

BAB III

Implementasi Sistem

Bab ini menjelaskan tentang implementasi dari sistem yang dikembangkan. Pembahasan mengenai bab ini terdiri atas : spesifikasi sistem yang meliputi perangkat keras dan perangkat lunak, implementasi prosedur yang digunakan dalam sistem, iterasi - iterasi pembuatan sistem.

3.1 Spesifikasi Sistem

Bagian spesifikasi sistem menjelaskan mengenai komponen - komponen yang dibangun untuk membangun sistem. Komponen - komponen yang dibahas meliputi komponen perangkat keras, perangkat lunak, jenis kamera yang digunakan, dan bentuk simulasi yang dilakukan.

3.1.1 Perangkat keras

Sistem pemetaan citra *long range* CCTV pada citra digital dengan menggunakan teori pengukuran *fuzzy* Sugeno dikerjakan pada komputer dekstop dengan spesifikasi perangkat keras sebagai berikut:

- Processor : Intel(R) CORE(TM)2 CPU 2 GHz
- Memori : 1 GB RAM
- *Harddisk* : 80 GB

3.1.2 Perangkat lunak

Perangkat lunak yang digunakan untuk mengembangkan sistem adalah sebagai berikut:

- Sistem Operasi : Microsoft Windows XP Professional SP 2

- IDE : Matlab, Java Development Kit 1.6

3.1.3 Kamera

Kamera untuk pengambilan citra mempunyai deskripsi sebagai berikut:

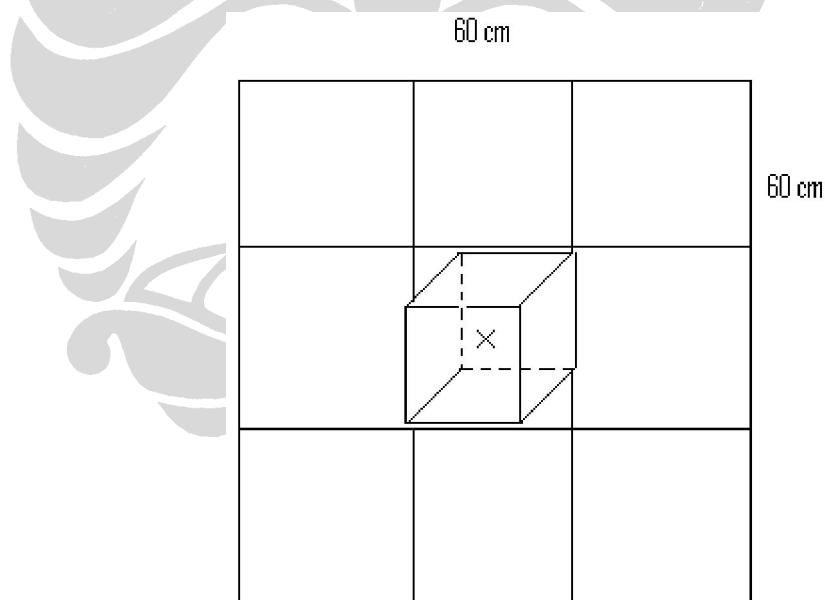
- Jenis kamera : Exiclim S50 Casio

- Dimensi citra : 1280 x 960

- Flash : Automatic

3.1.4 Simulasi ruangan

Berikut adalah model percobaan yang dibuat oleh penulis:



Gambar 3.1 Model Simulasi

Penulis melakukan eksperimen pada sebuah bidang seluas 60 cm x 60 cm. Objek X yang dicitrakan diletakkan secara acak pada bidang

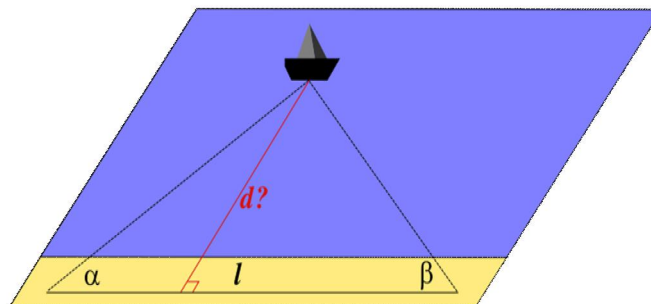
tersebut. Hasil digital pemotretan model percobaan dapat dilihat pada halaman lampiran.

3.2 Implementasi Prosedur

Prosedur dalam pengimplementasian sistem dapat dibagi dalam 2 tahap. Tahap pertama adalah prosedur untuk menentukan letak koordinat piksel pada gambar citra dari atas dengan pemotretan sejajar dengan sumbu y objek yang dicitrakan. Tahapan kedua adalah prosedur untuk menentukan letak koordinat piksel pada gambar citra dari atas dengan pemotretan mempunyai ketinggian tertentu terhadap objek yang dicitrakan. Pada tahapan kedua, *input* yang diberikan ke dalam sistem adalah sudut pemotretan berdasarkan sumbu Y . Hal ini disebabkan posisi *long range* CCTV yang lebih tinggi dari objek yang dicitrakan sehingga harus membentuk sudut pada sumbu Y untuk mendapatkan citra yang diinginkan. Pada simulasi yang dilakukan, pemotretan dilakukan dengan tinggi sebesar ± 30 cm dari permukaan lantai objek berada.

3.2.1 Tahap 1

Prosedur untuk menentukan letak koordinat piksel pada gambar citra dari atas dengan pemotretan sejajar sumbu y objek yang dicitrakan dapat menggunakan rumus sinus. Berikut adalah gambar penggunaan rumus sinus:



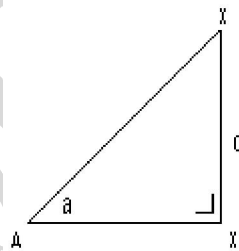
Gambar 3.2 Pencarian jarak

Kita dapat menghitung d dengan menggunakan rumus:

$$l = d/\tan \alpha + d/\tan \beta \quad (3.1)$$

$$d = l \tan \alpha \tan \beta : (\tan \alpha + \tan \beta) \quad (3.2)$$

l adalah jarak antara kamera pertama dengan kamera kedua. Apabila d ditemukan (jarak sumbu z objek pada dunia nyata) maka jarak x objek di dunia nyata dengan titik $(0,0)$ pada kamera 1 adalah :



Gambar 3.3 Model pemetaan

$$AX' = d : \tan \alpha. \quad (3.3)$$

AX' = jarak objek X pada koordinat nyata (sumbu x).

d = jarak objek X pada koordinat nyata (sumbu z).

Hasil pencitraan objek X pada gambar kamera 1 terletak pada koordinat:

$$\text{Piksel } x \text{ gambar: } (AX' : AB) * 1280 \quad (3.4)$$

$$\text{Piksel } Y \text{ gambar: } (d : AC) * 960 \quad (3.5)$$

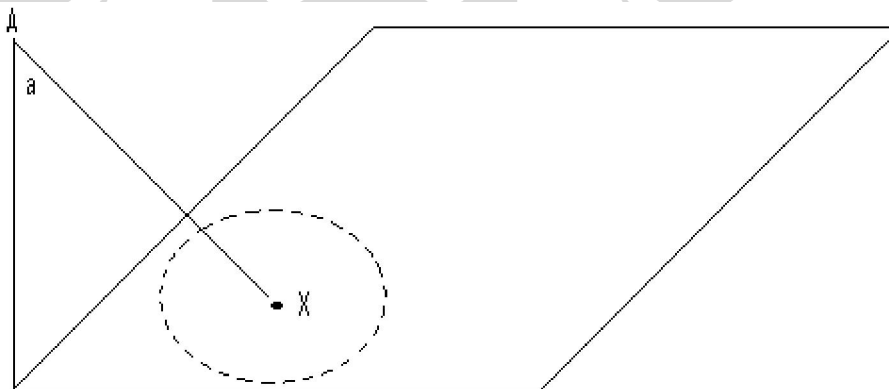
AB = Panjang kota yang dicitrakan

AC = Lebar kota yang dicitrakan

Penggunaan logika *fuzzy* terletak pada pengambilan sudut kamera melakukan pemotretan. Pada kasus ini *input* dari logika *fuzzy* yang dirancang berupa sudut sumbu x pemotretan, letak piksel x pada citra kamera 1 dan citra kamera 2.

3.2.2 Tahap 2

Setelah melakukan analisa pada tahapan pertama, maka kita memasuki tahapan kedua. Tahapan kedua adalah memotret objek dengan ketinggian tertentu; dalam simulasi yang dilakukan, tinggi objek pemotretan sebesar ± 30 cm. *Input* masukan dalam sistem bukanlah tinggi *long range* CCTV pemotretan melainkan sudut pemotretan berdasarkan sumbu y kepada objek yang dicitrakan. Semakin tinggi *long range* CCTV tersebut maka sudut yang dibentuk antara sumbu y dengan *long range* CCTV semakin kecil. Berikut adalah model penghitungan objek yang dicitrakan:



Gambar 3.4 Model penghitungan

Kasus di atas dapat disederhanakan dengan menganggap bahwa kamera A terletak sejajar sumbu y dengan objek X. Setelah mencari letak objek X pada piksel yang sejajar dengan sumbu y maka barulah kita

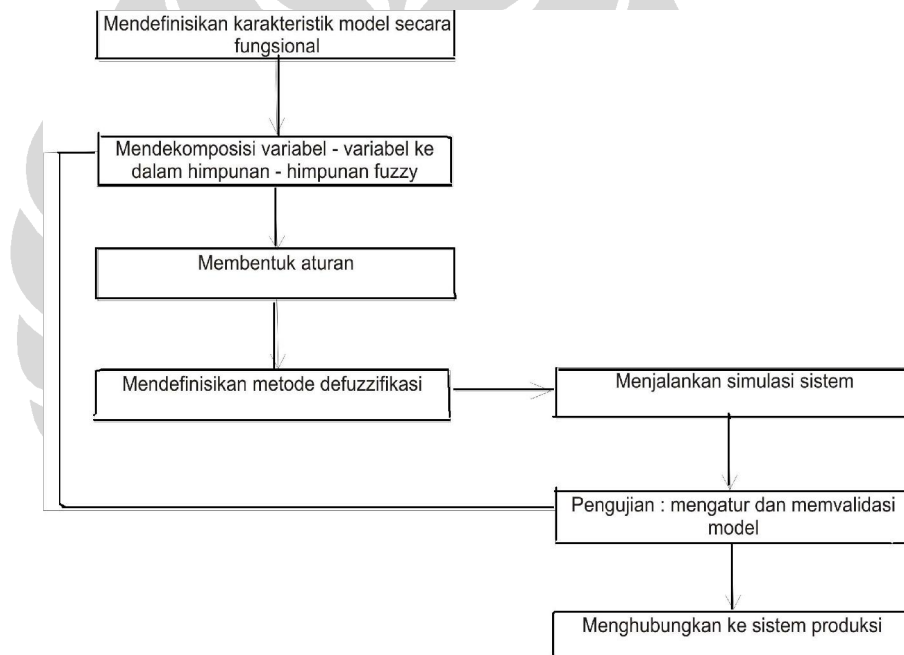
memperhitungkan sudut pemotretan terhadap sumbu y . Berikut adalah metode penghitungan piksel x, y pada citra yang baru:

$$\text{Piksel } x \text{ pada citra baru} = \text{piksel } x \text{ pada citra awal (sejajar sumbu } y)(3.6)$$

$$\text{Piksel } y \text{ pada citra baru} = y' + (\tan a * 960) \quad (3.7)$$

3.3 Logika Fuzzy

Logika fuzzy yang digunakan dapat dimodelkan seperti berikut :



Gambar 3.5 Langkah-langkah pembuatan logika fuzzy

Pada gambar tersebut setelah mendefinisikan karakteristik sistem yang dibuat, kita mendekomposisi variabel-variabel ke dalam himpunan fuzzy. Variabel yang dikomposisikan dalam himpunan fuzzy adalah variabel sudut pemotretan (berdasarkan sumbu x dan sumbu y). Setelah variabel tersebut dikomposisikan maka langkah kedua adalah membentuk aturan terhadap variabel – variabel tersebut. Terdapat 60 aturan yang digunakan untuk himpunan yang dibentuk. Metode defuzzifikasi yang dilakukan adalah

metode umum yaitu CoG (*Center of Gravity*). Apabila hasil yang didapatkan kurang bagus, maka dilakukan analisa di manakah letak kesalahan yang terjadi sehingga akurasi sistem dapat ditingkatkan.

