BAB 3
ANALISIS DAN PERANCANGAN SISTEM

Analisis adalah suatu proses untuk menguraikan kebutuhan – kebutuhan yang diperlukan untuk membangun sebuah sistem. Analisis yang dilakukan pada analisis performansi metode jaringan syaraf tiruan *learning vector quantization* untuk pendeteksian anak berbakat(*gifted child*) pada masa perkembangan yaitu analisis masalah, analisis data masukan, analisis metode, analisis *confusion matrix*, analisis kompleksitas waktu asimptotik algoritma *learning vector quantization* dan analisis kebutuhan perangkat lunak.

3.1 Analisis Masalah

Berdasarkan salah satu hasil studi literatur penggunaan metode *learning vector quantization* yang pernah dilakukan sebelumnya yaitu pengenalan suatu penyakit yang memiliki gejala beririsan antara jenis yang satu dengan jenis yang lainnya. Hal ini telah dibuktikan pada jurnal "Pengenalan Jenis Penyakit THT Menggunakan Jaringan *Learning Vector Quantization*" karya Enny Itje Sela dan Sri Hartati [3]. Pada jurnal ini, membahas mengenai metode *learning vector quantization* yang digunakan untuk mendiagnosis jenis penyakit THT pada bagian hidung berdasarkan gejala penyakit yang diolah menjadi 13 buah variabel masukan. Target keluaran berupa tiga jenis penyakit yang teridentifikasi oleh sistem, jenis penyakit yang dapat diidentifikasi oleh sisitem yaitu penyakit *Rinitas Kronis*, *Epitaksis* atau *Sinusitis*. Jumlah data yang digunakan untuk proses pembelajaran sebanyak 64 data. Dari penelitian ini, didapatkan kesimpulan bahwa metode *learning vector quantization* dapat digunakan untuk mendiagnosis penyakit THT dengan keakuratan 94% yang diperoleh dari pelatihan 60 data dan pengujian 29 data, *learning rate* 0,00001, target *error* minimum 0,1 serta epoh tercapai pada nilai 88 [3].

Pada masalah anak berbakat(*gifted child*), setiap anak memiliki gejala - gejala berdasarkan perilaku dan sosial emosinya. Gejala – gejala yang terjadi pada anak berbakat(*gifted child*) dapat berjumlah 10 sampai 17 gejala untuk setiap

tipenya. Gejala antara satu tipe anak berbakat(*gifted child*) beririsan dengan tipe anak berbakat(*gifted child*) yang lainnya. Hal ini dapat menyebabkan proses identifikasi menjadi sulit karena ada beberapa gejala yang mirip antara satu tipe dengan tipe yang lainnya. Tipe anak berbakat(*gifted child*) dikelompokkan menjadi enam tipe yaitu tipe *the successfuls, the challanging, the underground, the dropouts, the double labeled* atau *the autonomous learner*.

Dengan gejala - gejala yang beririsan seperti itu, digunakan metode *learning vector quantization* untuk mengidentifikasi anak berbakat(*gifted child*). Metode *learning vector quantization* ini memiliki kemampuan untuk belajar mengelompokkan pola secara tipikal yaitu mengelompokkan pola - pola ke dalam kelas - kelas pola. Metode ini juga dapat meringkas keseluruhan data pelatihan yang besar menjadi vektor kode berukuran lebih kecil untuk proses pengelompokkannya serta dapat menghasilkan nilai *error* yang lebih kecil. Berdasarkan masalah yang telah dipaparkan, maka perlu dilakukan suatu penelitian analisis performansi untuk mengetahui efektifitas dan efisiensi dari metode jaringan syaraf tiruan *learning vector quantization* untuk mendeteksi anak berbakat(*gifted child*) pada masa perkembangan..

3.2 Analisis Data Masukan

Data yang digunakan dalam proses pembelajaran dan pengujian untuk mendeteksi anak berbakat(*gifted child*) adalah data gejala – gejala dari enam tipe anak berbakat(*gifted child*) yang dijelaskan pada tabel 2.1 sampai dengan tabel 2.6. Berdasarkan tabel 2.1 sampai tabel 2.6, gejala - gejala tersebut disusun menjadi variabel X1 hingga X75 yang akan menjadi data masukan pada simulasi. Untuk setiap variabel akan diberikan nilai 0 atau 1 dimana nilai 0 untuk variabel tidak sedangkan nilai 1 untuk variabel iya. Untuk nilai dari seluruh variabel dijelaskan pada tabel 3.1 sampai dengan tabel 3.7.

Tabel 3.1 Variabel Input Gejala - Gejala Anak Berbakat(*Gifted Child*)

Gejala\Kelas	***The Successfuls*** **(1)**	***The Challanging*** **(2)**	***The Underground*** **(3)**	***The Dropouts*** **(4)**	***The Double Labeled*** **(5)**	***The Autonomous Learner*** **(6)**	Variabel	Nilai
Berprestasi (X1)	✓						Ya	1
							Tidak	0
Baik dalam akademis(X2)	✓						Ya	1
							Tidak	0
Mencari pengakuan orang(X3)	✓						Ya	1
							Tidak	0
Bukan pengambil resiko(X4)	✓						Ya	1
							Tidak	0
Bisa menerima dan menyesuaikan(X5)	✓						Ya	1
							Tidak	0
Tidak mandiri(X6)	✓						Ya	1
							Tidak	0
Menentang guru(X7)		✓					Ya	1
							Tidak	0
Suka menanyakan aturan(X8)		✓					Ya	1
							Tidak	0
Jujur dan terus terang(X9)		✓					Ya	1
							Tidak	0
Moodnya tidak menentu(X10)		✓					Ya	1
							Tidak	0
Melakukan pekerjaan yang tidak tetap(X11)		✓		✓	✓		Ya	1
							Tidak	0
Kreatif(X12)		✓		✓		✓	Ya	1
							Tidak	0

Tabel 3.2 Variabel Input Gejala - Gejala Anak Berbakat(*Gifted Child*) (Lanjutan)

Gejala\Kelas	*The Successfuls* (1)	*The Challanging* (2)	*The Underground* (3)	*The Dropouts* (4)	*The Double Labeled* (5)	*The Autonomous Learner* (6)	Variabel	Nilai
Tetap pada keyakinannya(X13)		✓				✓	Ya	1
							Tidak	0
Bersaing(X14)		✓					Ya	1
							Tidak	0
Tidak menunjukkan prestasi yang baik(X15)		✓					Ya	1
							Tidak	0
Sering terlibat konflik(X16)		✓					Ya	1
							Tidak	0
Menyembunyikan kemampuan(X17)			✓				Ya	1
							Tidak	0
Menjauh dari orang-orang yang mendukungnya(X18)			✓				Ya	1
							Tidak	0
Menolak tantangan(X19)			✓				Ya	1
							Tidak	0
Ingin memiliki hubungan sosial(X20)			✓				Ya	1
							Tidak	0
Berganti - ganti teman(X21)			✓				Ya	1
							Tidak	0
Jarang hadir di kelas(X22)				✓			Ya	1
							Tidak	0
Tidak bisa menyelesaikan tugas(X23)				✓	✓		Ya	1
							Tidak	0
Terbujuk dengan ketertarikan luar(X24)				✓			Ya	1
							Tidak	0

Tabel 3.3 Variabel Input Gejala - Gejala Anak Berbakat(*Gifted Child*) (Lanjutan)

Gejala\Kelas	*The Successfuls* (1)	*The Challanging* (2)	*The Underground* (3)	*The Dropouts* (4)	*The Double Labeled* (5)	*The Autonomous Learner* (6)	Variabel	Nilai
Suka bermimpi di kelas(X25)				✓			Ya	1
							Tidak	0
Mengasingkan diri(X26)				✓			Ya	1
							Tidak	0
Mengkritik diri sendiri dan orang lain(X27)				✓			Ya	1
							Tidak	0
Pengganggu(X28)				✓	✓		Ya	1
							Tidak	0
Berada di tengah atau di bawah anak – anak yang lain(X29)				✓	✓		Ya	1
							Tidak	0
Tidak berprestasi(X30)				✓	✓		Ya	1
							Tidak	0
Tulisan tangannya jelek(X31)					✓		Ya	1
							Tidak	0
Memiliki kemampuan bersosialisasi yang baik(X32)						✓	Ya	1
							Tidak	0
Membuat tujuannya sendiri(X33)						✓	Ya	1
							Tidak	0
Bekerja tanpa meminta ijin(X34)						✓	Ya	1
							Tidak	0
Mengikuti area yang sangat disukainya(X35)						✓	Ya	1
							Tidak	0
Pengambil resiko(X36)						✓	Ya	1
							Tidak	0

Tabel 3.4 Variabel Input Gejala - Gejala Anak Berbakat(*Gifted Child*) (Lanjutan)

Gejala\Kelas	*The Successfuls* (1)	*The Challanging* (2)	*The Underground* (3)	*The Dropouts* (4)	*The Double Labeled* (5)	*The Autonomous Learner* (6)	Variabel	Nilai
Efektif dalam bekerja(X37)						✓	Ya	1
							Tidak	0
Mandiri(X38)						✓	Ya	1
							Tidak	0
Mempunyai jiwa kepemimpinan yang besar(X39)						✓	Ya	1
							Tidak	0
Cepat bosan(X40)	✓	✓					Ya	1
							Tidak	0
Konsep diri yang positif(X41)	✓					✓	Ya	1
							Tidak	0
Cemas(X42)	✓		✓				Ya	1
							Tidak	0
Suka merasa bersalah(X43)	✓		✓				Ya	1
							Tidak	0
Motivasi ekstrinsik(X44)	✓						Ya	1
							Tidak	0
Bertanggung jawab kepada orang lain(X45)	✓						Ya	1
							Tidak	0
Mengurangi perasaan sendiri dan membenarkan emosi mereka(X46)	✓		✓				Ya	1
							Tidak	0
Mempunyai sifat mencela diri sendiri(X47)	✓						Ya	1
							Tidak	0
Frustasi(X48)		✓		✓	✓		Ya	1
							Tidak	0

Tabel 3.5 Variabel Input Gejala - Gejala Anak Berbakat(*Gifted Child*) (Lanjutan)

Gejala\Kelas	*The Successfuls* (1)	*The Challanging* (2)	*The Underground* (3)	*The Dropouts* (4)	*The Double Labeled* (5)	*The Autonomous Learner* (6)	Variabel	Nilai
Kesadaran harga diri yang rendah(X49)		✓			✓		Ya	1
							Tidak	0
Tidak sabaran(X50)		✓					Ya	1
							Tidak	0
Membela diri sendiri(X51)		✓		✓			Ya	1
							Tidak	0
Sangat sensitif(X52)		✓					Ya	1
							Tidak	0
Tidak yakin perannya dalam bergaul(X53)		✓	✓				Ya	1
							Tidak	0
Tertekan(X54)			✓	✓			Ya	1
							Tidak	0
Bingung(X55)			✓				Ya	1
							Tidak	0
Merasa tidak aman(X56)			✓				Ya	1
							Tidak	0
Dendam(X57)				✓			Ya	1
							Tidak	0
Marah(X58)				✓	✓		Ya	1
							Tidak	0
Suka marah tiba-tiba(X59)				✓			Ya	1
							Tidak	0
Konsep diri yang negatif(X60)				✓			Ya	1
							Tidak	0

Tabel 3.6 Variabel Input Gejala - Gejala Anak Berbakat(*Gifted Child*) (Lanjutan)

Gejala\Kelas	*The Successfuls* (1)	*The Challanging* (2)	*The Underground* (3)	*The Dropouts* (4)	*The Double Labeled* (5)	*The Autonomous Learner* (6)	Variabel	Nilai
Tidak berdaya(X61)					✓		Ya	1
							Tidak	0
Tidak menyadari kemampuannya(X62)					✓		Ya	1
							Tidak	0
Merasa tidak dihargai(X63)					✓		Ya	1
							Tidak	0
Merasa tidak dibantu(X64)					✓		Ya	1
							Tidak	0
Merasa diasingkan(X65)					✓		Ya	1
							Tidak	0
Percaya diri(X66)						✓	Ya	1
							Tidak	0
Menerima diri sendiri(X67)						✓	Ya	1
							Tidak	0
Antusias(X68)						✓	Ya	1
							Tidak	0
Diterima oleh orang lain(X69)						✓	Ya	1
							Tidak	0
Menerima orang lain(X70)						✓	Ya	1
							Tidak	0
Keingintahuan dan belajar yang tinggi(X71)						✓	Ya	1
							Tidak	0
Didukung dengan positif(X72)						✓	Ya	1
							Tidak	0

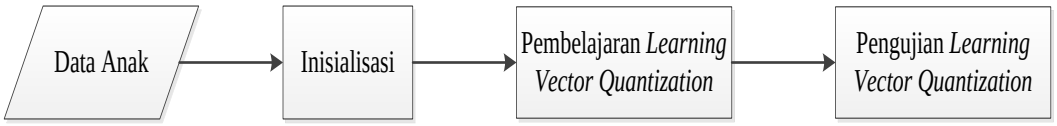
Tabel 3.7 Variabel Input Gejala - Gejala Anak Berbakat(*Gifted Child*) (Lanjutan)

Gejala\Kelas	*The Successfuls* (1)	*The Challanging* (2)	*The Underground* (3)	*The Dropouts* (4)	*The Double Labeled* (5)	*The Autonomous Learner* (6)	Variabel	Nilai
Bisa menerima kegagalan(X73)						✓	Ya	1
							Tidak	0
Kekuatan pribadi(X74)						✓	Ya	1
							Tidak	0
Motivasi yang instrinsik(X75)						✓	Ya	1
							Tidak	0

Data masukan memiliki batasan minimum gejala yang harus dipilih pada pilihan yang telah disediakan simulasi. Data masukan yang harus dipilih seluruhnya yaitu sebanyak 75 buah gejala. Sedangkan minimum gejala yang terdeteksi oleh simulasi yaitu sebanyak 20 buah gejala.

3.3 Analisis Metode

Block diagram menggambarkan setiap bagian dari alur kerja. *Block diagram* jaringan syaraf tiruan *learning vector quantization* untuk pendeteksian anak berbakat(*gifted child*) pada masa perkembangan akan dijelaskan pada gambar 3.1.



Gambar 3.1 *Block Diagram* Jaringan Syaraf Tiruan *Learning Vector Quantization* Untuk Pendeteksian Anak Berbakat(*gifted child*) pada Masa Perkembangan

Parameter yang digunakan untuk tahapan pembelajaran yaitu maksimal epoh (iterasi), *learning rate*(α) dan *error* minimum. Sedangkan parameter yang digunakan untuk tahapan pengujian yaitu nilai bobot akhir yang dihasilkan dari tahapan pembelajaran.

Berikut adalah penjelasan dari setiap tahapan pada *block diagram* gambar 3.1, yang terdiri dari tahapan inisialisasi, pembelajaran *learning vector quantization*, dan pengujian *learning vector quantization*.

