BAB 3

ANALISIS DAN KEBUTUHAN

3.1. Analisis

Analisis adalah proses mengurai konsep ke dalam bagian – bagian yang lebih sederhana. Sehingga struktur logisnya menjadi jelas. Analisis merupakan metode untuk menguji, menilai dan memahami sistem pemikiran yang kompleks dengan memecahkannnya ke dalam unsur – unsur yang lebih sederhana sehingga hubungan antar unsur menjadi lebih jelas.

3.1.1. Analisis Masalah

Jaringan syaraf tiruan merupakan salah satu metode untuk klasifikasi seperti pada kasus klasifikasi citra berdasarkan warna RGB untuk menentukan tingat kematangan buah. Beberapa metode jaringan syaraf tiruan yang digunakan untuk klasifikasi adalah ANFIS, namun terbukti bahwa klasifikasi tingkat kematangan buah berdasarkan warna RGB dengan ANFIS masih memiliki kegagalan, metode jaringan syaraf tiruan lainnya yang digunakan untuk melakukan klasifikasi berdasarkan warna RGB dari suatu citra adalah metode LVQ yang memiliki ketepatan yang baik dalam klasifikasi. Untuk mengetahui persentase hasil klasifikasi mana yang akan lebih baik antara ANFIS dan LVQ, maka kedua metode jaringan syaraf tiruan tersebut akan dibandingkan dengan data masukan yang sama dan parameter yang sama yaitu warna RGB.

3.1.2. Analisis Masukan Data

Data yang akan berfungsi sebagai data masukan adalah citra tomat dengan kondisi pada *stage* 1 hingga *stage* 6. Citra tomat tersebut memiliki parameter yang sama yang akan digunakan untuk klasifikasi yaitu ketiga unsur warna *Red-Green-Blue* (RGB) dari setiap pixel dengan nilai kedalaman warna antara 0 – 255.

Jumlah citra tomat yang digunakan sebagai data masukan adalah 120 citra sebagai data pembelajaran dan 24 citra sebagai data pengujian. Tiap citra diolah dengan pengolahan citra sebagai cara untuk mendapatkan nilai RGB dari citra tersebut.

Dalam penelitian tugas akhir ini pengolahan citra yang digunakan untuk analisis data adalah sebagai berikut :

1. Gambar masukan sebanyak 144 data berupa citra tomat dimana 120 data sebagai data pembelajaran dan 24 data sebagai data uji.
2. Semua citra melalui proses pemotongan citra (*cropping*) dari ukuran sebenarnya menjadi ukuran 300x300 pixel.
3. Kemudian citra hasil *cropping* melalui proses ekstraksi ciri untuk mendapatkan nilai RGB dari citra tomat dengan cara mengambil nilai nilai R, G dan B dari tiap pixel lalu dibagi dengan total R,G dan B dari citra.
4. Nilai RGB tomat disimpan dalam *database*.
5. Nilai RGB yang ada dalam *database* digunakan untuk proses pembelajaran dan pengujian LVQ dan ANFIS.

Berdasarkan analisis data yang telah dilakukan terhadap 144 citra didapatkan *range* nilai RGB untuk tiap *stage* pada Tabel 3.1.

Tabel 3.1 *Range* Setiap *Stage*

Stage	*Red*			*Green*			*Blue*		
	Min	*Max*	*Avg*	*Min*	*Max*	*Avg*	*Min*	*Max*	*Avg*
1	61	126	98	72	136	108	14	56	33
2	77	157	114	67	128	55	22	55	36
3	74	160	119	41	113	78	6	54	27
4	99	209	155	25	107	61	5	64	22
5	111	172	143	21	72	40	7	28	14
6	110	198	149	19	59	33	6	44	19

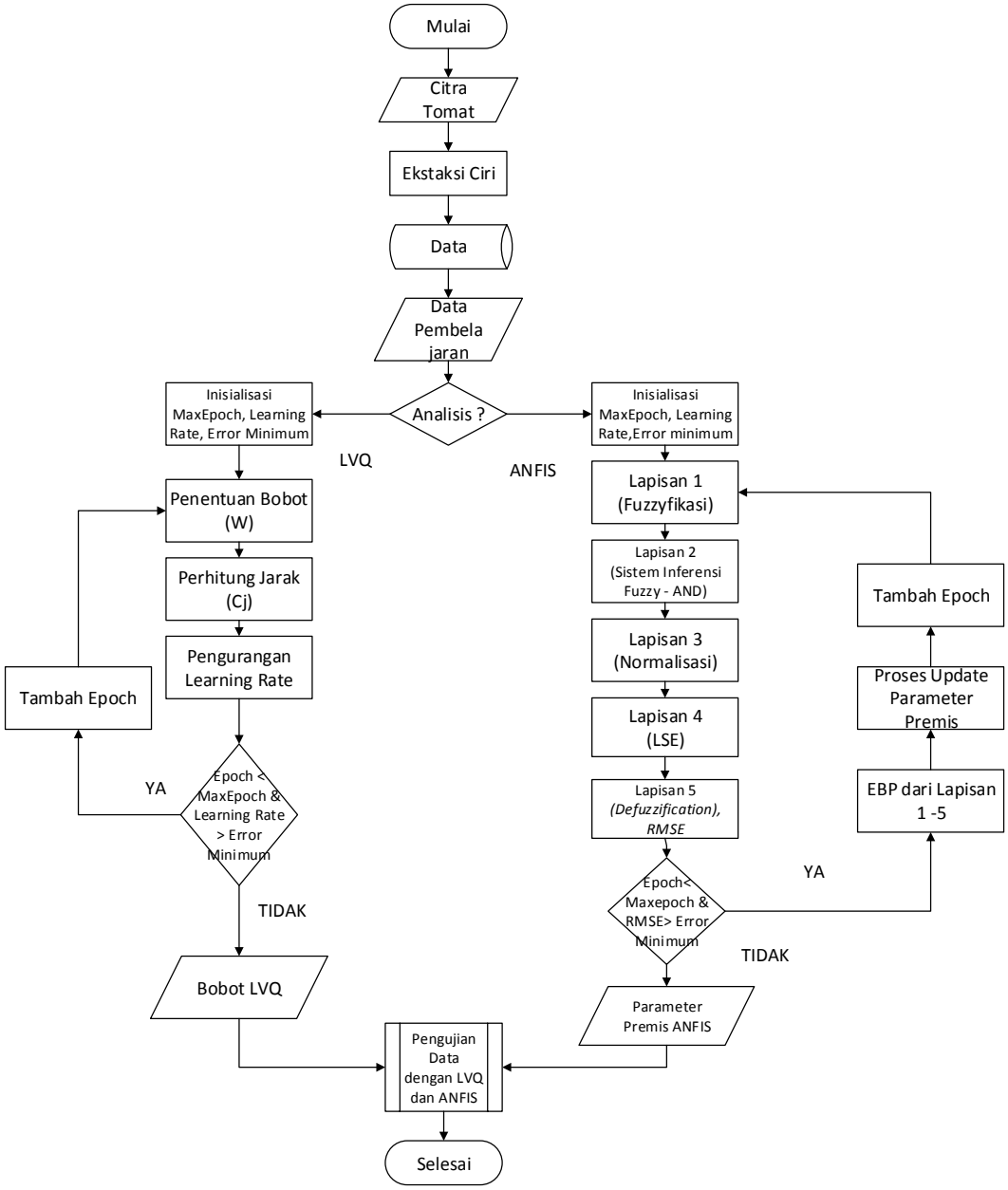
Setiap nilai *min* (*minimum*) hingga *max* (*maximum*) merupakan batasan nilai dari setiap unsur warna pada setiap *stage* dan *avg* (*average)* merupakan nilai rata – rata dari tiap unsur warna pada tiap *stage*. Batasan pada tiap stage digunakan untuk menentukan batasan untuk fungsi keanggotaan.

3.1.3. Analisis Sistem

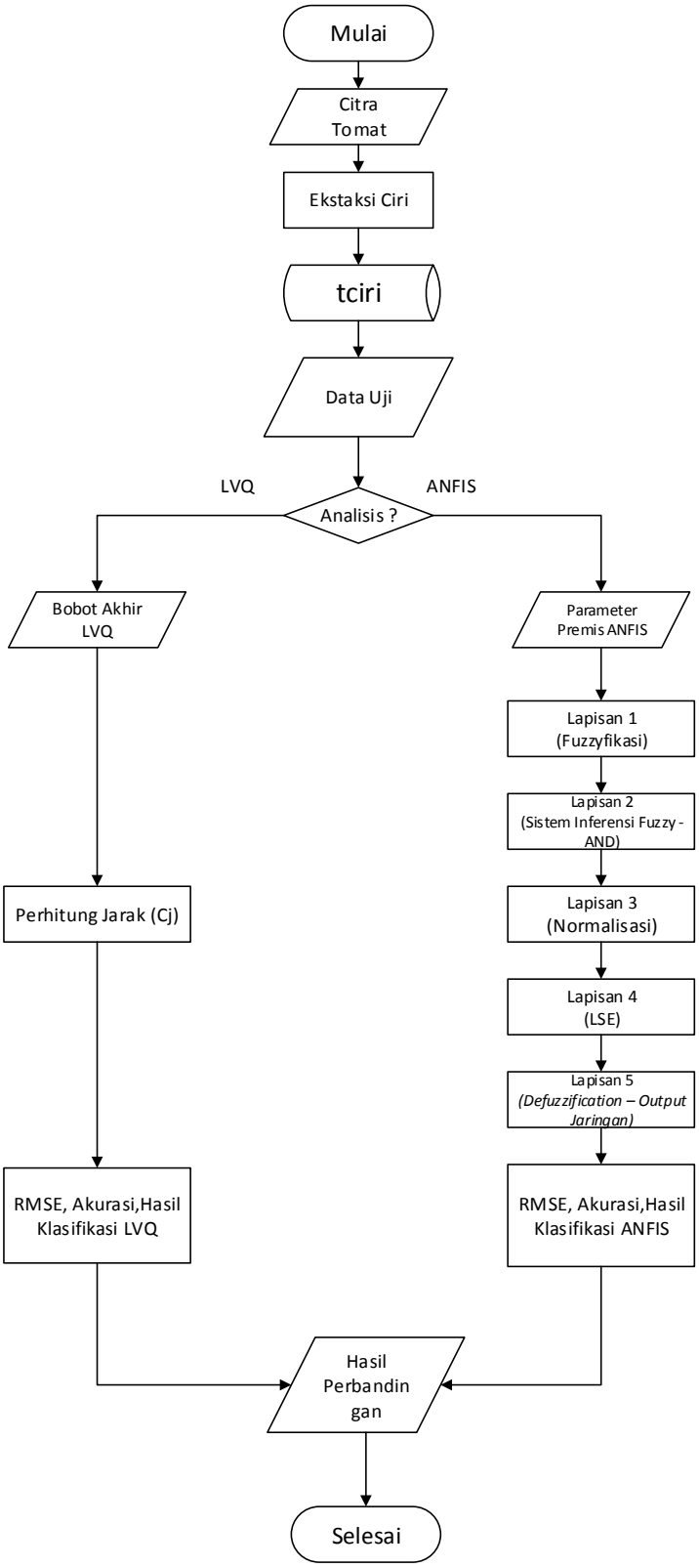
Perangkat lunak yang akan dibangun adalah sebuah simulator digunakan untuk menganalisis perbandingan tingkat akurasi hasil klasifikasi dari metode LVQ dan ANFIS pada kasus klasifikasi tingkat kematangan tomat. Adapun pada penjelasan isi simulator ini adalah sebagai berikut :

1. Pada sistem ini terdapat fungsionalitas pembelajaran data untuk melatih data-data citra tomat ke simulator serta fungsionalitas pengujian data untuk menguji keakuratan klasifikasi pada data uji.
2. Data yang diolah adalah data citra digital dari beberapa tomat yang akan digunakan sebagai data pembelajaran dan data uji. Data pembelajaran tersebut berguna untuk proses pembelajaran metode LVQ dan ANFIS. Sebelumnya citra digital tersebut akan melalui tahap ekstaksi ciri guna menghasilkan unsur RGB dari citra yang akan digunakan sebagai data untuk metode klasifikasi. Metode LVQ dan ANFIS akan menggunakan data tersebut untuk mengetahui tingkat kematangan tomat dari citra tomat tersebut. Setelah proses pembelajaran selesai akan didapatkan bobot dari masing – masing metode pembelajaran. Tahap selanjutnya adalah proses pengujian yang akan menguji data uji dengan bobot dari hasil pembelajaran untuk proses pengujian data dalam proses pengujian ini akan ditentukan persentase tingkat akurasi, RMSE serta hasil dari kedua metode klasifikasi yang digunakan.
3. Keluaran dari simulator ini adalah menunjukan metode mana yang akan menghasilkan persentase tingkat akurasi serta RMSE dari tiap metode beserta hasil klasifikasinya.

Adapun alur proses dari keseluruhan proses simulator dapat digambarkan kedalam suatu *flowchart* seperti pada Gambar 3.1 dan proses pengujian pada Gambar 3.2 berikut ini :



Gambar 3.1 Proses Pembelajaran Metode LVQ dan ANFIS



Gambar 3.2 Proses Pengujian Metode LVQ dan ANFIS

3.1.4. Analisis Metode

Pada langkah ini akan dilakukan analisis metode yang ada dalam penelitian ini yang meliputi ekstraksi ciri, metode LVQ dan metode ANFIS. Data masukan untuk proses pembelajaran pada analisis metode pada tugas akhir ini diasumsikan berjumlah 18 data dengan hasil ekstraksi ciri yang akan digunakan sebagai parameter masukan untuk tiap metode klasifikasi seperti pada Tabel 3.2 dan data uji yang berjumlah 18 data untuk proses pengujian pada Tabel 3.3 berikut ini :

Tabel 3.2 Data Pembelajaran

No Tomat	R-G-B (Vektor Masukan)	Vektor Target
1	[99 95 47]	1
2	[123 131 33]	1
3	[126 136 29]	1
4	[140 110 52]	2
5	[145 118 55]	2
6	[106 84 36]	2
7	[113 91 47]	3
8	[83 59 15]	3
9	[112 85 35]	3
10	[184 62 22]	4
11	[209 97 32]	4
12	[99 30 5]	4
13	[157 32 13]	5
14	[171 62 22]	5
15	[169 58 19]	5
16	[152 39 15]	6
17	[146 20 10]	6
18	[165 41 17]	6

Tabel 3.3 Data Uji

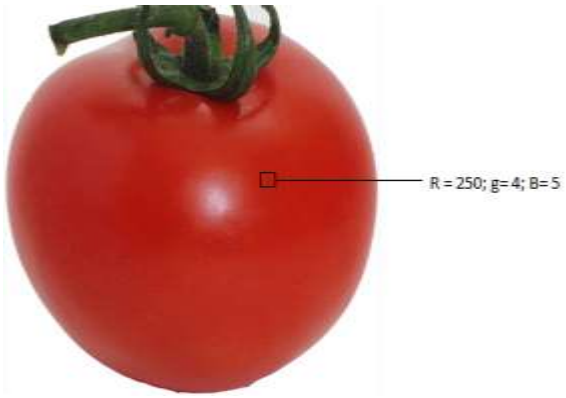
No Tomat	R-G-B (Vektor Masukan)	Stage Dugaan
1	[139 118 28]	2
2	[204 85 28]	4
3	[112 125 43]	1
4	[135 100 32]	3
5	[119 121 21]	1
6	[152 27 8]	6
7	[152 40 10]	5
8	[100 99 33]	1

9	[	207	38	16	]	6
10	[	181	79	15	]	4
11	[	147	120	52	]	3
12	[	92	72	24	]	2
13	[	114	109	42	]	1
14	[	140	116	54	]	2
15	[	74	86	18	]	1
16	[	137	117	27	]	1
17	[	174	119	17	]	3
18	[	167	52	22	]	6

Setiap nilai R, G dan B akan dijadikan sebagai satu data masukan untuk setiap data dalam bentuk bilangan real yang dijadikan vektor masukan dan *stage* yang dijadikan vektor target. Pada vektor target, angka 1 menunjukan *stage* 1, angka 2 menunjukan *stage* 2, angka 3 menujukan *stage* 3 dan seterusnya.

3.1.4.1. Ektraksi Ciri

Ekstaksi ciri dilakukan untuk mendapatkan nilai RGB dari sebuah citra yang akan digunakan sebagai parameter masukan oleh tiap metode analisis. Berikut ini merupakan salah satu contoh dari citra tomat yang akan dihitung nilai RGB-nya.



Gambar 3.3 Nilai RGB Pada Pixel Citra Tomat

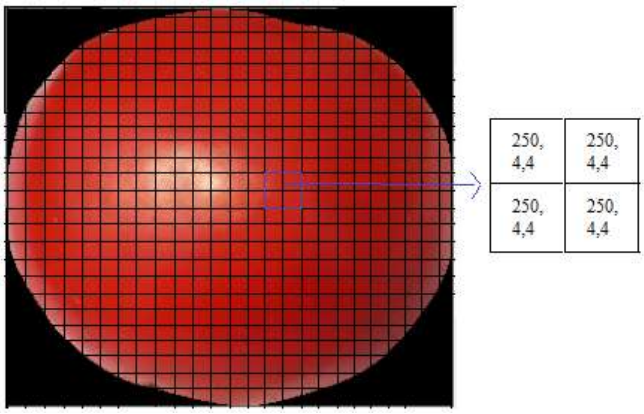
Citra yang digunakan pada sistem ini memiliki format jpg dengan ukuran 300x300 pixel yang melalui proses *cropping* seperti terlihat pada Gambar 3.4 berikut ini :



Gambar 3.4 Hasil Pemotongan Citra Tomat Ukuran 300x300 Pixel

Proses ekstraksi ciri yang dilakukan pada citra tomat, dapat dilakukan dengan langkah-langkah sebagai berikut [4] :

1. Mengambil tiap nilai R, G dan B pada tiap pixel.



Gambar 3.5 Nilai RGB pada Pixel

2. Hitung total R, G dan B dari semua pixel.

$$\text{Total R} = \sum \text{R}$$

$$\text{Total G} = \sum \text{G}$$

$$\text{Total B} = \sum \text{B}$$

3. Bagi tiap total R, G dan B dengan jumlah pixel.

$$R = \frac{Total\ R}{300x300}$$

$$G = \frac{Total\ G}{300x300}$$

$$B = \frac{Total\ B}{300x300}$$

3.1.4.2. Metode Learning Vector Quantization

Dalam pembahasan analisis metode *Learning Vector Quantization* akan dibahas mengenai proses pembelajaran dan pengujian menggunakan LVQ.

1. Pembelajaran

Tahap pertama yang akan dilakukan adalah inisialisasi nilai parameter awal yaitu *maxepoch, learning rate* dan *error* minimum yang akan dilakukan untuk metode pembelajaran LVQ. Berdasarkan penelitian sebelumnya yang dilakukan oleh Maharani [17], dalam kasus ini dipilih *learning rate* sebesar 0.05, *maxepoch* = 10 dan *epsilon* = 0.01. Tahap selanjutnya inisialisasi $\alpha = 0.05$, *MaxEpoch* = 10 dan *Eps* = 0.01.

Tahap selanjutnya adalah penentuan nilai bobot untuk tiap kelas yang dituju, pada tahap inisialisasi awal ini akan dilakukan pemisahan data yaitu enam data masukan dari salah satu kelas akan diinisialisasikan sebagai bobot awal tiap kelas yang akan dituju, bobot tersebut terdapat pada Tabel 3.4 sedangkan data sisanya pada Tabel 3.5 akan digunakan untuk pembelajaran.

Tabel 3.4 Vektor Bobot

Data ke -	Vektor Bobot (w)	Vektor Target (T)
1	[99 95 47]	1
4	[140 110 52]	2
7	[113 91 47]	3
10	[184 62 22]	4
13	[157 32 13]	5
16	[152 39 15]	6

Tabel 3.5 Vektor Data Pembelajaran

Data ke-	Vektor Pembelajaran(X)	Vektor Target(T)
1	[123 131 33]	1
2	[126 136 29]	1
3	[145 118 55]	2
4	[106 84 36]	2
5	[83 59 15]	3
6	[112 85 35]	3
7	[209 97 32]	4
8	[99 30 5]	4
9	[171 62 22]	5

10	[169 58 19]	5
11	[146 20 10]	6
12	[165 41 17]	6

Setelah dilakukan inisialisasi parameter awal dan bobot awal, tahap selanjutnya adalah pencarian jarak terkecil antar data pembelajaran terhadap tiap bobot dalam setiap *epoch*.

Epoch ke-1:

Data ke – 1 : [123 131 33]

Jarak pada

Bobot ke-1 [99 95 47]

$C_1= \sqrt{(123-99)^2+ (131-95)^2+ (33-47)^2}=45.475$

Bobot ke-2 [140 110 52]

$C_2= \sqrt{(123-140)^2+ (131-110)^2+ (33-52)^2}=33.03$

Bobot ke-3 [113 91 47]

$C_3= \sqrt{(123-113)^2+ (131-91)^2+ (33-47)^2}=43.543$

Bobot ke-4 [184 62 22]

$C_4= \sqrt{(123-184)^2+ (131-62)^2+ (33-22)^2}=92.752$

Bobot ke-5 [157 32 13]

$C_5= \sqrt{(123-157)^2+ (131-32)^2+ (33-13)^2}=106.57$

Bobot ke-6 [152 39 15]

$C_6= \sqrt{(123-152)^2+ (131-39)^2+ (33-15)^2}=98.172$

Jarak terkecil adalah pada bobot ke 2, dan target data ke-1 adalah *stage* 1, karena tidak sesuai maka perbaharui bobot ke-2 dengan persamaan 2.5.

$W_{2,1} = 140-0.05*(123-140)= 140.85$

$W_{2,2} = 110-0.05*(131-110)= 108.95$

$W_{2,3} = 52-0.05*(33-52)= 52.95$

Proses perhitungan jarak tersebut dilakukan pada setiap nilai *x* terhadap setiap bobot hasil dari perhitungan jarak tersebut dapat dilihat pada tabel Jarak Terhadap Bobot LVQ (tabel dilihat di Lampiran D), setelah berada pada data terakhir akan dihitung pengurangan nilai *learning rate* dan penambahan nilai *epoch* sebagai berikut :

$\alpha=\alpha-(0.1*\alpha)=0.05-(0.1*0.05)= 0.045$

Bobot yang didapat dari *epoch* ke-1 adalah

$W_1 = [99.8 \;\; 96.8 \;\; 48.6]$

$W_2 = [141.7629 \;\; 108.1176 \;\; 54.1901]$

$W_3 = [113.2825 \;\; 91.0325 \;\; 46.9225]$

$W_4 = [186.8106 \;\; -64.1294 \;\; 22.7012]$

$W_5 = [157.1775 \;\; -32.28 \;\; 13.0575]$

$W_6 = [154.65 \;\; -41.55 \;\; 17.6]$

Pembelajaran dilakukan pada *epoch* berikutnya hingga mencapai kondisi *maxepoch* = 10 atau nilai *learning rate* < *eps* dengan cara yang sama. Setelah pembelajaran selesai didapatkan bobot akhir yang akan digunakan untuk menentukan kelas dari data masukan yang akan diuji berikutnya.

$W_1 = [106.971 \;\; 107.878 \;\; 44.988]$
$W_2 = [146.252 \;\; 105.237 \;\; 59.309]$
$W_3 = [108.708 \;\; 85.7652 \;\; 40.755]$
$W_4 = [195.278 \;\; 73.1054 \;\; 25.257]$
$W_5 = [165.094 \;\; 35.1579 \;\; 14.469]$
$W_6 = [162.630 \;\; 36.2525 \;\; 19.653]$

2. Pengujian

Pada tahap ini akan dilakukan pengujian LVQ terhadap data uji dengan menggunakan bobot hasil pembelajaran yang telah dilakukan.