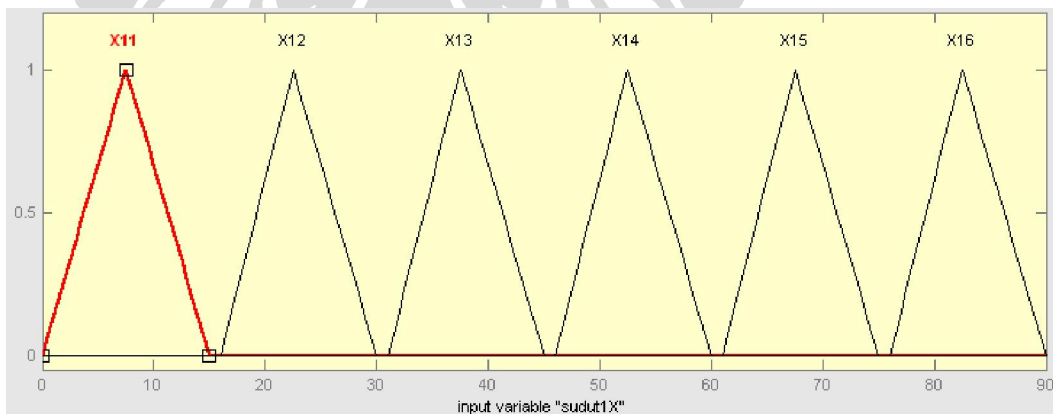
3.3.1 Iterasi 1

Iterasi 1 melakukan dekomposisi variabel input ke dalam *membership function trapmf* (segitiga). Ada 8 masukan input dari 2 kamera yang diujikan. 4 input dari masing-masing kamera yaitu :

1. Sudut X pemotretan,
2. Sudut Y pemotretan,
3. Koordinat X hasil pemotretan,
4. Korrrdinat Y hasil pemotretan.

Iterasi 1 dilakukan pada program Matlab. Perancangan *fuzzy inference system* mencakup *input* masukan, *output* keluaran dan *rules – rules* yang diperlukan. Terdapat 8 masukan dalam perancangan *input*:

1. Sudut X kamera 1

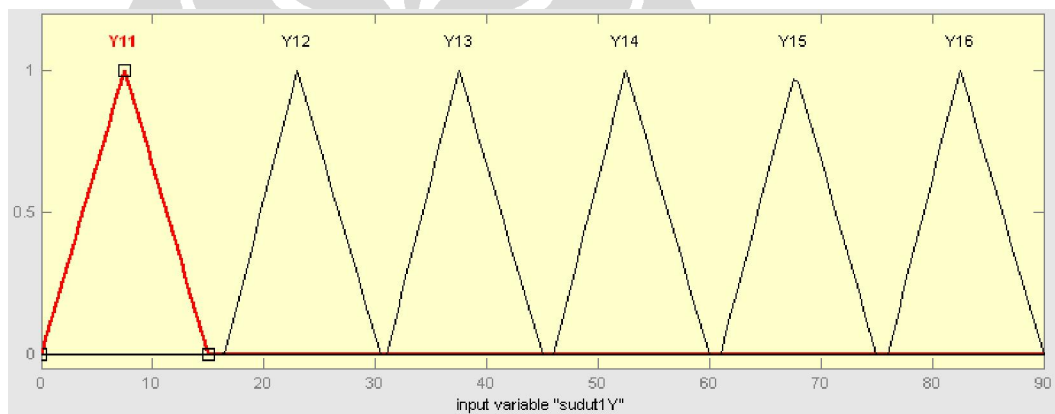


Gambar 3.6 *Membership function sudut1X*

Domain himpunan pada *input* tersebut:

1. $X_{11} = \text{trimf}(x,[0 \ 7.5 \ 15])$
2. $X_{12} = \text{trimf}(x,[16 \ 22.5 \ 30])$
3. $X_{13} = \text{trimf}(x,[31 \ 37.5 \ 45])$
4. $X_{14} = \text{trimf}(x,[46 \ 52.5 \ 60])$
5. $X_{15} = \text{trimf}(x,[61 \ 67.5 \ 75])$
6. $X_{16} = \text{trimf}(x,[76 \ 82.5 \ 90])$

2. Sudut Y kamera 1

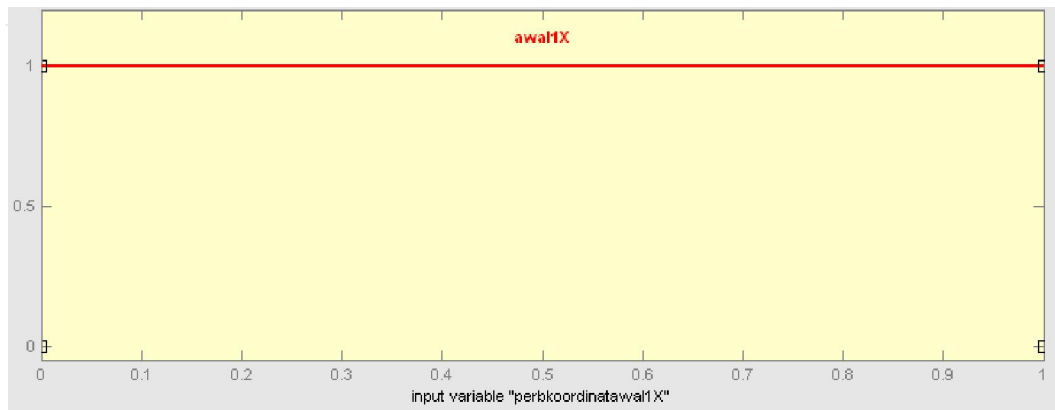


Gambar 3.7 Membership function sudut1Y

Domain himpunan pada *input* tersebut:

1. $Y_{11} = \text{trimf}(y,[0 \ 7.5 \ 15])$
2. $Y_{12} = \text{trimf}(y,[16 \ 22.5 \ 30])$
3. $Y_{13} = \text{trimf}(y,[31 \ 37.5 \ 45])$
4. $Y_{14} = \text{trimf}(y,[46 \ 52.5 \ 60])$
5. $Y_{15} = \text{trimf}(y,[61 \ 67.5 \ 75])$
6. $Y_{16} = \text{trimf}(y,[76 \ 82.5 \ 90])$

3. Perbandingan koordinat awal x dengan dimensi gambar pada citra kamera 1

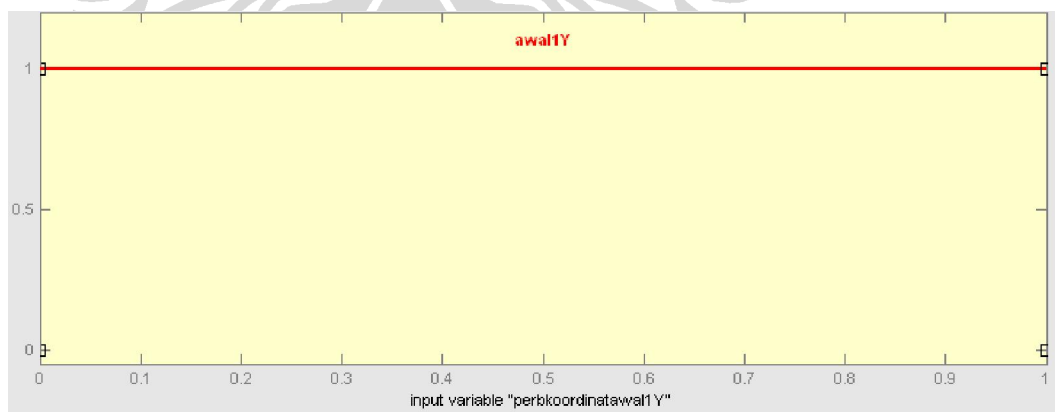


Gambar 3.8 *Membership function* perbkoordinataawal1X

Domain himpunan pada *input* tersebut:

$$\text{awal1X} = \text{trapmf}(x, [0 \ 1 \ 1 \ 1])$$

4. Perbandingan koordinat awal Y dengan dimensi gambar pada citra kamera 1

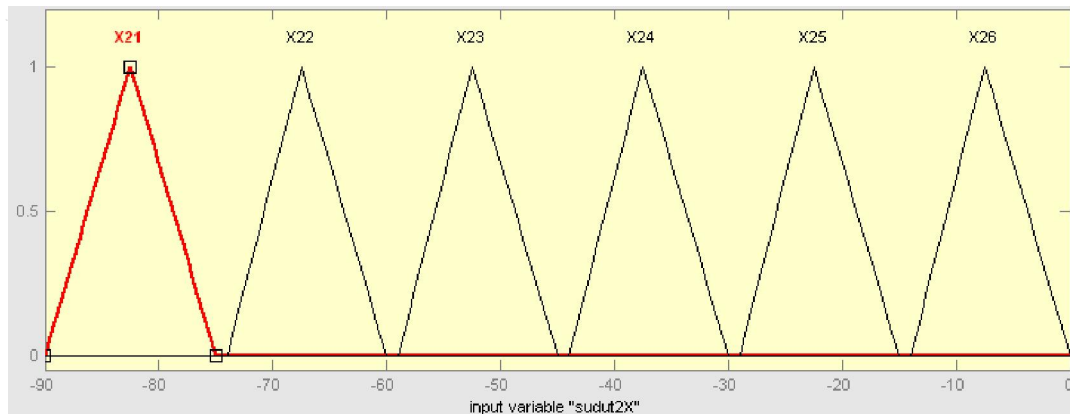


Gambar 3.9 *Membership function* perbkoordinataawal1Y

Domain himpunan pada *input* tersebut:

awal1Y = trapmf(y,[0 1 1 1])

5. Sudut x kamera 2

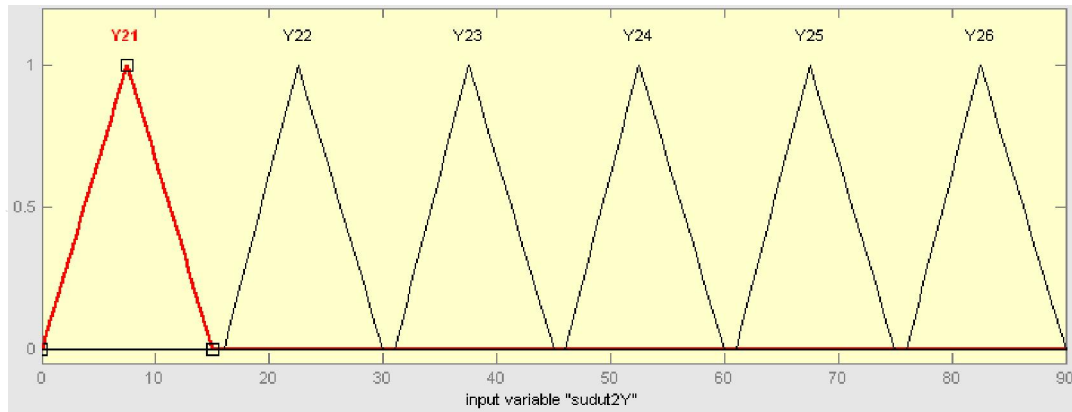


Gambar 3.10 Membership function sudut2X

Domain himpunan pada *input* tersebut:

1. $X21 = \text{trimf}(x, [-90 -82.5 -75])$
2. $X22 = \text{trimf}(x, [-74 -67.5 -60])$
3. $X23 = \text{trimf}(x, [-61 -52.5 -45])$
4. $X24 = \text{trimf}(x, [-46 -37.5 -30])$
5. $X25 = \text{trimf}(x, [-29 -22.5 -15])$
6. $X26 = \text{trimf}(x, [-14 -7.5 0])$

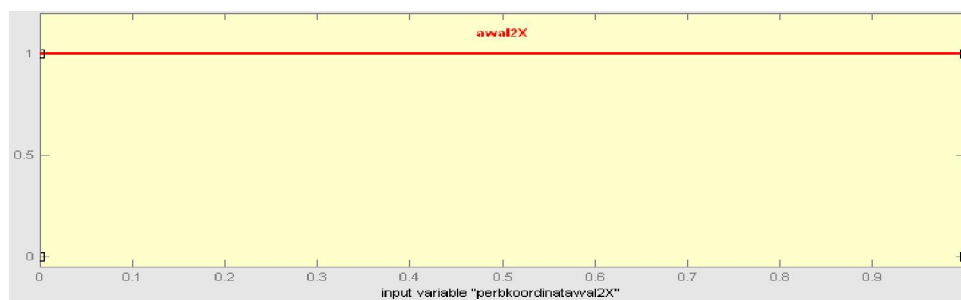
6. Sudut Y kamera 2



Gambar 3.11 *Membership function sudut2Y*

Domain himpunan pada *input* tersebut:

1. $Y11 = \text{trimf}(x,[0 \ 7.5 \ 15])$
2. $Y12 = \text{trimf}(x,[16 \ 22.5 \ 30])$
3. $Y13 = \text{trimf}(x,[31 \ 37.5 \ 45])$
4. $Y14 = \text{trimf}(x,[46 \ 52.5 \ 60])$
5. $Y15 = \text{trimf}(x,[61 \ 67.5 \ 75])$
6. $Y16 = \text{trimf}(x,[76 \ 82.5 \ 90])$
7. Perbandingan koordinat awal x dengan dimensi gambar pada citra kamera 2

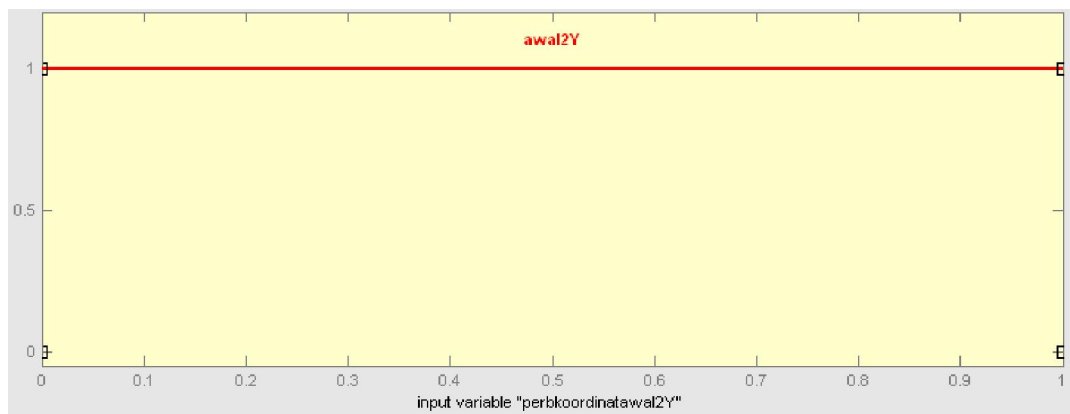


Gambar 3.12 *Membership function perbkoordinatawal2X*

Domain himpunan pada *input* tersebut:

$$\text{awal2X} = \text{trapmf}(x, [0 \ 1 \ 1 \ 1])$$

8. Perbandingan koordinat awal Y dengan dimensi gambar pada citra kamera 2



Gambar 3.13 *Membership function* perbkoordinataawal2Y

Domain himpunan pada *input* tersebut:

$$\text{awal2Y} = \text{trapmf}(y, [0 \ 1 \ 1 \ 1])$$

Nilai *output* (x) yang digunakan dalam sistem *fuzzy* yang dirancang adalah:

- § flat1 = 0.01755 * X11 + awal1X * 1280
- § flat2 = 0.01841 * X12 + awal1X * 1280
- § flat3 = 0.02046 * X13 + awal1X * 1280
- § flat4 = 0.02482 * X14 + awal1X * 1280
- § flat5 = 0.03577 * X15 + awal1X * 1280
- § flat6 = 0.09207 * X16 + awal1X * 1280
- § flat7 = -0.01755 * X21 + awal2X * 1280
- § flat8 = -0.01841 * X22 + awal2X * 1280

- § flat9 = -0.02046 * X23 + awal2X * 1280
- § flat10 = -0.02482 * X24 + awal2X * 1280
- § flat11 = -0.03577 * X25 + awal2X * 1280
- § flat12 = -0.09207 * X26 + awal2X * 1280
- § unflat11 = 0.02046 * X13 + -0.02046 * X23 + awal1X * 1280
- § unflat12 = 0.02046 * X13 + -0.02046 * X23 + awal2X * 1280
- § unflat21 = 0.02046 * X13 + -0.02482 * X24 + awal1X * 1280
- § unflat22 = 0.02046 * X13 + -0.02482 * X24 + awal2X * 1280
- § unflat31 = 0.02046 * X13 + -0.03577 * X25 + awal1X * 1280
- § unflat32 = 0.02046 * X13 + -0.03577 * X25 + awal2X * 1280
- § unflat41 = 0.02482 * X14 + -0.02046 * X23 + awal1X * 1280
- § unflat42 = 0.02482 * X14 + -0.02046 * X23 + awal2X * 1280
- § unflat51 = 0.02482 * X14 + -0.02482 * X24 + awal1X * 1280
- § unflat52 = 0.02482 * X14 + -0.02482 * X24 + awal2X * 1280
- § unflat61 = 0.02482 * X14 + -0.03577 * X25 + awal1X * 1280
- § unflat62 = 0.02482 * X14 + -0.03577 * X25 + awal2X * 1280
- § unflat71 = 0.03577 * X15 + -0.02046 * X23 + awal1X * 1280
- § unflat72 = 0.03577 * X15 + -0.02046 * X23 + awal2X * 1280
- § unflat81 = 0.03577 * X15 + -0.02482 * X24 + awal1X * 1280
- § unflat82 = 0.03577 * X15 + -0.02482 * X24 + awal2X * 1280
- § unflat91 = 0.03577 * X15 + -0.03577 * X25 + awal1X * 1280
- § unflat92 = 0.03577 * X15 + -0.03577 * X25 + awal1X * 1280

Sedangkan nilai *output* (Y) yang dirancang dalam sistem *fuzzy* ini adalah:

- § flat1 = 0.01755 * Y11 + awal1Y * 960
- § flat2 = 0.01841 * Y12 + awal1Y * 960
- § flat3 = 0.02046 * Y13 + awal1Y * 960
- § flat4 = 0.02482 * Y14 + awal1Y * 960
- § flat5 = 0.03577 * Y15 + awal1Y * 960
- § flat6 = 0.09207 * Y16 + awal1Y * 960
- § flat7 = 0.01755 * Y21 + awal2Y * 960
- § flat8 = 0.01841 * Y22 + awal2Y * 960

- § flat9 = 0.02046 * Y23 + awal2Y * 960
- § flat10 = 0.02482 * Y24 + awal2Y * 960
- § flat11 = 0.03577 * Y25 + awal2Y * 960
- § flat12 = 0.09207 * Y26 + awal2Y * 960
- § unflat11 = 0.01841 * Y12 + 0.01841 * Y22 + awal1Y * 960
- § unflat12 = 0.01841 * Y12 + 0.01841 * Y22 + awal2Y * 960
- § unflat21 = 0.01841 * Y12 + 0.02046 * Y23 + awal1Y * 960
- § unflat22 = 0.01841 * Y12 + 0.02046 * Y23 + awal2Y * 960
- § unflat31 = 0.01841 * Y12 + 0.02482 * Y24 + awal1Y * 960
- § unflat32 = 0.01841 * Y12 + 0.02482 * Y24 + awal2Y * 960
- § unflat41 = 0.02046 * Y13 + 0.01841 * Y22 + awal1Y * 960
- § unflat42 = 0.02046 * Y13 + 0.01841 * Y22 + awal2Y * 960
- § unflat51 = 0.02046 * Y13 + 0.02046 * Y23 + awal1Y * 960
- § unflat52 = 0.02046 * Y13 + 0.02046 * Y23 + awal2Y * 960
- § unflat61 = 0.02046 * Y13 + 0.02482 * Y24 + awal1Y * 960
- § unflat62 = 0.02046 * Y13 + 0.02482 * Y24 + awal2Y * 960
- § unflat71 = 0.02482 * Y14 + 0.01841 * Y22 + awal1Y * 960
- § unflat72 = 0.02482 * Y14 + 0.01841 * Y22 + awal2Y * 960
- § unflat81 = 0.02482 * Y14 + 0.02046 * Y23 + awal1Y * 960
- § unflat82 = 0.02482 * Y14 + 0.02046 * Y23 + awal2Y * 960
- § unflat91 = 0.02482 * Y14 + 0.02482 * Y24 + awal1Y * 960
- § unflat92 = 0.02482 * Y14 + 0.02482 * Y24 + awal2Y * 960

Rules yang ditetapkan dalam sistem ini sebanyak 60 rules. Berikut adalah aturan –aturan dalam sistem ini:

- If (sudut1X is X11) and (perbkoordinataawal1X is awal1X) then (koordinataakhirX is flat1)
- If (sudut1X is X12) and (perbkoordinataawal1X is awal1X) then (koordinataakhirX is flat2)
- If (sudut1X is X13) and (perbkoordinataawal1X is awal1X) then (koordinataakhirX is flat3)
- If (sudut1X is X14) and (perbkoordinataawal1X is awal1X) then (koordinataakhirX is flat4)

- If (sudut1X is X15) and (perbkoordinataawal1X is awal1X) then (koordinataakhirX is flat5)
- If (sudut1X is X16) and (perbkoordinataawal1X is awal1X) then (koordinataakhirX is flat6)
- If (sudut2X is X21) and (perbkoordinataawal2X is awal2X) then (koordinataakhirX is flat7)
- If (sudut2X is X22) and (perbkoordinataawal2X is awal2X) then (koordinataakhirX is flat8)
- If (sudut2X is X23) and (perbkoordinataawal2X is awal2X) then (koordinataakhirX is flat9)
- If (sudut2X is X24) and (perbkoordinataawal2X is awal2X) then (koordinataakhirX is flat10)
- If (sudut2X is X25) and (perbkoordinataawal2X is awal2X) then (koordinataakhirX is flat11)
- If (sudut2X is X26) and (perbkoordinataawal2X is awal2X) then (koordinataakhirX is flat12)
- If (sudut1X is X13) and (perbkoordinataawal1X is awal1X) and (sudut2X is X23) then (koordinataakhirX is unflat11)
- If (sudut1X is X13) and (sudut2X is X23) and (perbkoordinataawal2X is awal2X) then (koordinataakhirX is unflat12)
- If (sudut1X is X13) and (perbkoordinataawal1X is awal1X) and (sudut2X is X24) then (koordinataakhirX is unflat21)
- If (sudut1X is X13) and (sudut2X is X24) and (perbkoordinataawal2X is awal2X) then (koordinataakhirX is unflat22)
- If (sudut1X is X13) and (perbkoordinataawal1X is awal1X) and (sudut2X is X25) then (koordinataakhirX is unflat31)
- If (sudut1X is X13) and (sudut2X is X25) and (perbkoordinataawal2X is awal2X) then (koordinataakhirX is unflat32)
- If (sudut1X is X14) and (perbkoordinataawal1X is awal1X) and (sudut2X is X23) then (koordinataakhirX is unflat41)
- If (sudut1X is X14) and (sudut2X is X23) and (perbkoordinataawal2X is awal2X) then (koordinataakhirX is unflat42)

- If (sudut1X is X14) and (perbkoordinataawal1X is awal1X) and (sudut2X is X24) then (koordinataakhirX is unflat51)
- If (sudut1X is X14) and (sudut2X is X24) and (perbkoordinataawal2X is awal2X) then (koordinataakhirX is unflat52)
- If (sudut1X is X14) and (perbkoordinataawal1X is awal1X) and (sudut2X is X25) then (koordinataakhirX is unflat61)
- If (sudut1X is X14) and (sudut2X is X25) and (perbkoordinataawal2X is awal2X) then (koordinataakhirX is unflat62)
- If (sudut1X is X15) and (perbkoordinataawal1X is awal1X) and (sudut2X is X23) then (koordinataakhirX is unflat71)
- If (sudut1X is X15) and (sudut2X is X23) and (perbkoordinataawal2X is awal2X) then (koordinataakhirX is unflat72)
- If (sudut1X is X15) and (perbkoordinataawal1X is awal1X) and (sudut2X is X24) then (koordinataakhirX is unflat81)
- If (sudut1X is X15) and (sudut2X is X24) and (perbkoordinataawal2X is awal2X) then (koordinataakhirX is unflat82)
- If (sudut1X is X15) and (perbkoordinataawal1X is awal1X) and (sudut2X is X25) then (koordinataakhirX is unflat91)
- If (sudut1X is X15) and (sudut2X is X25) and (perbkoordinataawal2X is awal2X) then (koordinataakhirX is unflat92)
- If (sudut1Y is Y11) and (perbkoordinataawal1Y is awal1Y) then (koordinataakhirY is flat1)
- If (sudut1Y is Y12) and (perbkoordinataawal1Y is awal1Y) then (koordinataakhirY is flat2)
- If (sudut1Y is Y13) and (perbkoordinataawal1Y is awal1Y) then (koordinataakhirY is flat3)
- If (sudut1Y is Y14) and (perbkoordinataawal1Y is awal1Y) then (koordinataakhirY is flat4)
- If (sudut1Y is Y15) and (perbkoordinataawal1Y is awal1Y) then (koordinataakhirY is flat5)
- If (sudut1Y is Y16) and (perbkoordinataawal1Y is awal1Y) then (koordinataakhirY is flat6)