1. Inisialisasi

Inisialisasi merupakan tahapan untuk menentukan nilai bobot awal, maksimum epoh atau maksimum iterasi(MaxEpoh), *learning rate*(*α*), *error* minimum(eps), jumlah data pembelajaran, jumlah kelas dan nilai awal epoh yang dibutuhkan untuk melakukan proses pembelajaran *learning vector quantization*. Berikut adalah inisialisasi nilai – nilai yang dijadikan contoh kasus yaitu :

Tabel 3.8 Contoh Bobot Awal w_1 yang Dijadikan Contoh Kasus

w	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	31	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**
	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
	46	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**
	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	61	**62**	**63**	**64**	**65**	**66**	**67**	**68**	**69**	**70**	**71**	**72**	**73**	**74**	**75**
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

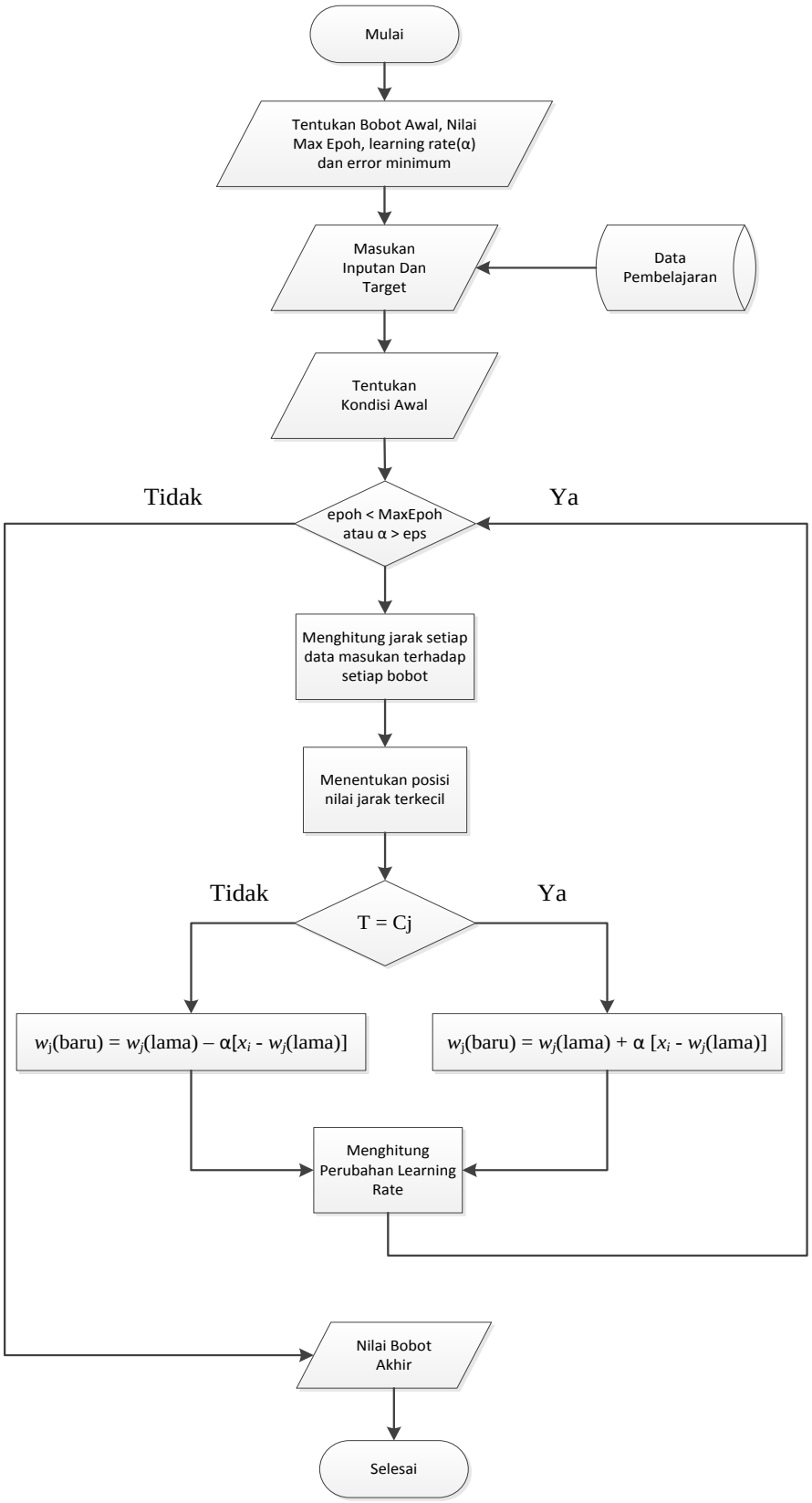
Tabel 3.9 Contoh Bobot Awal w_2 yang Dijadikan Contoh Kasus

w	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
2	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	31	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**
	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	46	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**
	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0
	61	**62**	**63**	**64**	**65**	**66**	**67**	**68**	**69**	**70**	**71**	**72**	**73**	**74**	**75**
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- MaxEpoh = 2;
- *Learning rate*(α) = 0,1;
- Eps = 0,09;
- Jumlah data pembelajaran = 2;
- Jumlah kelas = 2;
- Epoh = 0.

2. Pembelajaran *Learning Vector Quantization*

Setelah melakukan inisialisasi, tahapan selanjutnya yaitu melakukan pembelajaran *learning vector quantization* yang mana dari pembelajaran ini akan menghasilkan nilai bobot akhir atau nilai bobot baru serta nilai *learning rate* (α) yang baru. *Flowchart* pada proses pembelajaran *learning vector quantization* akan dijelaskan pada gambar 3.2.



Gambar 3.2 *Flowchart* Pembelajaran *Learning Vector Quantization*

Adapun penjelasan langkah – langkah tahapan pembelajaran *learning vector quantization* yaitu sebagai berikut :

1. **Menentukan bobot awal, nilai MaxEpoh, *learning rate*(α) dan *error* minimum;**
2. **Masukan Inputan dan Target;**
3. **Tentukan kondisi awal;**
4. **Cek kondisi epoh dan *learning rate*(α)**

   Jika epoh < MaxEpoh, maka lakukan perhitungan jarak setiap data masukan(x) terhadap setiap bobot(w_i), mencari jarak minimum, perhitungan perubahan nilai bobot dan perhitungan perubahan nilai *learning rate* (α). Jika kondisi epoh < MaxEpoh tidak terpenuhi, maka cek kondisi α > eps.

   Jika α > eps, maka lakukan perhitungan jarak setiap data masukan(x) terhadap setiap bobot(w_i), mencari jarak minimum, perhitungan perubahan nilai bobot dan perhitungan perubahan nilai *learning rate* (α). Jika kondisi α > eps tidak terpenuhi, maka perulangan berhenti.

   Contoh kasus :

   Diketahui nilai – nilai inisialisasi awal yang ada di tahapan inisialisasi, yaitu MaxEpoh = 2, α = 0,1, eps = 0,09 dan epoh = 0.

   Cek kondisi : epoh < maxEpoh = 0 < 2 maka

   epoh = epoh + 1 => 0 + 1 = 1 dan lakukan perhitungan jarak setiap data masukan(x) terhadap setiap bobot(w_i), mencari jarak minimum, perhitungan perubahan nilai bobot serta perhitungan perubahan nilai *learning rate* (α) sampai kedua kondisi (epoh < MaxEpoh) atau (α > eps) tidak terpenuhi dan perulangan berhenti.

5. Menghitung jarak setiap data masukan(x) terhadap setiap data bobot

Hitung setiap data masukan dari i=0 sampai n terhadap setiap bobot(w_i). Rumus yang digunakan untuk menghitung jarak setiap data masukan(x) terhadap setiap bobot(w_i) yaitu :

$$\text{Jarak} = \|x_{ij} - w_{ij}\| \quad \ldots\ldots(3.1)$$

Contoh kasus :

- Untuk data pembelajaran ke-1

Tabel 3.10 Contoh Data Pembelajaran Ke -1 yang Dijadikan Contoh Kasus

Data ke-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Target Kelas
	1	1	1	1	0	1	0	0	0	1	0	0	0	1	0	
	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	31	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	
1	0	1	0	1	0	0	0	1	0	1	1	1	0	1	1	1
	46	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	
	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	
	61	**62**	**63**	**64**	**65**	**66**	**67**	**68**	**69**	**70**	**71**	**72**	**73**	**74**	**75**	
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	

- Jarak terhadap :

a. Bobot ke-1 (w_1)

$$\text{Jarak} = \|x_{1j} - w_{1j}\|$$

$$\sqrt{\begin{array}{l}(1-1)^2+(1-1)^2+(1-1)^2+(1-1)^2+(0-1)^2+(1-1)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+\\(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+\\(0-0)^2+(1-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(1-1)^2+(1-1)^2+(1-1)^2+(0-1)^2+(1-1)^2+(1-1)^2+\\(0-1)^2+(1-1)^2+(0-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+\\(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+\end{array}}$$

$$\text{Jarak} = \sqrt{1+1+1+1+1+1+1+1+1+1+1+1}$$

$$= \sqrt{12} = 3{,}464$$

b. Bobot ke-2 (w_2)

Jarak $= \|x_{1j} - w_{2j}\|$

$$\sqrt{\begin{matrix}(1-0)^2+(1-0)^2+(1-0)^2+(1-0)^2+(0-0)^2+(1-0)^2+(0-1)^2+(0-1)^2+(0-1)^2+(1-1)^2+(0-1)^2+(0-1)^2+(0-1)^2+(1-1)^2+(0-1)^2+\\(0-1)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+\\(0-0)^2+(1-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(1-1)^2+(1-0)^2+(1-0)^2+(0-0)^2+(1-0)^2+(1-0)^2+\\(0-0)^2+(1-0)^2+(0-1)^2+(0-1)^2+(1-1)^2+(0-1)^2+(1-1)^2+(0-1)^2+(0-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+\\(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+\end{matrix}}$$

$$\text{Jarak} = \sqrt{\begin{matrix}1+1+1+1+1+1+1+1+1+1+\\1+1+1+1+1+1+1+1+1+1+\\1+1+1+1+1+1+1+1+1+1\end{matrix}}$$

$= \sqrt{27} = 5{,}196$

➢ Untuk data pembelajaran ke-2

Tabel 3.11 Contoh Data Pembelajaran Ke -2 yang Dijadikan Contoh Kasus

Data ke-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Target Kelas
2	0	0	0	0	0	0	1	1	0	1	1	1	0	1	1	2
	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	
	1	0	0	1	1	0	0	1	0	0	0	0	1	0	0	
	31	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	
	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
	46	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	
	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	
	61	**62**	**63**	**64**	**65**	**66**	**67**	**68**	**69**	**70**	**71**	**72**	**73**	**74**	**75**	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

- Jarak terhadap :

a. Bobot ke-1 (w_1)

Jarak $= \|x_{2j} - w_{1j}\|$

$$\sqrt{\begin{array}{c}(0-1)^2+(0-1)^2+(0-1)^2+(0-1)^2+(0-0{,}9)^2+(0-1)^2+(1-0)^2+(1-0)^2+(0-0)^2+(1-0{,}1)^2+(1-0)^2+(1-0)^2+(0-0)^2+(1-0{,}1)^2+(1-0)^2+\\(1-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+\\(0-0)^2+(0-0{,}1)^2+(0-0)^2+(0-0{,}1)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0{,}1)^2+(0-0)^2+(1-1)^2+(0-1)^2+(0-1)^2+(0-0{,}9)^2+(0-1)^2+(0-1)^2+\\(0-0{,}9)^2+(0-1)^2+(1-0)^2+(1-0)^2+(1-0{,}1)^2+(0-0)^2+(1-0{,}1)^2+(1-0)^2+(0-0)^2+(0-0{,}1)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+\\(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-0{,}1)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+\end{array}}$$

$$\text{Jarak} = \sqrt{\begin{array}{c}1+1+1+1+0{,}81+1+1+\\1+0{,}81+1+1+0{,}81+1+1+\\1+1+1+1+0{,}01+0{,}01+0{,}01+\\1+1+0{,}81+1+1+0{,}81+1+\\1+1+0{,}81+0{,}81+1+0{,}01+0{,}81\end{array}}$$

$$= \sqrt{29{,}52} = 5{,}433$$

b. Bobot ke-2 (w_2)

$$\text{Jarak} = \|x_{2j} - w_{2j}\|$$

$$\sqrt{\begin{array}{c}(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-1)^2+(1-1)^2+(0-1)^2+(1-1)^2+(1-1)^2+(1-1)^2+(0-1)^2+(1-1)^2+(1-1)^2+\\(1-1)^2+(0-0)^2+(0-0)^2+(1-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+\\(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-1)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+\\(0-0)^2+(0-0)^2+(1-1)^2+(1-1)^2+(1-1)^2+(0-1)^2+(1-1)^2+(1-1)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+\\(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+\end{array}}$$

$$\text{Jarak} = \sqrt{1+1+1+1+1+1+1}$$

$$\text{Jarak} = \sqrt{7} = 2{,}645$$

6. Menentukan posisi nilai jarak terkecil(C_j)

Posisi nilai jarak terkecil didapat dari hasil perhitungan jarak setiap data masukan(x) terhadap setiap bobot(w_i) yang paling terkecil. Setelah itu, bandingkan posisi nilai jarak terkecil tersebut(C_j) dengan target kelas (T) yang ada pada tabel.

Contoh kasus :

- Untuk data pembelajaran ke-1

  Nilai jarak terkecil terdapat pada jarak antara data ke-1 dengan bobot ke-1 (w_1) yaitu 3,464 maka posisi nilai jarak terkecil (C_j) = 1. Berdasarkan pada tabel 3.4, target kelas (T) dari data ke – 1 = 1. Maka, dapat disimpulkan bahwa target sesuai dengan hasil perhitungan.

- Untuk data pembelajaran ke-2

  Nilai jarak terkecil terdapat pada jarak antara data ke-2 dengan bobot ke-2 (w_2) yaitu 2,645 maka posisi nilai jarak terkecil(C_j) = 2. Berdasarkan pada tabel 3.5, target kelas (T) dari data ke – 2 = 2. Maka, dapat disimpulkan bahwa target sesuai dengan hasil perhitungan.

7. Cek nilai target (T) dengan hasil posisi nilai jarak terkecil (Cj)

Pengecekan ini dilakukan untuk melakukan perubahan bobot(w_j) terhadap bobot dengan jarak terkecil.

Jika T = C_j, maka w_j(baru) = w_j(lama) + α [x_i - w_j(lama)].....(3.2)

Jika T ≠ Cj maka w_j(baru) = w_j(lama) – α[x_i - w_j(lama)].....(3.3)

Contoh kasus :

Dilihat dari hasil contoh kasus untuk data pembelajaran ke-1 pada langkah c, didapatkan hasil T = C_j yaitu 1 = 1. Maka perhitungan yang digunakan adalah

$w_{1,1} = w_{1,1} + 0{,}1(x_{1,1} - w_{1,1}) = 1 + 0{,}1(1\text{-}1) = 1;$
$w_{1,2} = w_{1,2} + 0{,}1(x_{1,2} - w_{1,2}) = 1 + 0{,}1(1\text{-}1) = 1;$
$w_{1,3} = w_{1,3} + 0{,}1(x_{1,3} - w_{1,3}) = 1 + 0{,}1(1\text{-}1) = 1;$
...
$w_{1,75} = w_{1,75} + 0{,}1(x_{1,75} - w_{1,75}) = 0 + 0{,}1(0\text{-}0) = 0.$

Setelah dilakukan perhitungan, maka didapat hasil perhitungannya seperti yang terlihat pada tabel 3.12. Hasil pada tabel 3.12 ini yang akan dijadikan sebagai nilai w_1 yang baru.

Tabel 3.12 Nilai Bobot w_1 Baru

w	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	0,9	1	0	0	0	0,1	0	0	0	0,1	0
	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	31	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**
	0	0,1	0	0,1	0	0	0	0,1	0	1	1	1	0,9	1	1
	46	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**
	0,9	1	0	0	0,1	0	0,1	0	0	0,1	0	0	0	0	0
	61	**62**	**63**	**64**	**65**	**66**	**67**	**68**	**69**	**70**	**71**	**72**	**73**	**74**	**75**
	0	0	0	0	0	0	0	0,1	0	0	0	0	0	0	0

Sedangkan untuk data pembelajaran ke – 2, didapatkan hasil T = C_j yaitu 2 = 2. Maka perhitungan yang digunakan adalah

$w_{2,1} = w_{2,1} + 0{,}1(x_{2,1} - w_{2,1}) = 0 + 0{,}1(0\text{-}0) = 0;$
$w_{2,2} = w_{2,2} + 0{,}1(x_{2,2} - w_{2,2}) = 0 + 0{,}1(0\text{-}0) = 0;$
$w_{2,3} = w_{2,3} + 0{,}1(x_{2,3} - w_{2,3}) = 0 + 0{,}1(0\text{-}0) = 0;$
...
$w_{2,75} = w_{2,75} + 0{,}1(x_{2,75} - w_{2,75}) = 0 + 0{,}1(0\text{-}0) = 0.$

Dari perhitungan tersebut, maka didapat hasil perhitungannya seperti yang terlihat pada tabel 3.13. Hasil pada tabel 3.13 ini yang akan dijadikan sebagai nilai w_2 yang baru.

Tabel 3.13 Nilai Bobot w_2 Baru

w	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	0	0	0	0	0	1	1	0,9	1	1	1	0,9	1	1
	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
	1	0	0	0,1	0,1	0	0	0,1	0	0	0	0	0,1	0	0
	31	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**
	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	46	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**
	0	0	1	1	1	0,9	1	1	0	0	0	0	0	0	0
	61	**62**	**63**	**64**	**65**	**66**	**67**	**68**	**69**	**70**	**71**	**72**	**73**	**74**	**75**
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8. Menghitung perubahan *learning rate* (α)

Proses ini dilakukan setelah perulangan selesai. Rumus yang digunakan untuk melakukan pengurangan *learning rate* (α) adalah sebagai berikut :

$$\alpha = \alpha - (\alpha * 0{,}1);$$

Contoh kasus :

Setelah epoh ke-1, didapat nilai α = 0,1 – (0,1 * 0,1) = 0,09.