Data Uji 1 [139 118 28]

Jarak pada

Bobot ke-1 [106.971 107.878 44.988]

$C_1= \sqrt{(139\text{-}106.971)^2+ (118\text{-}107.878)^2+ (28\text{-}44.988)^2}=37.6418$

Bobot ke-2 [146.252 105.237 59.309]

$C_2= \sqrt{(139\text{-}146.252)^2+ (118\text{-}105.237)^2+ (28\text{-}59.309)^2}=34.5795$

Bobot ke-3 [108.708 85.7652 40.755]

$C_3= \sqrt{(139\text{-}108.708)^2+ (118\text{-}85.7652)^2+ (28\text{-}40.755)^2}=46.0367$

Bobot ke-4 [195.278 73.1054 25.257]

$C_4= \sqrt{(139\text{-}195.278)^2+ (118\text{-}73.1054)^2+ (28\text{-}25.257)^2}=72.0435$

Bobot ke-5 [165.094 35.1579 14.469]

$C_5= \sqrt{(139\text{-}165.094)^2+ (118\text{-}35.1579)^2+ (28\text{-}14.469)^2}=87.9022$

Bobot ke-6 [162.630 36.2525 19.653]

$C_6= \sqrt{(139\text{-}162.630)^2+ (118\text{-}36.2525)^2+ (28\text{-}19.653)^2}=85.5026$

Berdasarkan pengujian LVQ menunjukan bahwa data uji ke-1 tersebut masuk ke dalam *stage* 2. Pengujian pada data uji lainnya menunjukan hasil seperti pada Tabel 3.6 berikut ini.

Tabel 3.6 Pengujian *Learning Vector Quantization*

Data	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**Min**	***Stage***	***Stage LVQ***
1	37.64	34.58	46.04	72.04	87.9	85.5	34.58	2	2
2	101.13	68.74	96.14	15	64.66	64.48	15	4	4
3	17.96	42.78	39.44	99.72	108.19	104.81	17.96	1	1
4	31.88	30	31.15	66.35	73.6	70.57	30	3	2
5	29.87	49.59	41.69	90.17	97.65	95.33	29.87	1	1
6	99.68	93.74	80	65.55	16.73	18.29	16.73	6	5
7	88.65	81.98	70.1	56.58	14.66	14.84	14.66	5	5
8	16.47	53.58	17.64	99.04	93.04	89.65	16.47	1	1
9	125.42	100.43	112.05	38.15	42.03	44.55	38.15	6	4
10	84.93	62.12	77.04	18.54	46.64	46.76	18.54	4	4
11	42.41	16.49	52.58	72.42	94.52	91.13	16.49	3	2
12	44.18	72.76	27.37	103.29	82.41	79.28	27.37	2	3
13	7.72	36.8	23.86	90.42	93.92	90.31	7.72	1	1

14	35.19	13.53	45.48	75.64	93.42	89.73	13.53	2	2
15	47.9	85.42	41.5	122.18	104.38	101.65	41.5	1	3
16	36.17	35.61	44.33	72.98	87.43	85.04	35.61	1	2
17	73.48	52.43	77.02	51.26	84.35	83.57	51.26	3	4
18	85.17	68.24	69.93	35.44	18.55	16.51	16.51	6	6

Setelah didapatkan hasil klasifikasi dari LVQ, tahap selanjutnya adalah perhitungan nilai RMSE untuk mengetahui besar *error* yang ada pada pengujian LVQ tersebut.

$$\text{RMSE}=\sqrt{\frac{\sum_{i=1}^{n}\left(Y_i-(\sum \overline{w_i} f_i)\right)^2}{n}}$$

$$RMSE=\sqrt{\frac{(2-2)^2+(4-4)^2+(1-1)^2+(3-2)^2\ldots+(6-6)^2}{18}}=\sqrt{\frac{14}{18}}=0.881917$$

Berdasarkan hasil proses pengujian LVQ terhadap 18 data pengujian yang telah dilakukan dengan hasil bobot yang dari proses pembelajaran dengan *maxepoch* = 10, *learning rate* = 0.05 dan *eps* = 0.01 didapatkan persentase akurasi yang dapat dilakukan oleh LVQ adalah 55.55556 % dengan RMSE = 0.881917.

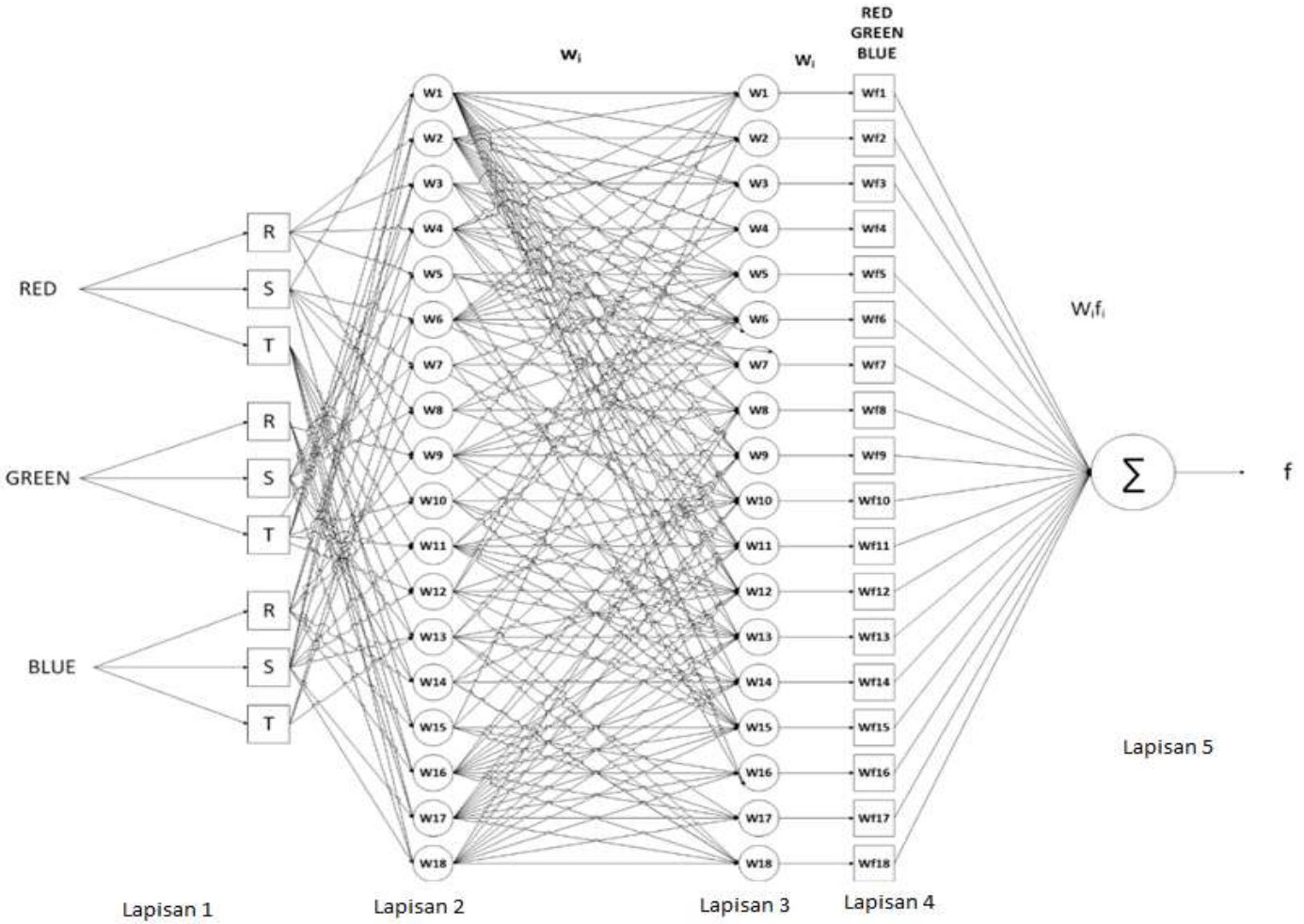
3.1.4.3. Metode Adaptive Neurro Fuzzy Inference System

Dalam pembahasan analisis metode *Adaptive Neuro Fuzzy Inference System* ini akan dibahas mengenai proses pembelajaran dan pengujian menggunakan ANFIS.

1. Pembelajaran

Tahap awal yang dilakukan adalah inisialisasi nilai parameter awal yaitu *maxepoch* dan *error* minimal. Selanjutnya setiap vektor masukan pada Tabel 3.4 yaitu setiap nilai R, G dan B akan dijadikan sebagai satu variabel masukan yang akan direpresentasikan dalam fungsi keanggotaan *gbell*. Setiap variabel masukan tersebut dibagi kedalam 3 *membership function(mf)* yaitu, rendah, sedang dan tinggi dan terdapat 18 *rules*, 18 *membership function* yang bertipe sama serta mempunyai satu keluaran tunggal hasil proses *deffuzyfication*. Variabel keluaran adalah tingkat kematangan yaitu *stage* 1, *stage* 2, *stage* 3, *stage* 4, *stage* 5 dan *stage* 6.

Data masukan tersebut akan melalui pembelajaran ANFIS yang dijabarkan dalam proses seperti pada Gambar 3.6 sebagai berikut :



Gambar 3.6 Struktur ANFIS Alur Maju (*FeedForward*)

1. Lapisan 1 (*Fuzzification*)

Terdapat ketentuan untuk tiap *membership function* dari *red*, *green* dan *blue* dengan persamaan 2.7, fungsi keanggotaannya adalah sebagai berikut :

a. Membership function untuk *red*

Rendah (R) : gbell(x; 50; 1; 60)

Sedang (S) : gbell(x; 40; 1; 130)

Tinggi (T) : gbell(x; 70; 1; 210)

Data ke -1

R = 99; G = 95; B = 47

$\mu_R\,(233) = gbell(99,50,1,60) = \frac{1}{1+\left|\frac{99-60}{50}\right|^2}$ = 0.62174

$\mu_S\,(233) = gbell(99,40,1,130) = \frac{1}{1+\left|\frac{99-130}{40}\right|^2}$ = 0.62476

$\mu_T\,(233) = gbell(99,70,1,210) = \frac{1}{1+\left|\frac{99-210}{70}\right|^2}$= 0.28454

b. Membership function untuk *green*

Rendah (R) : gbell(x; 130; 1; 10)

Sedang (S) : gbell(x; 65; 1; 75)

Tinggi (T) : gbell(x;130; 1; 140)

Data ke -1

R = 99; G = 95; B = 47

$\mu_R\,(233) = gbell(95,40,1,30) = \frac{1}{1+\left|\frac{95-10}{130}\right|^2}$ = 0.70052

$\mu_S\,(233) = gbell(95,40,1,90) = \frac{1}{1+\left|\frac{95-75}{65}\right|^2}$ = 0.91351

$\mu_T\,(233) = gbell(95,40,1,150) = \frac{1}{1+\left|\frac{95-140}{130}\right|^2}$= 0.893

c. Membership function untuk *blue*

Rendah (R) : gbell(x; 40; 1; 0)

Sedang (S) : gbell(x; 20; 1; 35)

Tinggi (T) : gbell(x; 40; 1; 70)

Data ke -1

R = 99; G = 95; B = 47

$$\mu_R\,(233) = gbell(47,40,1,0) = \frac{1}{1+\left|\frac{49-0}{40}\right|^2} = 0.42006$$

$$\mu_S\,(233) = gbell(47,20,1,35) = \frac{1}{1+\left|\frac{49-35}{20}\right|^2} = 0.73529$$

$$\mu_T\,(233) = gbell(47,40,1,70) = \frac{1}{1+\left|\frac{49-70}{40}\right|^2} = 0.73153$$

Fuzzyfication dilakukan pada setiap data yang dimasukan. Berdasarkan perhitungan *membership function* yang telah dilakukan pada 18 data yang telah dilakukan didapatkan hasil pada tabel *Membership Function* (tabel dilihat di Lampiran D).

2. Lapisan 2 (*System Inference Fuzzy*)

Terdapat 18 aturan (*rule*) yang akan digunakan untuk menentukan nilai keluaran pada lapisan ini, adapun 18 aturan tersebut pada Tabel 3.7 sebagai berikut ini :

Tabel 3.7 *Rule* ANFIS

No	Aturan (*Rule*)
1	*IF Red* = Sedang and *Green* = Tinggi and *Blue* = Sedang, *THEN* f = $P_1Red+Q_1Green+R_1Blue+S_1$
2	*IF Red* = Rendah and *Green* = Tinggi and *Blue* = Sedang, *THEN* f = $P_2Red+Q_2Green+R_2Blue+S_2$
3	*IF Red* = Rendah and *Green* = Tinggi and *Blue* = Rendah, *THEN* f = $P_3Red+Q_3Green+R_3Blue+S_3$
4	*IF Red* = Rendah and *Green* = Sedang and *Blue* = Rendah, *THEN* f = $P_4Red+Q_4Green+R_4Blue+S_4$
5	*IF Red* = Rendah and *Green* = Sedang and *Blue* = Sedang, *THEN* f = $P_5Red+Q_5Green+R_5Blue+S_5$
6	*IF Red* = Sedang and *Green* = Tinggi and *Blue* = Tinggi, *THEN* f = $P_6Red+Q_6Green+R_6Blue+S_6$
7	*IF Red* = Sedang and *Green* = Sedang and *Blue* = Tinggi, *THEN* f = $P_7Red+Q_7Green+R_7Blue+S_7$
8	*IF Red* = Sedang and *Green* = Sedang and *Blue* = Sedang, *THEN* f = $P_8Red+Q_8Green+R8Blue+S_8$
9	*IF Red* = Rendah and *Green* = Rendah and *Blue* = Rendah, *THEN* f = $P_9Red+Q_9Green+R_9Blue+S_9$
10	*IF Red* = Sedang and *Green* = Sedang and *Blue* = Rendah, *THEN* f = $P_{10}Red+Q_{10}Green+R_{10}Blue+S_{10}$
11	*IF Red* = Tinggi and *Green* = Tinggi and *Blue* = Sedang, *THEN* f = $P_{11}Red+Q_{11}Green+R_{11}Blue+S_{11}$
12	*IF Red* = Tinggi and *Green* = Sedang and *Blue* = Sedang, *THEN* f = $P_{12}Red+Q_{12}Green+R_{12}Blue+S_{12}$
13	*IF Red* = Tinggi and *Green* = Tinggi and *Blue* = Tinggi, *THEN* f = $P_{13}Red+Q_{13}Green+R_{13}Blue+S_{13}$
14	*IF Red* = Tinggi and *Green* = Sedang and *Blue* = Rendah, *THEN* f = $P_{14}Red+Q_{14}Green+R_{14}Blue+S_{14}$
15	*IF Red* = Sedang and *Green* = Rendah and *Blue* = Rendah, *THEN* f = $P_{15}Red+Q_{15}Green+R_{15}Blue+S_{14}$
16	*IF Red* = Tinggi and *Green* = Rendah and *Blue* = Rendah, *THEN* f = $P_{16}Red+Q_{16}Green+R_{16}Blue+S_{16}$
17	*IF Red* = Tinggi and *Green* = Rendah and *Blue* = Sedang, *THEN* f = $P_{17}Red+Q_{17}Green+R_{17}Blue+S_{17}$
18	*IF Red* = Tinggi and *Green* = Sedang and *Blue* = Tinggi, *THEN* f = $P_{18}Red+Q_{18}Green+R_{18}Blue+S_{18}$

Pada langkah ini akan dihitung nilai W_i dari tiap *membership function* nilai R, G dan B dengan menggunakan persamaan 2.8 sebagai berikut :

Data ke – 1 :

$W_1 = \mu_S(Red).\mu_T(Green).\mu_S(Blue)$ $= \mu_S(99).\mu_T(95).\mu_S(47)$
$=\min(0.624756, 0.892999, 0.735294)$
$= 0.624756$

$W_2 = \mu_R(Red).\mu_T(Green).\mu_S(Blue)$ $= \mu_R(99).\mu_T(95).\mu_S(47)$
$= \min(0.621736,0.892999,0.735294)$
$= 0.621736$

$W_3 = \mu_R(Red).\mu_T(Green).\mu_R(Blue)$ $= \mu_R(99).\mu_T(95).\mu_R(47)$
$= \min(0.621736,0.892999, 0.420058)$
$= 0.420058$

$W_4 = \mu_R(Red).\mu_S(Green).\mu_R(Blue)$ $= \mu_R(99).\mu_S(95).\mu_R(47)$
$= \min(0.621736,0.913514,0.420058)$
$= 0.420058$

$W_5 = \mu_R(Red).\mu_S(Green).\mu_S(Blue)$ $= \mu_R(99).\mu_S(95).\mu_S(47)$
$= \min(0.621736,0.913514, 0.735294)$
$= 0.621736$

$W_6 = \mu_S(Red).\mu_T(Green).\mu_T(Blue)$ $= \mu_S(99).\mu_T(95).\mu_T(47)$
$= \min(0.624756,0.892999, 0.751527)$
$= 0.624756$

$W_7 = \mu_S(Red).\mu_S(Green).\mu_T(Blue)$ $= \mu_S(99).\mu_S(95).\mu_T(47)$
$= \min(0.624755,0.913514, 0.751527)$
$= 0.624756$

$W_8 = \mu_S(Red).\mu_S(Green).\mu_S(Blue)$ $= \mu_S(99).\mu_S(95).\mu_S(47)$
$= \min(0.624756,0.913514,0.735294)$
$= 0.624756$

$W_9 = \mu_R(Red).\mu_R(Green).\mu_R(Blue)$ $= \mu_R(99).\mu_R(95).\mu_R(47)$
$= \min(0.621736,0.700518, 0.420058)$
$= 0.420058$

$W_{10} = \mu_S(Red).\mu_S(Green).\mu_R(Blue)$ $= \mu_S(99).\mu_S(95).\mu_R(47)$
$= \min(0.624756,0.913514, 0.420058)$
$= 0.420058$

$W_{11} = \mu_T(Red).\mu_T(Green).\mu_S(Blue)$ $= \mu_T(99).\mu_T(95).\mu_S(47)$
$= \min(0.284536,0.892999,0.735294)$
$= 0.284536$

$W_{12} = \mu_T(Red).\mu_S(Green).\mu_S(Blue)$ $= \mu_T(99).\mu_S(95).\mu_S(47)$
$= \min(0.284536,0.913514,0.735294)$
$= 0.284536$

$W_{13} = \mu_T(Red).\mu_T(Green).\mu_T(Blue)$ $= \mu_T(99).\mu_T(95).\mu_T(47)$
$= \min(0.284536,0.892999,0.751527)$
$= 0.284536$

$W_{14} = \mu_T(Red).\mu_R(Green).\mu_R(Blue)$ $= \mu_T(99).\mu_R(95).\mu_R(47)$
$= \min(0.284536, 0.700518,0.420058)$
$= 0.284536$

$W_{15} = \mu_S(Red).\mu_R(Green).\mu_R(Blue)$ $= \mu_S(99).\mu_R(95).\mu_R(47)$
$= \min(0.624756,0.700518,0.420058)$
$= 0.420058$

$W_{16} = \mu_T(Red).\mu_R(Green).\mu_R(Blue)$ $= \mu_T(99).\mu_R(95).\mu_R(47)$
$= \min(0.284536,0.700518,0.420058)$
$= 0.284536$

$W_{17} = \mu_T(Red).\mu_R(Green).\mu_S(Blue)$ $= \mu_T(99).\mu_R(95).\mu_S(47)$

W_{18}= μ_T (*Red*). μ_S(*Green*).μ_T(*Blue*)
= min (0.284536,0.700518,0.735294)
= 0.284536
= μ_T(99). μ_S(95).μ_T(47)
= min (0.284536,0.913514, 0.751527)
= 0.284536

Sistem inference fuzzy tersebut dilakukan pada tiap data masukan lain pada pembelajaran ANFIS, hasilnya dapat dilihat pada tabel *Output* Lapisan 2 (tabel dilihat di Lampiran D).

3. Lapisan 3 (*Normalisasi*)

Pada langkah ini akan dihitung derajat pengaktifan *firing strength* (W_i) dengan menggunakan persamaan 2.9 sebagai berikut :

$$\overline{W_1} = \frac{W_1}{W_1+\cdots+W_{18}} = \frac{0.624756}{7.83453}=0.07974$$
$$\overline{W_2} = \frac{W_2}{W_1+\cdots+W_{18}} = \frac{0.621736}{7.83453}=0.07936$$
$$\overline{W_3} = \frac{W_3}{W_1+\cdots+W_{18}} = \frac{0.621736}{7.83453}=0.07936$$
$$\overline{W_4} = \frac{W_4}{W_1+\cdots+W_{18}} = \frac{0.420058}{7.83453} =0.05362$$
$$\overline{W_5} = \frac{W_5}{W_1+\cdots+W_{18}} = \frac{0.621736}{7.83453} =0.07936$$
$$\overline{W_6} = \frac{W_6}{W_1+\cdots+W_{18}} = \frac{0.624756}{7.83453}=0.07974$$
$$\overline{W_7} = \frac{W_7}{W_1+\cdots W_{18}} = \frac{0.624756}{7.83453}=0.07974$$
$$\overline{W_8} = \frac{W_8}{W_1+\cdots+W_{18}} = \frac{0.624756}{7.83453}=0.07974$$
$$\overline{W_9} = \frac{W_9}{W_1+\cdots+W_{18}} = \frac{0.420058}{7.83453} =0.05362$$
$$\overline{W_{10}} = \frac{W_{10}}{W_1+\cdots+W_{18}} = \frac{0.420058}{7.83453} =0.05362$$
$$\overline{W_{11}} = \frac{W_{11}}{W_1+\cdots+W_{18}} = \frac{0.624756}{7.83453}=0.07974$$
$$\overline{W_{12}} = \frac{W_{12}}{W_1+\cdots+W_{18}} = \frac{0.284536}{7.83453} =0.03632$$
$$\overline{W_{13}} = \frac{W_{13}}{W_1+\cdots+W_{18}} = \frac{0.284536}{7.83453}=0.03632$$
$$\overline{W_{14}} = \frac{W_{14}}{W_1+\cdots W_{18}} = \frac{0.284536}{7.83453}=0.03632$$
$$\overline{W_{15}} = \frac{W_{15}}{W_1+\cdots+W_{18}} = \frac{0.420058}{7.83453}=0.05362$$
$$\overline{W_{16}} = \frac{W_{16}}{W_1+\cdots+W_{18}} = \frac{0.284536}{W_1+\cdots W_{18}} =0.03632$$
$$\overline{W_{17}} = \frac{W_{17}}{W_1+\cdots+W_{18}} = \frac{0.284536}{7.83453}=0.03632$$
$$\overline{W_{18}} = \frac{W_{18}}{W_1+\cdots+W_{18}} = \frac{0.284536}{7.83453} = 0.03632$$

Normalisasi dilakukan pada semua data masukan yaitu 18 data, hasilnya dapat dilihat pada tabel *Output* Lapisan 3 (tabel dilihat di Lampiran D).

