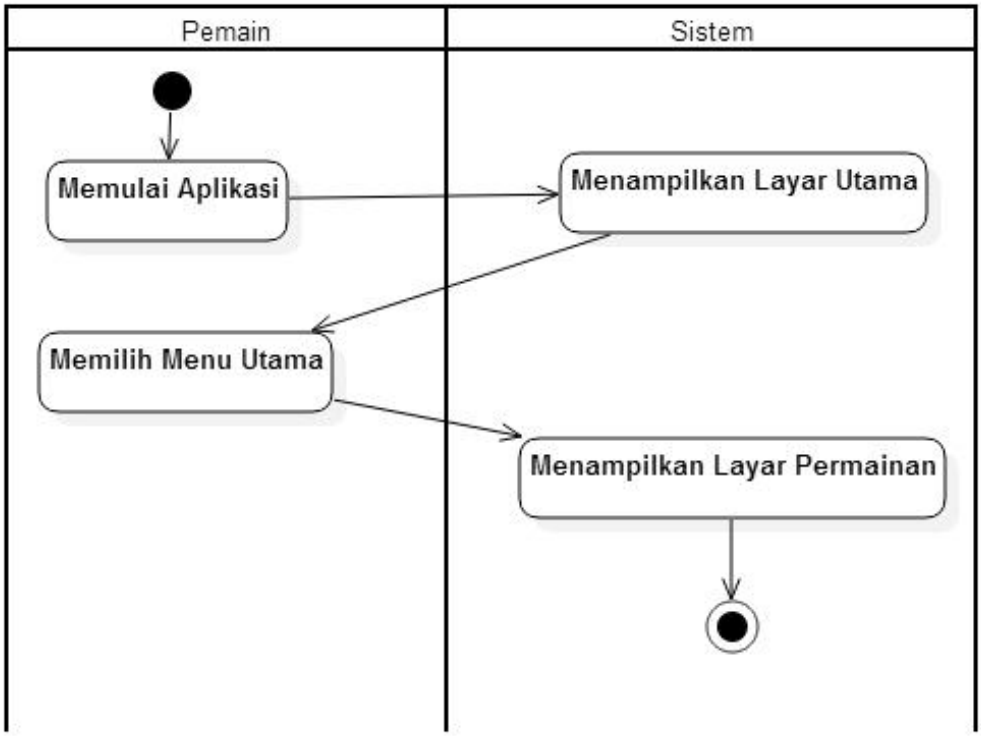
Identifikasi	
No	3
Nama	Memilih Petunjuk
Tujuan	Untuk melihat petunjuk aplikasi
Deskripsi	Proses untuk melihat petunjuk aplikasi
Aktor	Pengguna
Skenario Utama	
Kondisi awal	Aktor berada di menu utama
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu petunjuk	
	2. Menampilkan petunjuk penggunaan
3. Menutup tampilan petunjuk	
	4. Menampilkan menu utama
	5. Menampilkan menu perancangan tata letak
Kondisi akhir	Pengguna berada di menu petunjuk

3.6. Activity Diagram

Activity diagram digunakan untuk menggambarkan alur rangkaian proses yang terjadi di dalam sistem aplikasi serta menjelaskan bagaimana sebuah *use case* terjadi atau dipicu dari *use case* yang lain

1. Activity Diagram Memulai Aplikasi

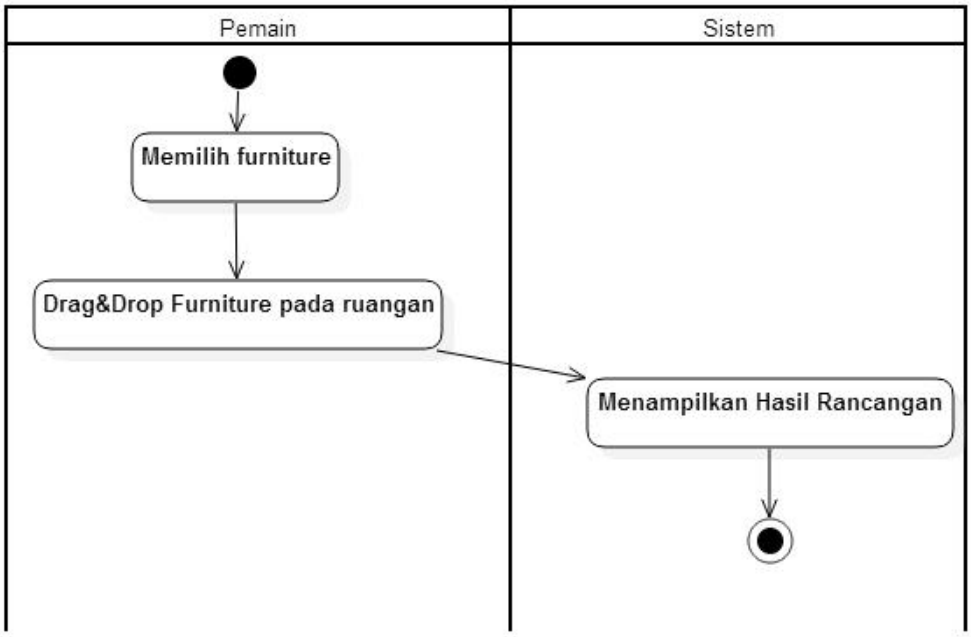
   Berikut adalah proses untuk memulai aplikasi simulasi denah yang digambarkan dengan diagram aktivitas.