- Sebagai contoh kasus yang digunakan dengan nilai awal

  MaxEpoh = 2;

  Learning rate (α) = 0,1;

  Error minimum = 0,09

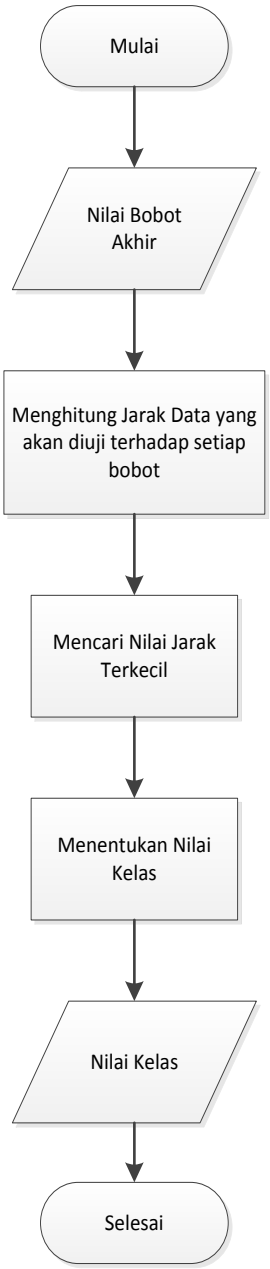
  Maka epoh yang dihasilkan yaitu sampai epoh ke-2 dengan nilai *learning rate* 0,081 dan hasil bobot akhir seperti yang terlihat pada tabel 3.14.

Tabel 3.14 Nilai Bobot Akhir

Bobot	Vektor Bobot																								
w_1	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23	X24	X25
	1	1	1	1	0,819	1	0	0	0	0,181	0	0	0	0,181	0	0	0	0	0	0	0	0	0	0	0
	X26	X27	X28	X29	X30	X31	X32	X33	X34	X35	X36	X37	X38	X39	X40	X41	X42	X43	X44	X45	X46	X47	X48	X49	X50
	0	0	0	0	0	0	0,181	0	0,181	0	0	0	0,181	0	1	1	1	0,819	1	1	0,819	1	0	0	0,181
	X51	X52	X53	X54	X55	X56	X57	X58	X59	X60	X61	X62	X63	X64	X65	X66	X67	X68	X69	X70	X71	X72	X73	X74	X75
	0	0,181	0	0	0,181	0	0	0	0	0	0	0	0	0	0	0	0	0,181	0	0	0	0	0	0	0
w_2	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23	X24	X25
	0	0	0	0	0	0	1	1	0,819	1	1	1	0,819	1	1	1	0	0	0,181	0,181	0	0	0,181	0	0
	X26	X27	X28	X29	X30	X31	X32	X33	X34	X35	X36	X37	X38	X39	X40	X41	X42	X43	X44	X45	X46	X47	X48	X49	X50
	0	0	0,181	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1
	X51	X52	X53	X54	X55	X56	X57	X58	X59	X60	X61	X62	X63	X64	X65	X66	X67	X68	X69	X70	X71	X72	X73	X74	X75
	0,819	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3. Pengujian *Learning Vector Quantization*

Setelah melakukan tahapan pembelajaran, tahapan selanjutnya yaitu tahapan pengujian *learning vector quantization* yang mana dari pengujian ini akan menghasilkan nilai kelas yang dikenali. *Flowchart* pada tahapan pengujian *learning vector quantization* akan dijelaskan pada gambar 3.3.



Gambar 3.3 *Flowchart* Pengujian *Learning Vector Quantization*

Adapun penjelasan langkah – langkah tahapan pengujian *learning vector quantization* yaitu sebagai berikut :

1. **Menghitung jarak data yang akan diuji terhadap setiap data bobot (w_j)**

   Data bobot yang digunakan di tahapan pengujian adalah nilai bobot akhir yang dihasilkan dari tahapan pembelajaran. Rumus yang digunakan untuk menghitung jarak data yang akan diuji terhadap setiap data bobot (w_j) adalah sebagai berikut :

   $$\text{Jarak} = \|x_{ij} - w_{ij}\| \quad \text{.....(3.1)}$$

   Contoh kasus :

Tabel 3.15 Contoh Data Pengujian yang Dijadikan Contoh Kasus

Data ke-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	0	1	1	0	0	0	1	0	1	1	0	0
	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	31	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**
	0	0	1	0	0	0	1	0	0	1	1	1	1	0	1
	46	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**
	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0
	61	**62**	**63**	**64**	**65**	**66**	**67**	**68**	**69**	**70**	**71**	**72**	**73**	**74**	**75**
	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0

- Jarak terhadap :

a. Bobot ke – 1 (w_1)

   Jarak $= \|x_{1j} - w_{1j}\|$

$$\sqrt{\begin{matrix}(1-1)^2+(1-1)^2+(1-1)^2+(0-1)^2+(1-0{,}819)^2+(1-1)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-0{,}181)^2+(0-0)^2+(1-0)^2+(1-0)^2+(0-0{,}181)^2+(0-0)^2+\\(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+\\(0-0)^2+(0-0{,}181)^2+(1-0)^2+(0-0{,}181)^2+(0-0)^2+(0-0)^2+(1-0)^2+(0-0{,}181)^2+(0-0)^2+(1-1)^2+(1-1)^2+(1-1)^2+(1-0{,}819)^2+(0-1)^2+(1-1)^2+\\(1-0{,}819)^2+(1-1)^2+(0-0)^2+(0-0)^2+(1-0{,}181)^2+(0-0)^2+(0-0{,}181)^2+(0-0)^2+(0-0)^2+(1-0{,}181)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+\\(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(1-0)^2+(0-0{,}181)^2+(1-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(0-0)^2\end{matrix}}$$

$$\text{Jarak} = \sqrt{\begin{matrix}1+0{,}032+0{,}670+1+1+0{,}032+0{,}032+1+\\0{,}032+1+0{,}032+0{,}032+1+0{,}032+0{,}670+\\0{,}032+0{,}670+1+1+0{,}032+1+1\end{matrix}}$$

$$\text{Jarak} = \sqrt{12{,}298} = 3{,}506$$

b. Bobot ke – 2 (w_2)

$$\text{Jarak} = \|x_{1j} - w_{2j}\|$$

$$\text{Jarak} = \sqrt{\begin{matrix}(1-0)^2+(1-0)^2+(1-0)^2+(0-0)^2+(1-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(0-0{,}819)^2+(1-1)^2+(0-1)^2+(1-1)^2+(1-0{,}819)^2+(0-1)^2+(0-1)^2+\\(0-1)^2+(0-0)^2+(0-0)^2+(0-0{,}181)^2+(0-0{,}181)^2+(0-0)^2+(0-0)^2+(0-0{,}181)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0{,}181)^2+(0-0)^2+(0-0)^2+\\(0-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(1-1)^2+(1-0)^2+(1-0)^2+(1-0)^2+(0-0)^2+(1-0)^2+\\(1-0)^2+(1-0)^2+(0-1)^2+(0-1)^2+(1-1)^2+(0-0{,}819)^2+(0-1)^2+(0-1)^2+(0-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+\\(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(1-0)^2+(0-0)^2+(1-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(0-0)^2+(1-0)^2+(0-0)^2\end{matrix}}$$

Jarak =

$$\sqrt{\begin{matrix}1+1+1+1+1+0{,}670+1+0{,}032+1+1+0{,}032+\\0{,}032+0{,}032+0{,}032+1+1+1+1+1+1+1+1+\\1+1+0{,}670+1+1+1+1+1+1+1\end{matrix}}$$

$$\text{Jarak} = \sqrt{26{,}5} = 5{,}147$$

2. Mencari nilai jarak terkecil

Dari hasil perhitungan data yang diuji terhadap setiap bobot(w_j), maka carilah nilai jarak terkecilnya.

Contoh kasus :

Berdasarkan hasil perhitungan antara data pengujian ke – 1 dengan setiap bobot(w_j), maka didapat nilai jarak terkecil terhadap bobot ke -1 (w_1) sebesar 3,506.

3. Menentukan nilai kelas atau target keluaran

Nilai kelas atau target keluaran didapat dari posisi nilai jarak terkecil.

Contoh kasus :

Untuk data pengujian ke -1, nilai kelasnya yaitu termasuk ke dalam kelas 1 karena posisi nilai jarak terkecilnya terdapat pada bobot ke -1.

3.4 Analisis *Cross Validation* Dan *Confusion Matrix*

Cross validation adalah salah satu cara untuk menemukan parameter terbaik dari satu model dengan cara menguji besarnya *error* pada tes. *Confusion matrix* digunakan untuk mengetahui informasi mengenai kelas asli dan kelas hasil deteksi yang dilakukan oleh sistem.

3.4.1 Analisis *Cross validation*

Cross validation yang dilakukan pada simulasi pendeteksian anak berbakat(*gifted child*) menggunakan nilai k = 5. Nilai 5 ini menunjukkan jumlah *fold* data. Masing – masing *fold* terdiri dari 10 data. Untuk setiap *fold* memiliki karakter – karakter seperti berikut :

a. D1

   D1 adalah kelompok data yang terdiri dari 10 data mulai dari id_anak ke-1 sampai id_anak ke-10. Karakter data pada kelompok ini terdiri dari 2 data tipe *the successfuls*, 2 data tipe *the challanging*, 2 data tipe *the underground*, 2 data tipe *the dropouts*, 1 data tipe the *double labeled* dan 1 data tipe the *autonomous learner*. Data anak berbakat(*gifted child*) untuk kelompok D1 dapat dilihat pada Lampiran C Data Anak Berbakat(*Gifted Child*) Tabel 1. Data Anak Berbakat(*Gifted Child*) Untuk Kelompok D1 sampai Tabel 3. Data Anak Berbakat(*Gifted Child*) Untuk Kelompok D1(Lanjutan).

b. D2

   D2 adalah kelompok data yang terdiri dari 10 data mulai dari id_anak ke-11 sampai id_anak ke-20. Karakter data pada kelompok ini terdiri dari 2 data tipe *the successfuls*, 2 data tipe *the challanging*, 1 data tipe *the underground*, 1 data tipe *the dropouts*, 2 data tipe the *double labeled* dan 2 data tipe the *autonomous learner*. Data anak berbakat(*gifted child*) untuk kelompok D2 dapat dilihat pada Lampiran C Data Anak Berbakat(*Gifted Child*) Tabel 4. Data Anak

Berbakat(*Gifted Child*) Untuk Kelompok D2 sampai Tabel 6. Data Anak Berbakat(*Gifted Child*) Untuk Kelompok D2(Lanjutan).

c. D3

D3 adalah kelompok data yang terdiri dari 10 data mulai dari id_anak ke-21 sampai id_anak ke-30. Karakter data pada kelompok ini terdiri dari 1 data tipe *the successfuls*, 1 data tipe *the challanging*, 2 data tipe *the underground*, 2 data tipe *the dropouts*, 2 data tipe the *double labeled* dan 2 data tipe the *autonomous learner*. Data anak berbakat(*gifted child*) untuk kelompok D3 dapat dilihat pada Lampiran C Data Anak Berbakat(*Gifted Child*) Tabel 7. Data Anak Berbakat(*Gifted Child*) Untuk Kelompok D3 sampai Tabel 9. Data Anak Berbakat(*Gifted Child*) Untuk Kelompok D3(Lanjutan).

d. D4

D4 adalah kelompok data yang terdiri dari 10 data mulai dari id_anak ke-31 sampai id_anak ke-40. Karakter data pada kelompok ini terdiri dari 2 data tipe *the successfuls*, 2 data tipe *the challanging*, 2 data tipe *the underground*, 2 data tipe *the dropouts*, 1 data tipe the *double labeled* dan 1 data tipe the *autonomous learner*. Data anak berbakat(*gifted child*) untuk kelompok D4 dapat dilihat pada Lampiran C Data Anak Berbakat(*Gifted Child*) Tabel 10. Data Anak Berbakat(*Gifted Child*) Untuk Kelompok D4 sampai Tabel 12. Data Anak Berbakat(*Gifted Child*) Untuk Kelompok D4(Lanjutan).

e. D5

D5 adalah kelompok data yang terdiri dari 10 data mulai dari id_anak ke-41 sampai id_anak ke-50. Karakter data pada kelompok ini terdiri dari 1 data tipe *the successfuls*, 2 data tipe *the challanging*, 1 data tipe *the underground*, 1 data tipe *the dropouts*, 2 data tipe the *double labeled* dan 3 data tipe the *autonomous learner*. Data anak

berbakat(*gifted child*) untuk kelompok D5 dapat dilihat pada Lampiran C Data Anak Berbakat(*Gifted Child*) Tabel 13. Data Anak Berbakat(*Gifted Child*) Untuk Kelompok D5 sampai Tabel 15. Data Anak Berbakat(*Gifted Child*) Untuk Kelompok D5(Lanjutan).

3.4.2 Analisis *Confusion Matrix*

Confusion matrix digunakan untuk mengetahui informasi mengenai kelas asli dan kelas hasil deteksi yang dilakukan oleh simulasi pendeteksian anak berbakat(*gifted child*). *Confusion matrix* pada simulasi pendeteksian anak berbakat(*gifted child*) menggunakan matriks 6 x 6 seperti yang dijelaskan pada tabel 3.16.

Tabel 3.16 *Confusion Matrix* Simulasi Pendeteksian Anak Berbakat(*Gifted Child*)

		Kelas Hasil Deteksi					
		Kelas 1	**Kelas 2**	**Kelas 3**	**Kelas 4**	**Kelas 5**	**Kelas 6**
Kelas Asli	**Kelas 1**	a	b	c	d	e	f
	Kelas 2	g	h	i	j	k	l
	Kelas 3	m	n	o	p	q	r
	Kelas 4	s	t	u	v	w	x
	Kelas 5	y	z	aa	bb	cc	dd
	Kelas 6	ee	ff	gg	hh	ii	jj

Keterangan :

a : Jumlah dari hasil kelas 1 yang terdeteksi benar sebagai kelas 1

b : Jumlah dari hasil kelas 1 yang terdeteksi salah sebagai kelas 2

c : Jumlah dari hasil kelas 1 yang terdeteksi salah sebagai kelas 3

d : Jumlah dari hasil kelas 1 yang terdeteksi salah sebagai kelas 4

e : Jumlah dari hasil kelas 1 yang terdeteksi salah sebagai kelas 5

f : Jumlah dari hasil kelas 1 yang terdeteksi salah sebagai kelas 6

g : Jumlah dari hasil kelas 2 yang terdeteksi salah sebagai kelas 1

h : Jumlah dari hasil kelas 2 yang terdeteksi benar sebagai kelas 2

i : Jumlah dari hasil kelas 2 yang terdeteksi salah sebagai kelas 3
j : Jumlah dari hasil kelas 2 yang terdeteksi salah sebagai kelas 4
k : Jumlah dari hasil kelas 2 yang terdeteksi salah sebagai kelas 5
l : Jumlah dari hasil kelas 2 yang terdeteksi salah sebagai kelas 6
m : Jumlah dari hasil kelas 3 yang terdeteksi salah sebagai kelas 1
n : Jumlah dari hasil kelas 3 yang terdeteksi salah sebagai kelas 2
o : Jumlah dari hasil kelas 3 yang terdeteksi benar sebagai kelas 3
p : Jumlah dari hasil kelas 3 yang terdeteksi salah sebagai kelas 4
q : Jumlah dari hasil kelas 3 yang terdeteksi salah sebagai kelas 5
r : Jumlah dari hasil kelas 3 yang terdeteksi salah sebagai kelas 6
s : Jumlah dari hasil kelas 4 yang terdeteksi salah sebagai kelas 1
t : Jumlah dari hasil kelas 4 yang terdeteksi salah sebagai kelas 2
u : Jumlah dari hasil kelas 4 yang terdeteksi salah sebagai kelas 3
v : Jumlah dari hasil kelas 4 yang terdeteksi benar sebagai kelas 4
w : Jumlah dari hasil kelas 4 yang terdeteksi salah sebagai kelas 5
x : Jumlah dari hasil kelas 4 yang terdeteksi salah sebagai kelas 6
y : Jumlah dari hasil kelas 5 yang terdeteksi salah sebagai kelas 1
z : Jumlah dari hasil kelas 5 yang terdeteksi salah sebagai kelas 2
aa : Jumlah dari hasil kelas 5 yang terdeteksi salah sebagai kelas 3
bb : Jumlah dari hasil kelas 5 yang terdeteksi salah sebagai kelas 4
cc : Jumlah dari hasil kelas 5 yang terdeteksi benar sebagai kelas 5
dd : Jumlah dari hasil kelas 5 yang terdeteksi salah sebagai kelas 6
ee : Jumlah dari hasil kelas 6 yang terdeteksi salah sebagai kelas 1
ff : Jumlah dari hasil kelas 6 yang terdeteksi salah sebagai kelas 2
gg : Jumlah dari hasil kelas 6 yang terdeteksi salah sebagai kelas 3
hh : Jumlah dari hasil kelas 6 yang terdeteksi salah sebagai kelas 4
ii : Jumlah dari hasil kelas 6 yang terdeteksi salah sebagai kelas 5
jj : Jumlah dari hasil kelas 6 yang terdeteksi benar sebagai kelas 6

Kelas yang digunakan pada *confusion matrix* simulasi anak berbakat(*gifted child*) adalah 6 kelas seperti yang dijelaskan pada tabel 3.17.