4. Lapisan 4 (*Least Square Estimator*)

Nilai derajat pengaktifan *firing strength* dari lapisan 3 akan digunakan sebagai nilai masukan untuk menentukan parameter konsekuen Θ (*theta*) dengan menggunakan *Least Square Estimate* (LSE), berikut langkah penentuan parameter konsekuen:

Hasil normalisasi dari lapisan 3 akan dibentuk dalam sebuah matriks A. Jumlah baris pada matriks A sebanyak jumlah masukan *x* yaitu 18.

$$A = \begin{bmatrix} R_i(\overline{W_1}) & G_i(\overline{W_1}) & B_i(\overline{W_1}) & \overline{W_1} & R_i(\overline{W_{20}}) & G_i(\overline{W_{20}}) & B_i(\overline{W_{20}}) & \overline{W_{20}} \\ & & & \dots & & & & \\ R_{18}(\overline{W_1}) & G_{18}(\overline{W_1}) & B_{18}(W_1) & W_1 & R_{18}(\overline{W_{20}}) & G_{18}(\overline{W_{20}}) & B_{18}(W_{20}) & W_{20} \end{bmatrix}$$

Data pada matriks A ada pada tabel RLSE (tabel dilihat di Lampiran D). Dari matriks A akan dicari nilai parameter konsekuen P_i, R_i, Q_i dan S_i menggunakan LSE dengan persamaan 2.13, proses untuk mendapatkan parameter premis dengan menjabarkan persamaan 2.13 dengan hasil proses yang dapat dilihat pada tabel Matriks A, Matriks A *Transpose*, Matriks A*A *Transpose*, Matriks Invers (A*A *Transpose*)* A *Transpose*, Matriks *Invers* (A*A *Transpose*)* A *Transpose** Y (tabel dilihat di Lampiran D). Hasil dari perhitungan RLSE akan dijadikan nilai konsekuen pada dengan hasil pada Tabel 3.8 sebagai berikut :

Tabel 3.8 Parameter Konsekuen

No.	**P_i**	**Q_i**	**R_i**	**S_i**
1	0.41498	-0.07985	0.12298	0.00584
2	-0.02262	-0.09889	0.00877	0.00401
3	-0.44813	-0.19357	-0.26497	0.00439
4	0.24386	0.34771	-0.12415	0.01293
5	-0.02262	-0.09889	0.00877	0.00401
6	-0.02162	0.40251	0.58122	0.00692
7	-0.5594	-0.06178	0.36979	0.00343
8	0.38664	-0.13739	0.18812	0.00596
9	0.24386	0.34771	-0.12415	0.01293
10	0.11665	-0.58776	-0.42396	-0.00107
11	0.23344	-0.27329	-0.24291	0.00421
12	0.2386	-0.2709	-0.24212	0.00423
13	-0.28437	-0.15787	-0.13322	0.00044
14	-0.46858	0.60243	0.06944	-0.00009

15	0.3641	-0.44685	-0.38025	0.00071
16	-0.46858	0.60243	0.06944	-0.00009
17	0.49493	-0.15194	-0.20287	0.00546
18	0.2386	-0.2709	-0.24212	0.00423

Setelah didapatkan nilai parameter konsekuen pada lapisan ini selanjutnya menghitung nilai keluaran lapisan 4 terhadap semua data masukan menggunakan persamaan 2.10, hasil dari perhitungan tersebut pada 18 data yang dimasukan dapat dilihat pada tabel *Output* lapisan 4 (tabel dilihat di Lampiran D). Adapun contoh perhitungan untuk data ke-1 sebagai berikut :

Data ke – 1

$\bar{w}_1 f_1$ = 0.0797 * 39.28357 = 3.13263
$\bar{w}_2 f_2$ = 0.0794 * -11.21754 = -0.8902
$\bar{w}_3 f_3$ = 0.0536 * -75.20254 = -4.03206
$\bar{w}_4 f_4$ = 0.0536 * 51.35151 = 2.75326
$\bar{w}_5 f_5$ = 0.0794 * -11.21754 = -0.8902
$\bar{w}_6 f_6$ = 0.0797 * 63.42156 = 5.05749
$\bar{w}_7 f_7$ = 0.0797 * -43.86652 = -3.49809
$\bar{w}_8 f_8$ = 0.0797 * 34.07301 = 2.71712
$\bar{w}_9 f_9$ = 0.0536 * 51.35151 = 2.75326
$\bar{w}_{10} f_{10}$ = 0.0536 * -64.21546 = -3.44298
$\bar{w}_{11} f_{11}$ = 0.0363 * -14.26495 = -0.51807
$\bar{w}_{12} f_{12}$ = 0.0363 * -13.48865 = -0.48988
$\bar{w}_{13} f_{13}$ = 0.0363 * -49.41109 = -1.79451
$\bar{w}_{14} f_{14}$ = 0.0363 * 14.10477 = 0.51226
$\bar{w}_{15} f_{15}$ = 0.0536 * -24.27536 = -1.30155
$\bar{w}_{16} f_{16}$ = 0.0363 * 14.10477 = 0.51226
$\bar{w}_{17} f_{17}$ = 0.0363 * 25.03427 = 0.90919
$\bar{w}_{18} f_{18}$ = 0.0363 * -13.48865 = -0.48988

5. Lapisan 5 (*Defuzzification*)

Setelah didapatkan hasil dari lapisan 4, langkah selanjutnya adalah defuzzifikasi dengan melakukan penjumlahan seperti pada persamaan 2.11, berikut hasil dari hasil defuzzifikasi dari 18 data yang dimasukan :

Tabel 3.9 *Output* Lapisan 5

No	$\sum \overline{w}_i f_i$
1	1.0000436
2	0.87657047
3	0.94695883
4	1.99996857
5	1.99988154
6	1.99985494
7	2.99982469
8	2.99996289
9	2.99996688
10	3.99976602
11	3.99993018
12	3.99994868
13	5.00002408
14	5.0000952
15	4.99997838
16	5.99196722
17	4.58240099
18	6.00024883

6. RMSE (*Root Mean Squared Error*)

RMSE merupakan metode alternatif untuk mengevaluasi teknik peramalan. RMSE adalah rata-rata kuadrat dari perbedaan nilai estimasi dengan nilai observasi suatu variabel. Jika nilai RMSE semakin kecil maka estimasi model atau variabel tersebut semakin valid. Berikut perhitungan RMSE pada tahap pertama dari 18 data yang dimasukan.

$$\text{RMSE}=\sqrt{\frac{\sum_{i=1}^{n}\left(Y_i-(\sum \overline{w}_i f_i)\right)^2}{n}}$$

$$RMSE=\sqrt{\frac{(1-(-1.0000436))^2+\ldots+(6-(-6.00024883))^2}{18}}=0.298461929$$

7. EBP (*Backprogpagation-Error*)

Prosedur pragagasi balik menghitung sinyal kesalahan secara mundur dari lapisan keluaran hingga lapisan masukan yang terdapat pada tahap maju, yaitu dari

lapisan 5 hingga lapisan 1, berikut penjelasan perhitungan sinyal kesalahan dengan EBP pada tiap lapisan :

1. Lapisan 5

Pada lapisan 5 ini akan dihitung nilai *error* pada simpul yang ada dengan persamaan 2.16.

$$\varepsilon_{5,1} = -2(Y - O_{5,1})$$

Data ke -1 :

$\varepsilon_{5,1} = -2(Y - O_{5,1})$ = -2(1-(-147.549) = -2(148.5) = 0.000087

Tabel 3.10 *Error* Lapisan 5

Data Ke -	$\boldsymbol{\varepsilon_{5,1}}$
1	0.000087
2	-0.246859
3	-0.106082
4	-0.000063
5	-0.000237
6	-0.00029
7	-0.000351
8	-0.000074
9	-0.000066
10	-0.000468
11	-0.00014
12	-0.000103
13	0.000048
14	0.00019
15	-0.000043
16	-0.016066
17	-2.835198
18	0.000498

2. Lapisan 4

Pada lapisan 4 perhitungan nilai *error* pada tiap simpul adalah dengan menggunakan persaman 2.17.

$$\varepsilon_{4,i} = \varepsilon_5 \left(\frac{\partial f_5}{\partial O_{4,i}}\right)$$

Data ke -1

$\varepsilon_{4,1} = \varepsilon_5 \left(\frac{\partial f_{5,1}}{\partial O_{4,i}}\right)$ = 0.000087 (1) = 0.000087

Nilai *error* pada simpul lainnya akan bernilai sama seperti perhitungan nilai *error* pada $\varepsilon_{4,1}$, hasilnya ada pada tabel *Error* Lapisan 4 (tabel dilihat di Lampiran D).

3. Lapisan 3

Pada lapisan 3 perhitungan nilai *error* pada tiap simpul adalah dengan menggunakan persaman 2.18 dan hasil nilai *error* pada tiap simpul pada lapisan 3 ada pada *Error* Lapisan 3.

$$\varepsilon_{3,i} = O_{3,i} * \varepsilon_{4,i}$$

Data ke – 1

$\varepsilon_{3,1} = O_{3,1} * \varepsilon_{4,1}$ = 0.0797 * 0.000087 = 7.00E-06

$\varepsilon_{3,2} = O_{3,2} * \varepsilon_{4,2}$ = 0.0794 * 0.000087 = 7.00E-06

$\varepsilon_{3,3} = O_{3,3} * \varepsilon_{4,3}$ = 0.0536 * 0.000087 = 5.00E-06

$\varepsilon_{3,4} = O_{3,4} * \varepsilon_{4,4}$ = 0.0536 * 0.000087 = 5.00E-06

$\varepsilon_{3,5} = O_{3,5} * \varepsilon_{4,5}$ = 0.0794 * 0.000087 = 7.00E-06

$\varepsilon_{3,6} = O_{3,6} * \varepsilon_{4,6}$ = 0.0797 * 0.000087 = 7.00E-06

$\varepsilon_{3,7} = O_{3,7} * \varepsilon_{4,7}$ = 0.0797 * 0.000087 = 7.00E-06

$\varepsilon_{3,8} = O_{3,8} * \varepsilon_{4,8}$ = 0.0797 * 0.000087 = 7.00E-06

$\varepsilon_{3,9} = O_{3,9} * \varepsilon_{4,9}$ = 0.0536 * 0.000087 = 5.00E-06

$\varepsilon_{3,10} = O_{3,10} * \varepsilon_{4,10}$ = 0.0536 * 0.000087 = 5.00E-06

$\varepsilon_{3,11} = O_{3,11} * \varepsilon_{4,11}$ = 0.0363 * 0.000087 = 3.00E-06

$\varepsilon_{3,12} = O_{3,12} * \varepsilon_{4,12}$ = 0.0363 * 0.000087 = 3.00E-06

$\varepsilon_{3,13} = O_{3,13} * \varepsilon_{4,13}$ = 0.0363 * 0.000087 = 3.00E-06

$\varepsilon_{3,14} = O_{3,14} * \varepsilon_{4,14}$ = 0.0363 * 0.000087 = 3.00E-06

$\varepsilon_{3,15} = O_{3,15} * \varepsilon_{4,15}$ = 0.0536 * 0.000087 = 5.00E-06

$\varepsilon_{3,16} = O_{3,16} * \varepsilon_{4,16}$ = 0.0363 * 0.000087 = 3.00E-06

$\varepsilon_{3,17} = O_{3,17} * \varepsilon_{4,17}$ = 0.0363 * 0.000087 = 3.00E-06

$\varepsilon_{3,18} = O_{3,18} * \varepsilon_{4,18}$ = 0.0363 * 0.000087 = 3.00E-06

4. Lapisan 2

Pada lapisan 2 perhitungan nilai *error* pada tiap simpul adalah dengan menggunakan persaman 2.19 dan hasil nilai *error* pada lapisan 2 ada pada tabel *Error* Lapisan 2 (tabel dilihat di Lampiran D).

$$\varepsilon_{2,i} = \left(\frac{W_i}{(w_1 + \cdots + w_{18})^2}\right)\varepsilon_{3,i}$$

Data ke – 1

$\varepsilon_{2,1}$ = 0.0101785 * 7.00E-06 = 7.00E-08

$\varepsilon_{2,2}$ = 0.0101293 * 7.00E-06 = 7.00E-08

$\varepsilon_{2,3}$ = 0.00684357 * 5.00E-06 = 3.00E-08

$\varepsilon_{2,4}$ = 0.00684357 * 5.00E-06 = 3.00E-08

$\varepsilon_{2,5}$ = 0.0101293 * 7.00E-06 = 7.00E-08

$\varepsilon_{2,6}$ = 0.0101785 * 7.00E-06 = 7.00E-08

$\varepsilon_{2,7}$ = 0.0101785 * 7.00E-06 = 7.00E-08

$\varepsilon_{2,8}$ = 0.0101785 * 7.00E-06 = 7.00E-08

$\varepsilon_{2,9}$ = 0.00684357 * 5.00E-06 = 3.00E-08

$\varepsilon_{2,10}$ = 0.00684357 * 5.00E-06 = 3.00E-08

$\varepsilon_{2,11}$ = 0.00463565 * 3.00E-06 = 1.00E-08

$\varepsilon_{2,12}$ = 0.00463565 * 3.00E-06 = 1.00E-08

$\varepsilon_{2,13}$ = 0.00463565 * 3.00E-06 = 1.00E-08

$\varepsilon_{2,14}$ = 0.00463565 * 3.00E-06 = 1.00E-08

$\varepsilon_{2,15}$ = 0.00684357 * 5.00E-06 = 3.00E-08

$\varepsilon_{2,16}$ = 0.00463565 * 3.00E-06 = 1.00E-08

$\varepsilon_{2,17}$ = 0.00463565 * 3.00E-06 = 1.00E-08

$\varepsilon_{2,18}$ = 0.00463565 * 3.00E-06 = 1.00E-08

5. Lapisan 1

Pada lapisan 1 perhitungan nilai *error* pada tiap simpul adalah dengan menggunakan persaman 2.20 dan hasil nilai *error* pada lapisan 1 ada pada tabel *Error* Lapisan 1 (tabel dilihat di Lampiran D).

$$\varepsilon_{1,i} = \frac{\sum \varepsilon_{2,i} * \mu(i)}{n}$$

Data ke - 1

$\varepsilon_{1,1}$ = 1.50E-07 / 5 = 3.00E-08
$\varepsilon_{1,2}$ = 2.20E-07 / 6 = 4.00E-08
$\varepsilon_{1,3}$ = 3.00E-08 / 7 = 0.00E+00
$\varepsilon_{1,4}$ = 7.00E-08 / 4 = 2.00E-08
$\varepsilon_{1,5}$ = 2.90E-07 / 8 = 4.00E-08
$\varepsilon_{1,6}$ = 2.40E-07 / 6 = 4.00E-08
$\varepsilon_{1,7}$ = 8.00E-08 / 7 = 1.00E-08
$\varepsilon_{1,8}$ = 2.50E-07 / 8 = 3.00E-08
$\varepsilon_{1,9}$ = 1.20E-07 / 3 = 4.00E-08

Setelah mendapatkan nilai *error* pada tiap lapisan, tahap selanjutnya adalah memperbaharui parameter premis dengan mencari nilai fungsi keanggotaan yang baru, fungsi turunan *g-bell* untuk tiap parameter premis hingga mendapat nilai parameter premis yang baru.

Untuk mendapatkan nilai fungsi keanggotaan yang baru adalah dengan menggunakan persamaan 2.21 dan hasil nilai fungsi keanggotaan yang baru ada pada tabel Fungsi Keanggotaan Baru (tabel dilihat di Lampiran D).

$$\mu_{i\ baru} = \mu_{i\ lama} + \varepsilon_i$$

Data ke -1

μ_{Rbar}	=	0.6217	+	3.00E-08	=	6.22E-01
μ_{Sbaru}	=	0.6248	+	4.00E-08	=	6.25E-01
μ_{Tbaru}	=	0.2845	+	0.00E+00	=	2.85E-01
μ_{Rbaru}	=	0.7005	+	2.00E-08	=	7.01E-01
μ_{Sbaru}	=	0.9135	+	4.00E-08	=	9.14E-01
μ_{Tbaru}	=	0.893	+	4.00E-08	=	8.93E-01
μ_{Rbaru}	=	0.4201	+	1.00E-08	=	4.20E-01
μ_{Sbaru}	=	0.7353	+	3.00E-08	=	7.35E-01
μ_{Bbaru}	=	0.7515	+	4.00E-08	=	7.52E-01

Untuk mendapatkan nilai turunan dari parameter *g-bell* adalah dengan menggunakan persamaan 2.22 hingga 2.14 dan hasil dari nilai turunan fungsi *g-bell* ada pada tabel Turunan Fungsi Kenggotaan *G-Bell* (tabel dilihat di Lampiran D).

a. Parameter Premis a

$$\frac{d\mu_i}{da} = \frac{2b}{a}\mu_i(1-\mu_i)$$

Data ke -1

$\frac{d\mu_1}{da}$	=	$\frac{2(1)}{50}$	6.22E-01	(1 -	6.22E-01	) =	9.40E-03
$\frac{d\mu_2}{da}$	=	$\frac{2(1)}{40}$	6.25E-01	(1 -	6.25E-01	) =	1.17E-02
$\frac{d\mu_3}{da}$	=	$\frac{2(1)}{70}$	2.85E-01	(1 -	2.85E-01	) =	5.80E-03
$\frac{d\mu_4}{da}$	=	$\frac{2(1)}{130}$	7.01E-01	(1 -	7.01E-01	) =	3.20E-03
$\frac{d\mu_5}{da}$	=	$\frac{2(1)}{65}$	9.14E-01	(1 -	9.14E-01	) =	2.40E-03
$\frac{d\mu_6}{da}$	=	$\frac{2(1)}{130}$	8.93E-01	(1 -	8.93E-01	) =	1.50E-03
$\frac{d\mu_7}{da}$	=	$\frac{2(1)}{40}$	4.20E-01	(1 -	4.20E-01	) =	1.22E-02
$\frac{d\mu_8}{da}$	=	$\frac{2(1)}{20}$	7.35E-01	(1 -	7.35E-01	) =	1.95E-02
$\frac{d\mu_9}{da}$	=	$\frac{2(1)}{40}$	7.52E-01	(1 -	7.52E-01	) =	9.30E-03

b. Parameter Premis b

$$\frac{d\mu_i}{db} = \begin{cases} -2\ln\left|\frac{x-c}{a}\right| \mu_i\,(1-\mu_i) & x \neq c \\ 0 & x = c \end{cases}$$

Data ke - 1

$$\frac{d\mu_1}{db} = -2\ln\left|\frac{99-60}{50}\right| \quad 6.22E\text{-}01 \quad (1- \quad 6.22E\text{-}01 \quad) = 0.1169$$

$$\frac{d\mu_2}{db} = -2\ln\left|\frac{99-130}{40}\right| \quad 6.25E\text{-}01 \quad (1- \quad 6.25E\text{-}01 \quad) = 0.1195$$

$$\frac{d\mu_3}{db} = -2\ln\left|\frac{99-210}{70}\right| \quad 2.85E\text{-}01 \quad (1- \quad 2.85E\text{-}01 \quad) = -0.1877$$

$$\frac{d\mu_4}{db} = -2\ln\left|\frac{95-10}{130}\right| \quad 7.01E\text{-}01 \quad (1- \quad 7.01E\text{-}01 \quad) = 0.1783$$

$$\frac{d\mu_5}{db} = -2\ln\left|\frac{95-75}{65}\right| \quad 9.14E\text{-}01 \quad (1- \quad 9.14E\text{-}01 \quad) = 0.1862$$

$$\frac{d\mu_6}{db} = -2\ln\left|\frac{95-140}{130}\right| \quad 8.93E\text{-}01 \quad (1- \quad 8.93E\text{-}01 \quad) = 0.2027$$

$$\frac{d\mu_7}{db} = -2\ln\left|\frac{47-0}{40}\right| \quad 4.20E\text{-}01 \quad (1- \quad 4.20E\text{-}01 \quad) = -0.0786$$

$$\frac{d\mu_8}{db} = -2\ln\left|\frac{47-35}{20}\right| \quad 7.35E\text{-}01 \quad (1- \quad 7.35E\text{-}01 \quad) = 0.1989$$

$$\frac{d\mu_9}{db} = -2\ln\left|\frac{47-70}{40}\right| \quad 7.52E\text{-}01 \quad (1- \quad 7.52E\text{-}01 \quad) = 0.2067$$

c. Parameter Premis c

$$\frac{d\mu_i}{dc} = \begin{cases} -\frac{2b}{x-c}\mu_i(1-\mu_i) & x \neq c \\ 0 & x = c \end{cases}$$

Data ke -1

$$\frac{d\mu_i}{dc} = -\frac{2(1)}{99-60} \quad 6.22E\text{-}01 \quad (1- \quad 6.22E\text{-}01 \quad) = -0.0121$$

$$\frac{d\mu_i}{dc} = -\frac{2(1)}{99-130} \quad 6.25E\text{-}01 \quad (1- \quad 6.25E\text{-}01 \quad) = 0.0151$$

$$\frac{d\mu_i}{dc} = -\frac{2(1)}{99-210} \quad 2.85E\text{-}01 \quad (1- \quad 2.85E\text{-}01 \quad) = 0.0037$$

$\frac{d\mu_i}{dc} = -\frac{2(1)}{95-10}$ 7.01E-01 (1- 7.01E-01) = -0.0049
$\frac{d\mu_i}{dc} = -\frac{2(1)}{95-75}$ 9.14E-01 (1- 9.14E-01) = -0.0079
$\frac{d\mu_i}{dc} = -\frac{2(1)}{95-140}$ 8.93E-01 (1- 8.93E-01) = 0.0042
$\frac{d\mu_i}{dc} = -\frac{2(1)}{47-0}$ 4.20E-01 (1- 4.20E-01) = -0.0104
$\frac{d\mu_i}{dc} = -\frac{2(1)}{47-35}$ 7.35E-01 (1- 7.35E-01) = -0.0324
$\frac{d\mu_i}{dc} = -\frac{2(1)}{47-70}$ 7.52E-01 (1- 7.52E-01) = 0.0162

Sehingga parameter fungsi keanggotaan fuzzy yang baru adalah fungsi keanggotaan fuzzy yang lama ditambah dengan turunan persamaan yang terdapat pada persamaan 2.25 hingga 2.26 dan hasil parameter fungsi keanggotaan fuzzy yang baru ada pada tabel Fungsi Keanggotaan Fuzzy Baru (tabel dilihat di Lampiran D).

$$a_{baru}=a_{lama}+\frac{d\mu_i}{da}$$

$$b_{baru}=b_{lama}+\frac{d\mu_i}{db}$$

$$c_{baru}=c_{lama}+\frac{d\mu_i}{dc}$$

Data ke – 1 (Keanggotaan *Red*)

a_{baru} = 50 + 9.41E-03 = 5.00E+01
b_{baru} = 1 + 1.17E-01 = 1.12E+00
c_{baru} = 60 + -1.21E-02 = 6.00E+01
a_{baru} = 40 + 1.17E-02 = 4.00E+01
b_{baru} = 1 + 1.20E-01 = 1.12E+00
c_{baru} = 130 + 1.51E-02 = 1.30E+02
a_{baru} = 70 + 5.82E-03 = 7.00E+01
b_{baru} = 1 + -1.88E-01 = 8.12E-01
c_{baru} = 210 + 9.41E-03 = 2.10E+02

Berdasarkan tabel *Error* Lapisan 1 (tabel dilihat di Lampiran D) untuk mendapatkan nilai parameter a,b dan c digunakan persamaan 2.28 hingga 2.30

dengan pembagi sejumlah data masukan yaitu 18, sehingga didapat nilai parameter a,b dan c yang dapat digunakan untuk *epoch* selanjutnya.

$$a_{baru} = \frac{\sum a_i}{18}$$

$$b_{baru} = \frac{\sum b_i}{18}$$

$$c_{baru} = \frac{\sum c_i}{18}$$

Data ke – 1 (Keanggotaan *Red*)

a_{baru} = 900.1359893 / 18 = 50.00755496
b_{baru} = 16.09208325 / 18 = 0.89400463
c_{baru} = 1079.886627 / 18 = 59.99370151
a_{baru} = 720.1514459 / 18 = 40.00841366
b_{baru} = 19.62664647 / 18 = 1.09036925
c_{baru} = 2339.976643 / 18 = 129.9987024
a_{baru} = 1260.103829 / 18 = 70.0057683
b_{baru} = 17.86388809 / 18 = 0.99243823
c_{baru} = 3780.110683 / 18 = 210.006149

Sehingga untuk *epoch* pertama dihasilkan parameter premis seperti pada Tabel 3.11.

Tabel 3.11 Parameter Konsekuen Baru

Himpunan Fuzzy		**a**	**b**	**c**
Red	**Rendah**	50.00755	0.894	59.9937
	Sedang	40.00841	1.09037	129.9987
	Tinggi	70.00577	0.99244	210.00615
Green	**Rendah**	130.0022	1.14327	9.99596
	Sedang	65.00403	1.14377	75.00029
	Tinggi	130.00225	1.14209	140.00399
Blue	**Rendah**	40.00877	1.09717	-0.01299
	Sedang	20.01719	1.05519	35.00851
	Tinggi	40.01082	0.98965	70.0114

Premis pada Tabel 3.11 digunakan untuk perhitungan selanjutnya yang akan dilakukan selama memenuhi keadaan (*epoch* < *maxepoch*) dan (*RMSE* > *epsilon*),

setelah dilakukan perhitungan mencapai *epoch* ke-10, selanjutnya didapatkan premis akhir yang akan digunakan untuk proses pengujian seperti pada Tabel 3.12.