- If (sudut2Y is Y21) and (perbkoordinataawal2Y is awal2Y) then (koordinataakhirY is flat7)
- If (sudut2Y is Y22) and (perbkoordinataawal2Y is awal2Y) then (koordinataakhirY is flat8)
- If (sudut2Y is Y23) and (perbkoordinataawal2Y is awal2Y) then (koordinataakhirY is flat9)
- If (sudut2Y is Y24) and (perbkoordinataawal2Y is awal2Y) then (koordinataakhirY is flat10)
- If (sudut2Y is Y25) and (perbkoordinataawal2Y is awal2Y) then (koordinataakhirY is flat11)
- If (sudut2Y is Y26) and (perbkoordinataawal2Y is awal2Y) then (koordinataakhirY is flat12)
- If (sudut1Y is Y12) and (perbkoordinataawal1Y is awal1Y) and (sudut2Y is Y22) then (koordinataakhirY is unflat11)
- If (sudut1Y is Y12) and (sudut2Y is Y22) and (perbkoordinataawal2Y is awal2Y) then (koordinataakhirY is unflat12)
- If (sudut1Y is Y12) and (perbkoordinataawal1Y is awal1Y) and (sudut2Y is Y23) then (koordinataakhirY is unflat21)
- If (sudut1Y is Y12) and (sudut2Y is Y23) and (perbkoordinataawal2Y is awal2Y) then (koordinataakhirY is unflat22)
- If (sudut1Y is Y12) and (perbkoordinataawal1Y is awal1Y) and (sudut2Y is Y24) then (koordinataakhirY is unflat31)
- If (sudut1Y is Y12) and (sudut2Y is Y24) and (perbkoordinataawal2Y is awal2Y) then (koordinataakhirY is unflat32)
- If (sudut1Y is Y13) and (perbkoordinataawal1Y is awal1Y) and (sudut2Y is Y22) then (koordinataakhirY is unflat41)
- If (sudut1Y is Y13) and (sudut2Y is Y22) and (perbkoordinataawal2Y is awal2Y) then (koordinataakhirY is unflat42)
- If (sudut1Y is Y13) and (perbkoordinataawal1Y is awal1Y) and (sudut2Y is Y23) then (koordinataakhirY is unflat51)
- If (sudut1Y is Y13) and (sudut2Y is Y23) and (perbkoordinataawal2Y is awal2Y) then (koordinataakhirY is unflat52)

- If (sudut1Y is Y13) and (perbkoordinataawal1Y is awal1Y) and (sudut2Y is Y24) then (koordinataakhirY is unflat61)
- If (sudut1Y is Y13) and (sudut2Y is Y24) and (perbkoordinataawal2Y is awal2Y) then (koordinataakhirY is unflat62)
- If (sudut1Y is Y14) and (perbkoordinataawal1Y is awal1Y) and (sudut2Y is Y22) then (koordinataakhirY is unflat71)
- If (sudut1Y is Y14) and (sudut2Y is Y22) and (perbkoordinataawal2Y is awal2Y) then (koordinataakhirY is unflat72)
- If (sudut1Y is Y14) and (perbkoordinataawal1Y is awal1Y) and (sudut2Y is Y23) then (koordinataakhirY is unflat81)
- If (sudut1Y is Y14) and (sudut2Y is Y23) and (perbkoordinataawal2Y is awal2Y) then (koordinataakhirY is unflat82)
- If (sudut1Y is Y14) and (perbkoordinataawal1Y is awal1Y) and (sudut2Y is Y24) then (koordinataakhirY is unflat91)
- If (sudut1Y is Y14) and (sudut2Y is Y24) and (perbkoordinataawal2Y is awal2Y) then (koordinataakhirY is unflat92)

3.3.2 Iterasi 2

Pada iterasi kedua penulis melakukan banyak perubahan terhadap sistem *fuzzy* yang dirancang. Perubahan yang dilakukan oleh penulis mencakup: *membership function* dan nilai *output* sistem. *Membership function* yang digunakan oleh penulis adalah *membership function Gaussian*, sedangkan nilai *output* yang diberikan oleh sistem dibentuk oleh persamaan trigonometri.

Pengubahan nilai *membership function* dari fungsi segitiga (*trapmf*) menjadi Gaussian (*gaussmf*) disebabkan karena fungsi segitiga dapat memberikan nilai 0 pada sudut-sudut tertentu seperti : $0^0, 7.5^0, 15^0$, dll. Sedangkan fungsi Gaussian tidak memberikan nilai 0 pada setiap sudut manapun. Nilai *output* yang dihitung oleh sistem mengalami perubahan dari pembuatan nilai *output* perdsarkan fungsi persamaan linear menjadi fungsi persamaan trigonometri. Perubahan tersebut dilakukan karena rancangan

penghitungan yang diusulkan oleh penulis merupakan suatu fungsi trigonometri.

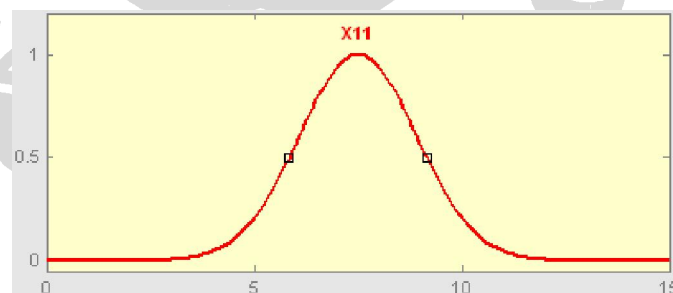
Penulis juga melakukan perancangan GUI agar memudahkan pengguna dalam menjalankan sistem ini. Penulis melakukan pemrograman menggunakan IDE Java Development Kit 1.6 untuk mengimplementasikan perubahan-perubahan yang dilakukan. Berikut adalah *membership function* program pada iterasi kedua:

1. *Membership function* X11 dibentuk dengan *Gaussian function*.

Berikut adalah *source* program X11 beserta grafik:

```
if((data_sudut1X>=0)&&(data_sudut1X<=15))
{
    return Math.pow(e,(-Math.pow((data_sudut1X - 7.5),2) / 4));
}
else
{
    return 0;
}
```

Grafik nilai sudut X11 dapat dilihat pada gambar di bawah ini:



Gambar 3.14 *Membership function* X11

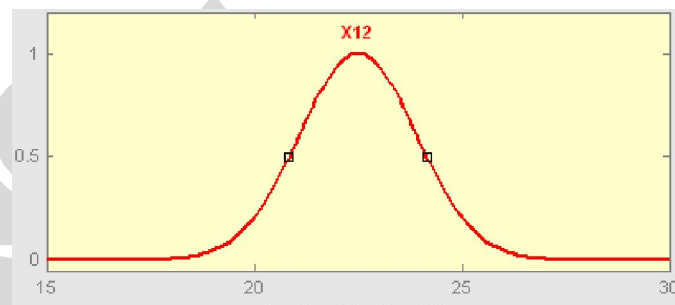
2. *Membership function* X12 dibentuk dengan *Gaussian function*.

Berikut adalah *source* program X12 beserta

grafik:

```
if((data_sudut1X>=15)&&(data_sudut1X<=30))  
{  
    return Math.pow(e,(-Math.pow((data_sudut1X - 22.5),2) / 4));  
}  
else  
{  
    return 0;  
}
```

Grafik nilai sudut X12 dapat dilihat pada gambar di bawah ini:



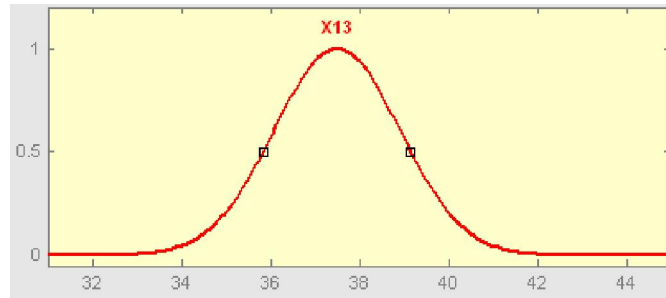
Gambar 3.15 *Membership function X12*

3. *Membership function X13* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program X13 beserta grafik:

```
if((data_sudut1X>=30)&&(data_sudut1X<=45))  
{  
    return Math.pow(e,(-Math.pow((data_sudut1X - 37.5),2) / 4));  
}  
else  
{  
    return 0;  
}
```

Grafik nilai sudut X13 dapat dilihat pada gambar di bawah ini:



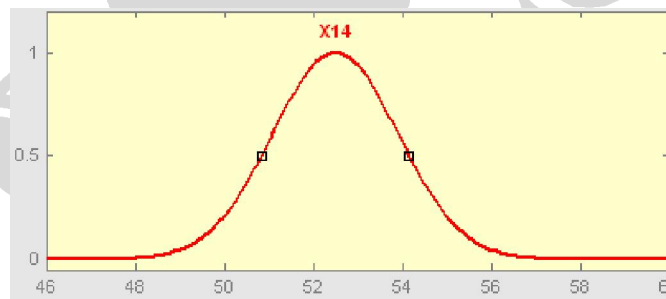
Gambar 3.16 *Membership function X13*

4. *Membership function X14* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program X14 beserta grafik:

```
if((data_sudut1X >= 45) && (data_sudut1X <= 60))  
{  
    return Math.pow(e, (-Math.pow((data_sudut1X - 52.5), 2) / 4));  
}  
else  
{  
    return 0;  
}
```

Grafik nilai sudut X14 dapat dilihat pada gambar di bawah ini:



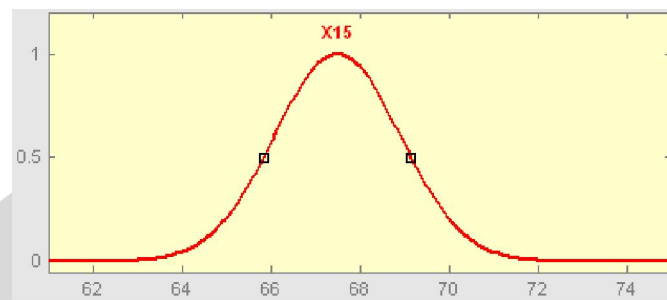
Gambar 3.17 *Membership function X14*

5. *Membership function X15* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program X15 beserta grafik:

```
if((data_sudut1X>=60)&&(data_sudut1X<=75))  
{  
    return Math.pow(e,(-Math.pow((data_sudut1X - 67.5),2) / 4));  
}  
else  
{  
    return 0;  
}
```

Grafik nilai sudut X15 dapat dilihat pada gambar di bawah ini:



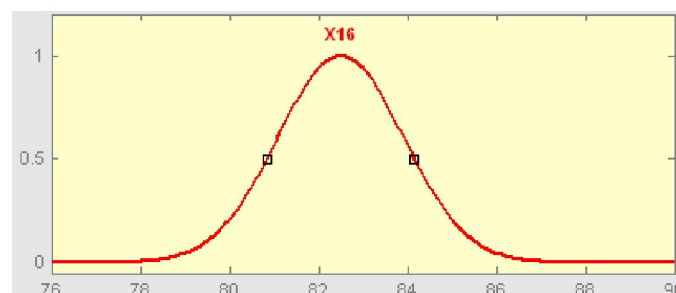
Gambar 3.18 *Membership function X15*

6. *Membership function X16* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program X16 beserta grafik:

```
if((data_sudut1X>=75)&&(data_sudut1X<=90))  
{  
    return Math.pow(e,(-Math.pow((data_sudut1X - 82.5),2) / 4));  
}  
else  
{  
    return 0;  
}
```

Grafik nilai sudut X16 dapat dilihat pada gambar di bawah ini:



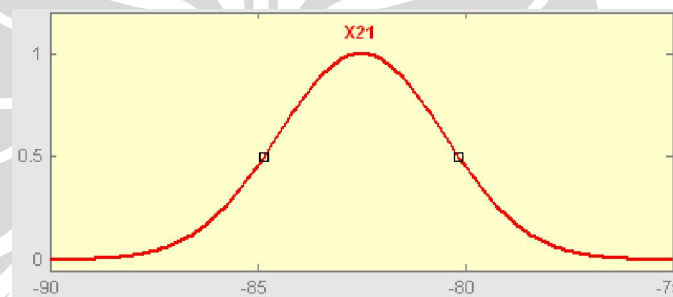
Gambar 3.19 *Membership function X16*

7. *Membership function X21* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program X21 beserta grafik:

```
if((data_sudut2X>=-90)&&(data_sudut2X<=-75))
{
    return Math.pow(e,(-Math.pow((data_sudut2X + 82.5),2) / 16));
}
else
{
    return 0;
}
```