Tabel 3.17 Kelas – Kelas Untuk Deteksi Anak Berbakat(*Gifted Child*)

Nama Kelas	Tipe Gifted
Kelas 1	*The Successfuls*
Kelas 2	*The Challanging*
Kelas 3	*The Underground*
Kelas 4	*The Dropouts*
Kelas 5	*The Double Labeled*
Kelas 6	*The Autonomous Learner*

Contoh kasus :

Dilakukan pembelajaran *learning vector quantization* dengan kombinasi parameter maksimal epoh = 2300, *learning rate* = 0,1 dan *error* minimum = 0,05.

Setelah dilakukan pembelajaran dan mendapatkan nilai bobot akhir, kemudian dilakukan pengujian *learning vector quantization* dan *confusion matrix.* Hasil *confusion matrix* dari contoh kasus dapat dilihat pada tabel 3.18.

Tabel 3.18 *Confusion Matrix* Contoh Kasus

		Kelas Hasil Deteksi					
		Kelas 1	**Kelas 2**	**Kelas 3**	**Kelas 4**	**Kelas 5**	**Kelas 6**
Kelas Asli	**Kelas 1**	1	0	0	0	0	0
	Kelas 2	0	2	0	0	0	0
	Kelas 3	0	0	1	0	0	0
	Kelas 4	0	0	0	0	1	0
	Kelas 5	0	0	0	0	2	0
	Kelas 6	1	0	0	0	0	2

Dari tabel 3.19 didapatkan hasil sebagai berikut :

a. Kelas 1 yang terdeteksi benar sebagai kelas 1 sebanyak 1 data
b. Kelas 2 yang terdeteksi benar sebagai kelas 2 sebanyak 2 data
c. Kelas 3 yang terdeteksi benar sebagai kelas 3 sebanyak 1 data
d. Kelas 4 yang terdeteksi benar sebagai kelas 4 tidak ada
e. Kelas 5 yang terdeteksi benar sebagai kelas 5 sebanyak 2 data
f. Kelas 6 yang terdeteksi benar sebagai kelas 6 sebanyak 2 data.

3.5 Analisis Kompleksitas Waktu Asimptotik Algoritma *Learning Vector Quantization*

Untuk mendapatkan nilai perhitungan kompleksitas waktu asimptotik algoritma *learning vector quantization* dapat dilakukan dengan cara menelusuri setiap langkah algoritma *learning vector quantization* pada *pseudocode*-nya. Di dalam *pseudocode* algoritma *learning vector quantization*, terdapat beberapa prosedur dan sebuah fungsi. Berikut adalah sebuah fungsi dan beberapa prosedur, yaitu :

1. *Pseudocode* prosedur pembelajaran *learning vector quantization* yang dijelaskan pada tabel 3.19;
2. *Pseudocode* prosedur pengujian *learning vector quantization* yang dijelaskan pada tabel 3.20.

Tabel 3.19 *Pseudocode* Prosedur Pembelajaran *Learning Vector Quantization*

	Isi pseudocode
1	`Procedure prosesPembelajaranLVQ (input epoh,MaxEpoh,kolom :`
2	`integer,eps,rate: float, x :`
3	`array[1..dataLatih] of float, TP`
4	`: array[1..databobot] of integer`
5	`output rate : float, w : array[1..databobot] of float)`
6	
7	`{I.S : bobot awal,MaxEpoh, rate, eps, epoh untuk pembelajaran`
8	`sudah terdefinisi}`
9	`{F.S : bobot akhir dari hasil pembelajaran}`
10	
11	`Kamus :`
12	`i,j,k,c : integer`
13	`dataLatih : integer`
14	`databobot : integer`
15	`H : array[1..databobot] of float`
16	`TK : array[1..databobot] of integer`
17	`jarak : float`
18	
19	`Algoritma :`
20	`while ((epoh < MaxEpoh) or (rate > eps) do`
21	`epoh ← epoh+1`
22	`for i ← 0 to dataLatih do`
23	`for j ← 0 to databobot do`
24	`jarak ← 0`
	`for k ← 0 to kolom do`

```
                    jarak ← jarak+Pow((x[i][k])-(w[j.x[k]),2)
                endfor
                H[j] ← sqrt(jarak)
            endfor
            c ← 0
            for j ← 1 to databobot do
                if (H[j] < H[c]) then
                    c ← j
                endif
            endfor
            TK[c] ← c
            if (TK[c]==TP[i]) then
                for k ← 0 to kolom do
                    w[c][k] ← w[c][k] + rate *(x[i][k]-w[c].x[k])
                endfor
            else
                for k ← 0 to kolom do
                    w[c][k] ← w[c][k] - rate *(x[i][k]-w[c].x[k])
                endfor
            endif
        endfor
        rate ← rate - (rate*o,1)
endwhile
endprocedure
```

Tabel 3.20 *Pseudocode* Prosedur Pengujian *Learning Vector Quantization*

Isi pseudocode	
1	Procedure processPengujianLVQ (input x : array[1..datauji]
2	of float, w :
3	array[1..databobot] of
4	float, kolom : integer,
5	output c : integer)
6	
7	{I.S : bobot akhir dan data pengujian sudah terdefinisi}
8	{F.S : menghasilkan target kelas}
9	
10	Kamus :
11	i,j,k : integer
12	H : array[1..databobot] of float
13	jarak : float
14	
15	Algoritma :
16	for i ← 0 to datauji do
17	for j ← 0 to databobot do
18	jarak ← 0
19	for k ← 0 to kolom do
20	jarak ← jarak +Pow((x[i][k])-(w[j].x[k]),2)
21	endfor
22	H[j] ← sqrt(jarak)
23	endfor
24	c ← 0
25	for j ← 1 to databobot do
26	if(H[j] < H[c]) then
27	c ← j
28	endif
29	endfor
30	endfor
31	endprocedure

Perhitungan kompleksitas waktu dengan perhitungan big-O dilakukan dengan menghitung seluruh tahap - tahap algoritma *learning vector quantization*, dimulai dari pembelajaran sampai pengujian jaringan syaraf tiruan. Perhitungan kompleksitas waktu ditunjukkan dalam dua tabel perhitungan Big-O algoritma *learning vector quantization* yang dijelaskan pada tabel 3.21 dan tabel 3.22.

Tabel 3.21 Perhitungan Big-O Prosedur Pembelajaran *Learning Vector Quantization*

No	Algoritma	Nilai Big-O
1	while ((epoh < MaxEpoh) or (rate > eps) do	O(*n*)
2	epoh ← epoh+1	O(1)
3	for i ← 0 to dataLatih do	O(*n*)
4	for j ← 0 to databobot do	O(*n*)
5	jarak ← 0	O(1)
6	for k ← 0 to kolom do	O(*n*)
7	jarak ← jarak +Pow((x[i][k])-(w[j].x[k]),2)	O(1)
8	Endfor	
9	H[j] ← sqrt(jarak)	O(1)
10	Endfor	
11	c ← 0	O(1)
12	for j ← 1 to databobot do	O(*n*)
13	if (H[j] < H[c]) then	O(1)
14	c ← j	O(1)
15	endif	
16	Endfor	
17	TK[c] ← c	O(1)
17	if (TK[c]==TP[i]) then	O(1)
18	for k ← 0 to kolom do	O(*n*)
19	w[c][k] ← w[c][k] + rate *(x[c][k]-w[c].x[k])	O(1)
20	Endfor	
21	Else	
22	for k ← 0 to kolom do	O(*n*)
23	w[c][k] ← w[c][k] - rate *(x[c][k]-w[c].x[k])	O(1)
24	Endfor	
25	Endif	
26	Endfor	
27	rate ← rate – (rate*o,1)	O(1)
28	endwhile	

Dari perhitungan pada tabel 3.21 didapatkan hasil f(n) = $n^4 + 3n^3 + 2n^2 + 2n$, maka didapatkan hasil T(n) = O(n^4). O(n^4) ini relatif mirip dengan *O*(n^2) yang mana *O*(n^2) berada dalam urutan spektrum kompleksitas waktu algoritma *polynomial*.

Tabel 3.22 Perhitungan Big-O Pengujian *Learning Vector Quantization*

No	Algoritma	Nilai Big-O
1	for i ← 0 to datauji do	O(n)
2	for j ← 0 to databobot do	O(n)
3	jarak ← 0	O(1)
4	for k ← 0 to kolom do	O(n)
5	jarak ← jarak +Pow((x[i][k])-(w[j].x[k]),2)	O(1)
7	Endfor	
8	H[j] ← sqrt(jarak)	O(1)
9	Endfor	
10	c ← 0	O(1)
11	for j ← 0 to databobot do	O(n)
12	if(H[j] < H[c]) then	O(1)
14	c ← k	O(1)
15	Endif	
16	Endfor	
17	Endfor	

Dari perhitungan pada tabel 3.22 didapatkan hasil f(n) = $n^3 + 2n^2 + n$, maka didapatkan hasil T(n) = *O*(n^3). *O*(n^3) ini adalah algoritma kubik yang mana berada dalam urutan spektrum kompleksitas waktu algoritma *polynomial*.

3.6 Analisis Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak dilakukan berdasarkan pada kebutuhan perangkat lunak pendeteksian anak berbakat (*gifted child*) untuk

penerapan analisis metode *learning vector quantization* serta hasil observasi. Analisis kebutuhan perangkat lunak dibagi menjadi dua bagian yaitu :

1. Kebutuhan Non Fungsional;
2. Kebutuhan Fungsional.

3.6.1 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan suatu langkah yang menguraikan kebutuhan - kebutuhan batasan sistem menjadi kebutuhan yang lebih detail berdasarkan fakta dan kebutuhan yang ada. Analisis kebutuhan non fungsional yang dilakukan dibagi dalam tiga tahap, yaitu :

1. Analisis Kebutuhan Perangkat Keras (*Hardware*);
2. Analisis Kebutuhan Perangkat Lunak (*Software*);
3. Analisis Kebutuhan Pengguna (*User*).

3.6.1.1 Analisis Kebutuhan Perangkat Keras (*Hardware*)

Spesifikasi perangkat keras minimum yang dibutuhkan agar dapat menjalankan simulasi pendeteksian anak berbakat(*gifted child*) pada masa perkembangan ini seperti yang dijelaskan pada tabel 3.23.

Tabel 3.23 Analisis Kebutuhan Perangkat Keras (*Hardware*)

No	Perangkat Keras	Spesifikasi
1	Prosesor	Kecepatan 1.5 GHz
2	Monitor	Monitor 17'', resolusi 1024x768
3	VGA On-board	Kecepatan 512 MB
4	Optical Drive DVD-ROM	Standard
5	Keyboard	Standard
6	Mouse	Standard
7	Memori	2048 MB

3.6.1.2 Analisis Kebutuhan Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam membangun dan menerapkan simulasi pendeteksian anak berbakat(*gifted child*) pada masa perkembangan ini dijelaskan pada tabel 3.24.

Tabel 3.24 Analisis Kebutuhan Perangkat Lunak (*Software*)

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 7 Ultimate
2	Tool Pembangun Perangkat Lunak	Microsoft Visual C# 2010 Express
3	Database	MySQL
4	Web server	WampServer 2.0
5	Pemodelan	Astah Professional
6	Perancangan	Balsamiq Mockup

3.6.1.3 Analisis Kebutuhan Pengguna (*User*)

Spesifikasi pengguna atau *user* yang akan menggunakan aplikasi pengidentifikasian anak berbakat(*gifted child*) pada masa perkembangan ini adalah sebagai berikut :

1. Pengguna dapat membaca;
2. Pengguna terbiasa dan mengerti menggunakan komputer;
3. Pengguna minimal mampu menggunakan keyboard dan mouse untuk berinteraksi dengan simulasi.

3.6.2 Analisis Kebutuhan Fungsional

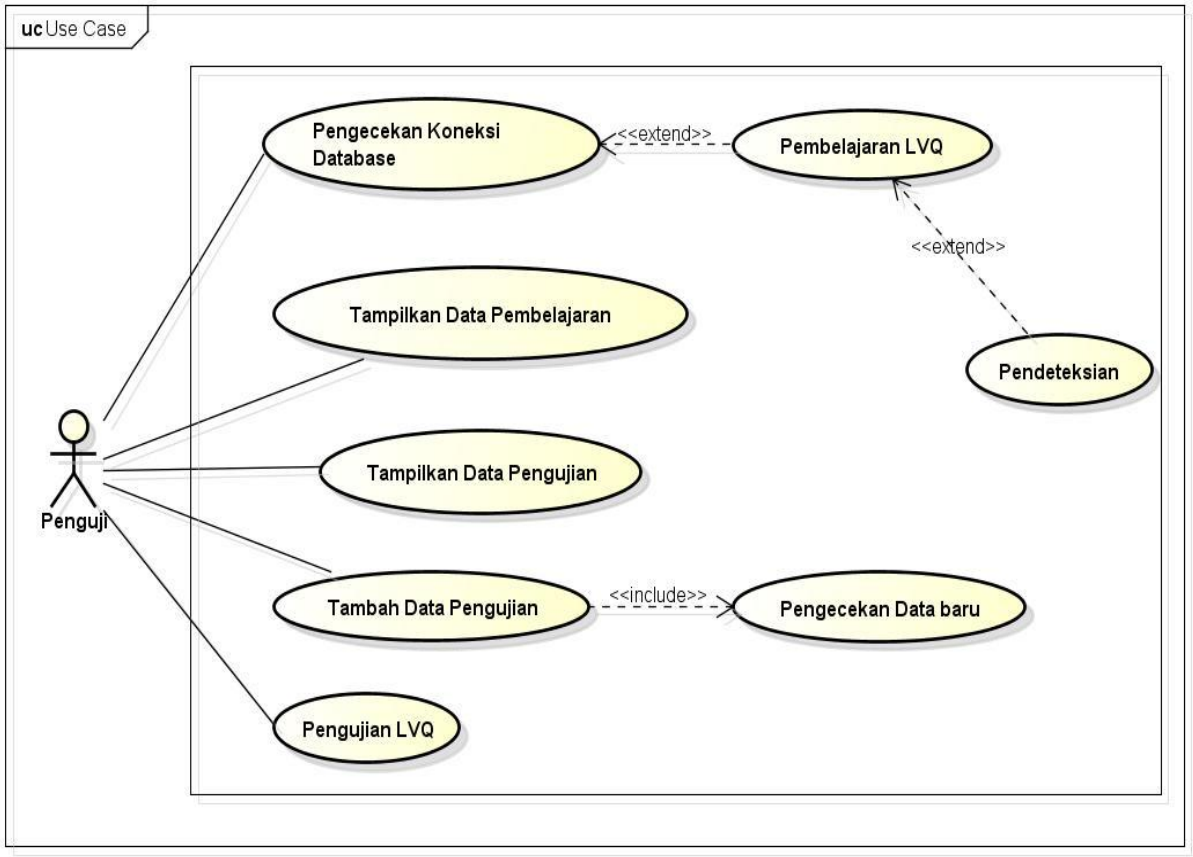
Analisis kebutuhan fungsional dapat didefinsikan sebagai penggambaran, perencanaan dan pembuatan sketsa yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Dengan kata lain, analisis kebutuhan fungsional yaitu menspesifikasikan tentang hal - hal yang akan dilakukan sistem ketika di implementasikan.

Alat bantu yang digunakan untuk menggambarkan sistem secara umum yang akan dibangun yaitu *Use Case Diagram*. Untuk menjelaskan bagaimana

aktivitas pada setiap proses maka digunakan *Activity Diagram*. *Class Diagram* untuk mengetahui struktur kelas – kelas dari suatu sistem dan *Sequence Diagram* untuk mengetahui gambaran interaksi yang terjadi antara sejumlah objek. Sedangkan alat bantu yang digunakan untuk perancangan dari sebuah sistem secara umum yang akan dibangun yaitu mulai dari perancangan struktur menu, perancangan antarmuka sampai perancangan jaringan semantik.

3.6.2.1 *Use Case Diagram*

Use Case Diagram adalah gambaran umum dari perancangan sistem yang akan dibangun. *Use case* diagram untuk simulasi pendeteksian anak berbakat(*gifted child*) dapat dilihat pada gambar 3.4.