Tabel 3.12 Premis untuk Pengujian

Himpunan Fuzzy		**a**	**b**	**c**
Red	**Rendah**	50.00755	0.894	59.9937
	Sedang	40.00841	1.09037	129.9987
	Tinggi	70.00577	0.99244	210.00615
Green	**Rendah**	130.0022	1.14327	9.99596
	Sedang	65.00403	1.14377	75.00029
	Tinggi	130.00225	1.14209	140.00399
Blue	**Rendah**	40.00877	1.09717	-0.01299
	Sedang	20.01719	1.05519	35.00851
	Tinggi	40.01082	0.98965	70.0114

2. Pengujian

Pada tahap pengujian akan menggunakan premis hasil proses pembelajaran ANFIS yang telah dilakukan. Dalam tahap pengujian ANFIS hanya menggunakan alur maju dengan perhitungan yang sama saat pembelajaran yaitu perhitungan *membership function*, *sistem inference fuzzy*, *normalisasi*, perhitungan LSE dan *defuzzyfikasi*.

1. Lapisan 1 (Fuzzification)

Terdapat ketentuan untuk tiap *membership function* dari *red*, *green* dan *blue* dengan persamaan 2.7, fungsi keanggotaannya adalah sebagai berikut :

a. Membership function untuk *red*

Rendah (R) : gbell(x; 50.00755; 0.894; 59.9937)

Sedang (S) : gbell(x; 40.00841; 1.09037; 129.9987)

Tinggi (T) : gbell(x; 70.00577; 0.99244; 210.00615)

Data ke -1

R = 139; G = 118; B = 28

$\mu_R(233) = gbell(139,50.00755,1.894,59.9937)$

$$= \frac{1}{1+\left|\frac{139-59.9937}{50.00755}\right|^2} = 0.62174$$

$\mu_S\,(233) = gbell(139,40.00841,1.09037,129.9987)$

$$= \frac{1}{1+\left|\frac{139-129.9987}{40.00841}\right|^2} = 0.62476$$

$\mu_T\,(233) = gbell(139,70.00577,0.99244,210.00615)$

$$= \frac{1}{1+\left|\frac{139-210.00615}{70.00577}\right|^2} = 0.28454$$

b. Membership function untuk *green*

Rendah (R) : gbell(x; 130.0022; 1.14327; 9.99596)

Sedang (S) : gbell(x; 65.00403; 1.14377; 75.00029)

Tinggi (T) : gbell(x;130.00225; 1.14209; 140.00399)

Data ke -1

R = 139; G = 118; B = 28

$\mu_R\,(233) = gbell(118,130.022,1.14327,9.99596)$

$$= \frac{1}{1+\left|\frac{118-9.99596}{130.022}\right|^2} = 0.70052$$

$\mu_S\,(233) = gbell(118,65.00403,1.14377,75.00029)$

$$= \frac{1}{1+\left|\frac{118-75.00029}{65.00403}\right|^2} = 0.91351$$

$\mu_T\,(233) = gbell(118,130.00225,1.14209,140.00399)$

$$= \frac{1}{1+\left|\frac{118-140.00399}{130.00225}\right|^2} = 0.893$$

c. Membership function untuk *blue*

Rendah (R) : gbell(x; 40.00877; 1.09717; -0.01299)

Sedang (S) : gbell(x; 20.01719; 1.05519; 35.00851)

Tinggi (T) : gbell(x; 40.01082; 0.98965; 70.0114)

Data ke -1

R = 139; G = 118; B = 28

$$\mu_R(233) = gbell(28, 40.00877, 1.09717, -0.01299)$$

$$= \frac{1}{1+\left|\frac{28-(-0.01299)}{40.00877}\right|^2} = 0.42006$$

$$\mu_S(233) = gbell(28, 20.01719, 1.05519, 35.00851)$$

$$= \frac{1}{1+\left|\frac{28-35.00851}{20.01719}\right|^2} = 0.73529$$

$$\mu_T(233) = gbell(28, 40.01082, 0.98965, 70.0114)$$

$$= \frac{1}{1+\left|\frac{28-70.0114}{40.01082}\right|^2} = 0.73153$$

Fuzzyfication dilakukan pada setiap data yang dimasukan. Berdasarkan perhitungan *membership function* yang telah dilakukan pada 18 data yang telah dilakukan didapatkan hasil pada tabel *Membership Function* Baru (tabel dilihat di Lampiran D).

2. Lapisan 2 (*System Inference Fuzzy*)

Terdapat 18 aturan (*rule*) pada tabel 3.10 yang akan digunakan untuk menentukan nilai keluaran pada lapisan ini. Perhitungan dilakukan sama dengan langkah perhitungan pada lapisan 2 proses pembelajaran dengan menggunakan operator *min* pada langkah ini akan dihitung nilai W_i dari tiap *membership function* nilai R, G dan B dengan menggunakan persamaan 2.8 dengan menghasilkan nilai keluaran pada tabel *Output* Lapisan 2 Pengujian (tabel dilihat di Lampiran D).

3. Lapisan 3 (*Normalisasi*)

Pada langkah ini akan dihitung derajat pengaktifan *firing strength* (W_i) dengan menggunakan persamaan 2.9 . Perhitungan dilakukan sama dengan langkah perhitungan pada lapisan 3 proses pembelajaran dan menghasilkan nilai keluaran pada tabel *Output* Lapisan 3 Pengujian (tabel dilihat di Lampiran D).

4. Lapisan 4 (*Least Square Estimator*)

Pada lapisan ini akan dilakukan perhitungan LSE untuk mendapatkan parameter konsekuen. Perhitungan yang dilakukan sama perhitungan pada lapisan

4 proses pembelajaran dan menghasilkan parameter konsekuen seperti pada Tabel 3.13.

Tabel 3.13 Parameter Konsekuen

No.	P_i	Q_i	R_i	S_i
1	1.066	-0.781	1.021	-0.009
2	0.391	1.312	-1.214	-0.009
3	-0.836	-0.197	-1.258	-0.034
4	-0.836	-0.197	-1.258	-0.034
5	0.391	1.312	-1.214	-0.009
6	0.933	-2.559	-0.911	-0.018
7	2.263	-0.592	-0.189	-0.001
8	-2.198	0.917	1.709	0.005
9	0.131	0.927	-1.023	-0.021
10	0.225	-2.927	-1.845	0.003
11	0.001	-0.219	0.327	-0.007
12	-0.252	-0.322	0.292	-0.009
13	-0.095	0.757	0.467	-0.011
14	-0.964	-0.062	0.131	-0.015
15	-0.088	3.098	-0.183	0.002
16	0.519	0.761	0.266	-0.007
17	0.66	0.058	0.417	-0.004
18	-0.252	-0.322	0.292	-0.009

Setelah didapatkan nilai parameter konsekuen pada lapisan ini selanjutnya menghitung nilai keluaran lapisan 4 terhadap semua data masukan menggunakan persamaan 2.10, hasil dari perhitungan tersebut pada 18 data yang dimasukan dapat dilihat pada tabel *Output* Lapisan 4 Pengujian (tabel dilihat di Lampiran D).

5. Lapisan 5 (*Defuzzification*)

Setelah didapatkan hasil dari lapisan 4, langkah selanjutnya adalah defuzzifikasi dengan melakukan penjumlahan seperti pada persamaan 2.11, berikut hasil dari hasil defuzzifikasi dari 18 data yang dimasukan ada pada Tabel 3.14.

Tabel 3.14 *Output* Lapisan 5

No	$\mathbf{O_5}$	Stage	Stage ANFIS
1	1.99999999989028	2	2

2	4.0000000001075	4	4
3	1.00000000020199	1	1
4	3.00000000010615	3	3
5	0.999999999765415	1	1
6	6.00000000005691	6	6
7	5.00000000015125	5	5
8	0.99999999991445	1	1
9	6.00000000005071	6	6
10	4.00000000011041	4	4
11	3.00000000006182	3	3
12	2.00000000003346	2	2
13	0.999999999960481	1	1
14	2.00000000002229	2	2
15	0.999999999864867	1	1
16	1.000000000059	1	1
17	3.0000000000398	3	3
18	6.00000000000912	6	6

Setelah didapatkan hasil klasifikasi dari ANFIS, tahap selanjutnya adalah perhitungan nilai RMSE untuk mengetahui besar *error* yang ada pada pengujian ANFIS tersebut.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}\left(Y_i-(\sum \overline{w_i} f_i)\right)^2}{n}}$$

$$RMSE = \sqrt{\frac{(2\text{-}1.99999999989028)^2+\ldots+(6\text{-}6.00000000000912)^2}{18}} = \sqrt{\frac{0.000000000000000021}{18}}$$

$$=0.00000000010783155206$$

Berdasarkan hasil proses pengujian ANFIS terhadap 18 data pengujian yang telah dilakukan dengan hasil premis yang dari proses pembelajaran dengan *maxepoch* = 10, *learning rate* = 0.05 dan *eps* = 0.01 didapatkan persentase akurasi yang dapat dilakukan oleh ANFIS adalah 100 % dengan RMSE = 0.00000000010783155206.

Berdasarkan pembelajaran dan pengujian yang telah dilakukan dengan menggunakan LVQ dan ANFIS dengan *maxepoch* = 10, *learning rate* = 0.05 dan *eps* = 0.01 didapatkan hasil klasifikasi yang dilakukan oleh LVQ adalah 55.55556 % dengan RMSE = 0.881917, sedangkan hasil klasifikasi yang dilakukan oleh ANFIS adalah 100 % dengan RMSE = 0.00000000010783155206. Sehingga dapat disimpulkan untuk parameter tersebut ANFIS lebih baik dibandingkan dengan LVQ dengan hasil klasifikasi seperti pada Tabel 3.15.

Tabel 3.15 Hasil Klasifikasi LVQ dan ANFIS

No	Stage	Stage LVQ	Keterangan	Stage ANFIS	Keterangan
1	2	2	Cocok	2	Cocok
2	4	4	Cocok	4	Cocok
3	1	1	Cocok	1	Cocok
4	3	2	Tidak Cocok	3	Cocok
5	1	1	Cocok	1	Cocok
6	6	5	Tidak Cocok	6	Cocok
7	5	5	Cocok	5	Cocok
8	1	1	Cocok	1	Cocok
9	6	4	Tidak Cocok	6	Cocok
10	4	4	Cocok	4	Cocok
11	3	2	Tidak Cocok	3	Cocok
12	2	3	Tidak Cocok	2	Cocok
13	1	1	Cocok	1	Cocok
14	2	2	Cocok	2	Cocok
15	1	3	Tidak Cocok	1	Cocok
16	1	2	Tidak Cocok	1	Cocok
17	3	4	Tidak Cocok	3	Cocok
18	6	6	Cocok	6	Cocok
Jumlah Cocok		**10**		**18**	
Persentase Akurasi (%)		**55.55556**		**100**	

3.1.5. Analisis Kebutuhan Non Fungsional

Analisis non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan simulator. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk simolator yang akan dibangun sampai dengan simolator tersebut diimplementasikan. Analisis kebutuhan

ini juga menentukan spesifikasi masukan yang diperlukan dalam simulator, keluaran yang akan dihasilkan simulator dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

Pada analisis kebutuhan simulator non fungsional ini dijelaskan analisis mengenai perangkat keras (*hardware*) dan perangkat lunak (*software*) sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan simulator yang akan diterapkan.

3.1.5.1. Kebutuhan Perangkat Keras

Perangkat keras adalah bagian dari komputer yang menerima intruksi dari perangkat lunak untuk menjalankan intruksi agar dapat menjalankan suatu sistem didalamnya. Adapun perangkat keras yang digunakan untuk membangun simulator ini adalah sebagai berikut :

a. *Processor* : Intel Pentium Core 2 Duo @ 2.2 Ghz
b. *Memory* : 1 GB
c. VGA : 512 MB
d. *Harddisk* : 256 GB
e. *Monitor* : 14 inch
f. Mouse dan Keyboard

3.1.5.2. Kebutuhan Perangkat Lunak

Perangkat lunak adalah bagian dari komputer yang memberikan intruksi kepada perangkat keras untuk menjalankan intruksi agar dapat menjalankan suatu sistem didalamnya. Adapun perangkat lunak yang digunakan untuk membangun simulator ini adalah sebagai berikut :

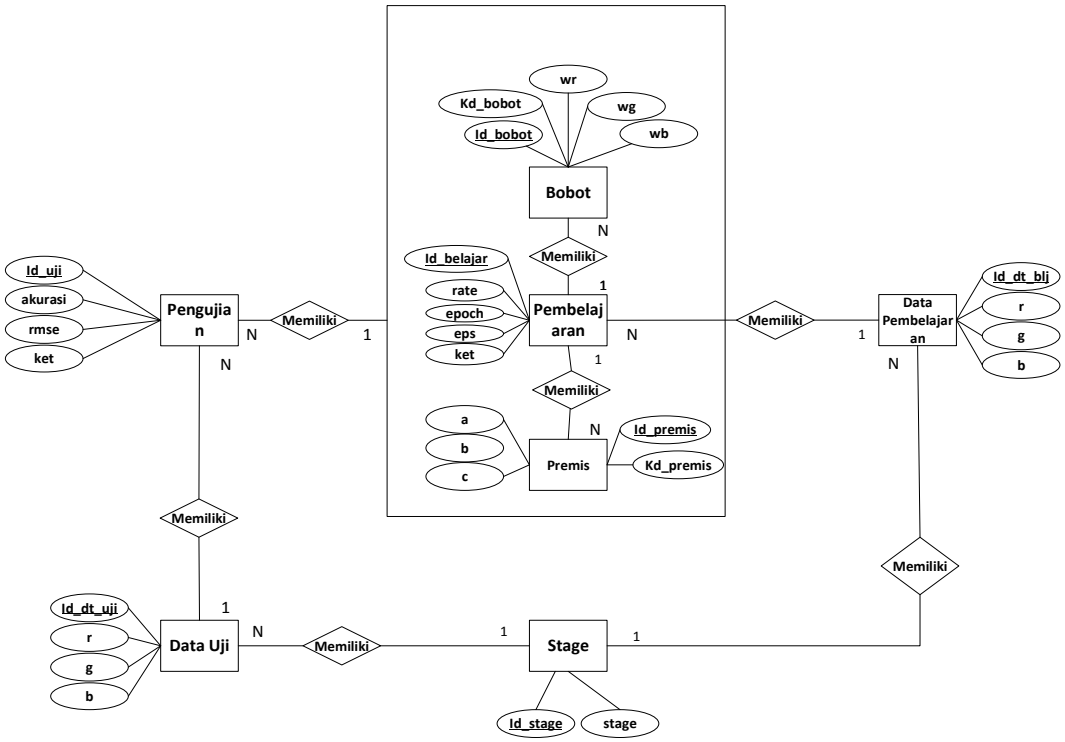
a. Sistem Operasi : Windows 7
b. Microsoft Visual Studio 2010
c. XAMPP server
d. Navicat

3.2. Pemodelan Sistem

Pemodelan sistem adalah bagian yang mendeklarasikan fungsi – fungsi yang dapat dilakukan oleh simulator berdasarkan analisis metode yang telah dilakukan sebelumnya dan mendeksripsikannya kedalam bentuk diagram diantaranya *Entity Relationship Diagram* (ERD), *usecase diagram, diagram scenario, activity diagram, sequence diagram, collaboration diagram* dan *class diagram.*

3.2.1. Entity Diagram Relational (ERD)

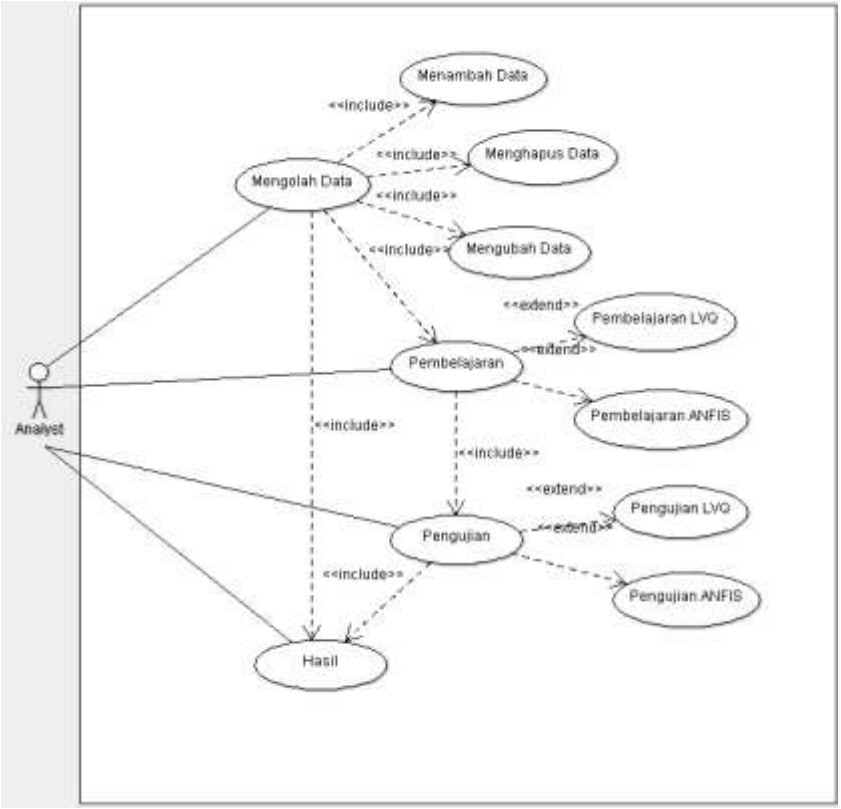
Entity Relationship Diagram (ERD) merupakan salah satu cara untuk mengolah *database* sehingga data tersebut dapat diketahui hubungan antara *file* dan teknik, ini dapat digunakan untuk mengatasi terjadinya redudansi data atau sejenisnya. Untuk lebih jelasnya bentuk *Entity Relationship Diagram (*ERD) tersebut dapat dilihat pada gambar di bawah ini :



Gambar 3.7 *Entity Diagram Relational*

3.2.2. Use Case Diagram

Use case diagram merupakan suatu aktivitas sistem dalam menanggapi permintaan pengguna dalam sebuah masalah dimana sistem yang digunakan oleh pengguna (*actor*). Adapun *use case diagram* tersebut dapat dilihat pada Gambar 3.8 berikut :



Gambar 3.8 *Use Case* Diagram

3.2.2.1. Definisi Actor

Actor diperlukan untuk terciptanya *use case diagram* untuk mempersentasikan seseorang atau perangkat yang berinteraksi dengan sistem. Berikut adalah *actor* yang terlibat dalam simulator yang akan dibangun :

Tabel 3.16 Definisi Actor

No	Actor	Deskripsi
1	*Analyst*	*Actor* dengan aturan ini mempunyai kewenangan untuk melakukan persiapan data olah, pembelajaran LVQ, pembelajarajan ANFIS dan perbandingan hasil LVQ dan ANFIS

3.2.2.2. Definisi Usecase

Bagian ini diisi dengan daftar *use case* dan deskripsi singkat mengenai *use case* pada simulator yang akan dibangun :

Tabel 3.17 Definisi Use Case

No	Use Case	Deskripsi
1	Menyiapkan Data	Fungsionalitas yang digunakan oleh *Analyst* untuk mengolah data pembelajaran dan data uji tomat didalam sistem
2	Menambah Data	Fungsionalitas yang digunakan oleh *Analyst* untuk menambah data pembelajaran dan data uji tomat kedalam sistem
3	Menghapus Data	Fungsionalitas yang digunakan oleh *Analyst* untuk menghapus data tomat dari data tomat yang sudah ada didalam sistem
4	Mengubah Data	Fungsionalitas yang digunakan oleh *Analyst* untuk mengubah data tomat dari data tomat yang sudah ada didalam sistem
5	Pembelajaran	Fungsionalitas yang digunakan oleh *analyst* untuk melakukan pemilihan pembelajaran yang akan dilakukan
6	Pembelajaran LVQ	Fungsionalitas yang digunakan oleh *Analyst* untuk melakukan pembelajaran data pembelajaran dengan menggunakan metode LVQ
5	Pembelajaran ANFIS	Fungsionalitas yang digunakan oleh An*a*lyst untuk melakukan pembelajaran data pembelajaran dengan menggunakan metode ANFIS
8	Pengujian	Fungsionalitas yang digunakan oleh *analyst* untuk melakukan pemilihan pengujian yang akan dilakukan
9	Pengujian LVQ	Fungsionalitas yang digunakan oleh *Analyst* untuk melakukan pengujian data uji dengan menggunakan metode LVQ
10	Pengujian ANFIS	Fungsionalitas yang digunakan oleh *Analyst* untuk melakukan pengujian data uji dengan menggunakan metode ANFIS
11	Hasil	Fungsionalitas yang digunakan oleh *Analyst* untuk melihat hasil pembelajaran berupa bobot dan premis serta hasil pengujian berupa hasil akurasi dan nilai *error* dari dari pengujian

3.2.3. Skenario Diagram

Bagian ini diisi dengan skenario untuk setiap *use case* pada rancangan yang menggambarkan urutan interaksi *actor* dengan simulator.