Gambar 3.7 Activity Diagram Proses Memulai Aplikasi

2. Activity Diagram Merancang Tata Letak

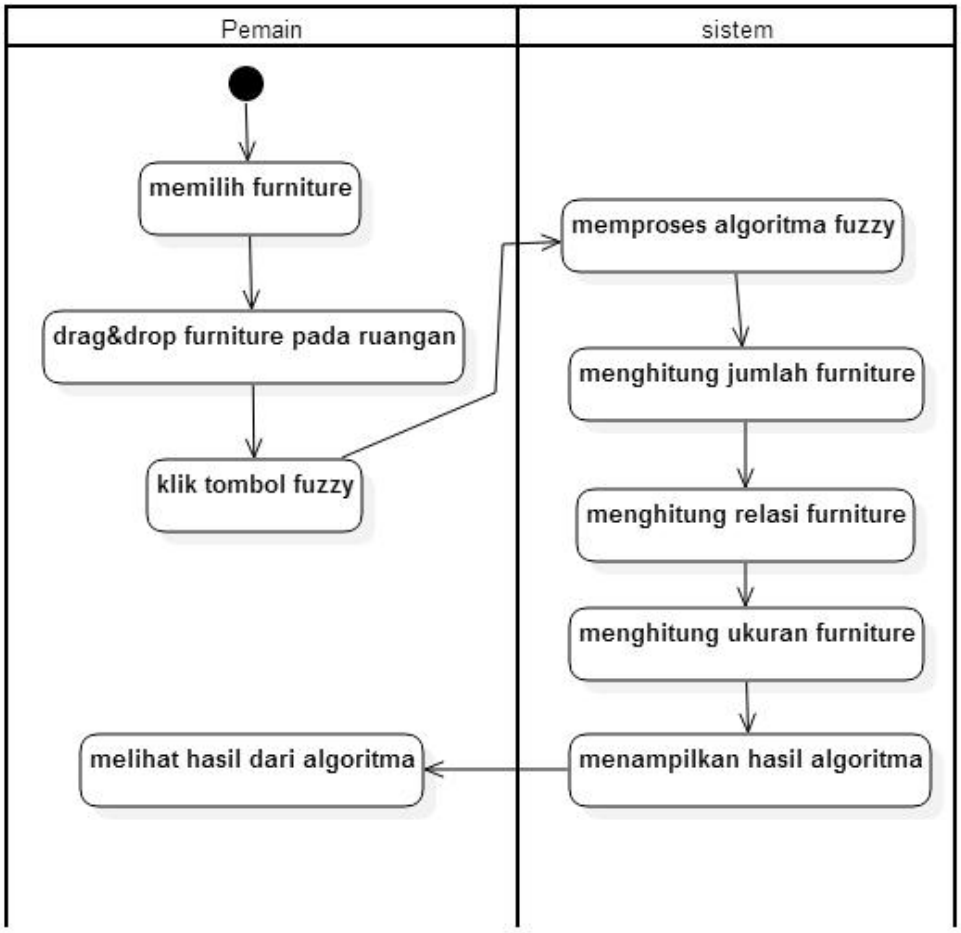
   Berikut adalah proses untuk merancang tata letak pada aplikasi simulasi denah yang digambarkan dengan diagram aktivitas.