Gambar 3.4 *Use Case* Simulasi Pendeteksian Anak Berbakat(*Gifted Child*)

3.6.2.1.1 Identifikasi Aktor

Aktor adalah objek yang akan melakukan peran apa saja dalam simulasi. Aktor yang berperan dalam menjalankan simulasi pendeteksian anak berbakat (*gifted child*) dijelaskan pada tabel 3.25.

Tabel 3.25 Identifikasi Aktor

No	Aktor	Deskripsi
A-01	Penguji	Aktor yang dapat menggunakan simulasi pendeteksian anak berbakat (*gifted child*) mulai dari pembelajaran LVQ, menampilkan data pembelajaran dan data pengujian, melakukan pendeteksian anak berbakat(*gifted child*) sampai dengan melakukan pengujian LVQ.

3.6.2.1.2 Identifikasi *Use Case*

Identifikasi *use case* digunakan untuk mendefinisikan fungsi dari setiap *use case* yang bekerja pada simulasi. Identifikasi *use case* pada simulasi pendeteksian anak berbakat(*gifted child*) dijelaskan pada tabel 3.26 sampai dengan tabel 3.27.

Tabel 3.26 Identifikasi *Use Case* Simulasi Pendeteksian Anak Berbakat(*gifted child*)

No	Use Case	Deskripsi
UC-01	Pengecekan Koneksi Database	Fungsionalitas untuk menghubungkan simulasi dengan database yang akan digunakan
UC-02	Pembelajaran LVQ	Fungsionalitas untuk melakukan pembelajaran terhadap data – data gejala anak berbakat(*gifted child*) yang sudah ada sebelumnya
UC-03	Pendeteksian	Fungsionalitas untuk melakukan pendeteksian anak berbakat(*gifted child*) terhadap data yang akan diuji
UC-04	Tampilkan Data Pembelajaran	Fungsionalitas untuk menampilkan data – data pembelajaran

Tabel 3.27 Identifikasi *Use Case* Simulasi Pendeteksian Anak Berbakat(*gifted child*) (Lanjutan)

No	Use Case	Deskripsi
UC-05	Tampilkan Data Pengujian	Fungsionalitas untuk menampilkan data – data pengujian
UC-06	Tambah Data Pengujian	Fungsionalitas untuk menambah data pengujian
UC-07	Pengecekan Data Baru	Fungsionalitas untuk mendapatkan kelas dan nama kelas target dari data pengujian baru
UC-08	Pengujian LVQ	Fungsionalitas untuk mengetahui akurasi dan optimasi parameter serta waktu pemrosesan dari metode LVQ dengan menggunakan *confusion matrix* dan *cross validation*

3.6.2.1.3 Skenario *Use Case*

Skenario *use case* digunakan untuk menggambarkan atau menjelaskan setiap *use case* yang ada pada simulasi. Gambaran *use case* di skenario berupa proses apa saja yang terjadi pada setiap bagian, dimana *user* memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh simulasi kepada *user*. Skenario *use case* dijelaskan pada tabel 3.28 sampai dengan tabel 3.41.

Tabel 3.28 Skenario *Use Case* Pengecekan Koneksi Database

Nama *Use Case*	Pengecekan Koneksi Database
***Use Case* Terkait**	Pembelajaran LVQ
Tujuan	Mengkoneksikan antara database yang akan digunakan dengan simulasi.
Kondisi Awal	Database belum terkoneksi dengan simulasi
Kondisi Akhir (jika berhasil)	Database berhasil terkoneksi dengan simulasi
Kondisi Akhir (jika gagal)	Database gagal terkoneksi dengan simulasi

Tabel 3.29 Skenario *Use Case* Pengecekan Koneksi Database (Lanjutan)

Aktor	Penguji	
Skenario Utama	**Langkah**	**Aksi**
	1	Penguji memilih *icon setting*
	2	Sistem menampilkan halaman *setting database*
	3	Penguji memasukkan nama *server*, nama *database*, nama *user* dan password
	4	Penguji memilih tombol cek koneksi
	5	Sistem memverifikasi nama *server*, nama *database*, nama *user* dan password dengan *database* yang akan digunakan
	6	Sistem menampilkan pesan "Simulasi Terkoneksi Dengan Database"
	7	Penguji memilih tombol simpan
	8	Database berhasil terkoneksi dengan simulasi
Skenario Alternatif	**Langkah**	**Aksi**
	5.1	Sistem menolak nama *server*, nama *database*, nama *user* dan password karena ketidakcocokan dengan nama *database* yang akan digunakan
	5.2	Sistem menampilkan pesan "Simulasi Tidak Terkoneksi Dengan Database"
	5.3	Database gagal terkoneksi dengan simulasi

Tabel 3.30 Skenario *Use Case* Pembelajaran LVQ

Nama *Use Case*	Pembelajaran LVQ
***Use Case* Terkait**	Pendeteksian
Tujuan	Melakukan pembelajaran terhadap data – data gejala anak berbakat(*gifted child*) yang sudah ada sebelumnya
Kondisi Awal	Nilai parameter – parameter masih kosong

Tabel 3.31 Skenario *Use Case* Pembelajaran LVQ (Lanjutan)

Kondisi Akhir (jika berhasil)	Berhasil menampilkan hasil nilai bobot akhir, banyaknya iterasi yang dilakukan dan perubahan setiap nilai *learning rate* di setiap iterasinya.	
Kondisi Akhir (jika gagal)	Gagal menampilkan hasil nilai bobot akhir, banyaknya iterasi yang dilakukan dan perubahan setiap nilai *learning rate* di setiap iterasinya.	
Aktor	Penguji	
Skenario Utama	**Langkah**	**Aksi**
	1	Penguji memilih menu pembelajaran
	2	Sistem menampilkan halaman pembelajaran
	3	Penguji memasukkan detail parameter
	4	Penguji memilih tombol belajar
	5	Sistem memeriksa kelengkapan dari detail parameter
	6	Sistem melakukan proses perhitungan pembelajaran LVQ
	7	Hasil nilai bobot akhir, banyaknya iterasi yang dilakukan dan perubahan setiap nilai *learning rate* di setiap iterasinya berhasil ditampilkan
Skenario Alternatif	**Langkah**	**Aksi**
	5.1	Sistem menolak kelengkapan dari detail parameter
	5.2	Sistem menampilkan pesan "Inputan Parameter Tidak Boleh Kosong"
	5.3	Gagal menampilkan hasil nilai bobot akhir, banyaknya iterasi yang dilakukan dan perubahan setiap nilai *learning rate* di setiap iterasinya.

Tabel 3.32 Skenario *Use Case* Pendeteksian

Nama *Use Case*	Pendeteksian	
***Use Case* Terkait**		
Tujuan	Melakukan pendeteksian terhadap data pengujian dan mendapatkan hasil deteksi	
Kondisi Awal	Data pengujian yang ingin diuji dan data gejala anak berbakat(*gifted child*) belum tersedia	
Kondisi Akhir (jika berhasil)	Hasil deteksi berhasil ditampilkan	
Kondisi Akhir (jika gagal)	Hasil deteksi gagal ditampilkan	
Aktor	Penguji	
Skenario Utama	**Langkah**	**Aksi**
	1 2	Penguji memilih menu deteksi Sistem menampilkan halaman deteksi
	3	Penguji memilih id data uji ke berapa yang akan di uji
	4	Sistem menampilkan id data uji yang akan di uji, kelas, nama kelas dan data gejala anak berbakat(*gifted child*) berdasarkan data uji yang dipilih
	5	Penguji memilih tombol deteksi
	6	Sistem melakukan proses perhitungan pengujian LVQ
	7	Hasil deteksi berhasil ditampilkan
Skenario Alternatif	**Langkah**	**Aksi**
	3.1	Penguji tidak memilih id data uji ke berapa yang akan di uji
	5.2	Hasil deteksi gagal ditampilkan

Tabel 3.33 Skenario *Use Case* Tampilkan Data Pembelajaran

Nama *Use Case*	Tampilkan Data Pembelajaran	
***Use Case* Terkait**		
Tujuan	Menampilkan data – data pembelajaran	
Kondisi Awal	Data – data pembelajaran belum tersedia	
Kondisi Akhir (jika berhasil)	Data – data pembelajaran berhasil ditampilkan	
Kondisi Akhir (jika gagal)	Data – data pembelajaran gagal ditampilkan	
Aktor	Penguji	
Skenario Utama	**Langkah**	**Aksi**
	1	Penguji memilih menu data
	2	Sistem menampilkan halaman data
	3	Penguji memilih tombol tampilkan data di grup pembelajaran
	4	Sistem menampilkan data – data pembelajaran di grup pembelajaran
	5	Data – data pembelajaran berhasil ditampilkan
Skenario Alternatif	**Langkah**	**Aksi**
	3.1	Penguji batal menampilkan data – data pembelajaran di grup pembelajaran
	3.2	Data – data pembelajaran gagal ditampilkan

Tabel 3.34 Skenario *Use Case* Tampilkan Data Pengujian

Nama *Use Case*	Tampilkan Data Pengujian
***Use Case* Terkait**	
Tujuan	Menampilkan data – data pengujian
Kondisi Awal	Data – data pengujian belum tersedia

Tabel 3.35 Skenario *Use Case* Tampilkan Data Pengujian (Lanjutan)

Kondisi Akhir (jika berhasil)	Data – data pengujian berhasil ditampilkan	
Kondisi Akhir (jika gagal)	Data – data pengujian gagal ditampilkan	
Aktor	Penguji	
Skenario Utama	**Langkah**	**Aksi**
	1	Penguji memilih menu data
	2	Sistem menampilkan halaman data
	3	Penguji memilih tombol tampilkan data di grup pengujian
	4	Sistem menampilkan data – data pembelajaran di grup pengujian
	5	Data – data pengujian berhasil ditampilkan
Skenario Alternatif	**Langkah**	**Aksi**
	3.1	Penguji batal menampilkan data – data pengujian di grup pengujian
	3.2	Data – data pengujian gagal ditampilkan

Tabel 3.36 Skenario *Use Case* Tambah Data Pengujian

Nama *Use Case*	Tambah Data Pengujian
***Use Case* Terkait**	Pengecekan Data Baru
Tujuan	Menambah data pengujian
Kondisi Awal	Data pengujian baru belum tersedia
Kondisi Akhir (jika berhasil)	Data pengujian berhasil ditambahkan
Kondisi Akhir (jika gagal)	Data pengujian gagal ditambahkan
Aktor	Penguji

Tabel 3.37 Skenario *Use Case* Tambah Data Pengujian (Lanjutan)

Skenario Utama	Langkah	Aksi
	1	Penguji memilih menu deteksi
	2	Sistem menampilkan halaman deteksi
	3	Penguji memilih data uji ke- dan pilih buat data baru
	4	Sistem menampilkan data gejala anak berbakat(*gifted child*) yang sudah diisi dengan nilai default "tidak"
	5	Penguji memasukkan data gejala anak berbakat(*gifted child*) berdasarkan gejala perilaku dan sosial emosi
	6	Penguji memilih tombol simpan
	7	Sistem memeriksa kelengkapan data gejala anak berbakat(*gifted child*) berdasarkan perilaku dan sosial emosi
	8	Sistem menampilkan pesan "Data Pengujian Dengan Id x Berhasil Disimpan"
	9	Data pengujian berhasil ditambahkan
Skenario Alternatif	**Langkah**	**Aksi**
	7.1	Sistem menolak kelengkapan data gejala anak berbakat(*gifted child*) berdasarkan perilaku dan sosial emosi
	7.2	Sistem menampilkan pesan "Data Pengujian Dengan Id x Gagal Disimpan"
	7.3	Data pengujian gagal ditambahkan

Tabel 3.38 Skenario *Use Case* Pengecekan Data Baru

Nama *Use Case*	Pengecekan Data Baru	
***Use Case* Terkait**	Tambah Data Pengujian	
Tujuan	Mendapatkan kelas dan nama kelas target dari data pengujian baru	
Kondisi Awal	Kelas dan nama kelas target dari data pengujian baru belum tersedia	
Kondisi Akhir (jika berhasil)	Kelas dan nama kelas target dari data pengujian baru berhasil ditampilkan	
Kondisi Akhir (jika gagal)	Kelas dan nama kelas target dari data pengujian baru gagal ditampilkan	
Aktor	Penguji	
Skenario Utama	**Langkah**	**Aksi**
	1	Penguji memasukkan data gejala anak berbakat(*gifted child*) berdasarkan gejala perilaku dan sosial emosi
	2	Penguji memilih tombol cek data baru
	3	Sistem memeriksa kelengkapan data gejala anak berbakat(*gifted child*) berdasarkan perilaku dan sosial emosi
	4	Sistem menampilkan kelas dan nama kelas target dari data pengujian baru
	5	Kelas dan nama kelas target dari data pengujian baru berhasil ditampilkan
Skenario Alternatif	**Langkah**	**Aksi**
	3.1	Sistem menolak kelengkapan data gejala anak berbakat(*gifted child*) berdasarkan perilaku dan sosial emosi karena nilai pilihan Iya kurang dari 20

Tabel 3.39 Skenario *Use Case* Pengecekan Data Baru (Lanjutan)

Skenario Alternatif	Langkah	Aksi
	3.2	Sistem menampilkan pesan "Nilai Pilihan Iya Kurang Dari 20"
	3.3	Sistem menolak kelengkapan data gejala anak berbakat(*gifted child*) berdasarkan perilaku dan sosial emosi karena kelas tidak terdeteksi
	3.4	Sistem menampilkan pesan "Kelas Tidak Diketahui Untuk Data Pengujian Dengan Id.."
	3.5	Kelas dan nama kelas target dari data pengujian baru gagal ditampilkan

Tabel 3.40 Skenario *Use Case* Pengujian LVQ

Nama *Use Case*	Pengujian LVQ	
***Use Case* Terkait**		
Tujuan	Mengetahui akurasi dan optimasi parameter dan waktu pemrosesan dari metode LVQ	
Kondisi Awal	Pembagian jumlah data pembelajaran dan pengujian sudah tersedia, Maksimum epoh(iterasi), *learning rate* dan *error* minimum belum tersedia	
Kondisi Akhir (jika berhasil)	Persentasi akurasi dan optimasi parameter dan waktu dari metode LVQ berhasil ditampilkan	
Kondisi Akhir (jika gagal)	Persentasi akurasi dan optimasi parameter dan waktu dari metode LVQ gagal ditampilkan	
Aktor	Penguji	
Skenario Utama	**Langkah**	**Aksi**
	1	Penguji memilih menu pengujian
	2	Sistem menampilkan halaman pengujian

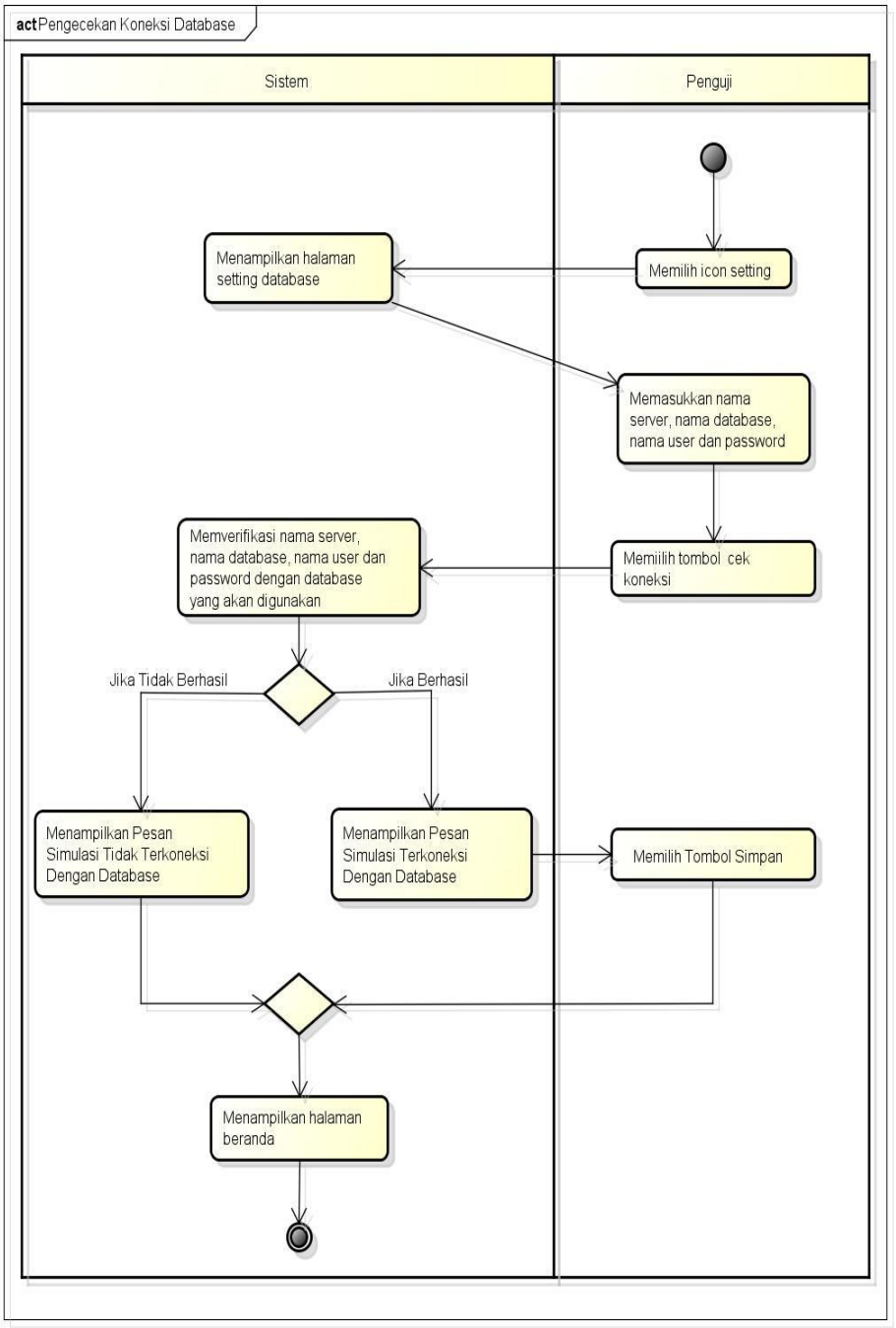
Tabel 3.41 Skenario *Use Case* Pengujian LVQ (Lanjutan)