Tabel 3.18 Menyiapkan Data

Use case Name	Menyiapkan Data
Related Requirements	-
Goal In Context	*Analyst* dapat mengolah citra tomat menjadi data yang akan digunakan sebagai masukan untuk proses pembelajaran dan pengujian jaringan syaraf tiruan.
Preconditions	*Analyst* mempunyai citra tomat
Succsessful End Condition	Data hasil pengolahan citra tomat berhasil diolah kedalam *database*

Failed End Condition	Data hasil pengolahan citra tomat tidak berhasil diolah kedalam *database*		
Primary Actors	Analyst		
	Trigger	Sistem menampilkan data yang sudah tersimpan dalam database	
Main Flow	Step	Action	
		Analyst	Simulator
	1	Memilih menu data	
	2		Menampilkan form data
	3	Mengolah data tomat yang ada dalam sistem	
	4		Memproses hasil olahan data yang dilakukan
	5		Menyimpan data
Extensions	Step	Branching Action	
	3.1	Menambah Data	
	3.2	Mengubah Data	
	3.3	Menghapus Data	
	5	Data tidak berhasil disimpan	

Tabel 3.19 Menambah Data

Use case Name	Menambah Data		
Related Requirements	Menyiapkan Data		
Goal In Context	*Analyst* dapat menambah data hasil pengolahan citra tomat yang akan digunakan untuk proses pembelajaran dan pengujian jaringan syaraf tiruan.		
Preconditions	Analyst mempunyai citra tomat		
Succsessful End Condition	Data hasil pengolahan citra tomat berhasil dimasukan kedalam sistem		
Failed End Condition	Data hasil pengolahan citra tomat tidak berhasil disimpan kedalam sistem		
Primary Actors	Analyst		
	Trigger	Data tomat berhasil disimpan kedalam *database*	
Main Flow	Step	Action	
		Analyst	Simulator
	1	Berada dalam menu data	
	2	Memilih jenis data	
	3		Menampilkan data yang sudah tersimpan sesuai jenis data yang dipilih
	4	Memilih untuk menambah data	
	5		Mengaktifkan menu tambah data
	6	Memasukan citra tomat	
	7		Menampilkan citra tomat yang dimasukan
	8	Memotong citra romat	
	9		Menampilkan citra yang telah dipotong
	10	Memilih tombol ekstraksi ciri RGB	

	11		Menhitung dan Menampilkan nilai RGB dari citra
	12	Memasukan stage dan kode	
	13		Data divalidasi kelengkapannya
	14		Data disimpan kedalam database
Extensions	Step	Branching Action	
	2.1	Memilih jenis data pembelajaran	
	2.2	Memilih jenis data pengujian	
	3.1	Menampilkan jenis data pembelajaran	
	3.2	Menampilkan jenis data pengujian	
	14.1	Jika belum lengkap maka kembali ke proses 13, data tidak disimpan	

Tabel 3.20 Menghapus Data

Use case Name	Menghapus Data		
Related Requirements	Menyiapkan Data		
Goal In Context	Analyst dapat menghapus data tomat yang sudah ada didalam sistem.		
Preconditions	Data tomat yang akan dihapus sudah ada dalam sistem		
Succsessful End Condition	Data tomat berhasil dihapus dari sistem		
Failed End Condition	Data tomat tidak berhasil dihapus dari sistem		
Primary Actors	Analyst		
	Trigger	Sistem menampilkan data yang sudah tersimpan dalam sistem	
Main Flow	Step	Action	
		Analyst	Simulator
	1	Berada dalam menu data	
	2	Memilih jenis data	
	3		Menampilkan data yang sudah tersimpan sesuai jenis data yang dipilih
	4	Memilih data yang akan dihapus	
	5		Menampilkan detail data yang dipilih
	6	Memilih menghapus data yang telah dipilih	
	7		Menghapus data yang dipilih
	8		Data terhapus
		Branching Action	
	2.1	Memilih jenis data pembelajaran	
	2.2	Memilih jenis data pengujian	
	3.1	Menampilkan jenis data pembelajaran	
	3.2	Menampilkan jenis data pengujian	
	8.1	Data tidak berhasil terhapus	

Tabel 3.21 Mengubah Data

Use case Name	Mengubah Data
Related Requirements	Menyiapkan Data

Goal In Context	Analyst dapat mengubah data tomat yang sudah ada didalam sistem.		
Preconditions	Data tomat yang akan diubah sudah ada dalam sistem		
Succsessful End Condition	Data tomat berhasil diubah dari sistem		
Failed End Condition	Data tomat tidak berhasil diubah dari sistem		
Primary Actors	Analyst		
	Trigger	Sistem menampilkan data yang sudah tersimpan dalam database	
Main Flow	Step	Action	
		Analyst	Simulator
	1	Berada dalam menu data	
	2	Memilih jenis data	
	3		Menampilkan data yang sudah tersimpan sesuai jenis data yang dipilih
	4	Memilih data yang akan diubah	
	5		Menampilkan detail data yang dipilih
	6	Memasukan perubahan data	
	7		Mengubah data yang dipilih
	8		Data berhasil diubah
	Branching Action		
	2.1	Memilih jenis data pembelajaran	
	2.2	Memilih jenis data pengujian	
	3.1	Menampilkan jenis data pembelajaran	
	3.2	Menampilkan jenis data pengujian	
	8.1	Data tidak berhasil diubah	

Tabel 3.22 Pembelajaran

Use case Name	Pembelajaran		
Related Requirements	Menyiapkan Data		
Goal In Context	Analyst dapat memilih pembelajaran LVQ atau ANFIS		
Preconditions	Analyst akan memilih salah satu metode pembelajaran		
Succsessful End Condition	Data pembelajaran akan diklasifikasi seseuai metode yang dipilih		
Failed End Condition	Data tidak dapat diklasifikasi		
Primary Actors	Analyst		
	Trigger	Sistem menampilkan data yang sudah tersimpan dalam database	
Main Flow	Step	Action	
		Analyst	Simulator
	1	Memilih menu pembelajaran	
	2		Menampilkan pilihan pembelajaran
	3	Memasukan nilai inisialisasi kebutuhan LVQ berupa maxepoch, learning rate dan *error* minimum	
	4		Memproses pembelajaran LVQ dan ANFIS
		Branching Action	

Tabel 3.23 Pembelajaran LVQ

Use case Name	Pembelajaran LVQ		
Related Requirements	Pembelajaran		
Goal In Context	Berhasil menghasilkan bobot baru untuk pengujian dari data yang diolah dengan menggunakan LVQ		
Preconditions	*Analyst* telah mengisi nilai awal *learning rate*, bobot awal, *maxepoch* dan *eps*		
Succsessful End Condition	Hasil klasifikasi sesuai dengan target kelas awal		
Failed End Condition	Hasil klasifikasi tidak sesuai dengan target kelas awal		
Primary Actors	*Analyst*		
	Trigger		
Main Flow	Step	Action	
		Analyst	Simulator
	1	Berada dalam menu pembelajaran	
	2	Menentukan bobot awal	
	3		Mengecek nilai inisialisasi
	4		Menghitung jarak data pada tiap bobot
	5		Perbaharui Bobot
	6		Penambahan Epoch dan Pengurangan *Learning Rate*
	7		Menghitung RMSE
	8		Menyimpan bobot kedalam *database*
	9		Menampilkan data hasil pembelajaran LVQ berupa klasifikasi dan bobot
	10	Melihat data hasil pembelajaran LVQ berupa klasifikasi dan bobot	
Extensions	Step	Branching Action	
	3.1	Jika nilai inisialisasi belum lengkap atau salah maka kembali ke proses inisialisasi oleh *analyst*	
	5.1	Perbaharui bobot jika epoch lebih kecil dari maxepoch atau learning rate lebih besar dari *error* minimum	

Tabel 3.24 Pembelajaran ANFIS

Use case Name	Pembelajaran ANFIS	
Related Requirements	Menyiapkan Data, Pembelajaran	
Goal In Context	Berhasil mengklasifikasikan data yang diolah dengan menggunakan ANFIS	
Preconditions	*Analyst* sudah mengisi nilai awal *maxepoch* dan *eps*	
Succsessful End Condition	Hasil klasifikasi sesuai dengan target kelas awal	
Failed End Condition	Hasil klasifikasi tidak sesuai dengan target kelas awal	
Primary Actors	*Analyst*	
	Trigger	
Main Flow	Step	Action

		Analyst	Simulator
	1	Berada dalam menu pembelajaran	
	2		Mengecek nilai inisialisasi
	3		Perhitungan *Membership Function*
	4		Perhitungan nilai bobot dengan menggunakan operator fuzzy "AND"
	5		Pengaktifan Firing Strength dengan t-normalisasi
	6		Perhitungan parameter konsekuen dengan LSE
	7		Defuzzification
	8		Perhitungan RMSE
	9		Perhitungan EBP lapian 5
	10		Perhitungan EBP lapian 4
	11		Perhitungan EBP lapian 3
	12		Perhitungan EBP lapian 2
	13		Perhitungan EBP lapian 1
	14		Update Premis
	15		Penambahan Epoch
	16		Pengecekan Kondisi Epoch dan nilai *Error*
	17		Menyimpan parameter premis kedalam database
	18		Menampilkan data hasil pembelajaran berupa klasifikasi dari tiap data yang diolah
	19	Melihat data hasil pembelajaran berupa klasifikasi dari tiap data yang diolah	
Extensions	Step	Branching Action	
	2.1	Jika nilai inisialisasi belum lengkap atau salah maka kembali ke proses inisialisasi oleh *analyst*	
	16.1	Jika memenuhi epoch < maxepoch atau nilai *error* > *epsilon* maka kembali ke langkah 3	
	16.2	Jika tidak memenuhi *epoch* < *maxepoch* atau nilai *error* > *epsilon* maka kembali ke langkah 17	

Tabel 3.25 Pengujian

Use case Name	Pengujian	
Related Requirements	Pembelajaran LVQ , Pembelajaran ANFIS	
Goal In Context	Dapat melakukan pengenalan stage pada data uji	
Preconditions	Data uji sudah ada dalam *database*	
Succsessful End Condition	Menampilkan data dalam stage hasil klasifikasi oleh kedua metode	
Failed End Condition	Gagal dalam klasifikasi	
Primary Actors	*Analyst*	
	Trigger	
Main Flow	Step	Action

		Analyst	Simulator
	1	Memilih menu pengujian	
	2		Menampilkan form pengujian
	3	Memilih data pembelajaran mana yang akan digunakan	
	4		Mengambil bobot dan premis hasil pembelajaran
Extensions	Step	Branching Action	

Tabel 3.26 Pengujian LVQ

Use case Name	Pengujian LVQ		
Related Requirements	Pembelajaran LVQ		
Goal In Context	Dapat melakukan pengenalan stage pada data uji dengan menggunakan metode LVQ		
Preconditions	Data uji dan bobot lvq sudah ada dalam *database*		
Succsessful End Condition	Menampilkan data dalam stage hasil klasifikasi oleh metode LVQ		
Failed End Condition	Gagal dalam klasifikasi		
Primary Actors	Analyst		
	Trigger		
Main Flow	Step	Action	
		Analyst	Simulator
	1	Berada dalam menu pengujian	
	2		Mengambil bobot hasil pembelajaran sesuai pembelajaran yang dipilih
	3		Memproses pengujian dengan LVQ
	4		Mencari jarak terkecil dari data data dengan tiap bobot dari tiap kelas
	5		Menampilkan hasil klasifikasi
	6		Perhitungan RMSE
	7		Perhitungan akurasi klasifikasi
	8	Melihat hasil klasifikasi	
Extensions	Step	Branching Action	

Tabel 3.27 Pengujian ANFIS

Use case Name	Pengujian ANFIS
Related Requirements	Pembelajaran ANFIS
Goal In Context	Dapat melakukan pengenalan stage pada data uji dengan menggunakan metode ANFIS
Preconditions	Data uji dan parameter premis sudah ada dalam database
Succsessful End Condition	Menampilkan data dalam stage hasil klasifikasi oleh metode LVQ
Failed End Condition	Gagal dalam klasifikasi
Primary Actors	Analyst

	Trigger		
Main Flow	Step	Action	
		Analyst	Simulator
	1	Berada dalam menu pengujian	
	2		Mengambil premis hasil pembelajaran sesuai pembelajaran yang dipilih
	3		Memproses pengujian dengan ANFIS
	4		Perhitungan Membership Function
	5		Perhitungan nilai bobot dengan menggunakan operator fuzzy “AND”
	6		Pengaktifan Firing Strength dengan t-normalisasi
	7		Perhitungan parameter konsekuen dengan LSE
	8		Defuzzification
	9		Perhitungan RMSE
	10		Perhitungan akurasi
	11	Melihat hasil klasifikasi	
Extensions	Step	Branching Action	

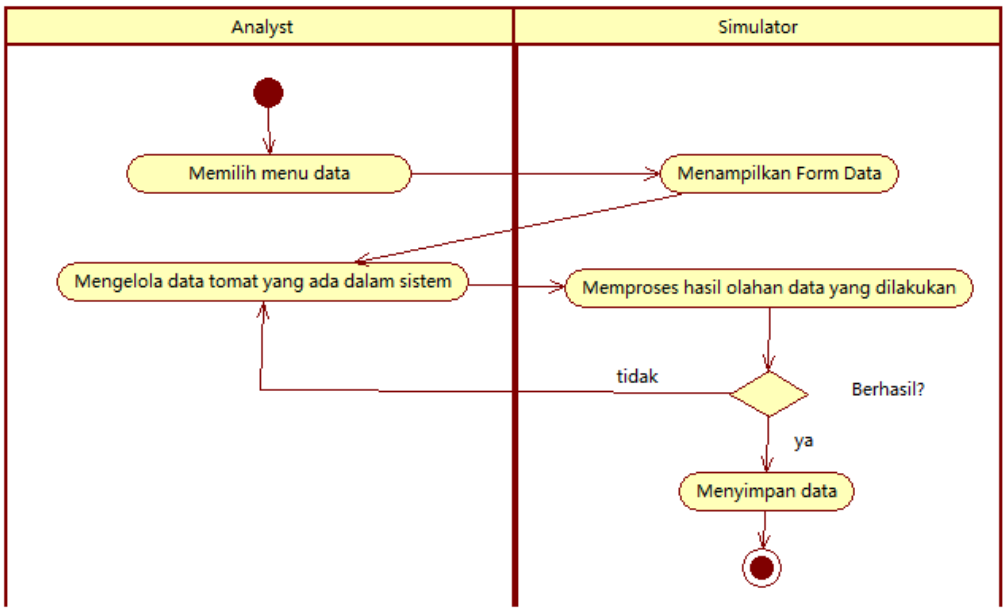
Tabel 3.28 Hasil

Use case Name	Hasil		
Related Requirements	Pembelajaran , Pengujian		
Goal In Context	Berhasil menampilkan hasil pembelajaran dan pengujian dari LVQ dan ANFIS		
Preconditions	Pembelajaran dan Pengujian sudha dilakukan		
Succsessful End Condition	Menampilkan hasil pembelajaran dan pengujian		
Failed End Condition	-		
Primary Actors	Analyst		
	Trigger		
Main Flow	Step	Action	
		Analyst	Simulator
	1	Memilih menu hasil	
	2		Menampilkan form hasil pembelajaran dan pengujian
	3	Memilih jenis Hasil yang akan dilihat	
	4		Mengambil data sesuai jenis hasil yang dipilih
	5		Menampilkan data hasil sesuai jenis yang dipilih
Extensions	Step	Branching Action	
	3.1	Jenis hasil pembelajaran	
	3.2	Jenis hasil pengujian	

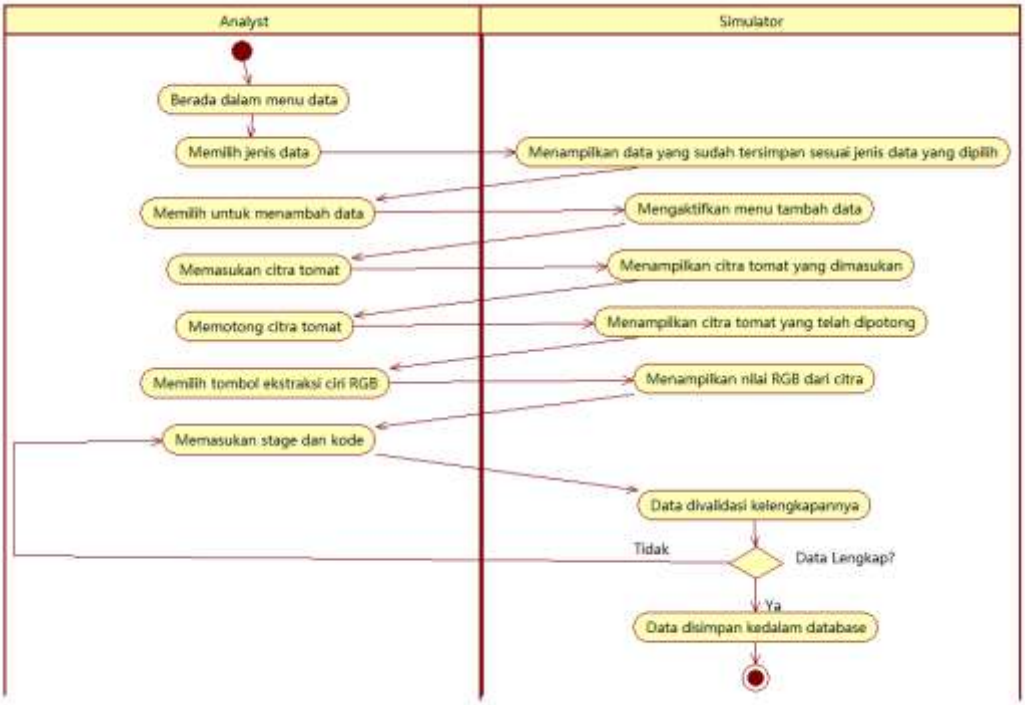
	4.1	Mengambil bobot LVQ dari hasil pembelajaran LVQ sesuai data yang dipilih, Mengambil premis hasil pembelajaran ANFIS sesuai data yang dipilih
	4.2	Mengambil akurasi dan rmse hasil pengujian LVQ dan ANFIS
	5.1	Menampilkan Bobot hasil pembelajaran LVQ dan premis hasil pembelajaran ANFIS
	5.2	Menampilkan hasil pengujian berupa akurasi dan RMSE dari LVQ dan ANFIS.

3.2.4. Activity Diagram

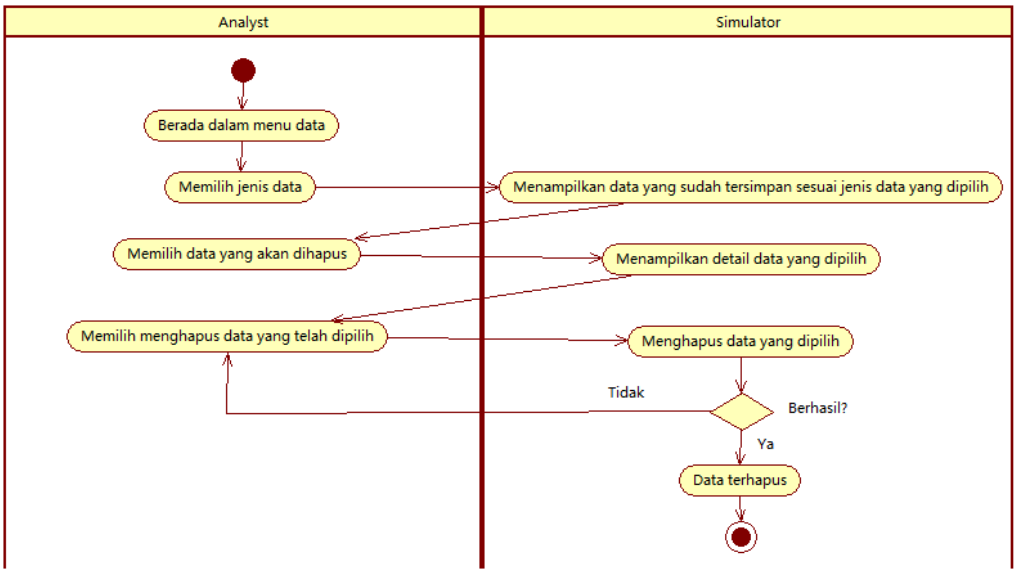
Diagram aktivitas atau *activity diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Adapun diagram aktivitas pada simulator yang akan dibangun adalah sebagai berikut :