Gambar 3.8 Activity Diagram Proses Merancang Tata Letak

3. Activity Diagram Penggunaan algoritma

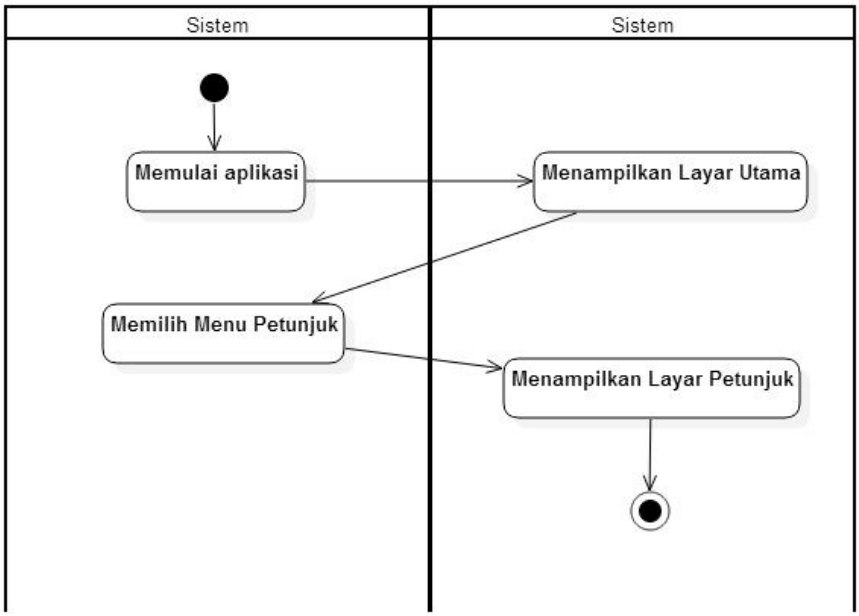
   Berikut adalah proses untuk menggunakan algoritma pada aplikasi simulasi denah yang digambarkan dengan diagram aktivitas.



Gambar 3.9 Activity Diagram Proses Merancang Tata Letak

4. Activity Diagram Menu Petunjuk

   Berikut adalah proses untuk menampilkan layar petunjuk penggunaan aplikasi yang digambarkan dengan diagram aktivitas.

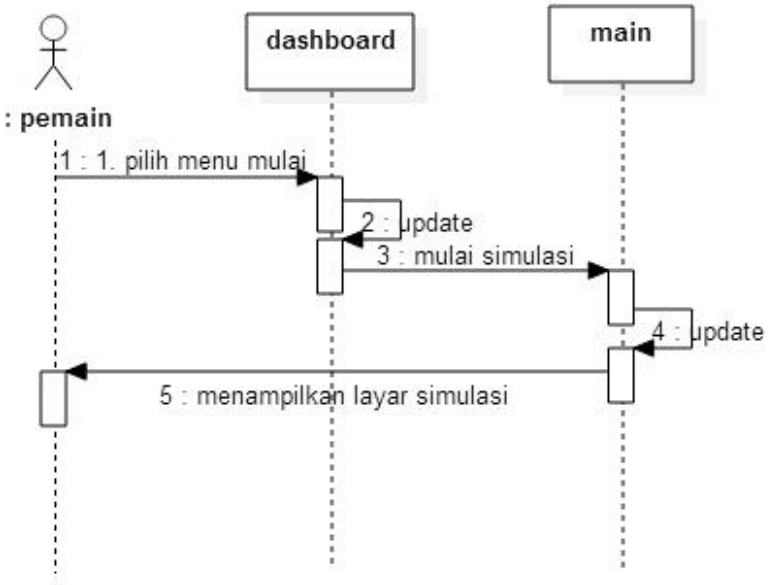


Gambar 3.10 Activity Diagram Proses Menampilkan Layar Petunjuk Penggunaan

3.7. Sequence Diagram

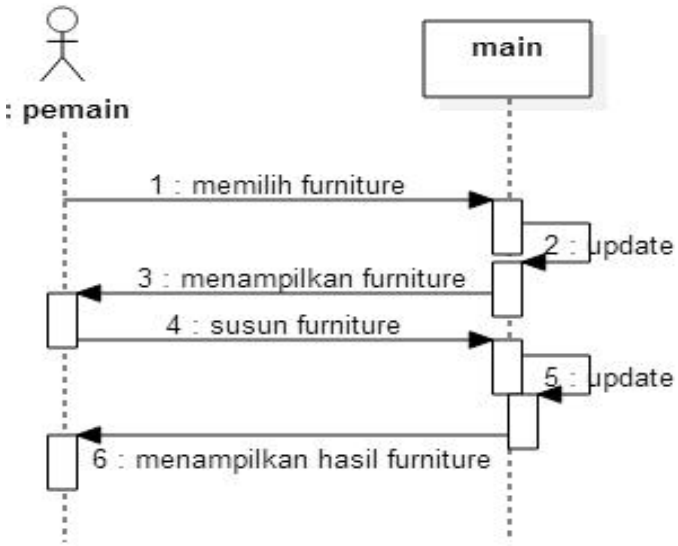
Diagram *sequence* menggambarkan urutan aktifitas yang terjadi di dalam sistem. Diagram ini menunjukkan pengiriman pesan-pesan yang melewati objek yang terlibat di dalam sistem serta tahapan langkah yang harus dilalui untuk mencapai *use case* tertentu:

3.7.1. Diagram *Sequence Memulai Aplikasi*



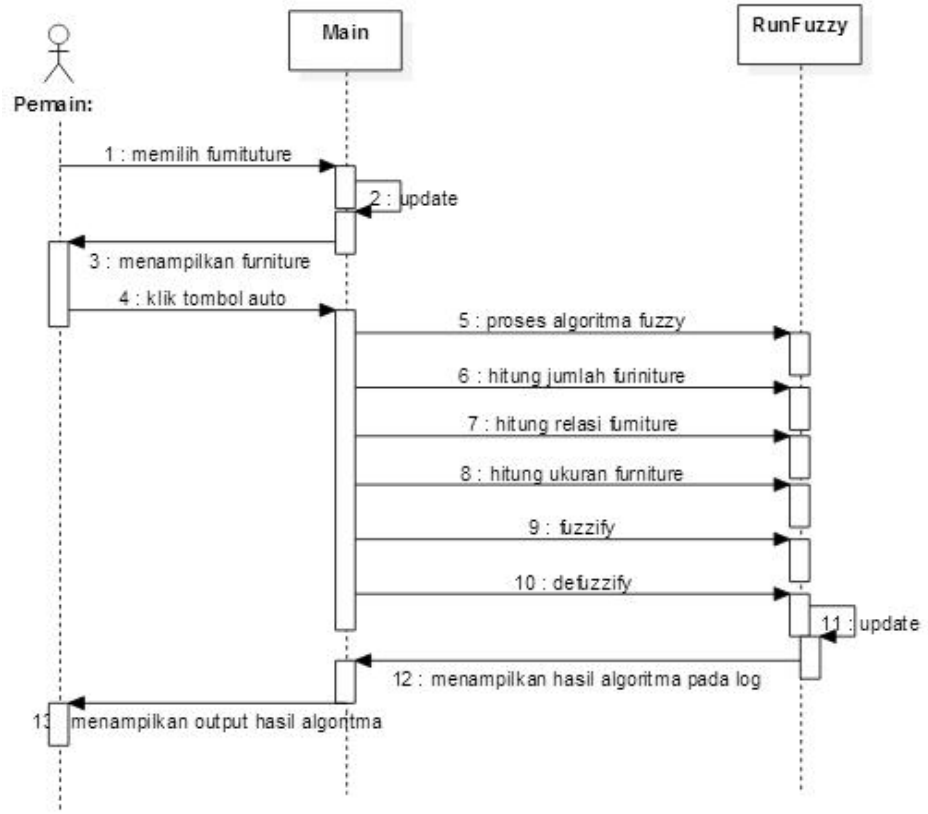
Gambar 3.11 Diagram *Sequence* Memulai Aplikasi

3.7.2. Diagram *Sequence* Merancang Tata Letak



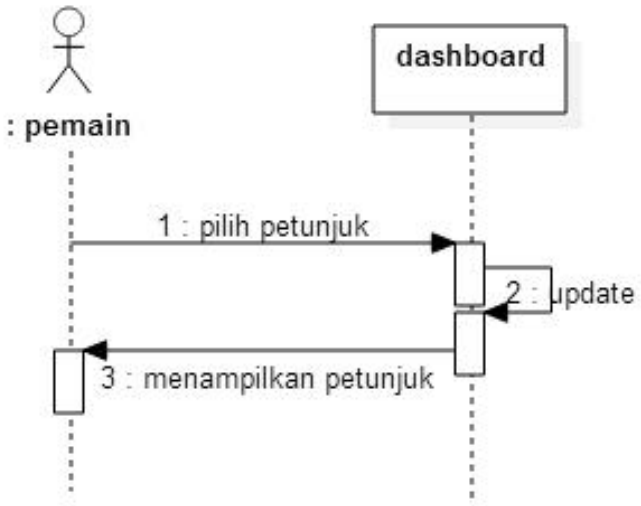
Gambar 3.12 Diagram *Sequence* Merancang Tata Letak