Grafik nilai sudut X21 dapat dilihat pada gambar di bawah ini:



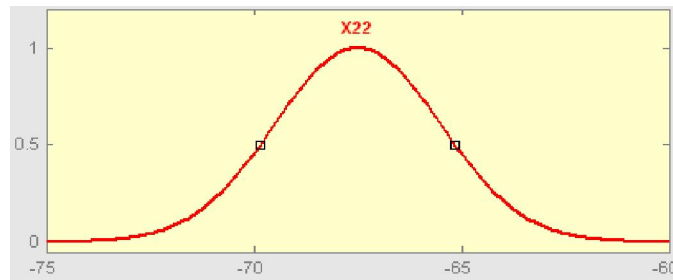
Gambar 3.20 *Membership function X21*

8. *Membership function X22* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program X22 beserta grafik:

```
if((data_sudut2X>=-75)&&(data_sudut2X<=-60))
{
    return Math.pow(e,(-Math.pow((data_sudut2X + 67.5),2) / 16));
}
else
{
    return 0;
}
```

Grafik nilai sudut X22 dapat dilihat pada gambar di bawah ini:



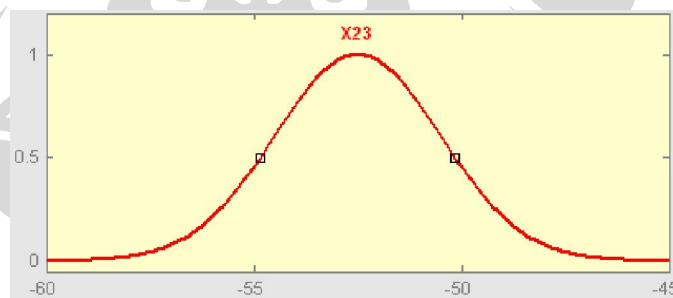
Gambar 3.21 *Membership function X22*

9. *Membership function X23* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program X23 beserta grafik:

```
if((data_sudut2X>=-60)&&(data_sudut2X<=-45))  
{  
    return Math.pow(e,(-Math.pow((data_sudut2X + 52.5),2) / 16));  
}  
else  
{  
    return 0;  
}
```

Grafik nilai sudut X23 dapat dilihat pada gambar di bawah ini:



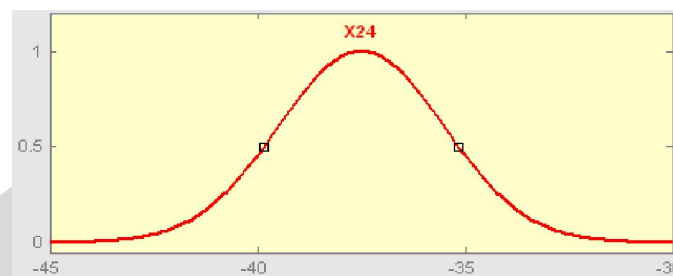
Gambar 3.22 *Membership function X23*

10. *Membership function X24* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program X24 beserta grafik:


```
if((data_sudut2X>=-45)&&(data_sudut2X<=-30))  
{  
    return Math.pow(e,(-Math.pow((data_sudut2X + 37.5),2) / 16));  
}  
else  
{  
    return 0;  
}
```

Grafik nilai sudut X24 dapat dilihat pada gambar di bawah ini:



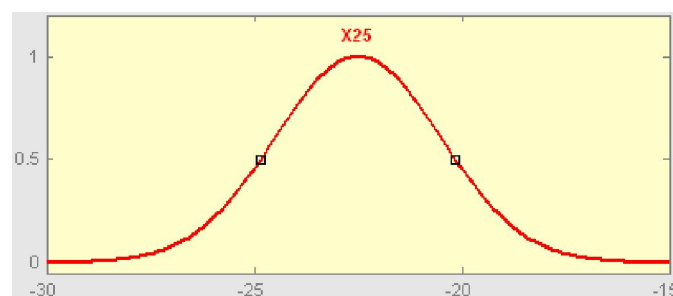
Gambar 3.23 *Membership function X24*

11. *Membership function X25* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program X25 beserta grafik:

```
if((data_sudut2X>=-30)&&(data_sudut2X<=-15))  
{  
    return Math.pow(e,(-Math.pow((data_sudut2X + 22.5),2) / 16));  
}  
else  
{  
    return 0;  
}
```

Grafik nilai sudut X25 dapat dilihat pada gambar di bawah ini:



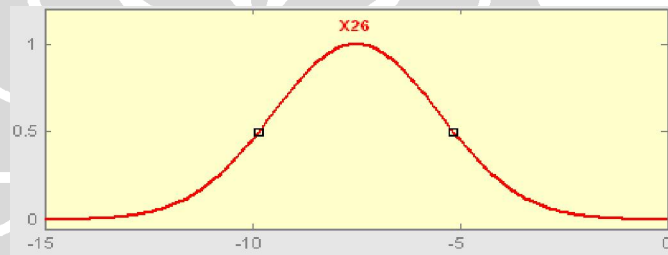
Gambar 3.24 *Membership function X25*

12. *Membership function X26* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program X26 beserta grafik:

```
if((data_sudut2X>=-15)&&(data_sudut2X<=0))
{
    return Math.pow(e,(-Math.pow((data_sudut2X + 7.5),2) / 16));
}
else
{
    return 0;
}
```

Grafik nilai sudut X26 dapat dilihat pada gambar di bawah ini:



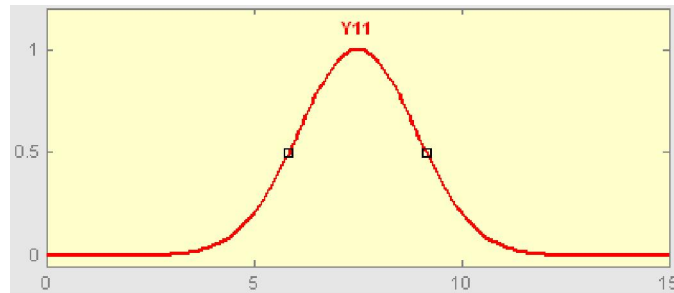
Gambar 3.25 *Membership function X26*

13. *Membership function Y11* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program Y11 beserta grafik:

```
if((data_sudut1Y>=0)&&(data_sudut1Y<=15))
{
    return Math.pow(e,(-Math.pow((data_sudut1Y - 7.5),2) / 16));
}
else
{
    return 0;
}
```

Grafik nilai sudut Y11 dapat dilihat pada gambar di bawah ini:



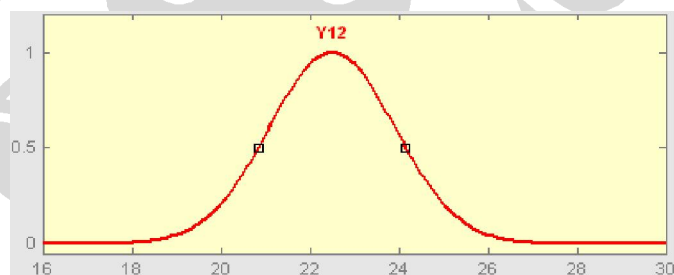
Gambar 3.26 *Membership function Y11*

14. *Membership function Y12* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program Y12 beserta grafik:

```
if((data_sudut1Y >= 15) && (data_sudut1Y <= 30))  
{  
    return Math.pow(e, (-Math.pow((data_sudut1Y - 22.5), 2) / 16));  
}  
else  
{  
    return 0;  
}
```

Grafik nilai sudut Y12 dapat dilihat pada gambar di bawah ini:



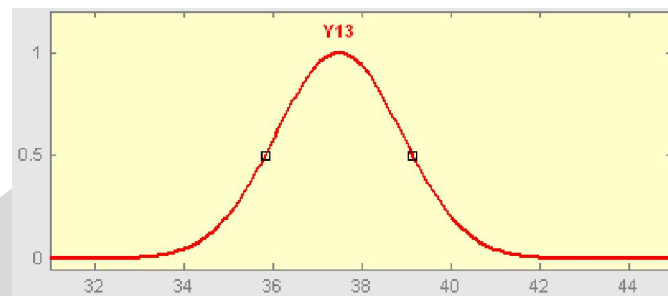
Gambar 3.27 *Membership function Y12*

15. *Membership function Y13* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program Y13 beserta grafik:

```
if((data_sudut1Y>=30)&&(data_sudut1Y<=45.5))  
{  
    return Math.pow(e,(-Math.pow((data_sudut1Y - 37.5),2) / 16));  
}  
else  
{  
    return 0;  
}
```

Grafik nilai sudut Y13 dapat dilihat pada gambar di bawah ini:



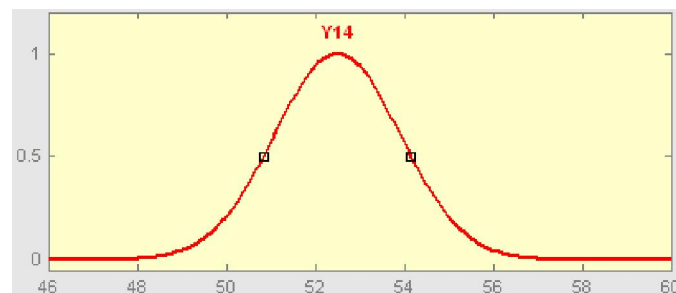
Gambar 3.28 *Membership function Y13*

16. *Membership function Y14* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program Y14 beserta grafik:

```
if((data_sudut1Y>=45)&&(data_sudut1Y<=60))  
{  
    return Math.pow(e,(-Math.pow((data_sudut1Y - 52.5),2) / 16));  
}  
else  
{  
    return 0;  
}
```

Grafik nilai sudut Y14 dapat dilihat pada gambar di bawah ini:



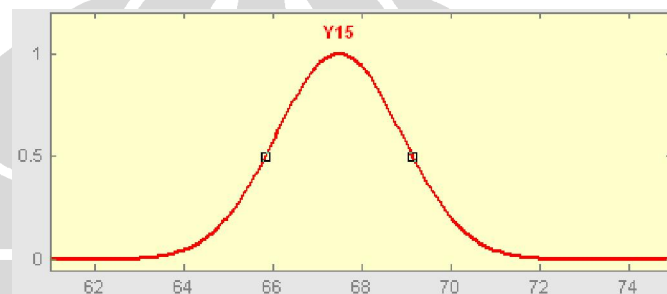
Gambar 3.29 *Membership function Y14*

17. *Membership function* Y15 dibentuk dengan *Gaussian function*.

Berikut adalah *source* program Y15 beserta grafik:

```
if((data_sudut1Y>=60)&&(data_sudut1Y<=90))
{
    return Math.pow(e,(-Math.pow((data_sudut1Y - 67.5),2) / 16));
}
else
{
    return 0;
}
```

Grafik nilai sudut Y15 dapat dilihat pada gambar di bawah ini:



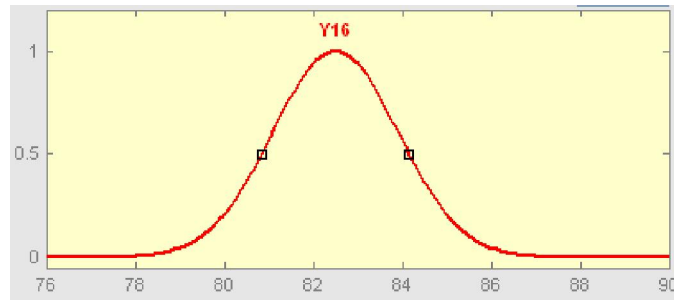
Gambar 3.30 *Membership function* Y15

18. *Membership function* Y16 dibentuk dengan *Gaussian function*.