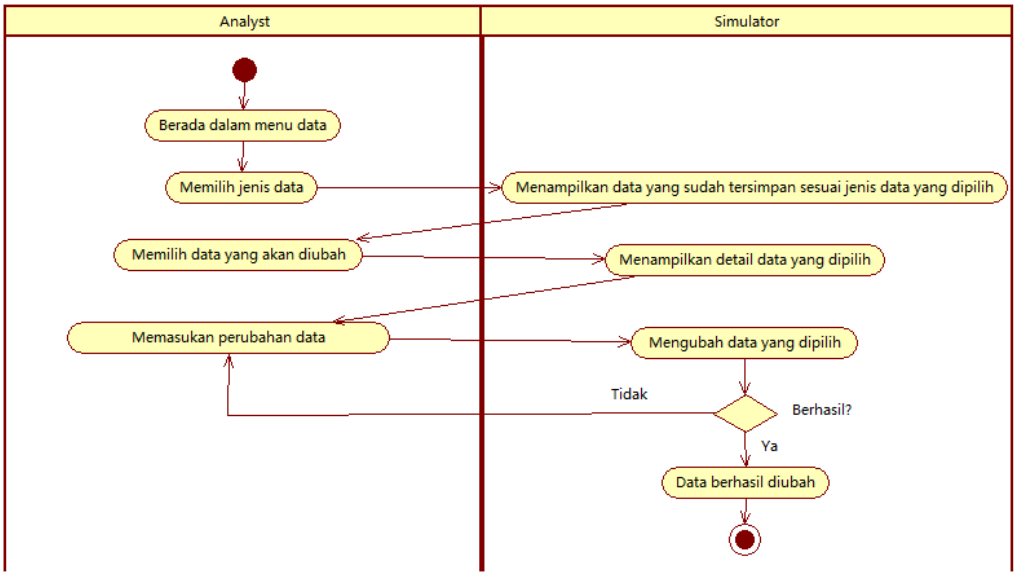
Gambar 3.9 Menyiapkan Data



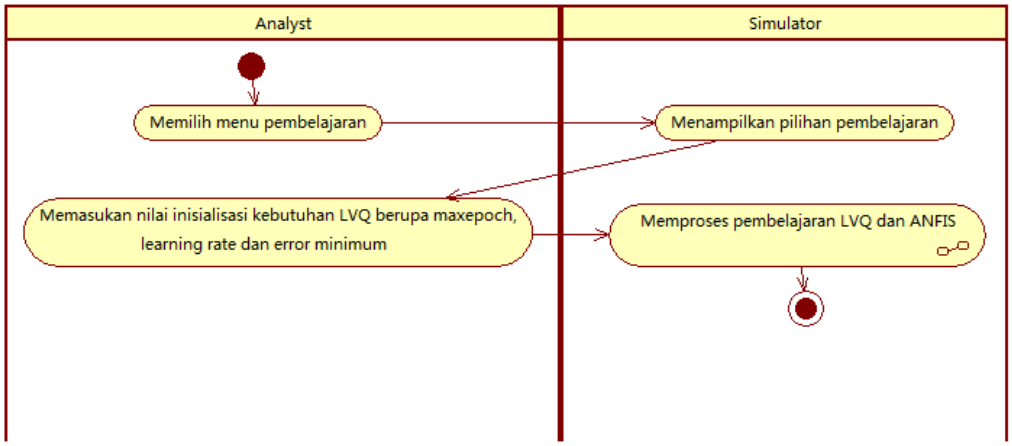
Gambar 3.10 Menambah Data



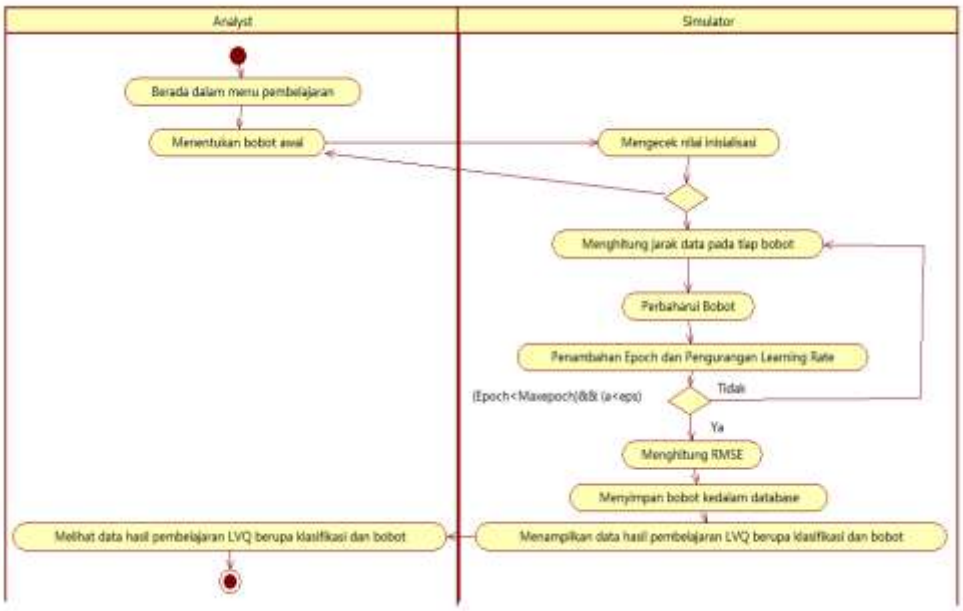
Gambar 3.11 Menghapus Data



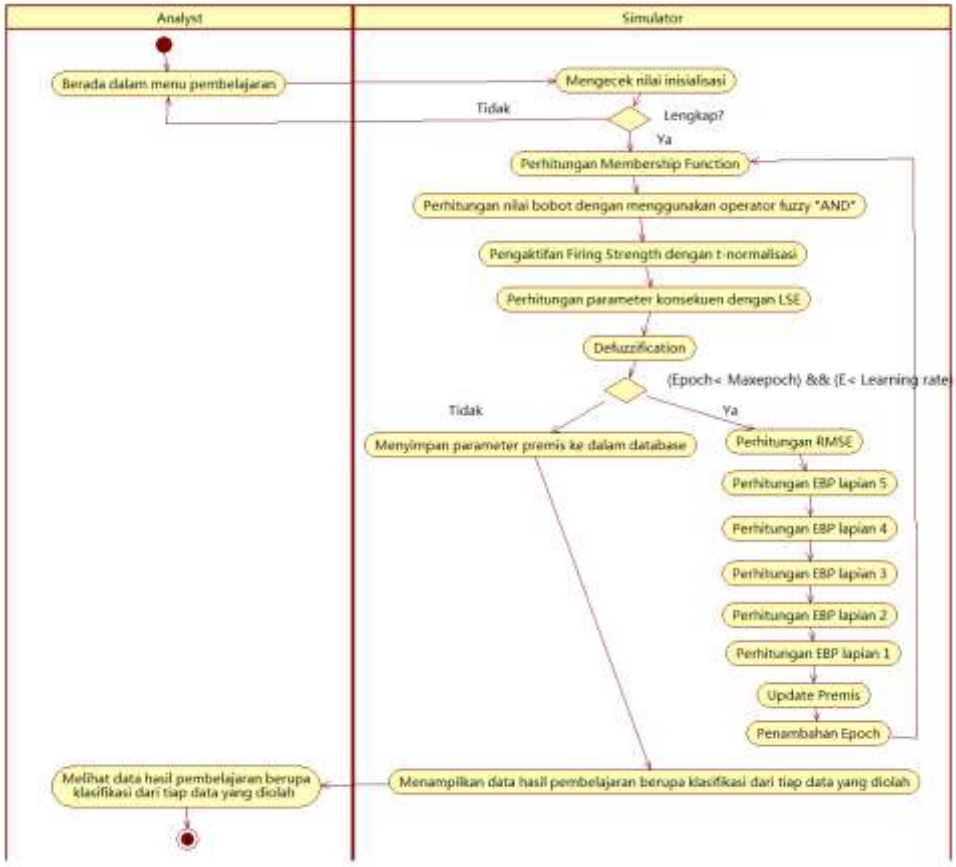
Gambar 3.12 Mengubah Data



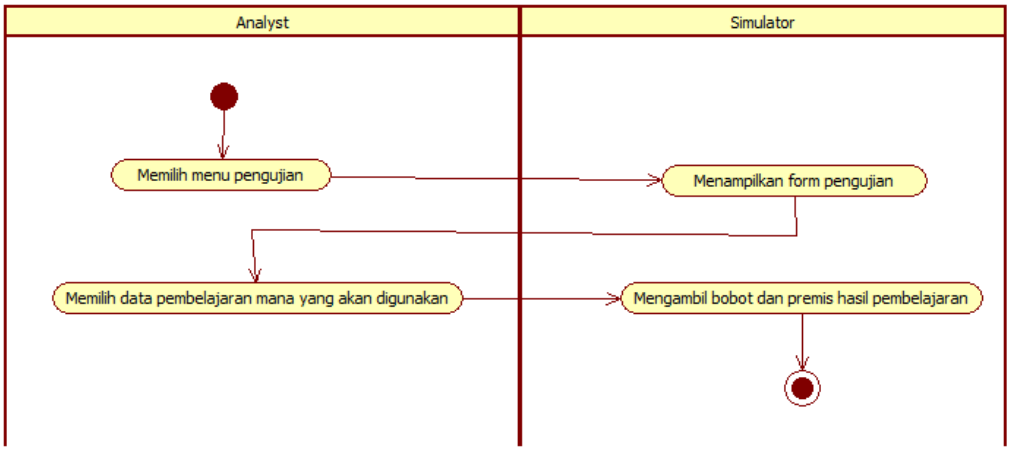
Gambar 3.13 Pembelajaran



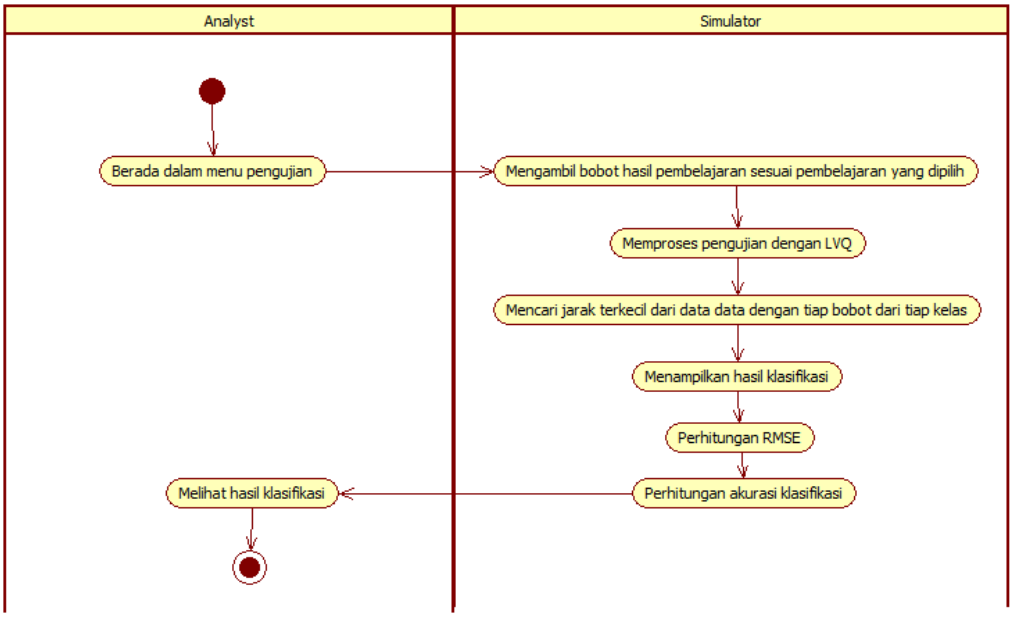
Gambar 3.14 Pembelajaran LVQ



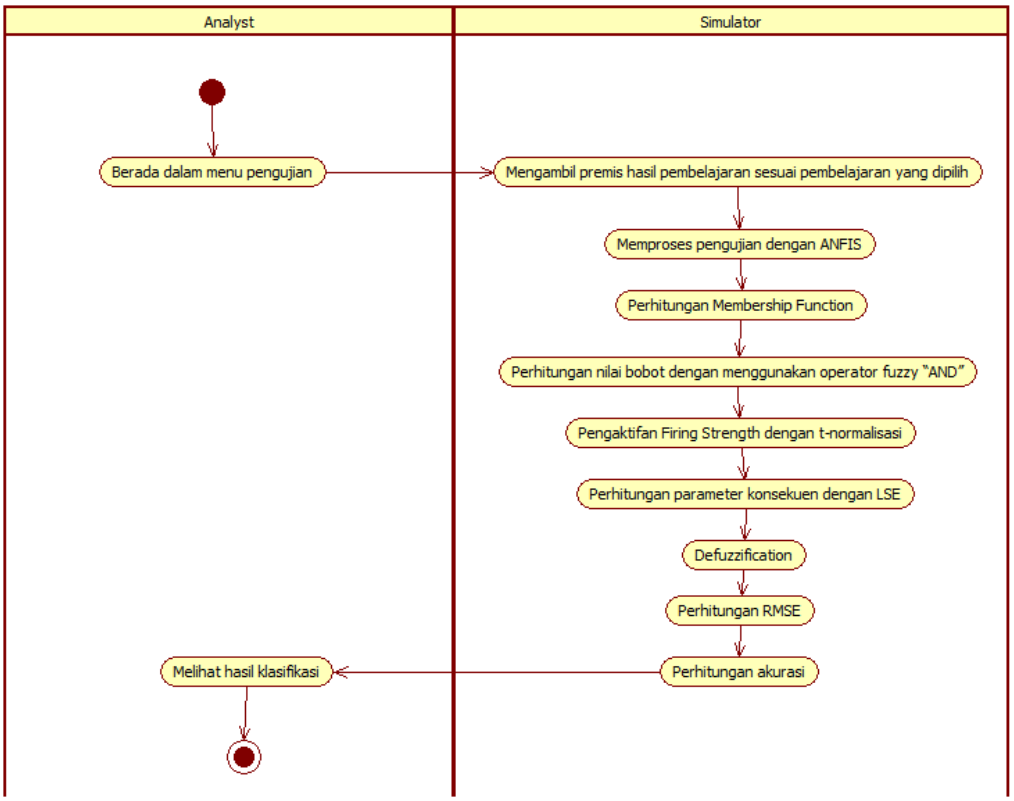
Gambar 3.15 Pembelajaran ANFIS



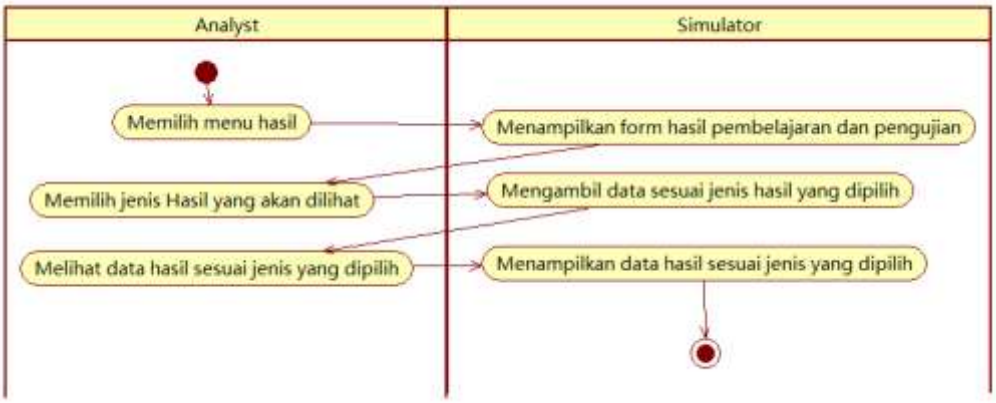
Gambar 3.16 Pengujian



Gambar 3.17 Pengujian LVQ



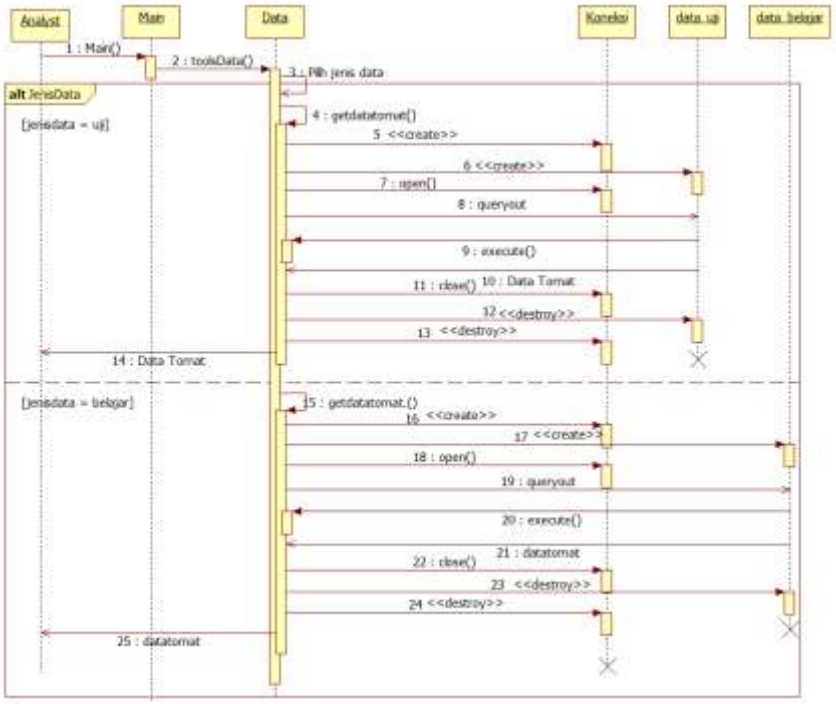
Gambar 3.18 Pengujian ANFIS



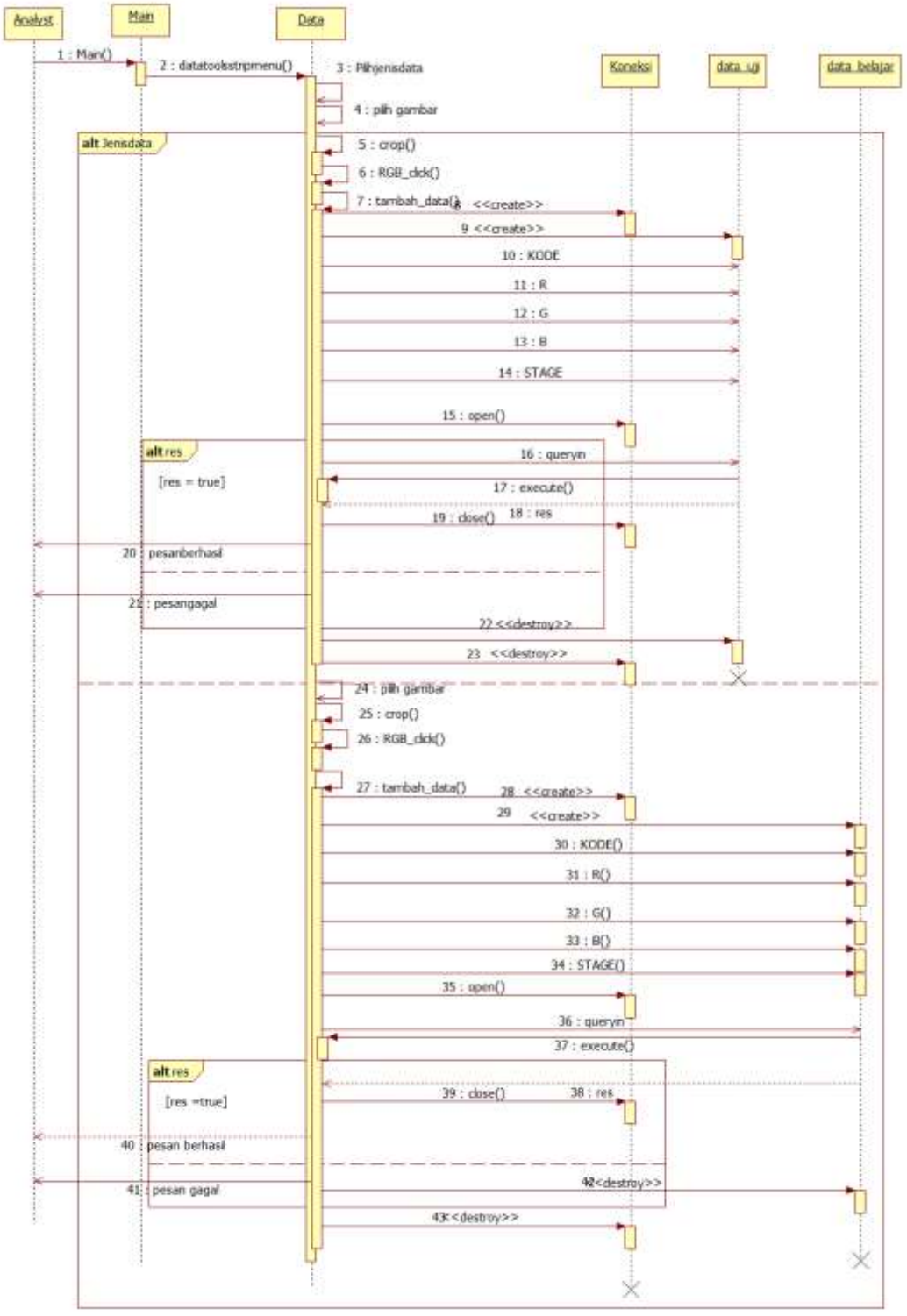
Gambar 3.19 Hasil

3.2.5. Sequence Diagram

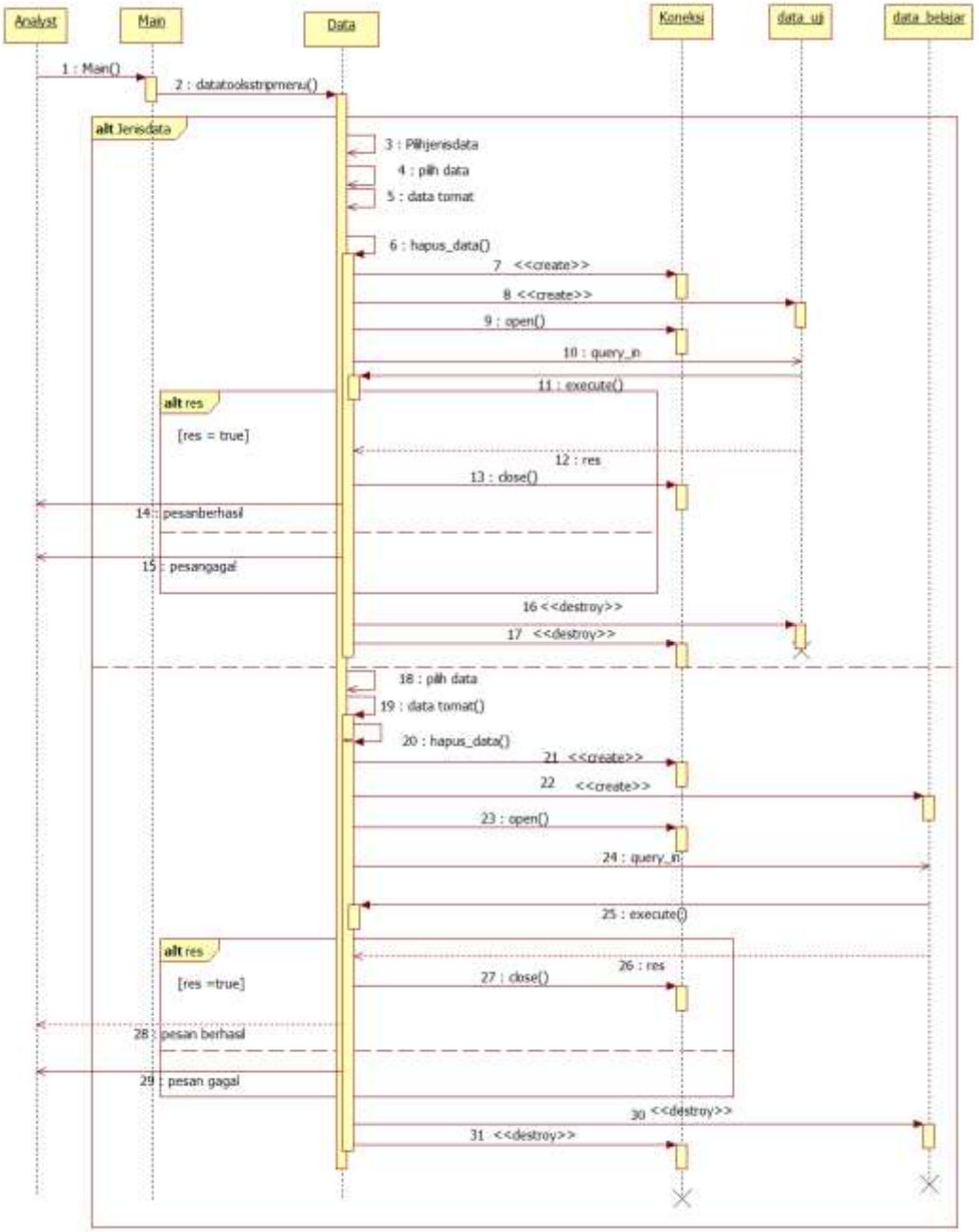
Diagram sekuen atau *sequence diagram* menggambarkan prilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Adapun diagram sekuen pada simulator yang akan dibangun adalah sebagai berikut :