3.7.3. Diagram *Sequence* Penggunaan Algoritma



Gambar 3.13 Diagram *Sequence* Penggunaan Algoritma

3.7.4. Diagram *Sequence* Menu Petunjuk

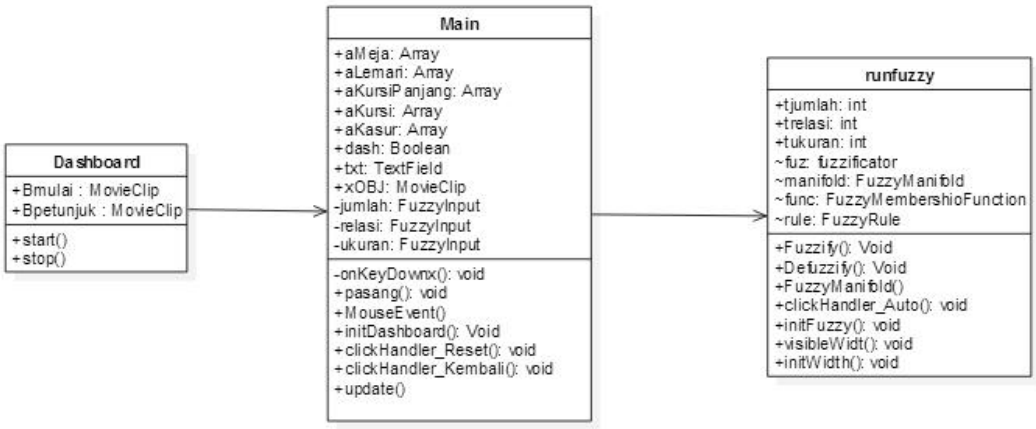


Gambar 3.14 Diagram *Sequence* Menu Petunjuk

3.8. Class Diagram

Diagram kelas atau atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem [7].

Berikut ini adalah perancangan struktur sistem yang terdapat pada aplikasi pengaturan tata letak furniture dalam denah , yang digambarkan dengan *class* diagram yang dapat dilihat pada gambar berikut:



Gambar 3.15 Class diagram

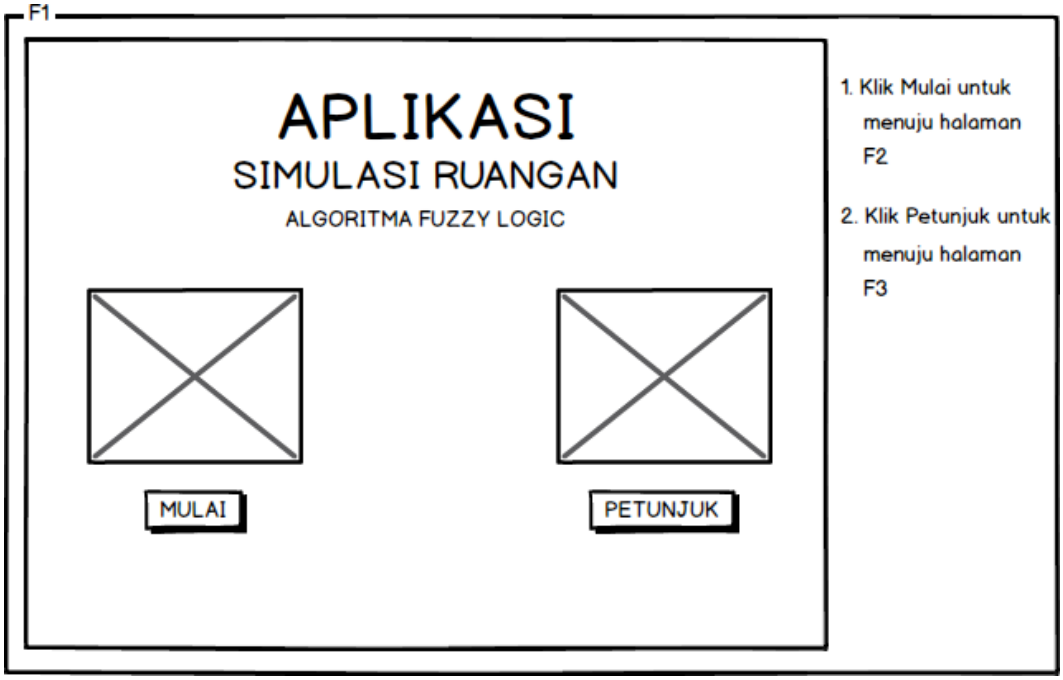
3.9. Perancangan Sistem

Perancangan adalah suatu bagian dari metodologi pengembangan suatu perangkat lunak yang dilakukan setelah tahapan untuk memberikan gambaran secara terperinci. Dalam tahap ini digambarkan rancangan sistem yang akan dibangun sebelum dilakukan pengkodean ke dalam suatu bahasa pemrograman.