Skenario Utama	**Langkah**	**Aksi**
	3	Penguji memasukkan detail parameter
	4	Sistem memeriksa kelengkapan dari detail parameter
	5	Sistem melakukan proses *confusion matrix*
	6	Persentasi akurasi dan optimasi parameter dan waktu dari metode LVQ berhasil ditampilkan
Skenario Alternatif	**Langkah**	**Aksi**
	4.1	Sistem menolak kelengkapan dari detail parameter
	4.2	Persentasi akurasi dan optimasi parameter dan waktu dari metode LVQ gagal ditampilkan

3.6.2.2 *Activity Diagram*

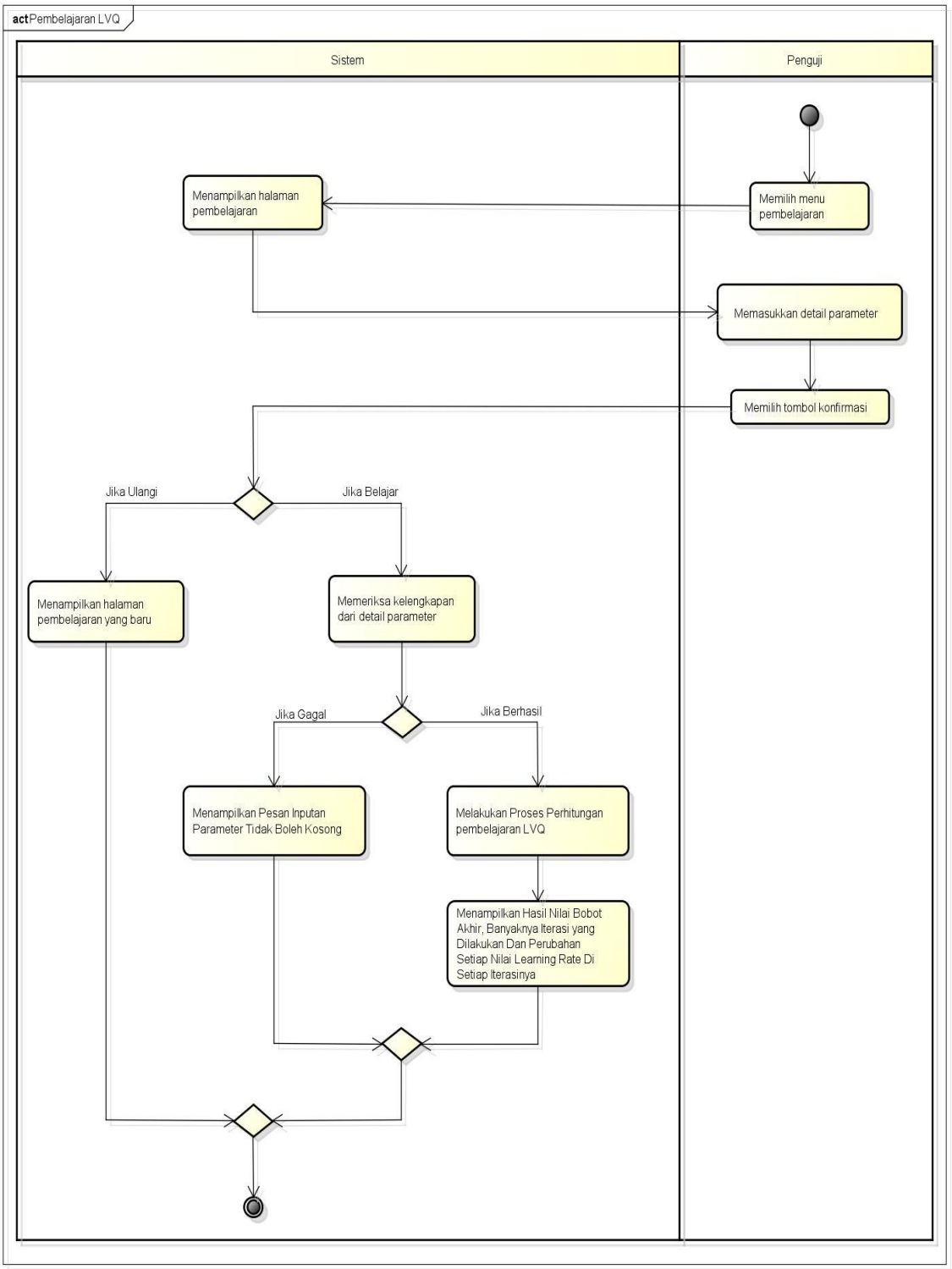
Diagram aktivitas (*Activity Diagram*) berfungsi untuk memodelkan alur kerja sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram aktivitas (*Activity Diagram*) simulasi pendeteksian anak berbakat(*gifted child*) akan dijelaskan pada gambar 3.5 sampai dengan gambar 3.12.

1. *Activity Diagram* Pengecekan Koneksi Database



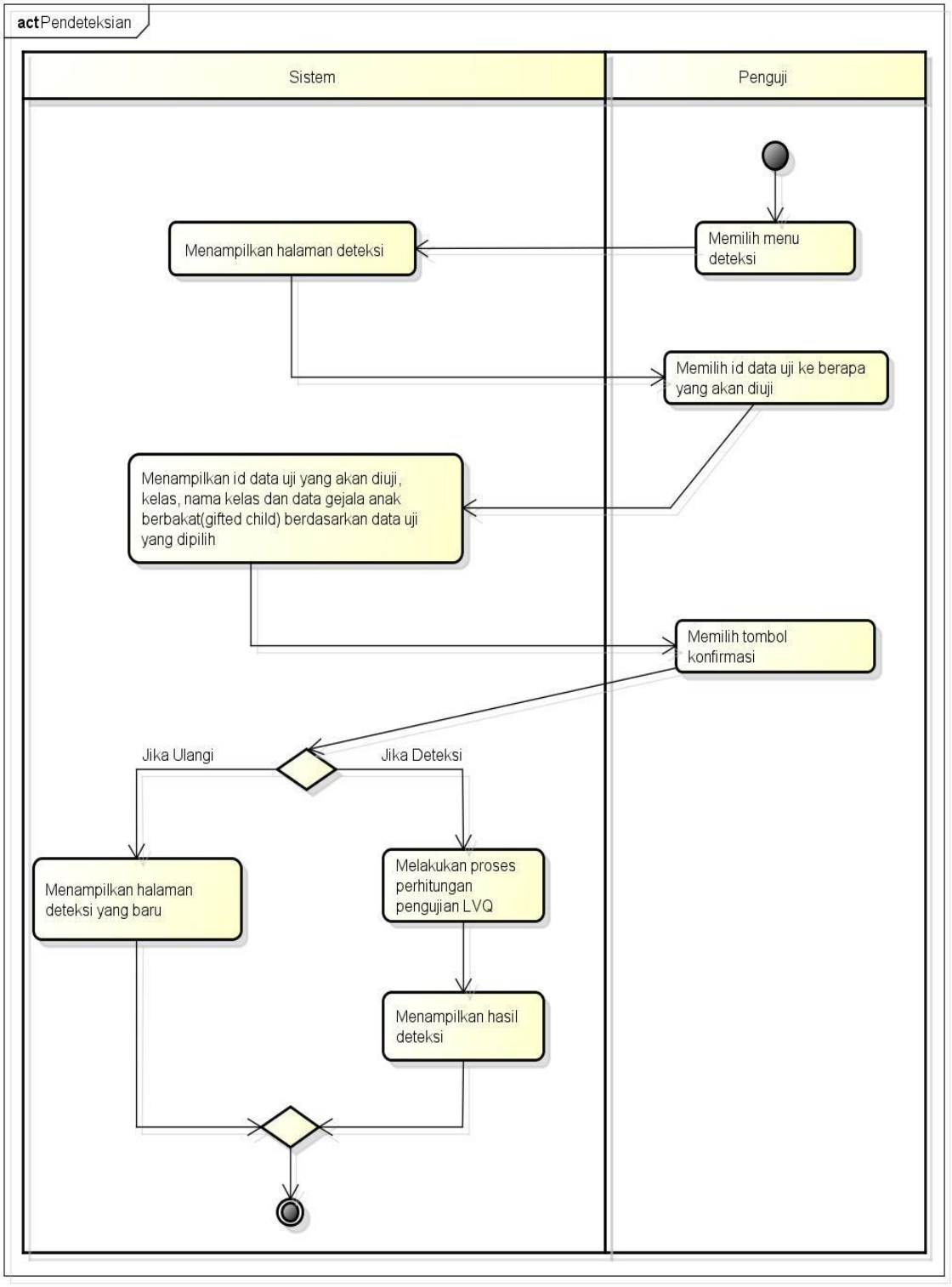
Gambar 3.5 *Activity Diagram* Pengecekan Koneksi Database

2. *Activity Diagram* Pembelajaran LVQ



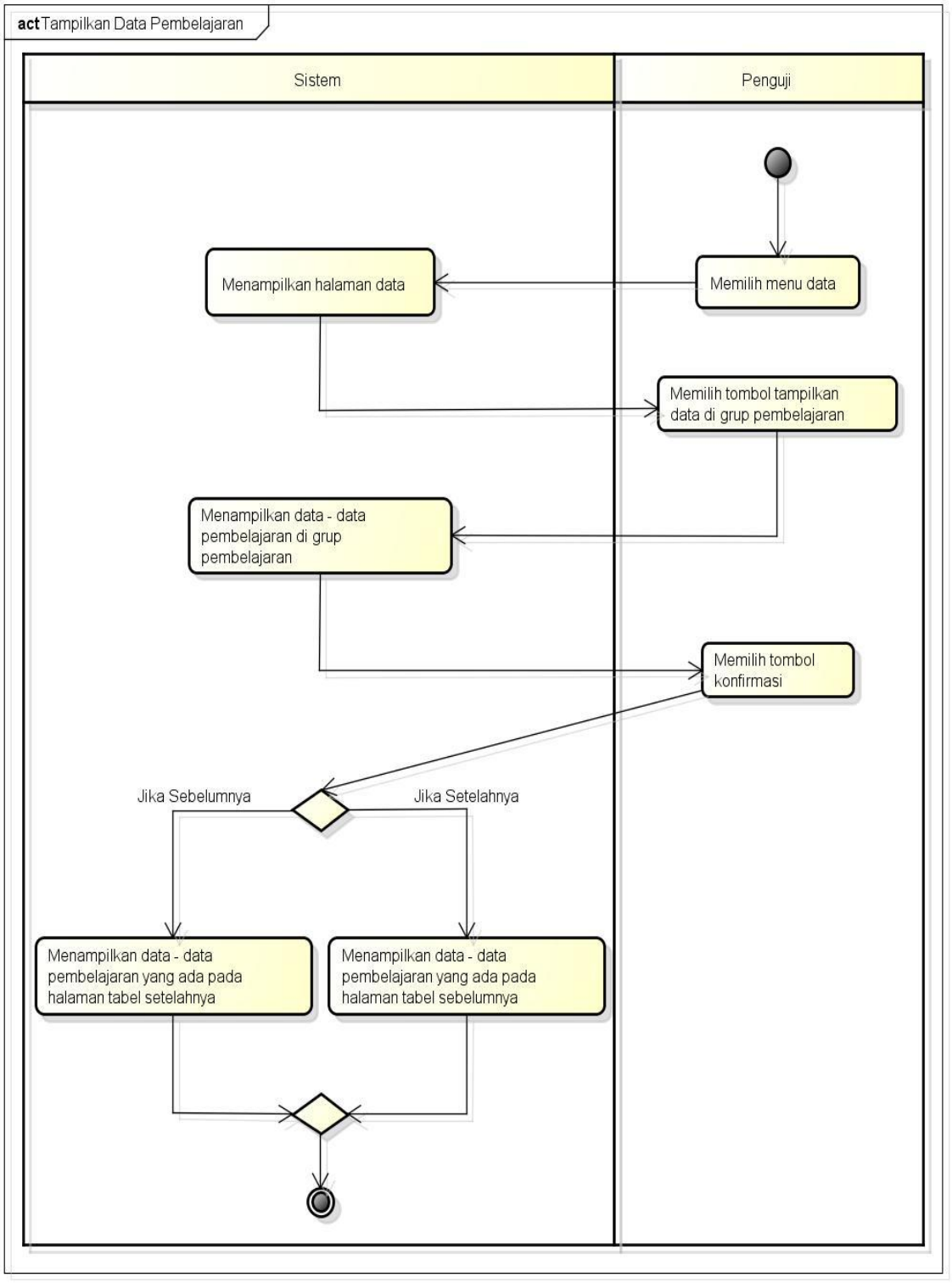
Gambar 3.6 *Activity Diagram* Pembelajaran LVQ

3. *Activity Diagram* Pendeteksian



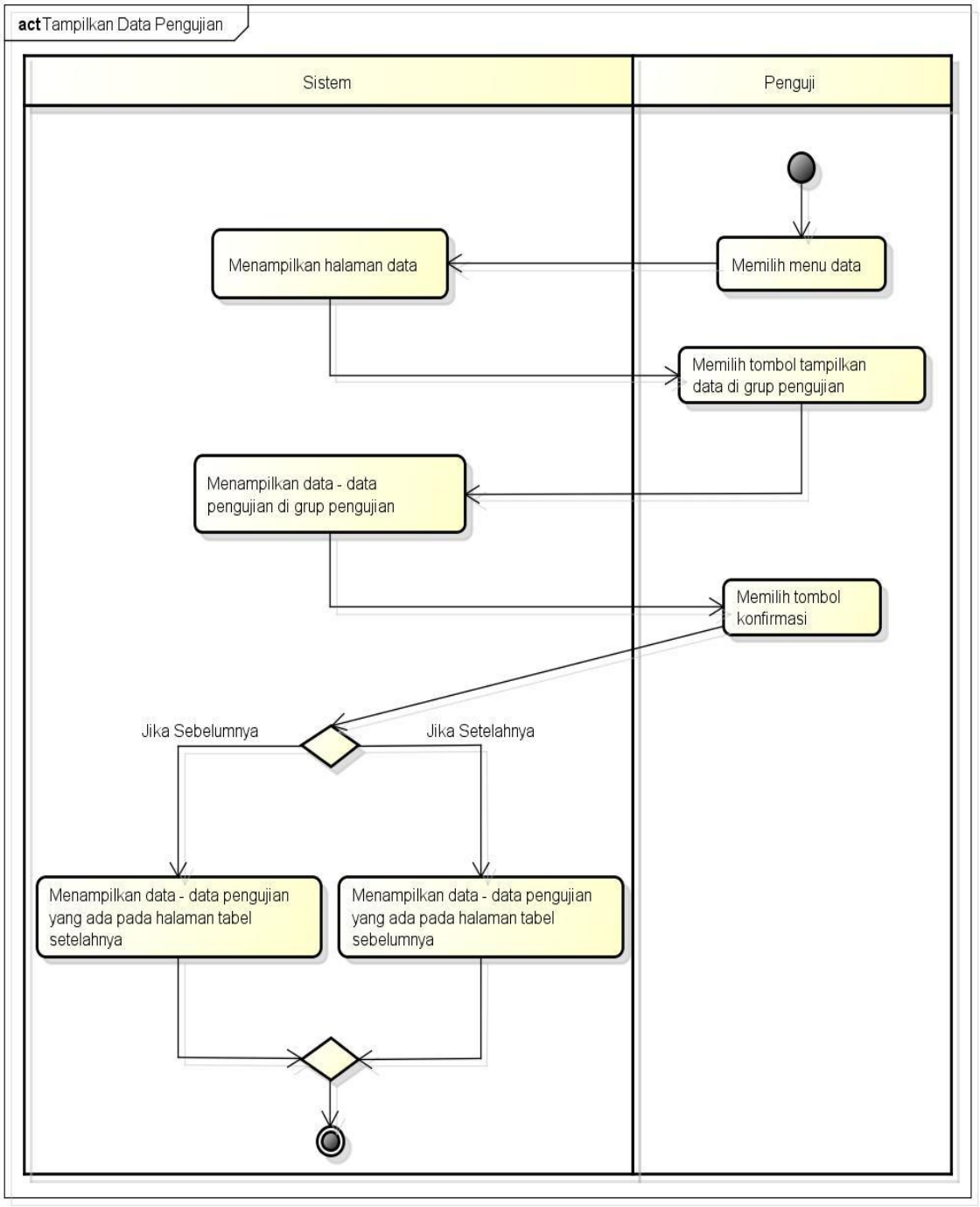
Gambar 3.7 *Activity Diagram* Pendeteksian