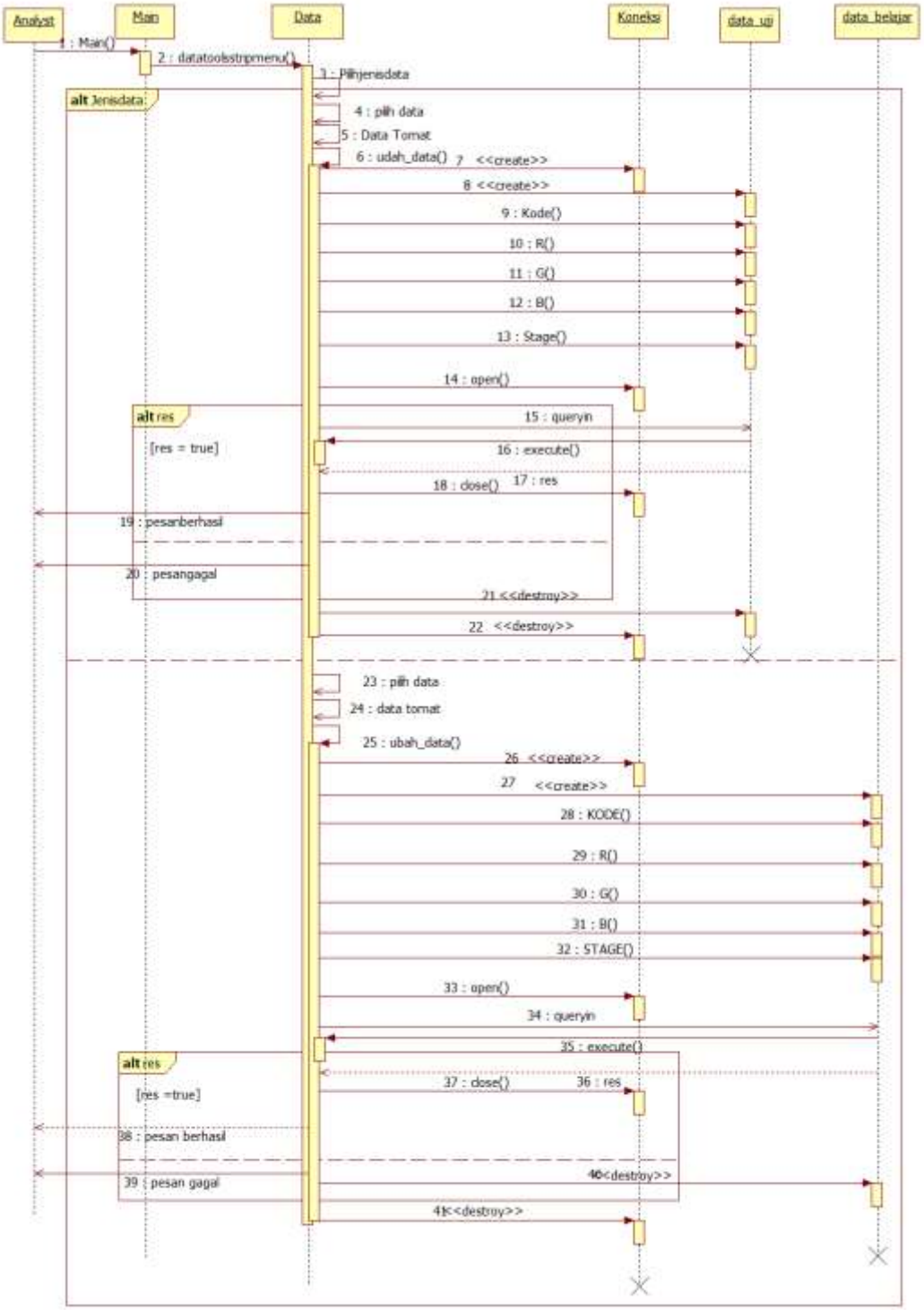
Gambar 3.20 Sequence Menyiapkan Data



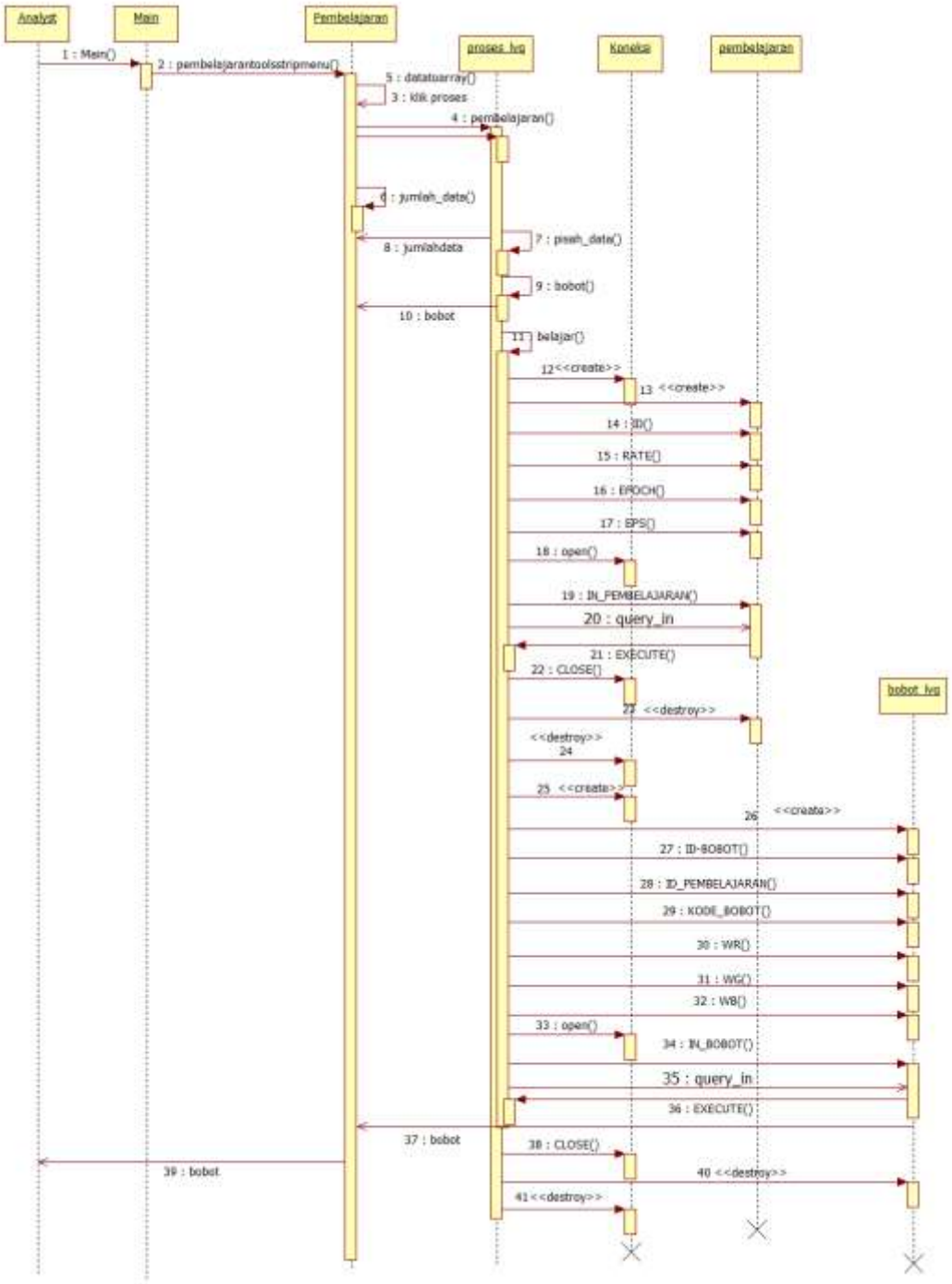
Gambar 3.21 Sequence Menambah Data



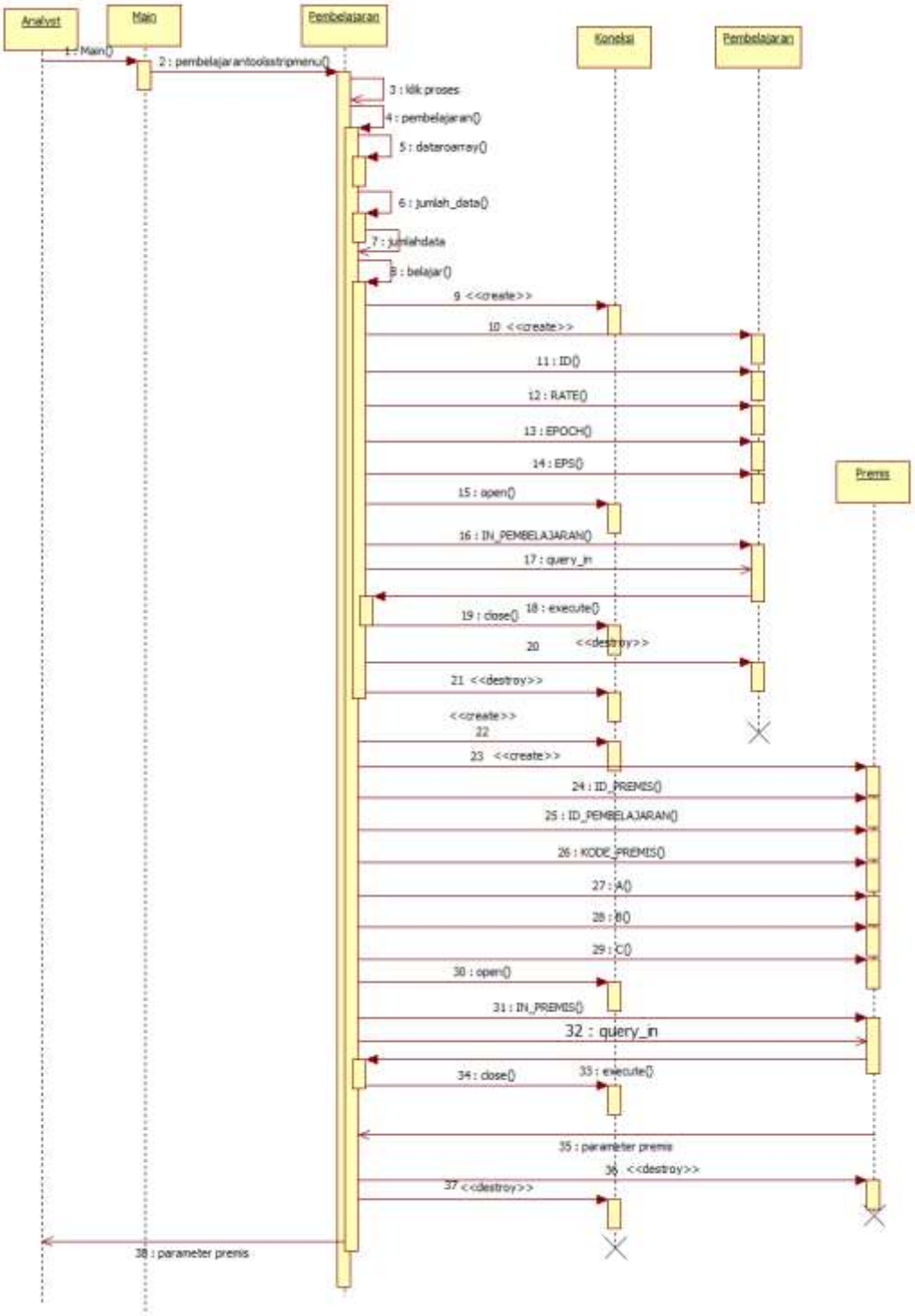
Gambar 3.22 Sequence Menghapus Data Tomat



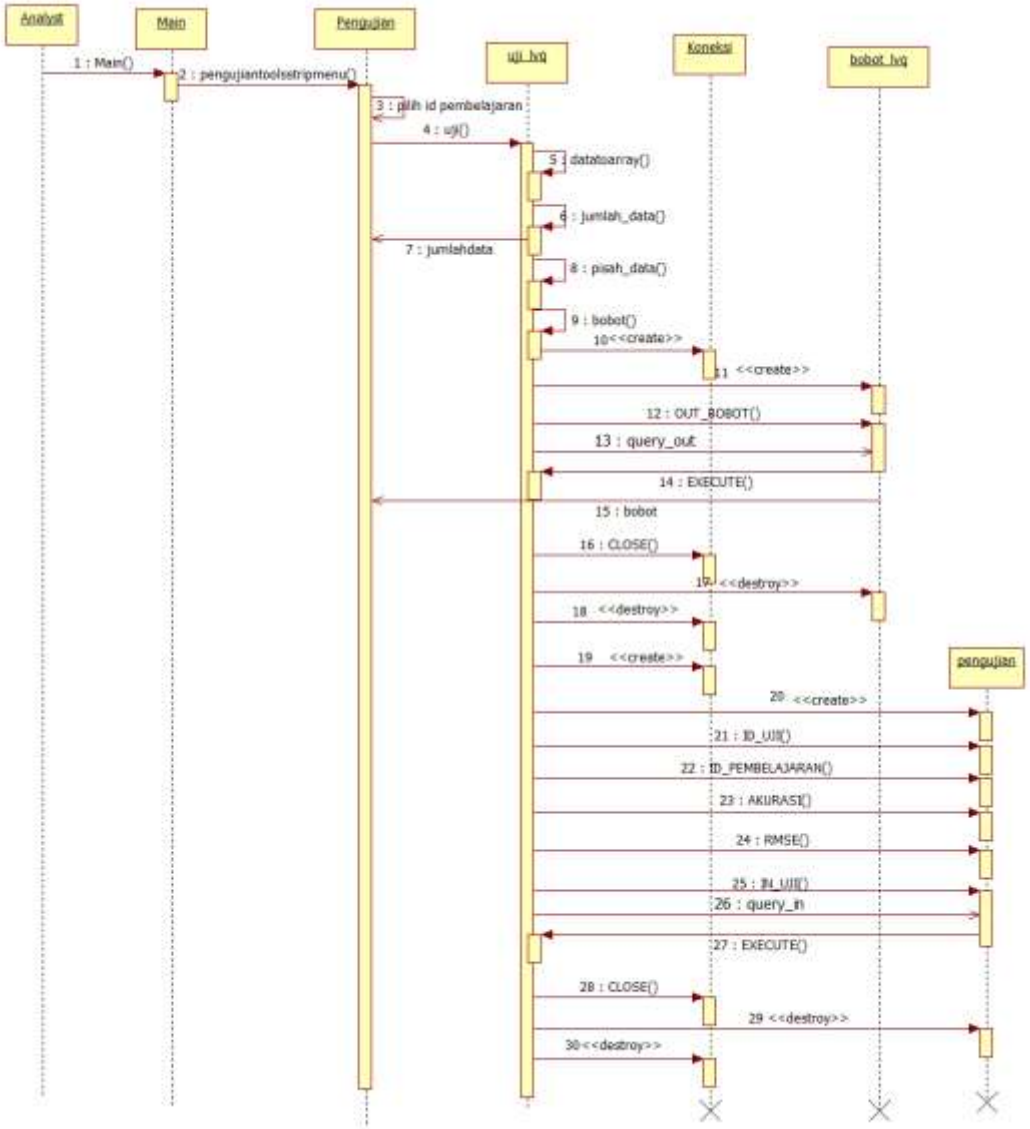
Gambar 3.23 Sequence Mengubah Data Tomat



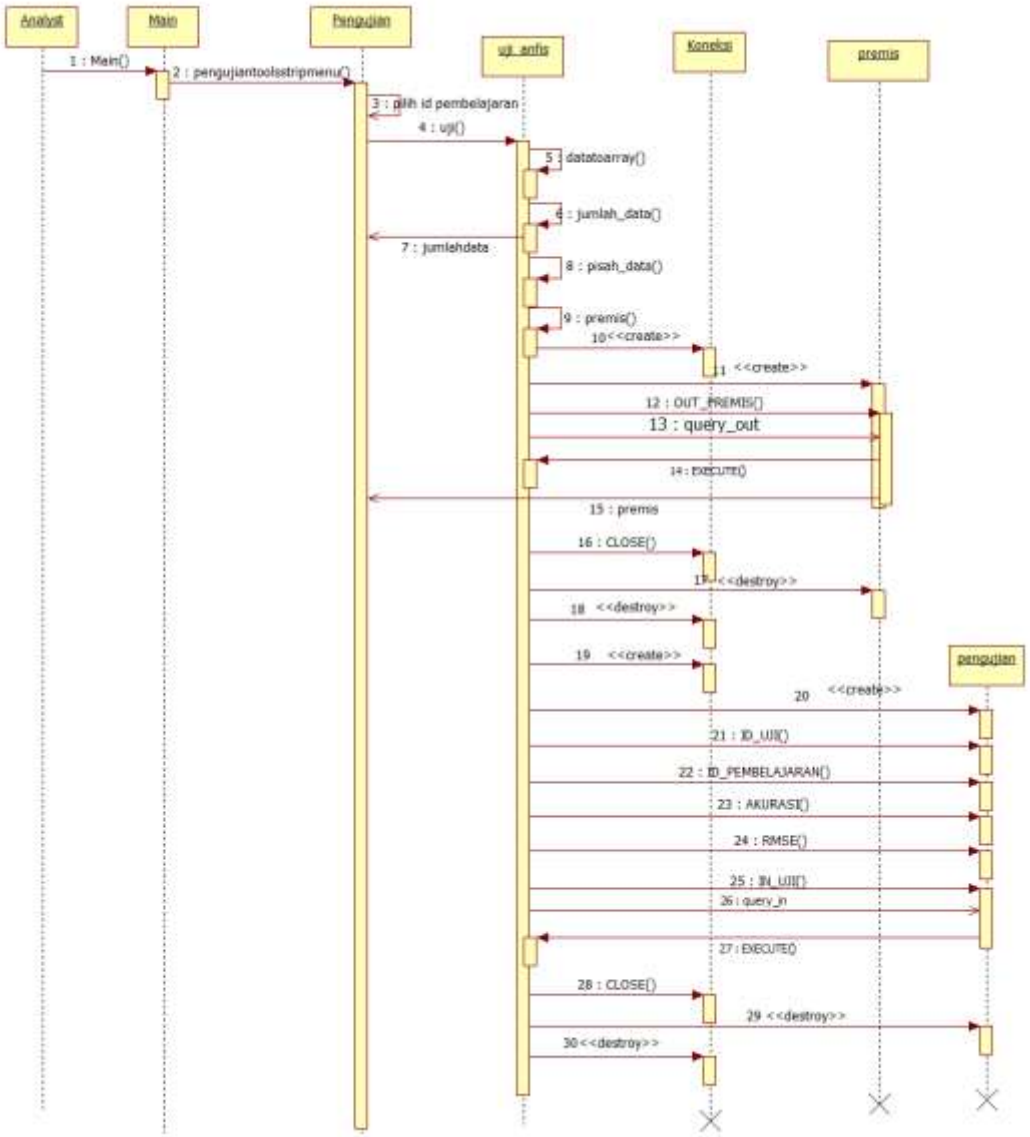
Gambar 3.24 Sequence Pembelajaran LVQ



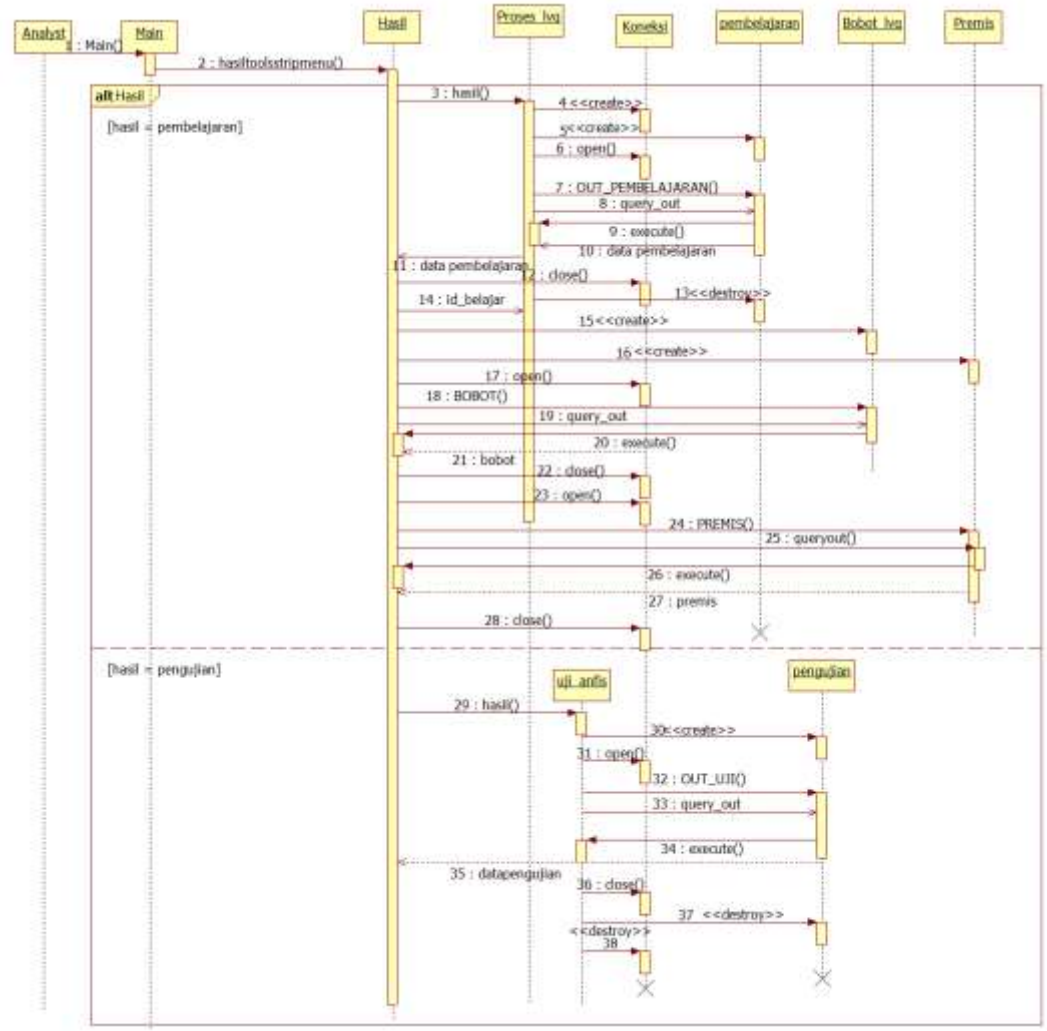
Gambar 3.25 Sequence Pembelajaran ANFIS



Gambar 3.26 Sequence Pengujian LVQ



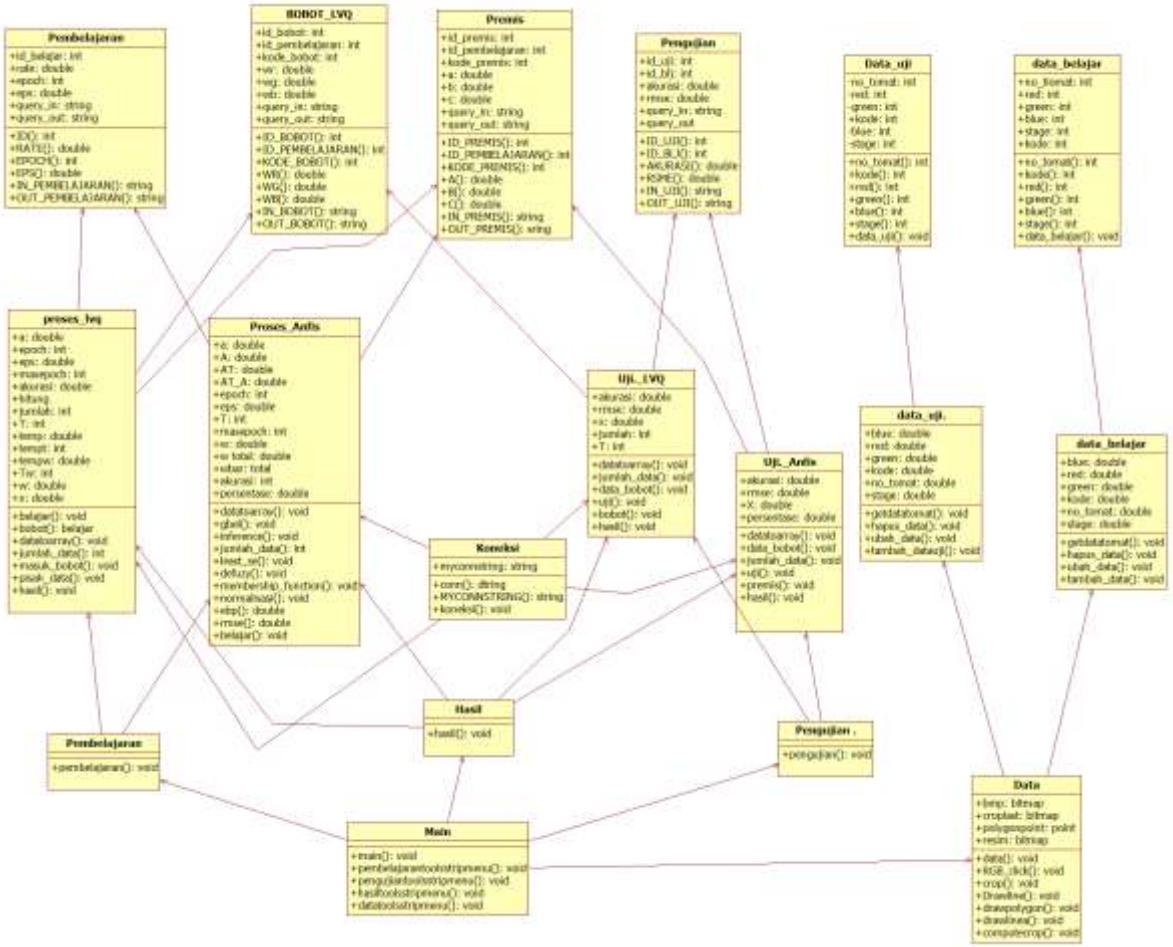
Gambar 3.27 Sequence Pengujian ANFIS



Gambar 3.28 Sequence Melihat Hasil

3.2.6. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. Adapun diagram kelas pada simulator yang akan dibangun digambarkan pada Gambar 3.29 berikut ini :



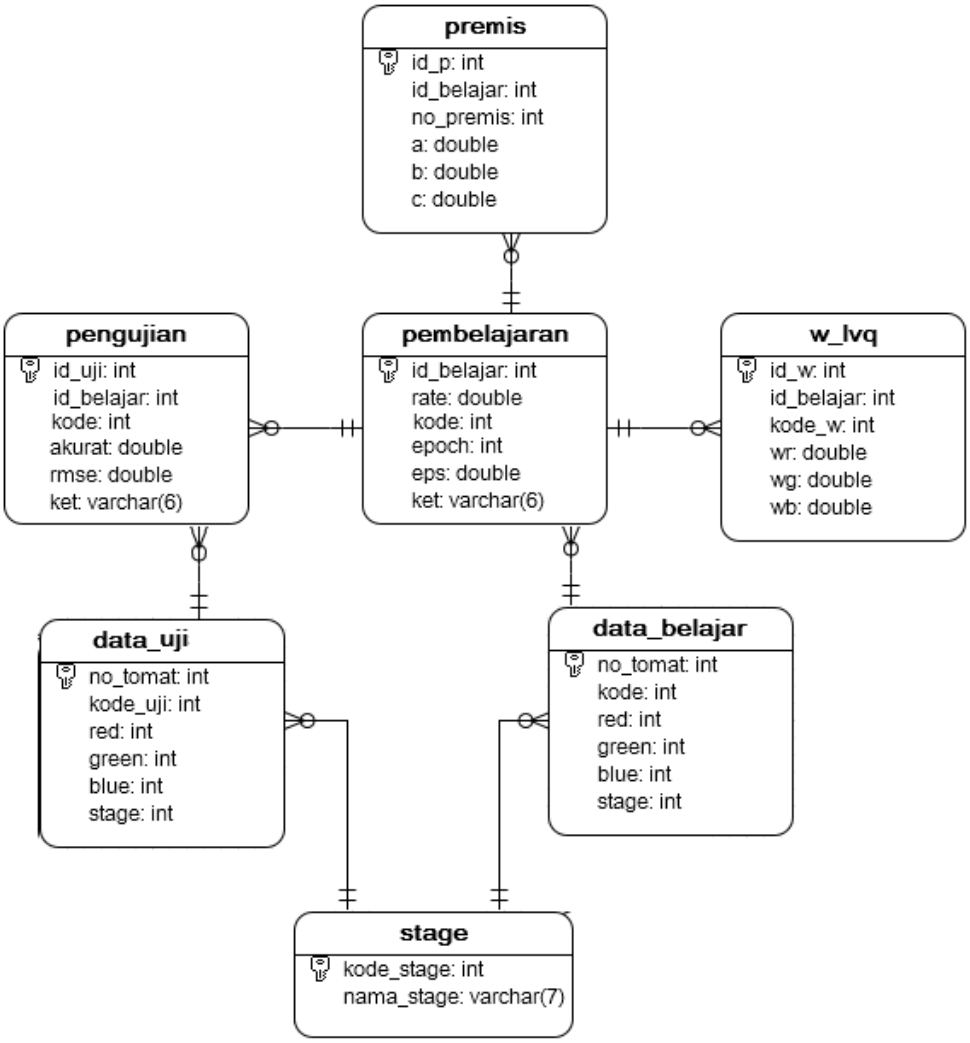
Gambar 3.29 Class Diagram

3.3. Perancangan Sistem

Perancangan sistem merupakan rancangan- rancangan mengenai sistem yang akan dibangun yang terdiri dari diagram relasi, rancangan menu, rancangan anatramuka, rancangan antarmuka pesan dan jaringan semantik.