Berikut adalah *source* program Y16 beserta grafik:

```
if((data_sudut1Y>=75)&&(data_sudut1Y<=90))
{
    return Math.pow(e,(-Math.pow((data_sudut1Y - 82.5),2) / 16));
}
else
{
    return 0;
}
```

Grafik nilai sudut Y16 dapat dilihat pada gambar di bawah ini:



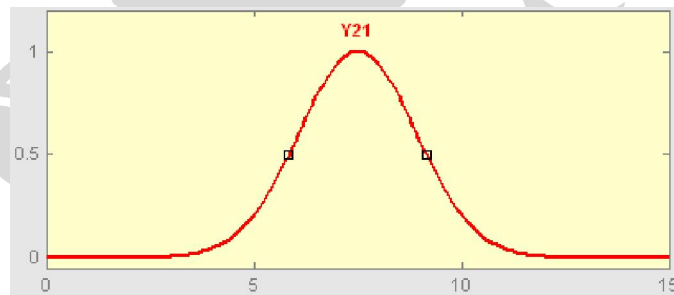
Gambar 3.31 *Membership function Y16*

19. *Membership function Y21* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program Y21 beserta grafik:

```
if((data_sudut2Y >= 0) && (data_sudut2Y <= 15))  
{  
    return Math.pow(e, (-Math.pow((data_sudut2Y - 7.5), 2) / 16));  
}  
else  
{  
    return 0;  
}
```

Grafik nilai sudut Y21 dapat dilihat pada gambar di bawah ini:



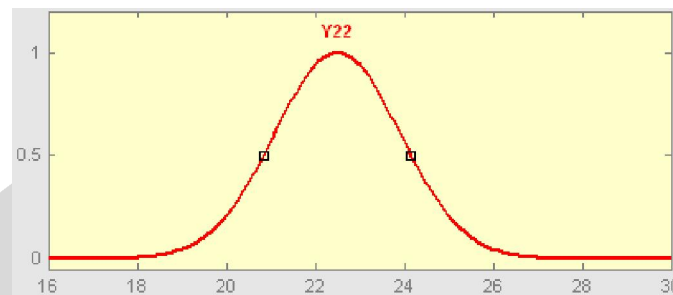
Gambar 3.32 *Membership function Y21*

20. *Membership function Y22* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program Y22 beserta grafik:

```
if((data_sudut2Y>=15)&&(data_sudut2Y<=30))
{
    return Math.pow(e,(-Math.pow((data_sudut2Y - 22.5),2) / 16));
}
else
{
    return 0;
}
```

Grafik nilai sudut Y22 dapat dilihat pada gambar di bawah ini:



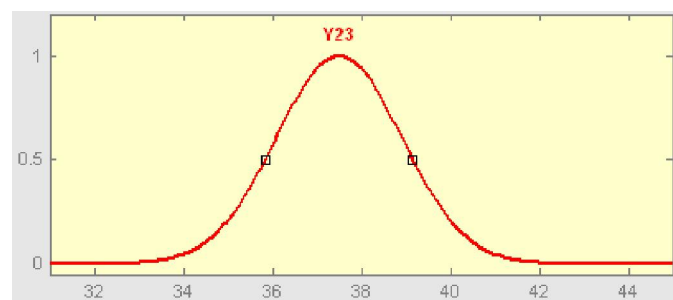
Gambar 3.33 *Membership function Y22*

21. *Membership function Y23* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program Y23 beserta grafik:

```
if((data_sudut2Y>=30)&&(data_sudut2Y<=45))
{
    return Math.pow(e,(-Math.pow((data_sudut2Y - 37.5),2) / 16));
}
else
{
    return 0;
}
```

Grafik nilai sudut Y23 dapat dilihat pada gambar di bawah ini:



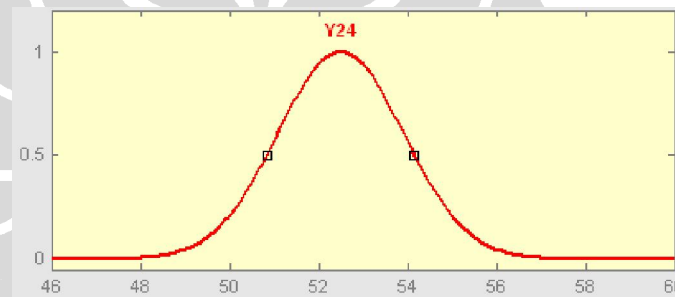
Gambar 3.34 *Membership function Y23*

22. *Membership function Y24* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program Y24 beserta grafik:

```
if((data_sudut2Y >= 45) && (data_sudut2Y <= 60))  
{  
    return Math.pow(e, (-Math.pow((data_sudut2Y - 52.5), 2) / 16));  
}  
else  
{  
    return 0;  
}
```

Grafik nilai sudut Y24 dapat dilihat pada gambar di bawah ini:



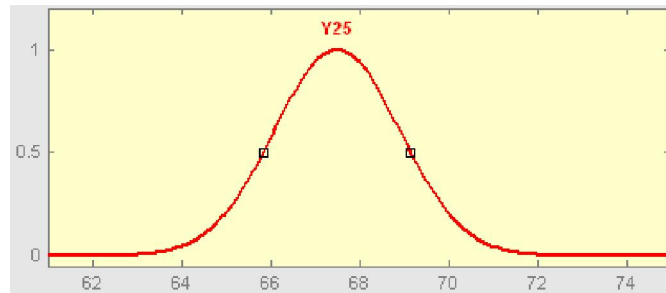
Gambar 3.35 *Membership function Y24*

23. *Membership function Y25* dibentuk dengan *Gaussian function*.

Berikut adalah *source* program Y25 beserta grafik:

```
if((data_sudut2Y >= 60) && (data_sudut2Y <= 75))  
{  
    return Math.pow(e, (-Math.pow((data_sudut2Y - 67.5), 2) / 16));  
}  
else  
{  
    return 0;  
}
```

Grafik nilai sudut Y25 dapat dilihat pada gambar di bawah ini:



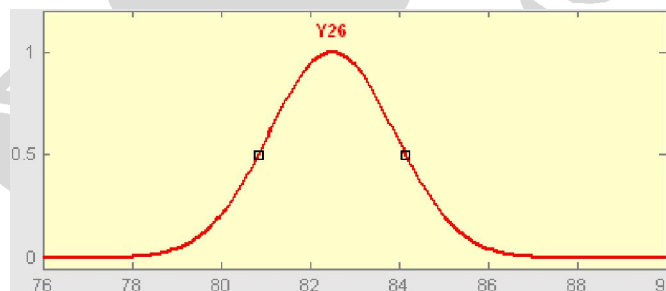
Gambar 3.36 Membership function Y25

24. Membership function Y26 dibentuk dengan Gaussian function.

Berikut adalah source program Y26 beserta grafik:

```
if((data_sudut2Y >= 75) && (data_sudut2Y <= 90))  
{  
    return Math.pow(e, (-Math.pow((data_sudut2Y - 82.5), 2) / 16));  
}  
else  
{  
    return 0;  
}
```

Grafik nilai sudut Y26 dapat dilihat pada gambar di bawah ini:



Gambar 3.37 Membership function Y26

Nilai *output* yang digunakan pada rancangan sistem iterasi kedua ini mengalami perubahan. Penulis membuat *output* tidak didapatkan dari persamaan linear ataupun konstan seperti pada rancangan *fuzzy* Sugeno lainnya melainkan dari persamaan trigonometri. Berikut adalah persamaan trigonometri *output* sistem:

$$\text{piksel X citra} = \tan \theta + (\text{pikselX} * 1280) \quad (3.8)$$

$$\text{piksel Y citra} = \tan \theta + (\text{pikselY} * 960) \quad (3.9)$$

if (sudut1X <= 45) && (sudut2X <= 45) && (2.4 <= (pikselX₁/pikselX₂) <= 2.7)

$$\text{piksel X citra} = \tan \theta + (\text{pikselX} * 1280) + ((\text{pikselX}_1 * \cos \theta_{1x}) - (\text{pikselX}_2 * \cos \theta_{2x}) / (\text{pikselX}_1/\text{pikselX}_2) * \cos \theta_{1x})$$

if (sudut1X <= 45) && (sudut2X <= 45) && (2.7 <= (pikselX₁/pikselX₂) <= 3.0)

$$\text{piksel X citra} = \tan \theta + (\text{pikselX} * 1280) + ((\text{pikselX}_1 * \cos \theta_{1x}) - (\text{pikselX}_2 * \cos \theta_{2x}) / (\text{pikselX}_1/\text{pikselX}_2))$$

if (sudut1Y >= 46) && (sudut2Y <= 15)

$$\text{piksel Y citra} = \tan \theta + (\text{pikselY} * 960) - ((\text{pikselY}_1 * \sin \theta_{1y}) - (\text{pikselY}_2 * \sin \theta_{2y}) * (\text{pikselY}_1)^2)$$

if(sudut1Y<=46) && (sudut2Y <=46) && (pikselY₁/960>=0.6) && (pikselY₂>=0.6)

$$\text{piksel Y citra} = \tan \theta + (\text{pikselY} * 960) - ((\text{pikselY}_1 / \text{pikselY}_2) * ((\text{pikselY}_1 + \text{pikselY}_2) / 2) * (\sin \theta_{1y} * \sin \theta_{2y})^2) + \text{pikselY}_1 - \text{pikselY}_2$$

Penulis melakukan persamaan tambahan disebabkan rumusan awal tidak tepat untuk beberapa piksel yang terletak pada citra *long range* CCTV. Dengan adanya persamaan tambahan tersebut maka hasil pemetaan menjadi lebih akurat.

BAB IV

UJI COBA DAN ANALISA

Bab ini berisi mengenai eksperimen yang dilakukan beserta analisa terhadap hasil yang didapatkan. Pembahasan meliputi data, skenario percobaan, dan hasil uji coba beserta analisa data yang didapatkan.

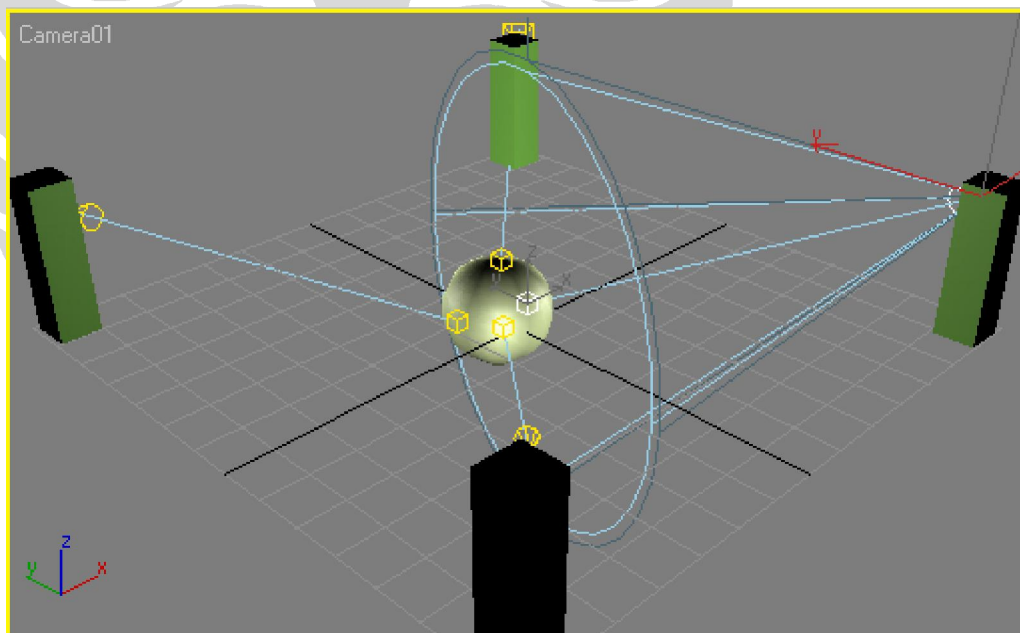
4.1 Data Uji Coba

Data uji coba dilakukan pada simulasi yang dibuat oleh penulis. Hal ini disebabkan karena sulitnya mencari data asli untuk melakukan percobaan. Pencitraan dilakukan oleh 2 kamera sebanyak 15 kali pemotretan. Berikut adalah sudut – sudut yang diambil oleh pemotretan:

1. Kamera dengan sudut 45 derajat terhadap sumbu X dan 0 derajat terhadap sumbu Y dari titik awal (kiri bawah);
2. Kamera dengan sudut 60 derajat terhadap sumbu X dan 0 derajat terhadap sumbu Y dari titik awal (kiri bawah);
3. Kamera dengan sudut 45 derajat terhadap sumbu X dan 45 derajat terhadap sumbu Y dari titik awal (kiri bawah);
4. Kamera dengan sudut 45 derajat terhadap sumbu X dan 60 derajat terhadap sumbu Y dari titik awal (kiri bawah);
5. Kamera dengan sudut 60 derajat terhadap sumbu X dan 45 derajat terhadap sumbu Y dari titik awal (kiri bawah);
6. Kamera dengan sudut 60 derajat terhadap sumbu X dan 60 derajat terhadap sumbu Y dari titik awal (kiri bawah);
7. Kamera dengan sudut -45 derajat terhadap sumbu X dan 0 derajat terhadap sumbu Y dari titik kedua (kanan bawah);
8. Kamera dengan sudut -60 derajat terhadap sumbu X dan 0 derajat terhadap sumbu Y dari titik kedua (kanan bawah);
9. Kamera dengan sudut -45 derajat terhadap sumbu X dan 45 derajat

- terhadap sumbu Y dari titik kedua (kanan bawah);
10. Kamera dengan sudut -45 derajat terhadap sumbu X dan 60 derajat terhadap sumbu Y dari titik kedua (kanan bawah);
 11. Kamera dengan sudut -60 derajat terhadap sumbu X dan 45 derajat terhadap sumbu Y dari titik kedua (kanan bawah);
 12. Kamera dengan sudut -60 derajat terhadap sumbu X dan 60 derajat terhadap sumbu Y dari titik kedua (kanan bawah);
 13. 3 gambar pemotretan dari atas objek yang dicitrakan. Pemotretan mencakup bidang beserta objek yang dicitrakan.