4. *Activity Diagram* Tampilkan Data Pembelajaran



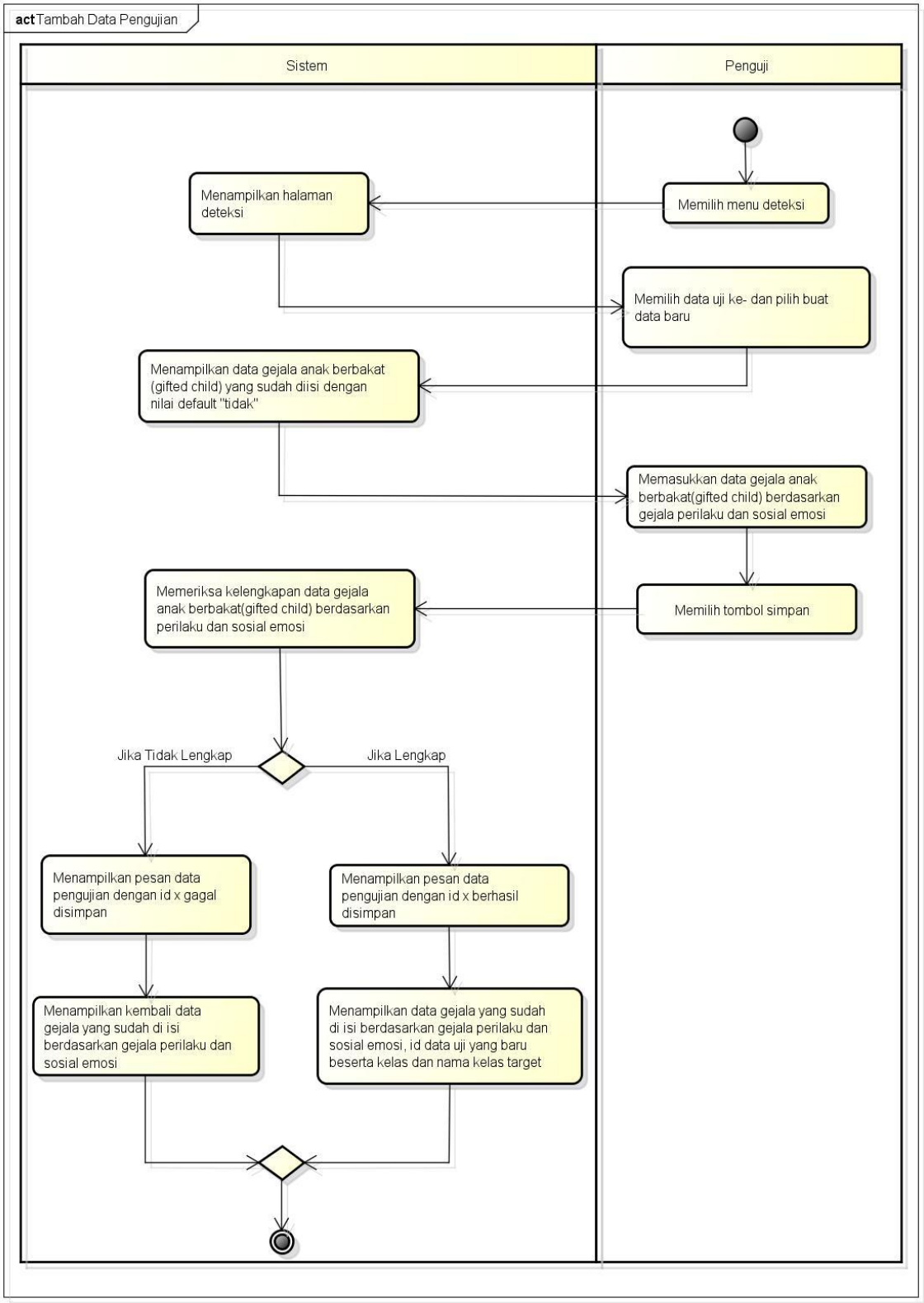
Gambar 3.8 *Activity Diagram* Tampilkan Data Pembelajaran

5. *Activity Diagram* Tampilkan Data Pengujian



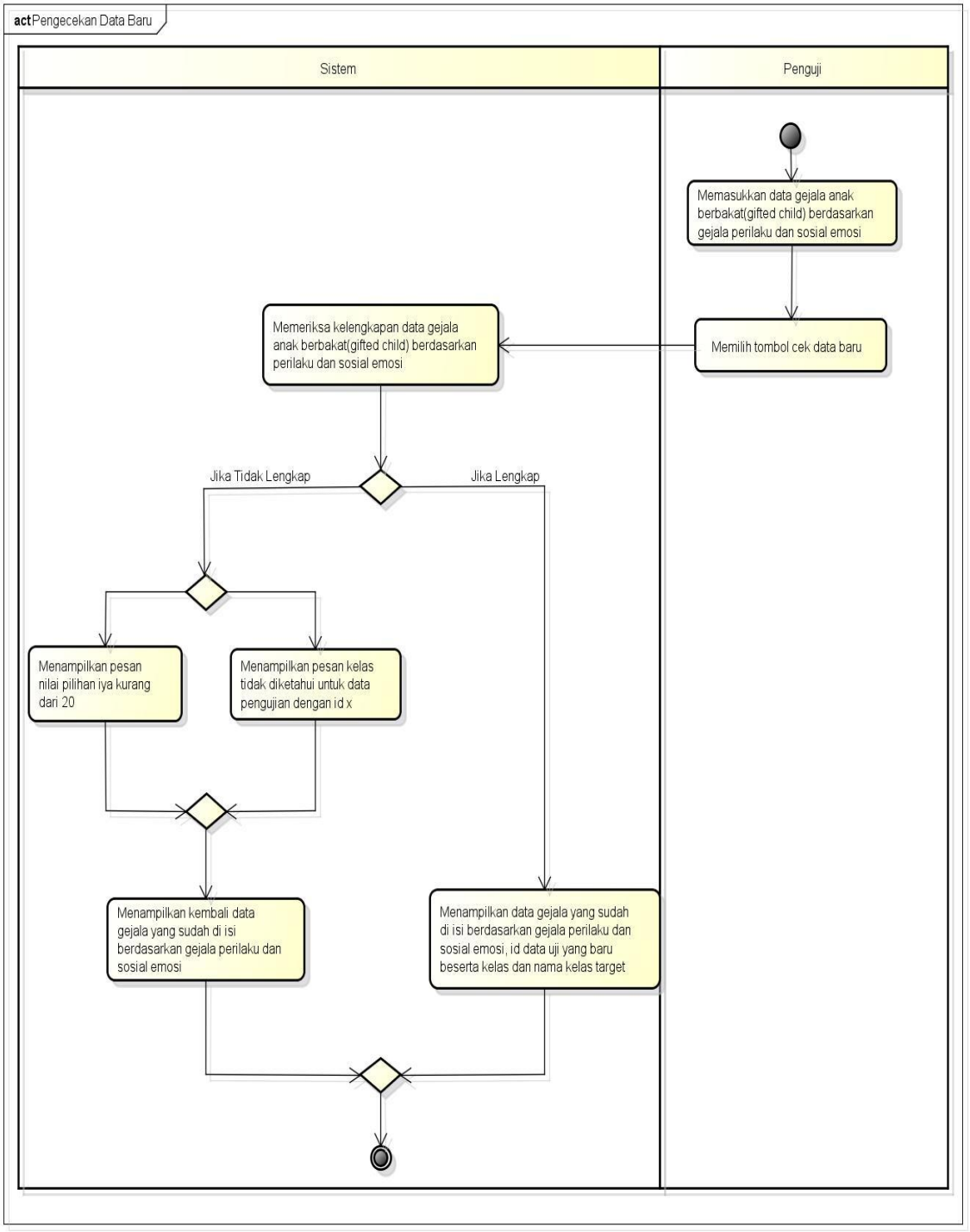
Gambar 3.9 *Activity Diagram* Tampilkan Data Pengujian

6. *Activity Diagram* Tambah Data Pengujian



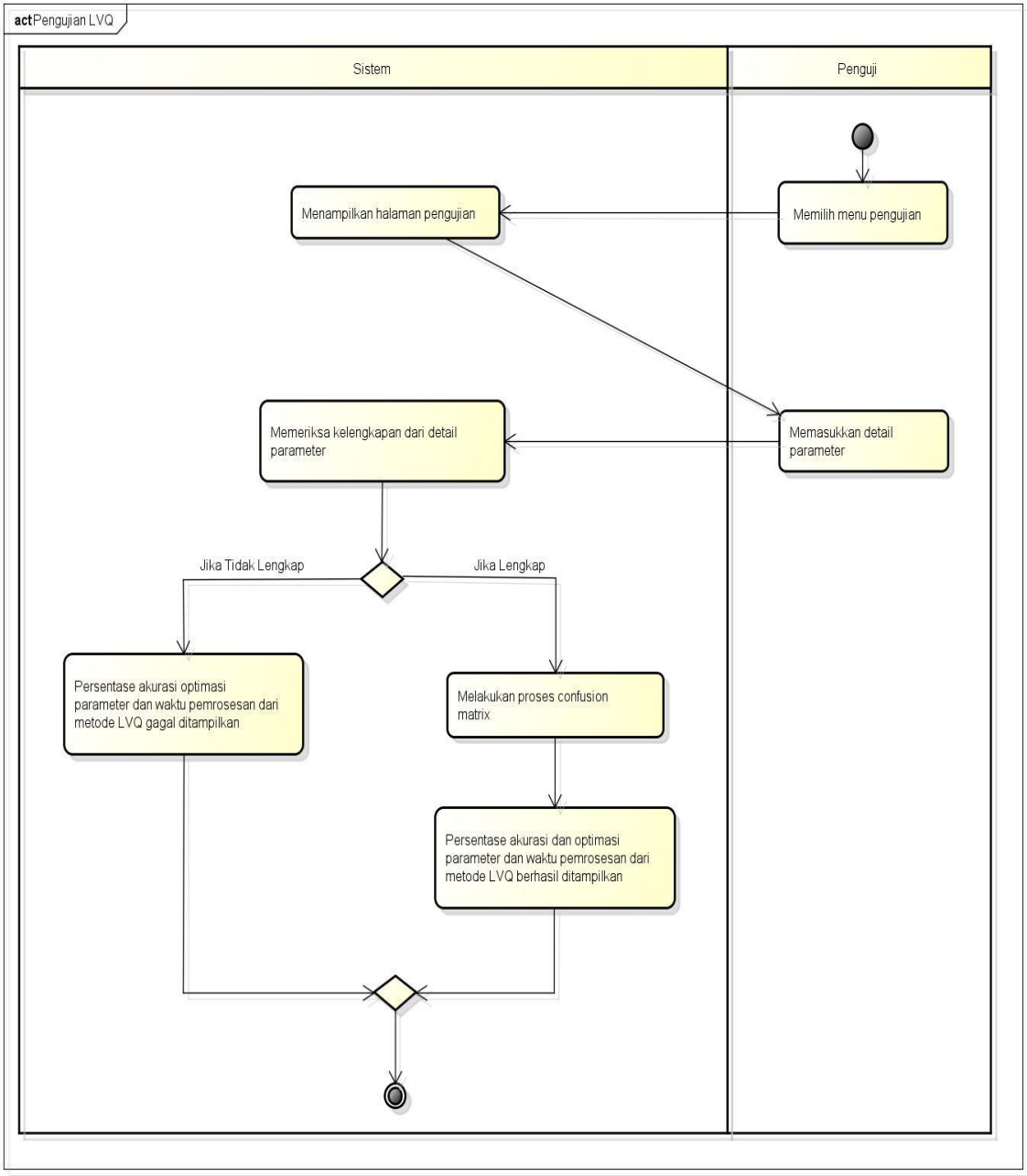
Gambar 3.10 *Activity Diagram* Tambah Data Pengujian

7. *Activity Diagram* Pengecekan Data Baru



Gambar 3.11 *Activity Diagram* Pengecekan Data Baru

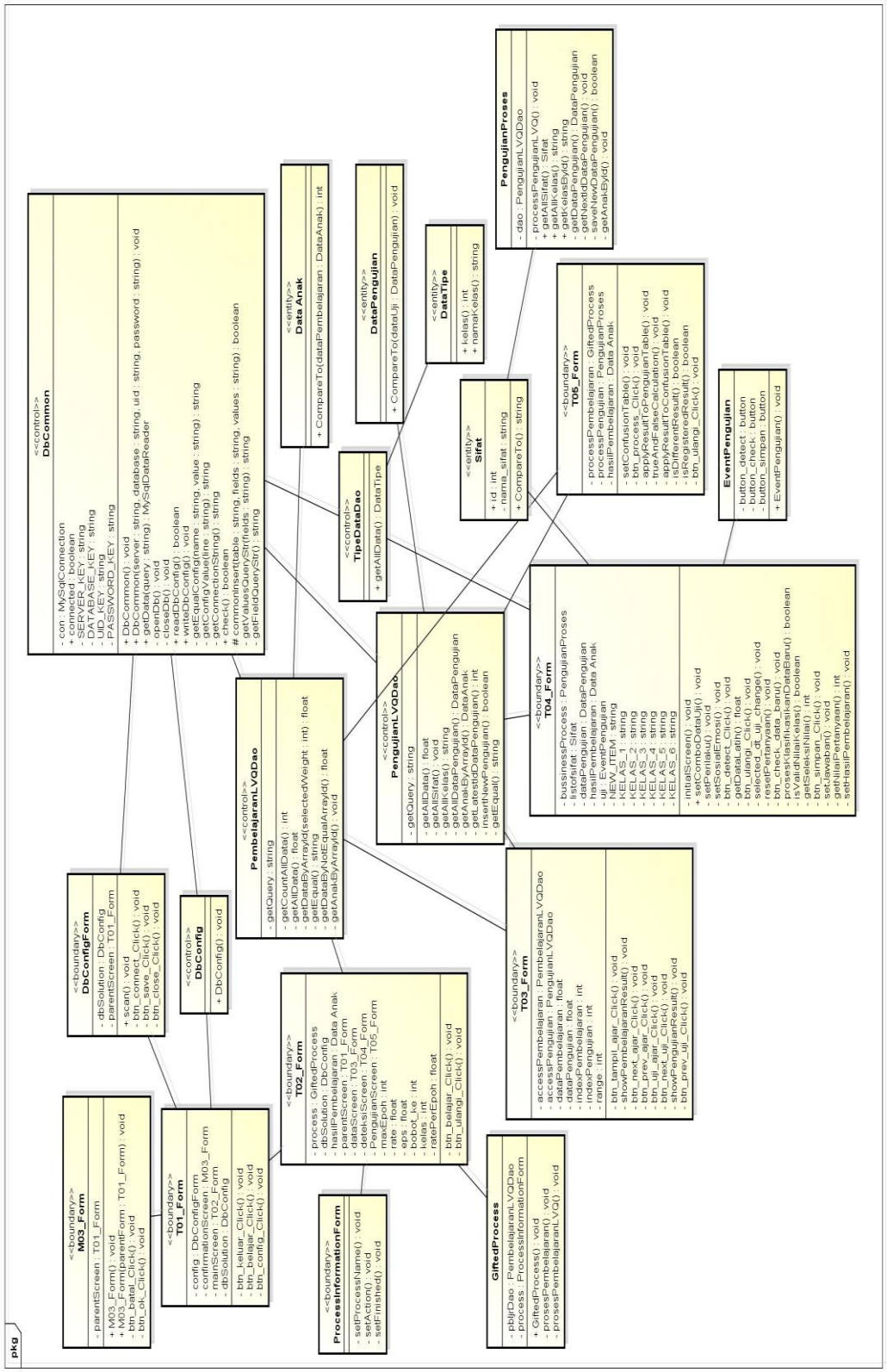
8. *Activity Diagram* Pengujian LVQ



Gambar 3.12 *Activity Diagram* Pengujian LVQ

3.6.2.3 *Class Diagram*

Class Diagram berfungsi untuk memvisualisasikan struktur kelas – kelas dari suatu aplikasi dan merupakan tipe diagram yang paling sering ditemukan dalam proses pemodelan aplikasi berbasis *object-oriented*. *Class Diagram* simulasi pendeteksian anak berbakat(*gifted child*) dijelaskan pada gambar 3.13.