3.9.1. Rancangan Antarmuka

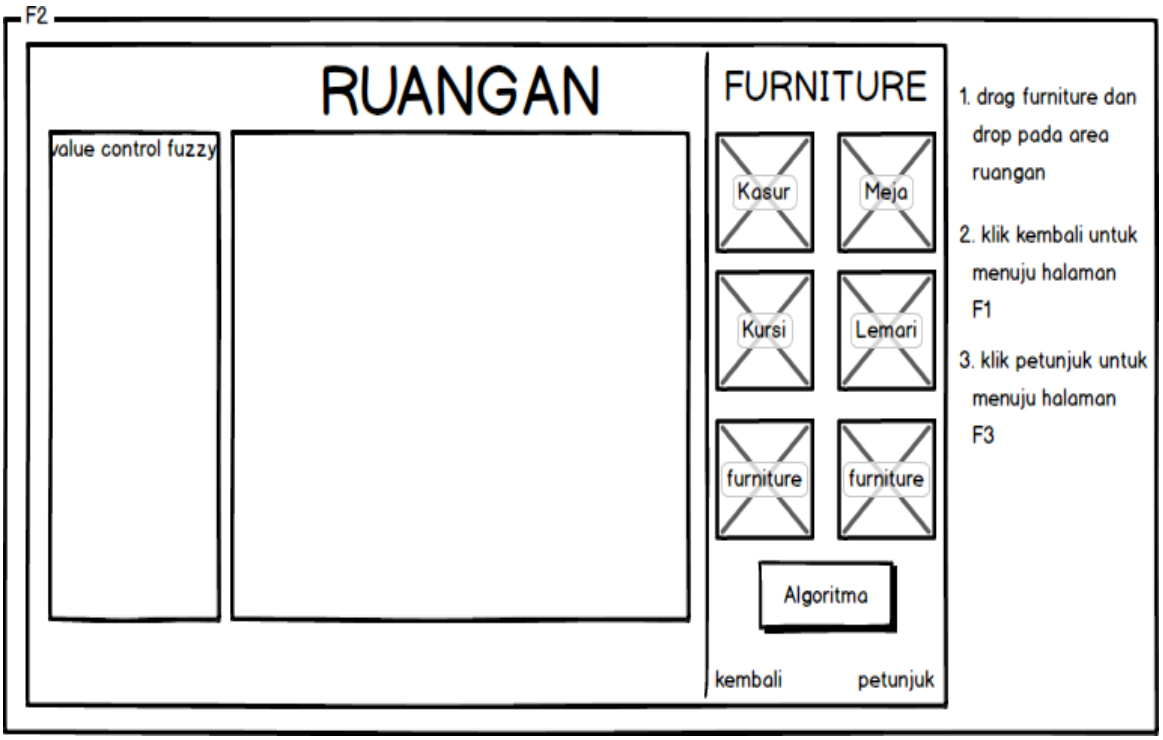
Perancangan antarmuka yang telah dibuat meliputi beberapa perancangan, diantaranya perancangan antarmuka menu utama. Tampilan Menu Utama digunakan untuk merancang tata letak *furniture* yang ingin diolah dan hasil perancangan