Gambar simulasi percobaan dapat dilihat pada bagian lampiran. Penulis juga menggambarkan pemodelan simulasi sebagai data percobaan. Berikut adalah gambar pemodelan simulasi data percobaan:



Gambar 4.1 Pemodelan simulasi

4.2 Skenario Percobaan

Skenario percobaan dilakukan dengan mengambil secara acak titik yang dicari pada kamera 1 (pemotretan dari titik awal), kamera 2

(pemotretan dari titik kedua) dan pemotretan dari atas. Setiap piksel yang berkorespondensi dicari secara manual. Piksel pada kamera 1 dan kamera 2 dimasukkan pada sistem hasil yang didapatkan dicocokkan dengan pemotretan citra yang dilakukan dari atas.

4.3 Hasil Uji Coba

Uji coba dilakukan dengan mengambil titik piksel secara random pada kamera 1 dan kamera 2 lalu dicocokkan hasilnya dengan citra digital (citra 1, citra 2, citra 3). Titik piksel beserta sudut pencitraan yang diambil pada kamera 1 dan kamera 2 dimasukkan ke dalam sistem, sistem melakukan penghitungan dan memberikan hasil.



Gambar di samping adalah gambar pemotretan kamera 1 ($45^0,0^0$). Piksel yang dicari diberikan lingkaran merah dan terletak pada nilai (669,521).



Gambar di samping adalah gambar pemotretan kamera 2 ($60^{06},0^0$). Piksel yang dicari diberikan lingkaran merah dan terletak pada nilai (869,520).



Gambar di samping adalah gambar pemotretan citra 1. Piksel yang dicari diberikan lingkaran merah dan terletak pada nilai (811,373).



Gambar di samping adalah gambar pemotretan citra 2. Piksel yang dicari diberikan lingkaran merah dan terletak pada nilai (832,319).



Gambar di samping adalah gambar pemotretan citra 3. Piksel yang dicari diberikan lingkaran merah dan terletak pada nilai (831,405).

Hasil yang diberikan sistem iterasi pertama dari kedua gambar kamera (kamera 1 dan kamera 2) terletak pada nilai (820,521) sedangkan hasil iterasi dua memberikan nilai (870,521). Masing-masing dari gambar tersebut berkorespondensi dengan baris pertama table 4.1.

Penulis menampilkan 2 tabel hasil yang dilakukan. Tabel 4.1 menggambarkan hasil percobaan 30 pengambilan piksel secara acak yang telah dilakukan terhadap sistem pada iterasi satu:

Tabel 4.1 Nilai rancangan sistem iterasi pertama

No	Kamera 1	Kamera 2	Citra 1	Citra 2	Citra 3	Hasil	Pemotretan
1	(669,521)	(869,520)	(811,373)	(832,319)	(831,405)	(820,521)	Kamera 1 (45 ⁰ , 0 ⁰) Kamera 2 (60 ⁰ , 60 ⁰)
2	(503,365)	(1073,299)	(1085,343)	(1109,292)	(1113,384)	(936,347)	Kamera 1 (45 ⁰ , 0 ⁰) Kamera 2 (60 ⁰ , 60 ⁰)
3	(364,349)	(412,451)	(509,496)	(530,449)	(530, 534)	(496,492)	Kamera 1 (60 ⁰ , 0 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)
4	(558,457)	(513,615)	(486, 527)	(506,477)	(506,560)	(514,615)	Kamera 1 (60 ⁰ , 0 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)
5	(1003, 455)	(409, 454)	(518,402)	(537,355)	(532,428)	(520,454)	Kamera 1 (45 ⁰ , 60 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)
6	(961,571)	(346,584)	(499,390)	(517,336)	(513,420)	(506,498)	Kamera 1 (45 ⁰ , 60 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)
7	(283,190)	(996,592)	(808,495)	(830,445)	(827,527)	(997,593)	Kamera 1 (60 ⁰ , 60 ⁰) Kamera 2 (60 ⁰ , 60 ⁰)
8	(488,641)	(379,695)	(407,266)	(427,214)	(430,291)	(320,696)	Kamera 1 (60 ⁰ , 60 ⁰) Kamera 2 (60 ⁰ , 60 ⁰)
9	(808,327)	(775,690)	(736,443)	(759,388)	(762,475)	(776,408)	Kamera 1 (45 ⁰ , 60 ⁰) Kamera 2 (60 ⁰ , 45 ⁰)
10	(959,574)	(478,799)	(500,392)	(521,344)	(523,427)	(479,588)	Kamera 1 (45 ⁰ , 60 ⁰) Kamera 2 (60 ⁰ , 45 ⁰)
11	(1124,429)	(1028,545)	(542,483)	(563,434)	(559,517)	(- ,545)	Kamera 1 (45 ⁰ , 0 ⁰) Kamera 2 (60 ⁰ , 0 ⁰)
12	(1052,467)	(859,577)	(499,390)	(525,340)	(522,431)	(809,524)	Kamera 1 (45 ⁰ , 0 ⁰) Kamera 2 (60 ⁰ , 0 ⁰)
13	(432,366)	(1028,545)	(542,483)	(563,434)	(559,517)	(584,545)	Kamera 1 (60 ⁰ , 0 ⁰) Kamera 2 (60 ⁰ , 60 ⁰)
14	(357,109)	(1018,273)	(557,445)	(578,392)	(572,489)	(563,501)	Kamera 1 (64 ⁰ , 0 ⁰) Kamera 2 (60 ⁰ , 0 ⁰)

15	(1021,808)	(383,908)	(508,391)	(529,343)	(529,432)	(380,-)	Kamera 1 (45 ⁰ , 45 ⁰) Kamera 2 (45 ⁰ , 45 ⁰)
16	(693,601)	(621,725)	(809,374)	(829,319)	(830,404)	(622,-)	Kamera 1 (45 ⁰ , 45 ⁰) Kamera 2 (45 ⁰ , 45 ⁰)
17	(1123,428)	(555,493)	(542,483)	(563,434)	(559,517)	(556,493)	Kamera 1 (45 ⁰ , 0 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)
18	(1090,262)	(555,317)	(541,468)	(559,416)	(557,507)	(552,320)	Kamera 1 (45 ⁰ , 0 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)
19	(810,681)	(688,683)	(699,400)	(716,347)	(718,432)	(689,684)	Kamera 1 (45 ⁰ , 45 ⁰) Kamera 2 (60 ⁰ , 45 ⁰)
20	(776,886)	(441,697)	(560,271)	(577,219)	(580,300)	(611,778)	Kamera 1 (45 ⁰ , 45 ⁰) Kamera 2 (60 ⁰ , 45 ⁰)
21	(832,553)	(998,593)	(807,494)	(830,441)	(826,527)	(916,563)	Kamera 1 (45 ⁰ , 45 ⁰) Kamera 2 (60 ⁰ , 60 ⁰)
22	(1231,767)	(807,898)	(494,577)	(513,527)	(512,608)	(622,657)	Kamera 1 (45 ⁰ , 45 ⁰) Kamera 2 (60 ⁰ , 60 ⁰)
23	(495,339)	(429,491)	(509,496)	(530,449)	(530,534)	(430,339)	Kamera 1 (60 ⁰ , 0 ⁰) Kamera 2 (45 ⁰ , 60 ⁰)
24	(430,365)	(449,467)	(542,485)	(563,438)	(559,520)	(450,468)	Kamera 1 (60 ⁰ , 0 ⁰) Kamera 2 (45 ⁰ , 60 ⁰)
25	(285,189)	(755,782)	(804,497)	(827,443)	(827,529)	(756,191)	Kamera 1 (60 ⁰ , 60 ⁰) Kamera 2 (45 ⁰ , 45 ⁰)
26	(488,639)	(114,950)	(411,267)	(427,215)	(433,293)	(115,640)	Kamera 1 (60 ⁰ , 60 ⁰) Kamera 2 (45 ⁰ , 45 ⁰)
27	(126,665)	(408,255)	(766,262)	(784,207)	(787,290)	(409,256)	Kamera 1 (60 ⁰ , 45 ⁰) Kamera 2 (45 ⁰ , 60 ⁰)
28	(209,559)	(540,281)	(807,376)	(828,320)	(832,407)	(541,282)	Kamera 1 (60 ⁰ , 45 ⁰) Kamera 2 (45 ⁰ , 60 ⁰)
29	(527,481)	(516,318)	(546,435)	(569,384)	(564,477)	(528,387)	Kamera 1 (60 ⁰ , 45 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)
30	(576,556)	(560,490)	(540,484)	(562,435)	(559,519)	(559,490)	Kamera 1 (60 ⁰ , 45 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)

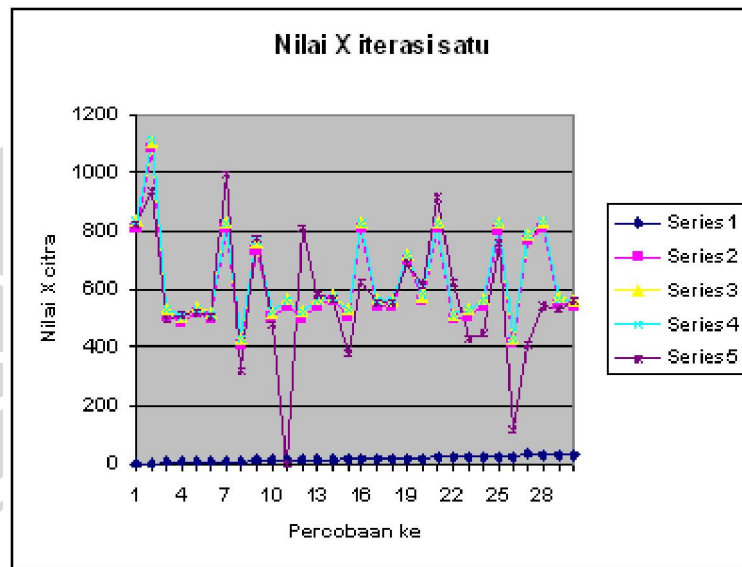
Sedangkan hasil rancangan sistem iterasi kedua dapat dilihat pada tabel dibawah ini :

Tabel 4.2 Nilai rancangan sistem iterasi kedua

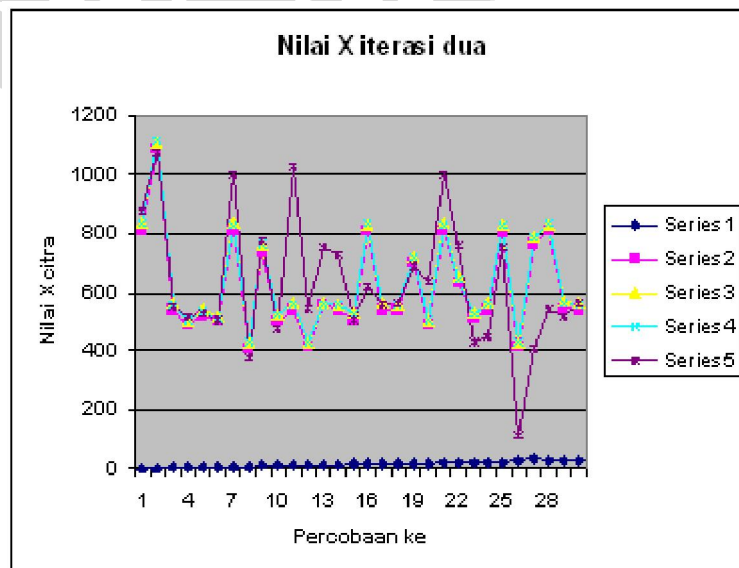
No	Kamera 1	Kamera 2	Citra 1	Citra 2	Citra 3	Hasil	Pemotretan
1	(669,521)	(869,520)	(811,373)	(832,319)	(831,405)	(870,521)	Kamera 1 (45 ⁰ , 0 ⁰) Kamera 2 (60 ⁰ , 60 ⁰)
2	(503,365)	(1073,299)	(1085,343)	(1109,292)	(1113,384)	(1074,322)	Kamera 1 (45 ⁰ , 0 ⁰) Kamera 2 (60 ⁰ , 60 ⁰)
3	(432,366)	(551,320)	(542,483)	(563,434)	(559,517)	(551,342)	Kamera 1 (60 ⁰ , 0 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)
4	(558,457)	(513,615)	(486,527)	(506,477)	(506,560)	(514,535)	Kamera 1 (60 ⁰ , 0 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)