3.3.1. Diagram Relasi

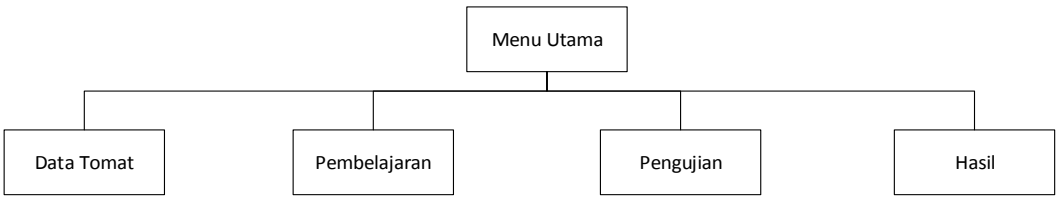
Diagram relasi merupakan rangkaian hubungan antara dua buah tabel atau lebih pada sistem basis data, yang memiliki dependensi atau ketergantungan antara satu tabel dengan tabel lain. Berikut adalah diagram relasi dari basis data pada simulator yang akan dibangun :



Gambar 3.30 Diagram Relasi

3.3.2. Rancangan Menu

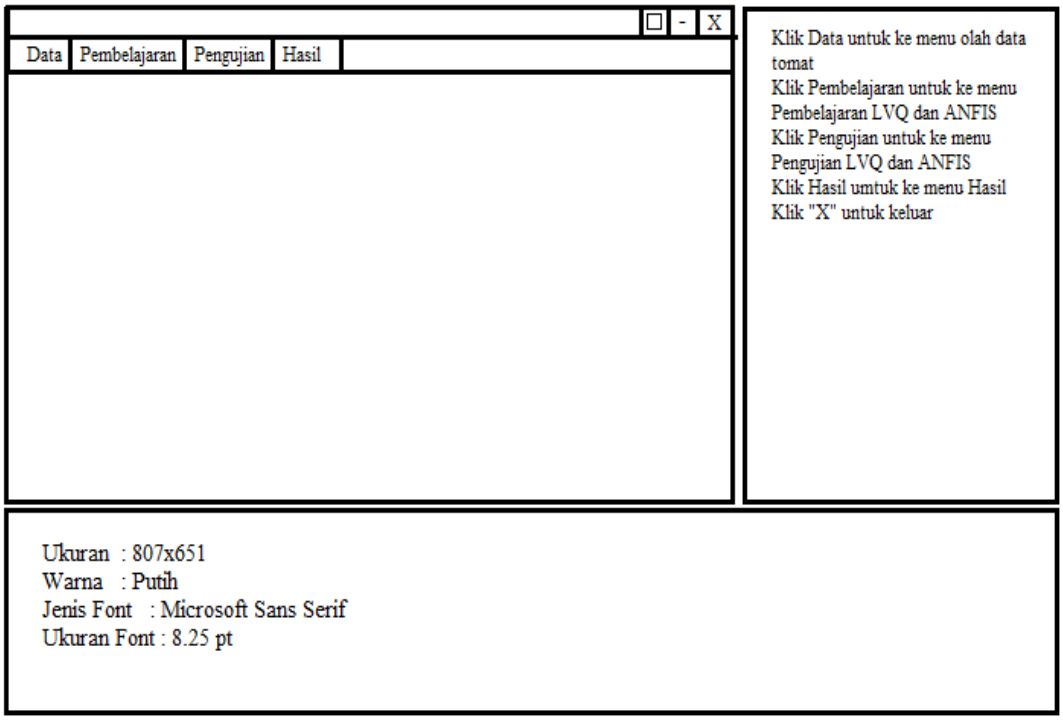
Rancangan menu dari simulator yang akan dibangun digambarkan pada Gambar 3.31 berikut ini :



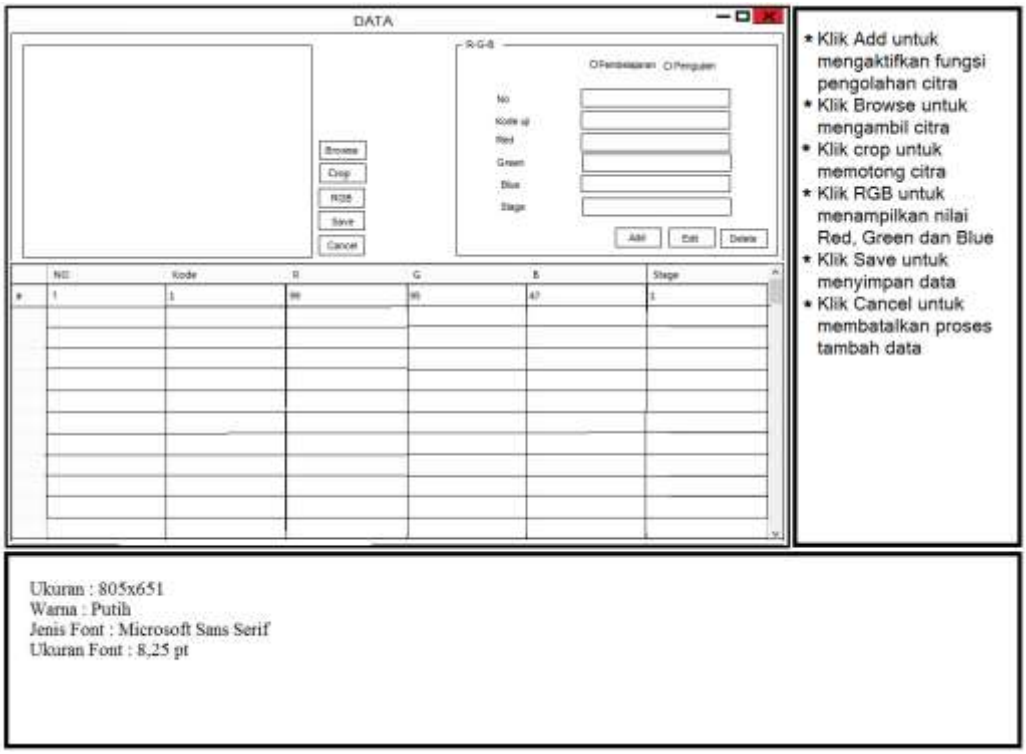
Gambar 3.31 Rancangan Menu

3.3.2.1. Rancangan Antarmuka

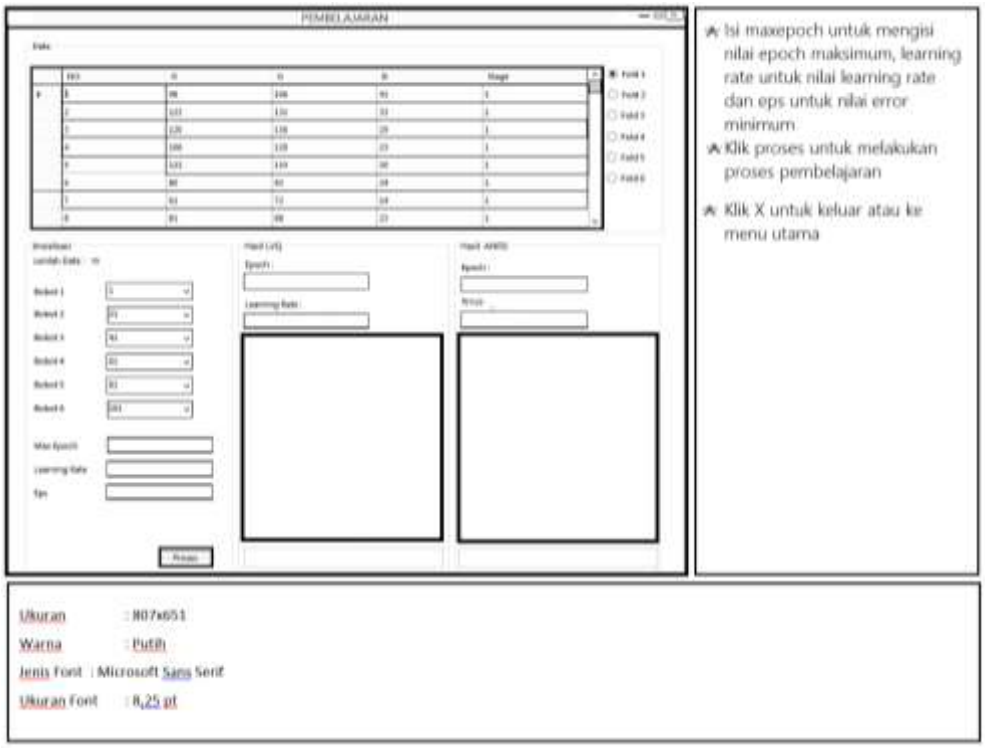
Rancangan antarmuka dari simulator yang akan dibangun adalah sebagai berikut :



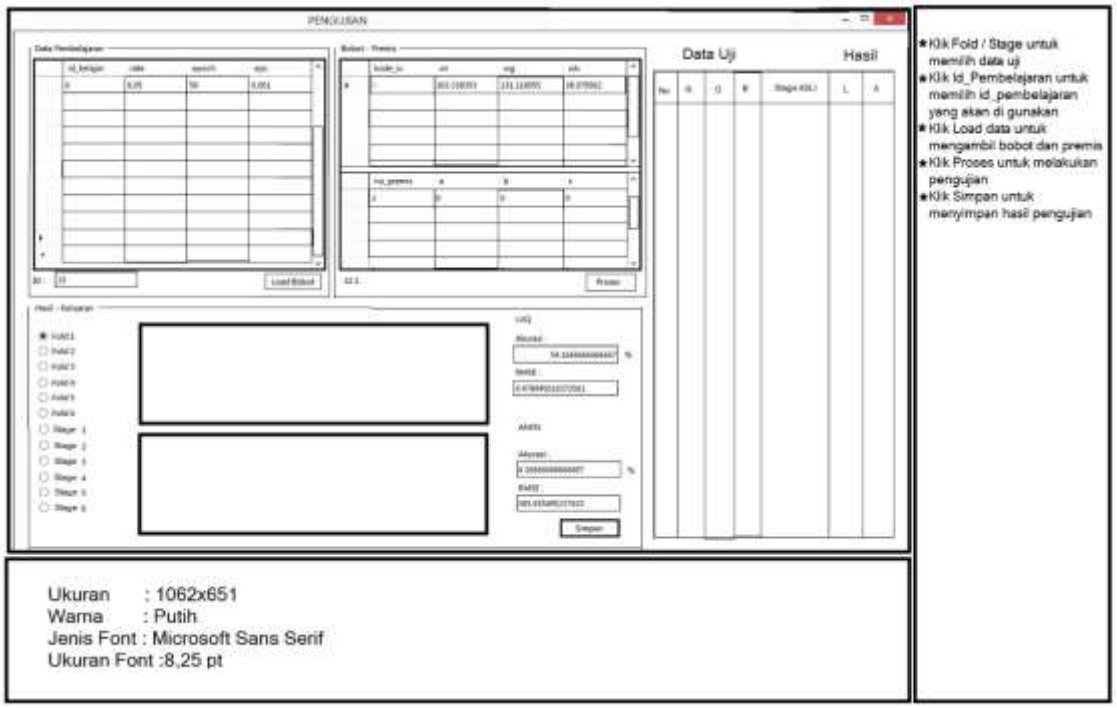
Gambar 3.32 Rancangan Halaman Main (T01)



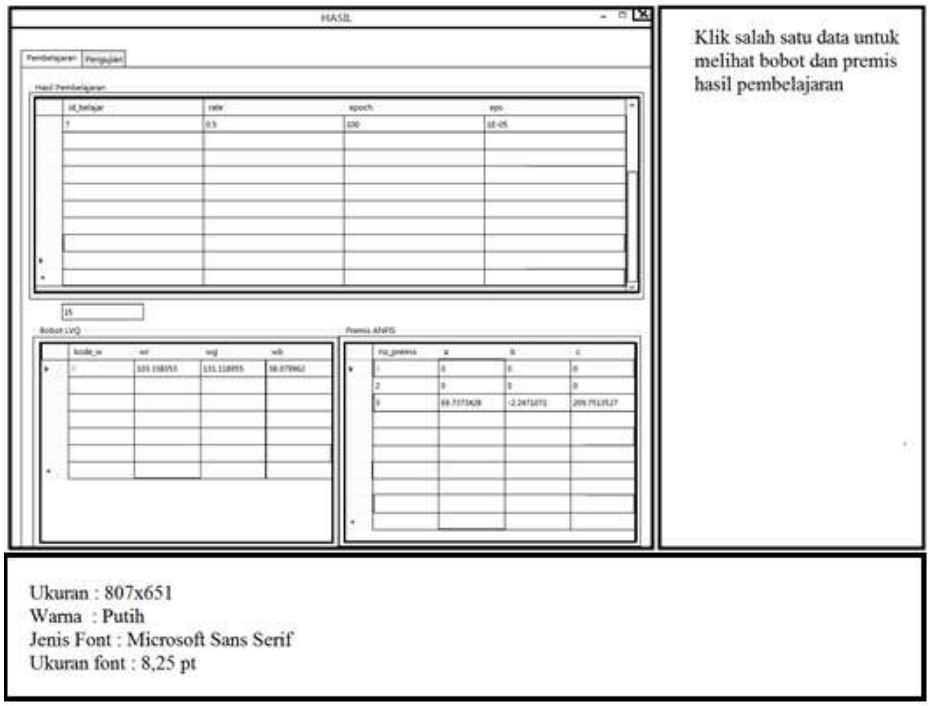
Gambar 3.33 Rancangan Halaman Olah Data (T02)



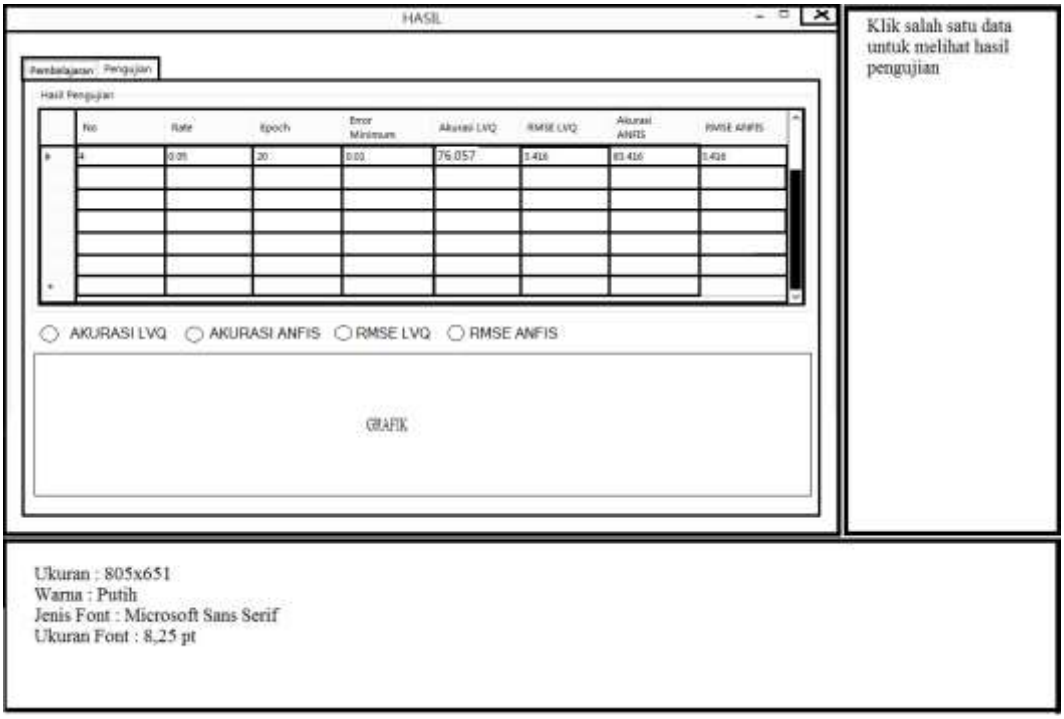
Gambar 3.34 Rancangan Halaman Pembelajaran (T03)



Gambar 3.35 Rancangan Halaman Pengujian (T04)



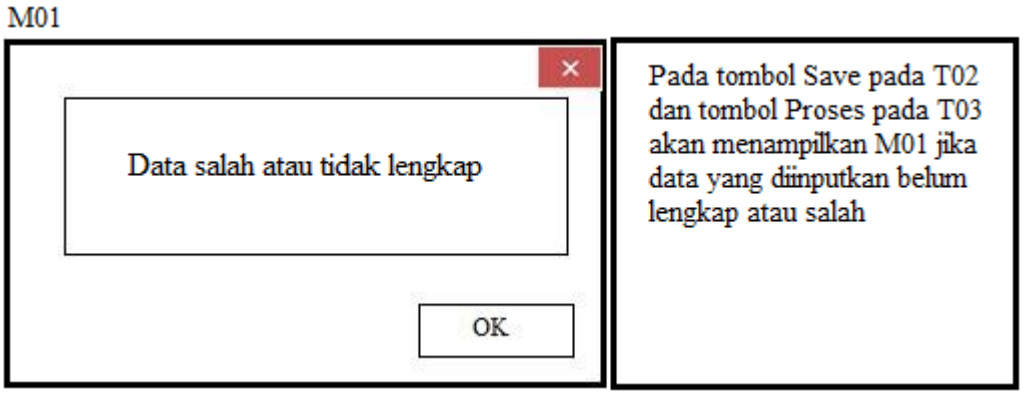
Gambar 3.36 Rancangan Halaman Hasil Bagian Pembelajaran (T05)



Gambar 3.37 Rancangan Halaman Hasil Bagian Pengujian (T06)

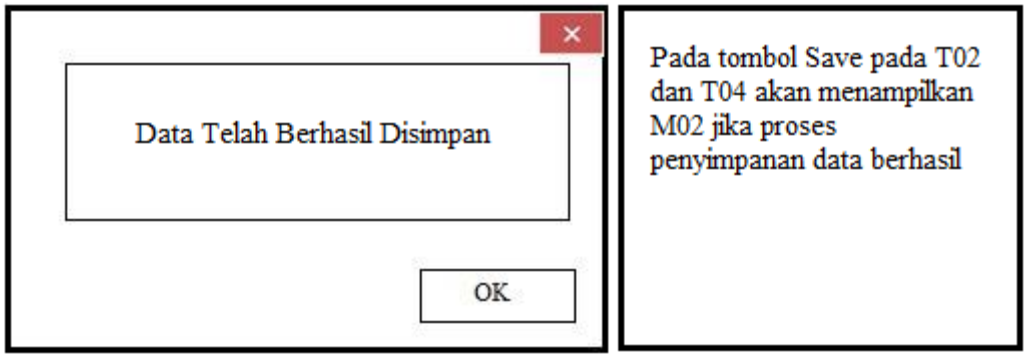
3.3.2.2. Rancangan Antamuka Pesan

Rancangan antarmuka pesan dari simulator yang akan dibangun adalah sebagai berikut :



Gambar 3.38 Pesan Data Salah Atau Belum Lengkap

M02



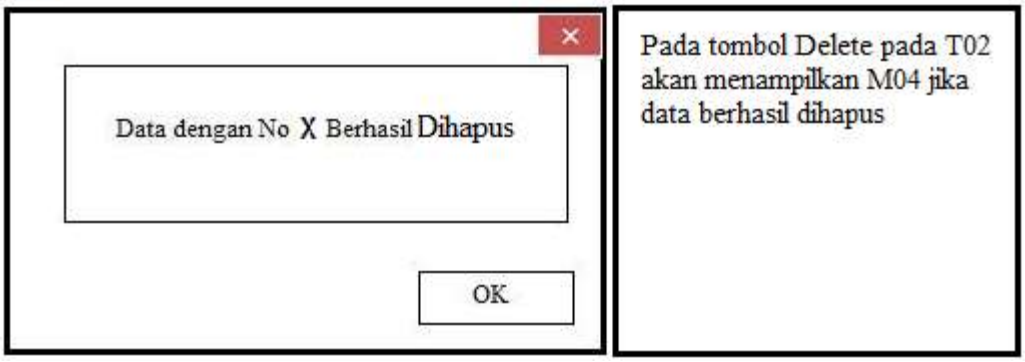
Gambar 3.39 Pesan Penyimpanan Data Berhasil

M03

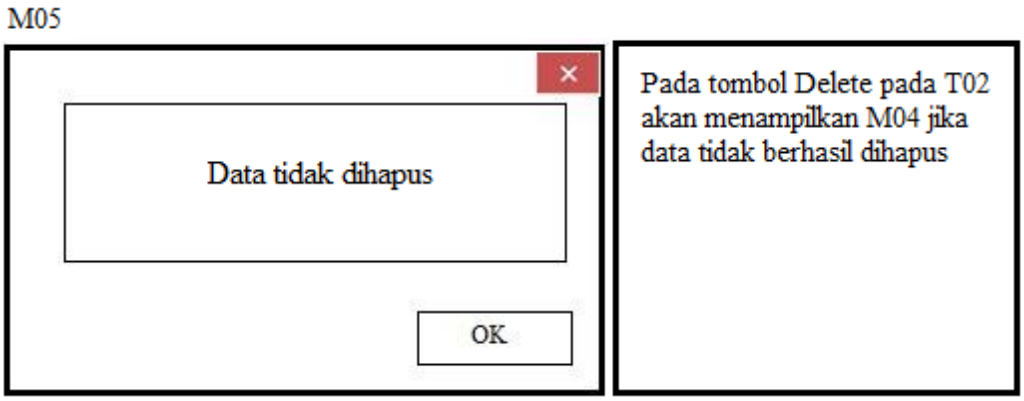


Gambar 3.40 Pesan Data Berhasil Diubah

M04



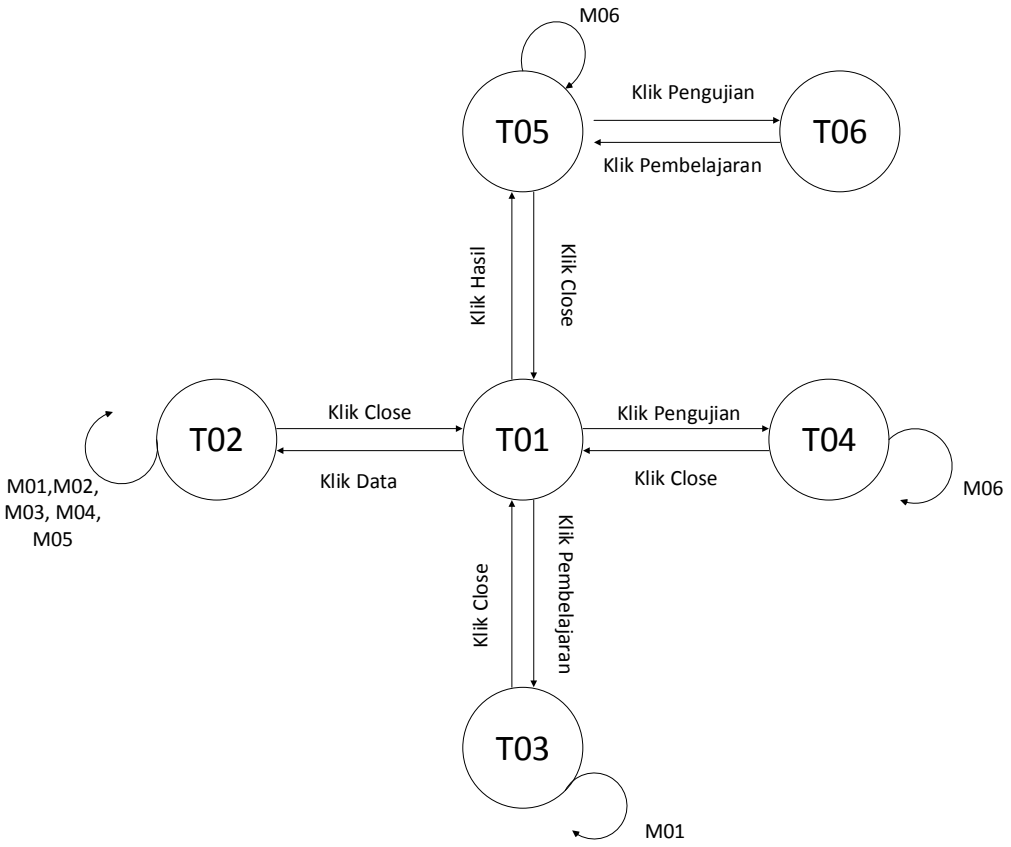
Gambar 3.41 Pesan Data Berhasil Dihapus



Gambar 3.42 Pesan Data Tidak Berhasil Dihapus

3.3.2.3. Jaringan Semantik

Jaringan semantik merupakan pengetahuan secara grafis yang menunjukkan hubungan antar berbagai objek. Jaringan semantik dari simulator yang akan dibangun pada Gambar 3.43 adalah sebagai berikut :



Gambar 3.43 Jaringan Semantik Simulator