1. Tampilan Layar Utama



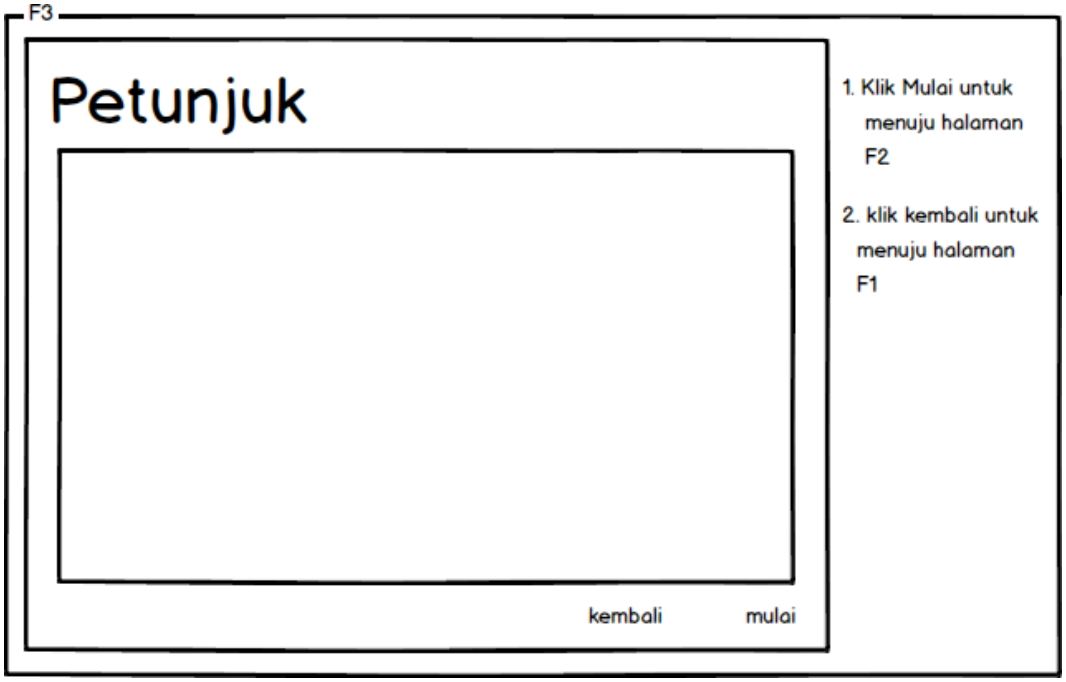
Gambar 3.16 Tampilan Layar Utama

2. Tampilan Layar perancangan tata letak



Gambar 3.17 Tampilan antarmuka merancang tata letak

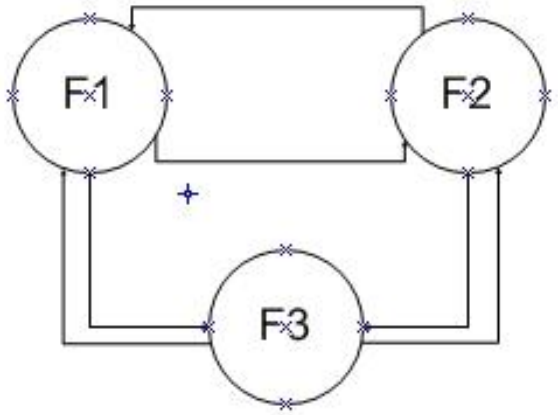
3. Tampilan Layar petunjuk



Gambar 3.18 Tampilan antarmuka halaman petunjuk

3.9.2. Rancangan Jaringan Semantik

Jaringan semantik menggambarkan hubungan navigasi menu dari satu halaman ke halaman lainnya (dari form satu ke form lainnya). Berikut adalah gambaran jaringan semantik yang ada pada aplikasi simulasi yang akan dibangun.