5	(1003, 455)	(409, 454)	(518,402)	(537,355)	(532,428)	(531,367)	Kamera 1 (45 ⁰ , 60 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)
6	(961,571)	(346,584)	(499,390)	(517,336)	(513,420)	(503,401)	Kamera 1 (45 ⁰ , 60 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)
7	(283,190)	(996,592)	(808,495)	(830,445)	(827,527)	(997,393)	Kamera 1 (60 ⁰ , 60 ⁰) Kamera 2 (60 ⁰ , 60 ⁰)
8	(488,641)	(379,695)	(407,266)	(427,214)	(430,291)	(380,670)	Kamera 1 (60 ⁰ , 60 ⁰) Kamera 2 (60 ⁰ , 60 ⁰)
9	(808,327)	(775,690)	(736,443)	(759,388)	(762,475)	(776, 510)	Kamera 1 (45 ⁰ , 60 ⁰) Kamera 2 (60 ⁰ , 45 ⁰)
10	(959,574)	(478,799)	(500,392)	(521,344)	(523,427)	(479, 688)	Kamera 1 (45 ⁰ , 60 ⁰) Kamera 2 (60 ⁰ , 45 ⁰)
11	(1124,429)	(1028,545)	(542,483)	(563,434)	(559,517)	(1029,486)	Kamera 1 (45 ⁰ , 0 ⁰) Kamera 2 (60 ⁰ , 0 ⁰)
12	(1030,609)	(546,618)	(411,268)	(425,217)	(430,296)	(547,613)	Kamera 1 (45 ⁰ , 0 ⁰) Kamera 2 (60 ⁰ , 0 ⁰)
13	(432,366)	(1028,545)	(559,517)	(563,434)	(559,517)	(755,409)	Kamera 1 (64 ⁰ , 0 ⁰) Kamera 2 (60 ⁰ , 0 ⁰)
14	(398,254)	(1013,425)	(541,469)	(559,417)	(559,504)	(731,339)	Kamera 1 (64 ⁰ , 0 ⁰) Kamera 2 (60 ⁰ , 0 ⁰)
15	(1021, 808)	(383,908)	(508,391)	(529,343)	(529,432)	(503,374)	Kamera 1 (45 ⁰ , 45 ⁰) Kamera 2 (45 ⁰ , 45 ⁰)
16	(693,601)	(621,725)	(809,374)	(829,319)	(830,404)	(621,229)	Kamera 1 (45 ⁰ , 45 ⁰) Kamera 2 (45 ⁰ , 45 ⁰)
17	(1123,428)	(555,493)	(542,483)	(563,434)	(559,517)	(556,460)	Kamera 1 (45 ⁰ , 0 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)
18	(1101,270)	(551,320)	(541,468)	(559,416)	(557,507)	(563, 501)	Kamera 1 (45 ⁰ , 0 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)
19	(810,681)	(688,683)	(699,400)	(716,347)	(718,432)	(689,396)	Kamera 1 (45 ⁰ , 45 ⁰) Kamera 2 (60 ⁰ , 45 ⁰)
20	(1239,496)	(636,571)	(487,500)	(508,443)	(503, 542)	(637,535)	Kamera 1 (45 ⁰ , 45 ⁰) Kamera 2 (60 ⁰ , 45 ⁰)
21	(832,553)	(998,593)	(807, 494)	(830, 441)	(826,527)	(999,575)	Kamera 1 (45 ⁰ , 45 ⁰) Kamera 2 (60 ⁰ ,60 ⁰)
22	(916,447)	(759,379)	(632,373)	(650,320)	(645,415)	(760,415)	Kamera 1 (45 ⁰ , 45 ⁰) Kamera 2 (60 ⁰ , 60 ⁰)
23	(495,339)	(429,491)	(509,496)	(530,449)	(530,534)	(429,441)	Kamera 1 (60 ⁰ , 0 ⁰) Kamera 2 (45 ⁰ , 60 ⁰)
24	(430,365)	(449,467)	(542,485)	(563,438)	(559,520)	(449,434)	Kamera 1 (60 ⁰ , 0 ⁰) Kamera 2 (45 ⁰ , 60 ⁰)
25	(285,189)	(755,782)	(804,497)	(827,443)	(827,529)	(755,487)	Kamera 1 (60 ⁰ , 60 ⁰) Kamera 2 (45 ⁰ ,45 ⁰)
26	(488,639)	(114,950)	(411,267)	(427,215)	(433,293)	(115,796)	Kamera 1 (60 ⁰ , 60 ⁰) Kamera 2 (45 ⁰ , 45 ⁰)
27	(126,665)	(408,255)	(766,262)	(784,207)	(787,290)	(408,462)	Kamera 1 (60 ⁰ ,45 ⁰) Kamera 2 (45 ⁰ , 60 ⁰)
28	(209,559)	(540,281)	(807,376)	(828,320)	(832,407)	(540,422)	Kamera 1 (60 ⁰ , 45 ⁰) Kamera 2 (45 ⁰ , 60 ⁰)
29	(527,481)	(516,318)	(546,435)	(569,384)	(564,477)	(516,427)	Kamera 1 (60 ⁰ , 45 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)
30	(576,556)	(560,490)	(540,484)	(562,435)	(559,519)	(561,534)	Kamera 1 (60 ⁰ , 45 ⁰) Kamera 2 (45 ⁰ , 0 ⁰)

Pada gambar di atas dapat dilihat bahwa rancangan sistem iterasi kedua mempunyai tingkat akurasi lebih tinggi. Tingkat akurasi pada rancangan sistem kedua sebesar 66.67% sedangkan tingkat akurasi dari rancangan sistem pertama sebesar : 56%. Penulis menggambarkan pula grafik yang dibentuk dari hasil pada tabel di atas. Berikut adalah tampilan grafik perbandingan antara nilai X iterasi pertama dengan iterasi kedua:

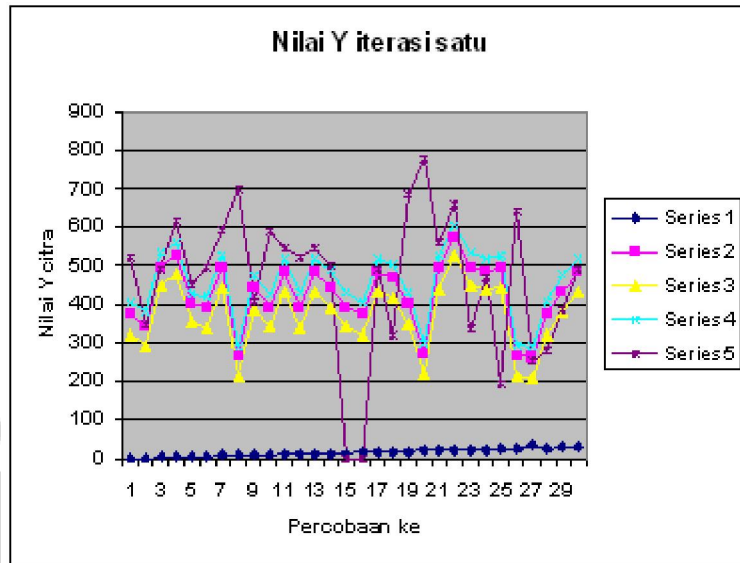


Gambar 4.2 Nilai X iterasi pertama

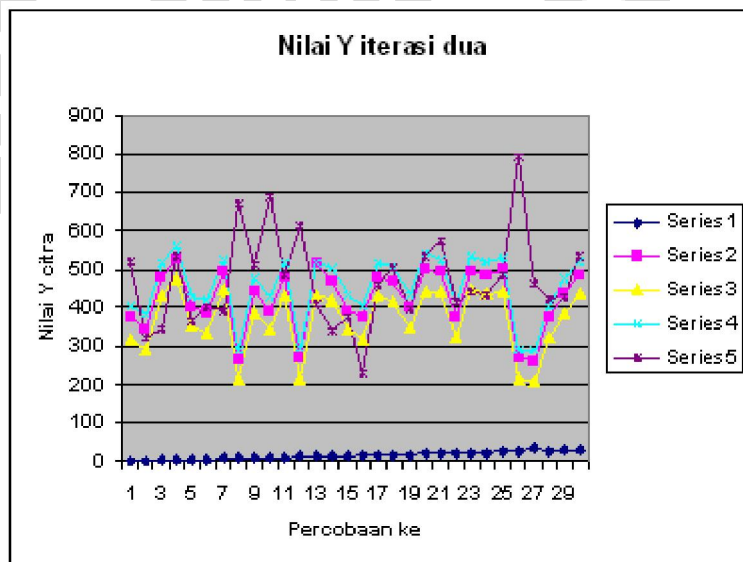


Gambar 4.3 Nilai X iterasi kedua

Perbandingan nilai Y iterasi pertama dengan iterasi kedua dapat dilihat pada gambar di bawah ini:



Gambar 4.4 Nilai Y iterasi pertama



Gambar 4.5 Nilai Y iterasi kedua

Dari hasil uji coba yang dilakukan, dapat disimpulkan, terjadinya ketidakcocokan piksel dapat disebabkan oleh hal-hal berikut:

1. Kesalahan sudut pemotretan yang dilakukan oleh penulis;
2. Tidak ada standard yang digunakan dalam pemotretan dari atas;
3. Kurang lengkapnya aturan *fuzzy* yang dipakai dalam permasalahan *stereo vision*;
4. Pemodelan *membership function* terhadap *domain* permasalahan.

4.4 Analisa

Dari hasil uji coba yang dilakukan, metode *fuzzy* yang digunakan tidak selalu memberikan hasil yang tepat pada pemotretan dari atas. Rentang toleransi piksel yang digunakan adalah sebesar ± 30 piksel (perbedaan pemotretan citra dari atas). Penulis menyadari bahwa ketidakcocokan piksel di atas selain pada sudut pemotretan terdapat pula pada aturan yang digunakan. Pendekatan keakurasian metode yang digunakan menggunakan pendekatan deskriptif yaitu menilai keakurasian berdasarkan data simulasi yang telah diujicobakan. Metode 1 memberikan keakuratan sebesar 56% sedangkan untuk metode kedua memberikan keakuratan sebesar 66 %. Penghitungan keakuratan berdasarkan letak koordinat akhir hasil *output* dengan pixel yang terdapat pada pemotretan dari atas. Apabila pixel *output* terletak antara ke-3 pixel pemotretan citra dari atas maka dapat dinyatakan bahwa *output* sesuai dengan hasil yang sebenarnya.

Aturan *fuzzy* yang digunakan pada iterasi pertama menggunakan metode Sugeno dan hasil *output* didapatkan dari persamaan linear. Persamaan linear tersebut kurang cocok digunakan dalam rumus *stereo vision (triangulation)* yang dipakai. Persamaan yang sesuai untuk metode *triangulation* adalah persamaan trigonometri. Pada *toolbox fuzzy logic* yang disediakan matlab, tidak terdapat perumusan *output* berdasarkan persamaan trigonometri; perumusan *output* yang terdapat pada toolbox tersebut adalah

perumusan *output* dari konstanta dan persamaan linear. Oleh karena itu, penulis berusaha untuk mencari persamaan linear yang mendekati persamaan trigonometri yang seharusnya.

Sedangkan aturan *fuzzy* yang digunakan pada iterasi kedua menggunakan metode trigonometri Sugeno, hasil *output* didapatkan dari persamaan trigonometri, dan pengubahan *membership function* sudut yang dibentuk antara citra *long range* CCTV dengan objek. Terdapat pula beberapa aturan tambahan untuk pemetaan citra pada sudut–sudut tertentu dengan piksel- piksel tertentu. Aturan–aturan tambahan tersebut masih dapat ditambahkan untuk mencapai tingkat akurasi yang lebih tinggi.