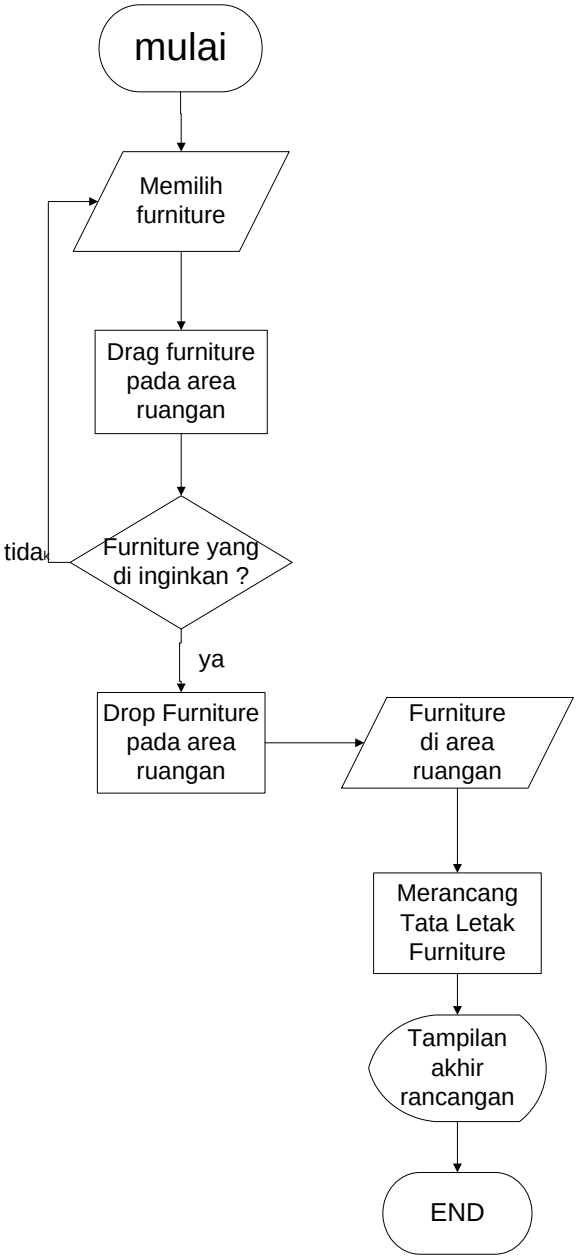


Gambar 3.19 Jaringan Semantik

3.9.3. Rancangan Prosedural

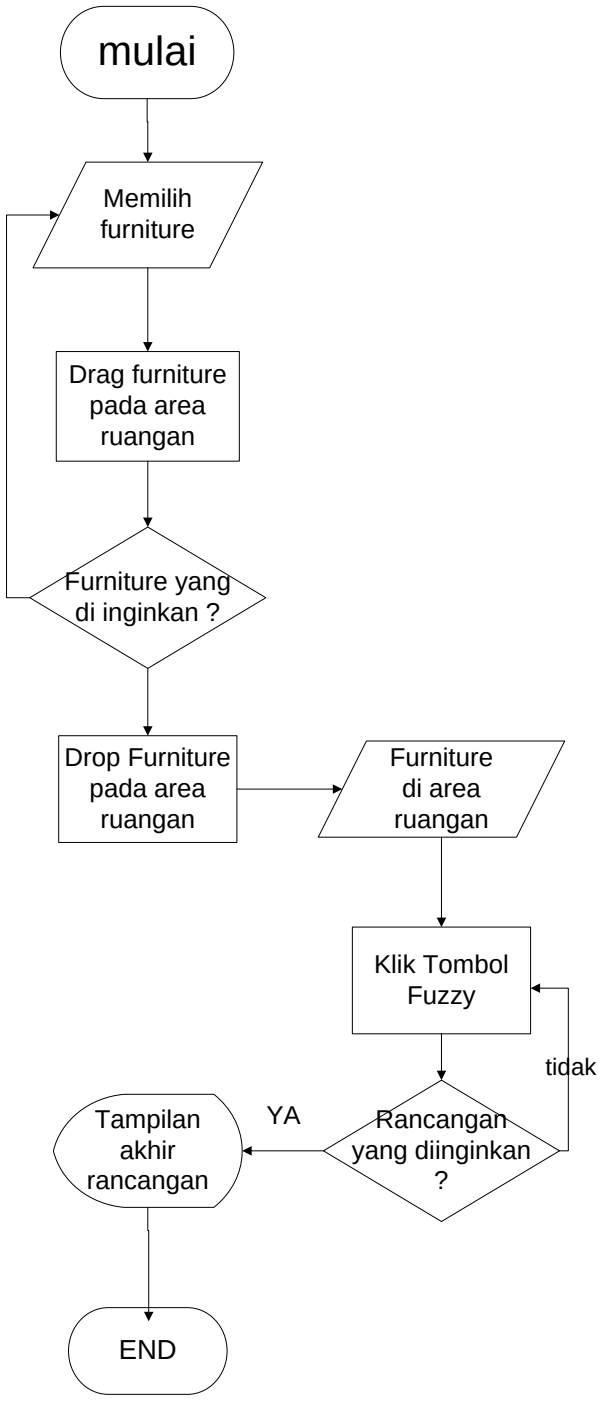
Perancangan prosedural digunakan untuk menggambarkan langkah-langkah atau alur dari setiap proses yang ada pada sistem. Berikut adalah rancangan prosedural dari aplikasi yang akan dibangun.

3.9.3.1 *Flowchart* merancang tata letak furniture oleh pengguna



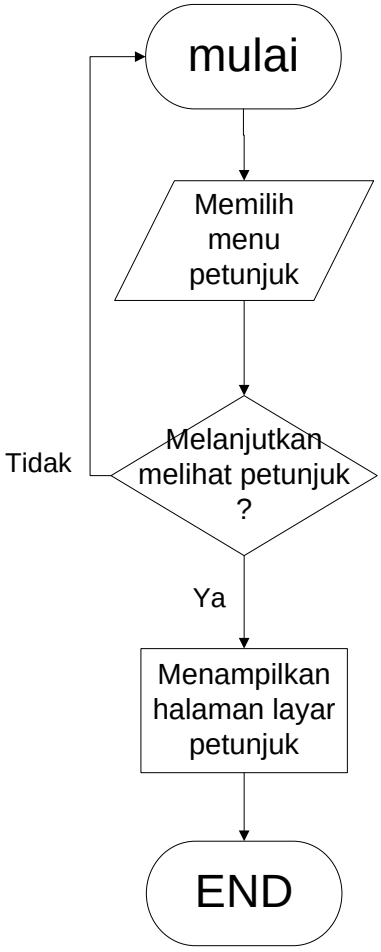
Gambar 3.20 Flowchart merancang tata letak *furniture* oleh pengguna

3.9.3.2 *Flowchart* merancang tata letak furniture oleh algoritma



Gambar 3.21 Flowchart merancang tata letak *furniture* oleh algoritma

3.9.3.3 *Flowchart* halaman petunjuk



Gambar 3.22 Flowchart Halaman petunjuk