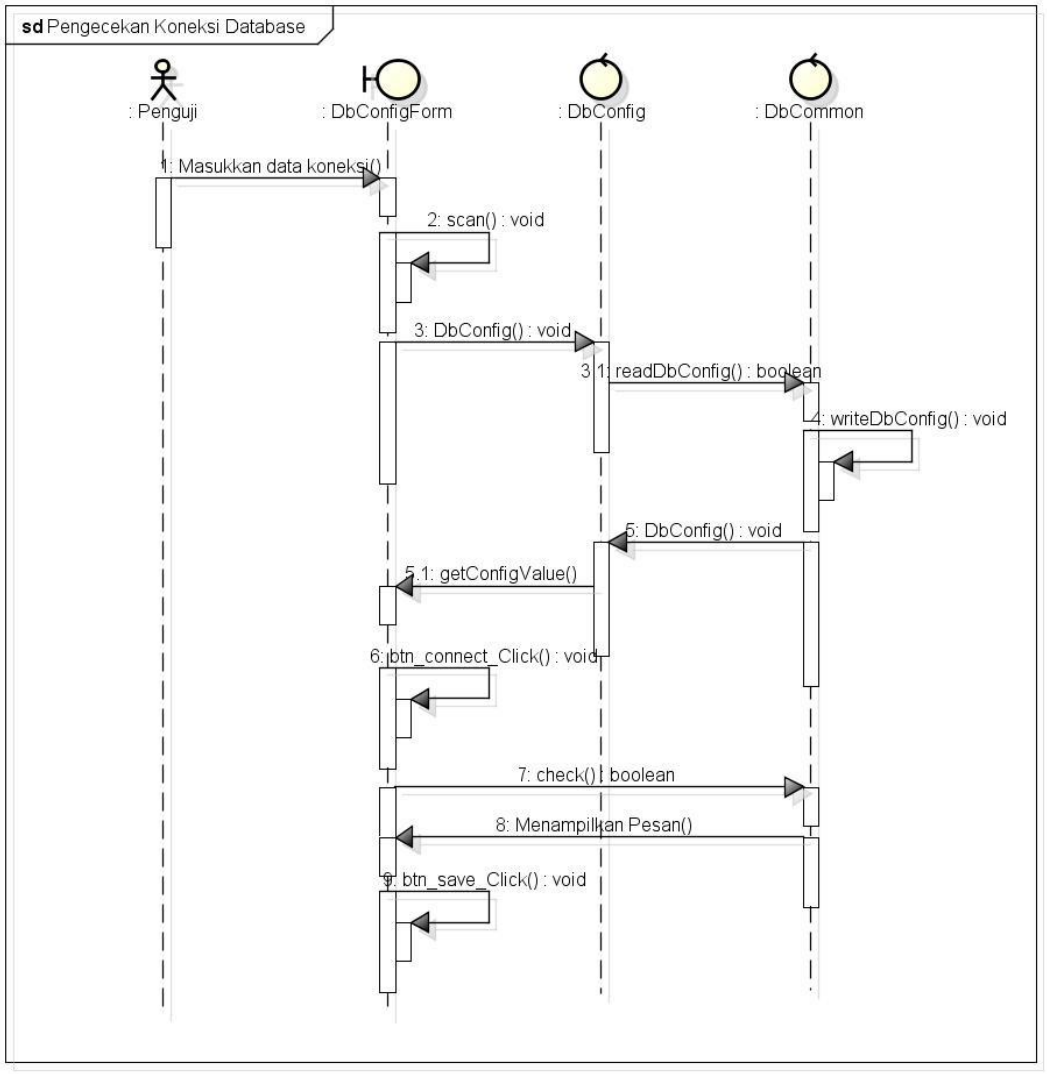


Gambar 3.13 *Class Diagram* Simulasi Pendeteksian Anak Berbakat(*Gifted Child*)

3.6.2.4 *Sequence Diagram*

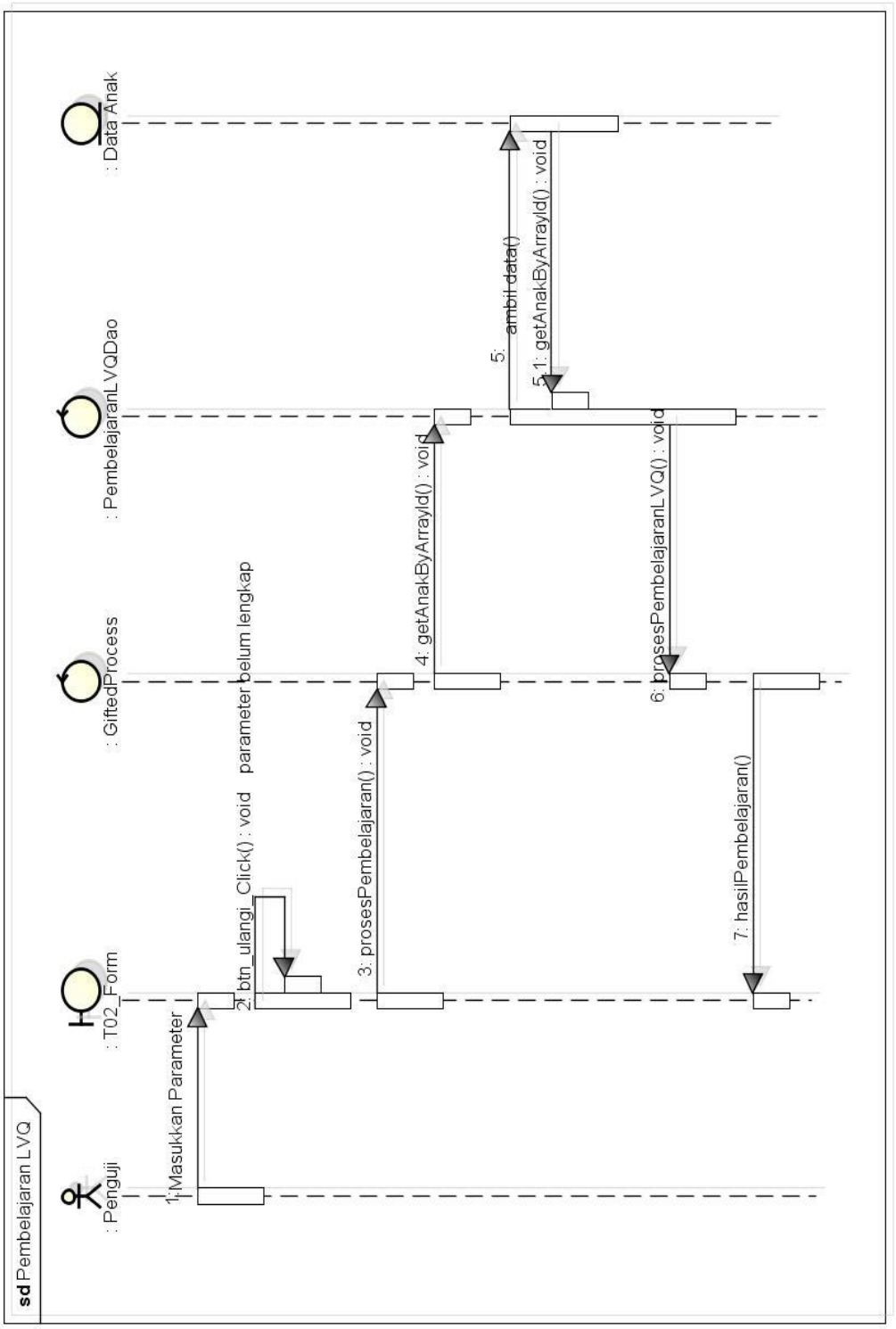
Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. *Sequence diagram* simulasi pendeteksian anak berbakat(*gifted child*) akan dijelaskan pada gambar 3.14 sampai dengan gambar 3.21.

1. ***Sequence Diagram* Pengecekan Koneksi Database**



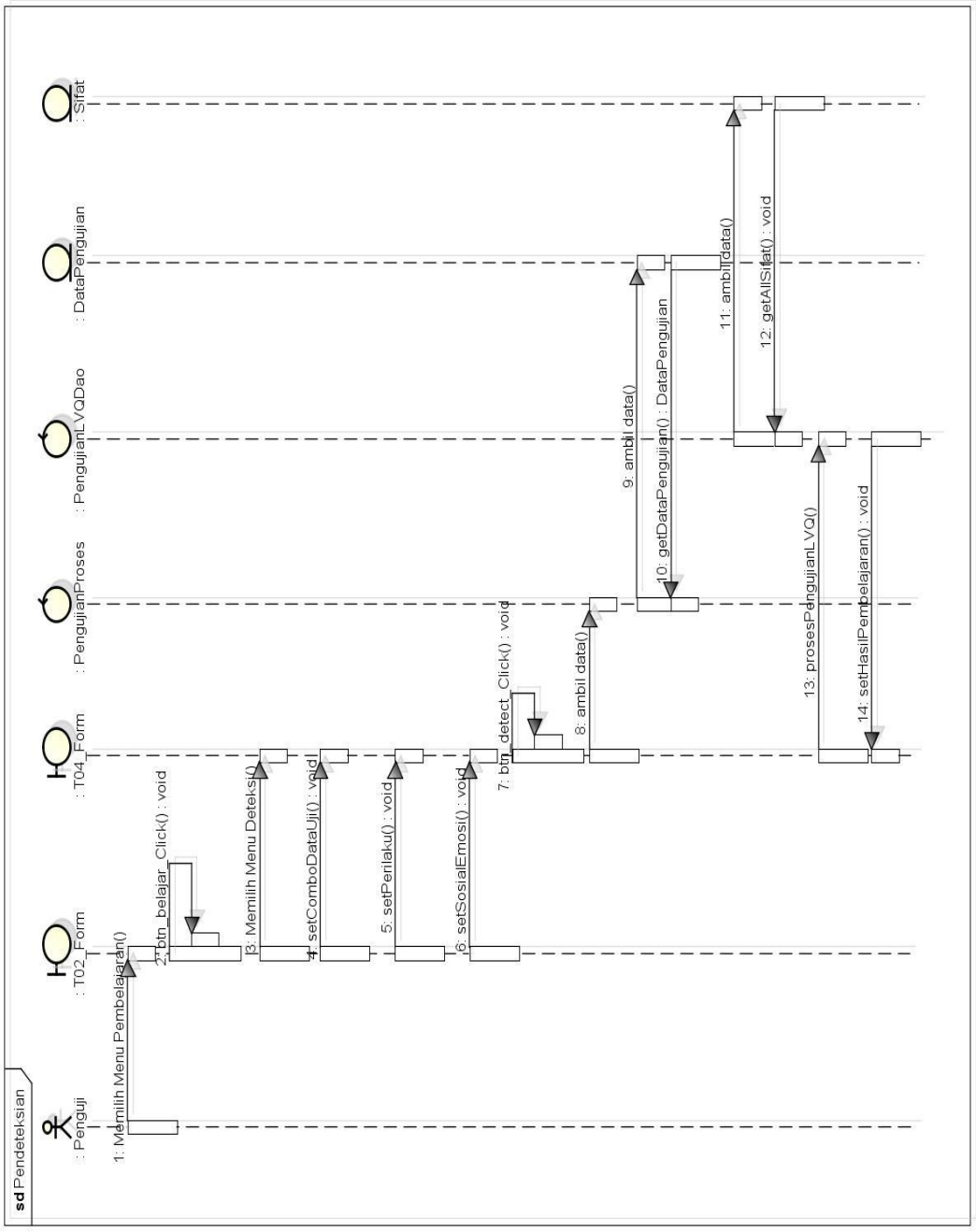
Gambar 3.14 *Sequence Diagram* Pengecekan Koneksi Database

2. *Sequence Diagram* Pembelajaran LVQ



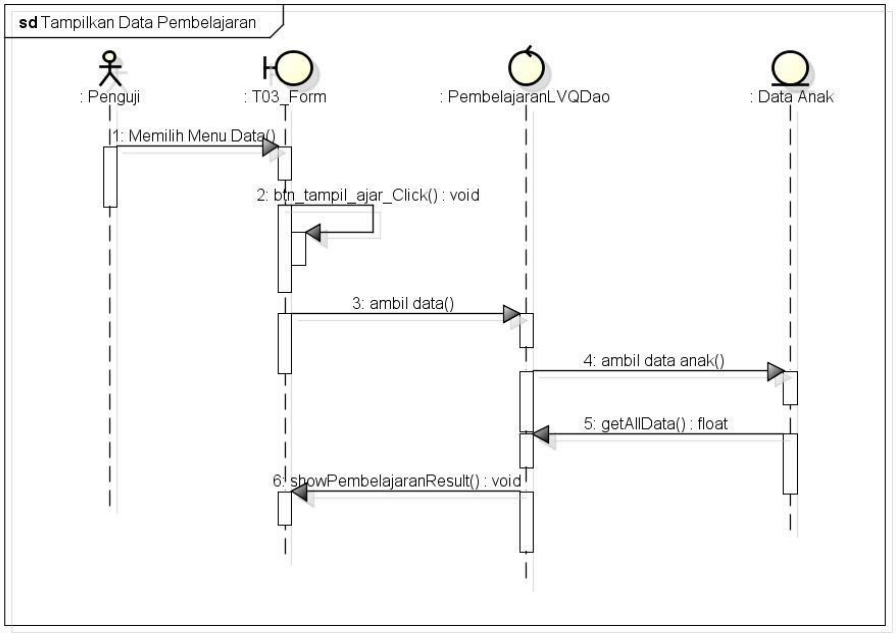
Gambar 3.15 *Sequence Diagram* Pembelajaran LVQ

3. *Sequence Diagram* Pendeteksian



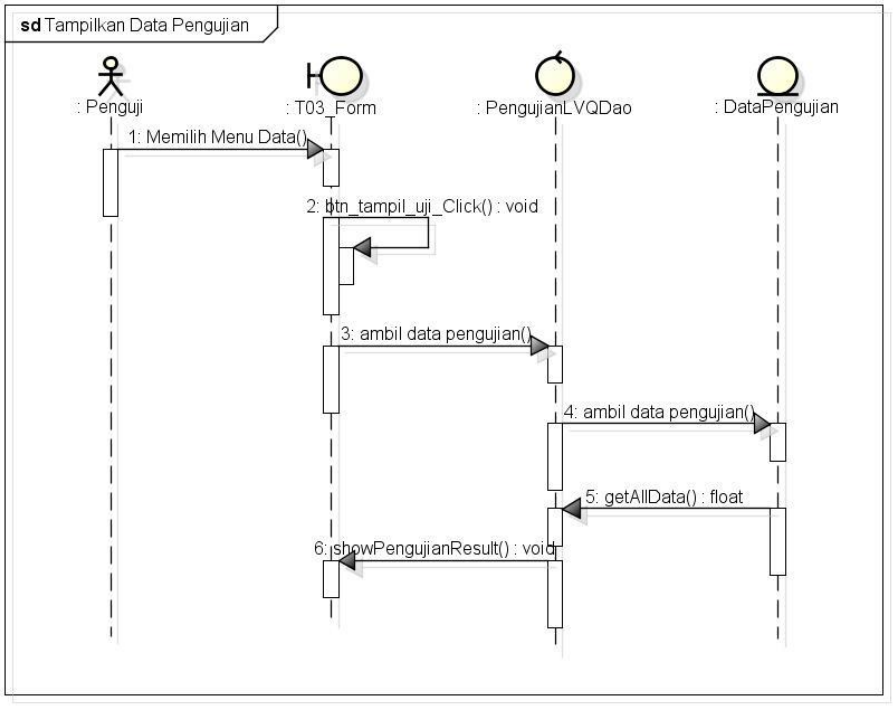
Gambar 3.16 *Sequence Diagram* Pendeteksian

4. *Sequence Diagram* Tampilkan Data Pembelajaran



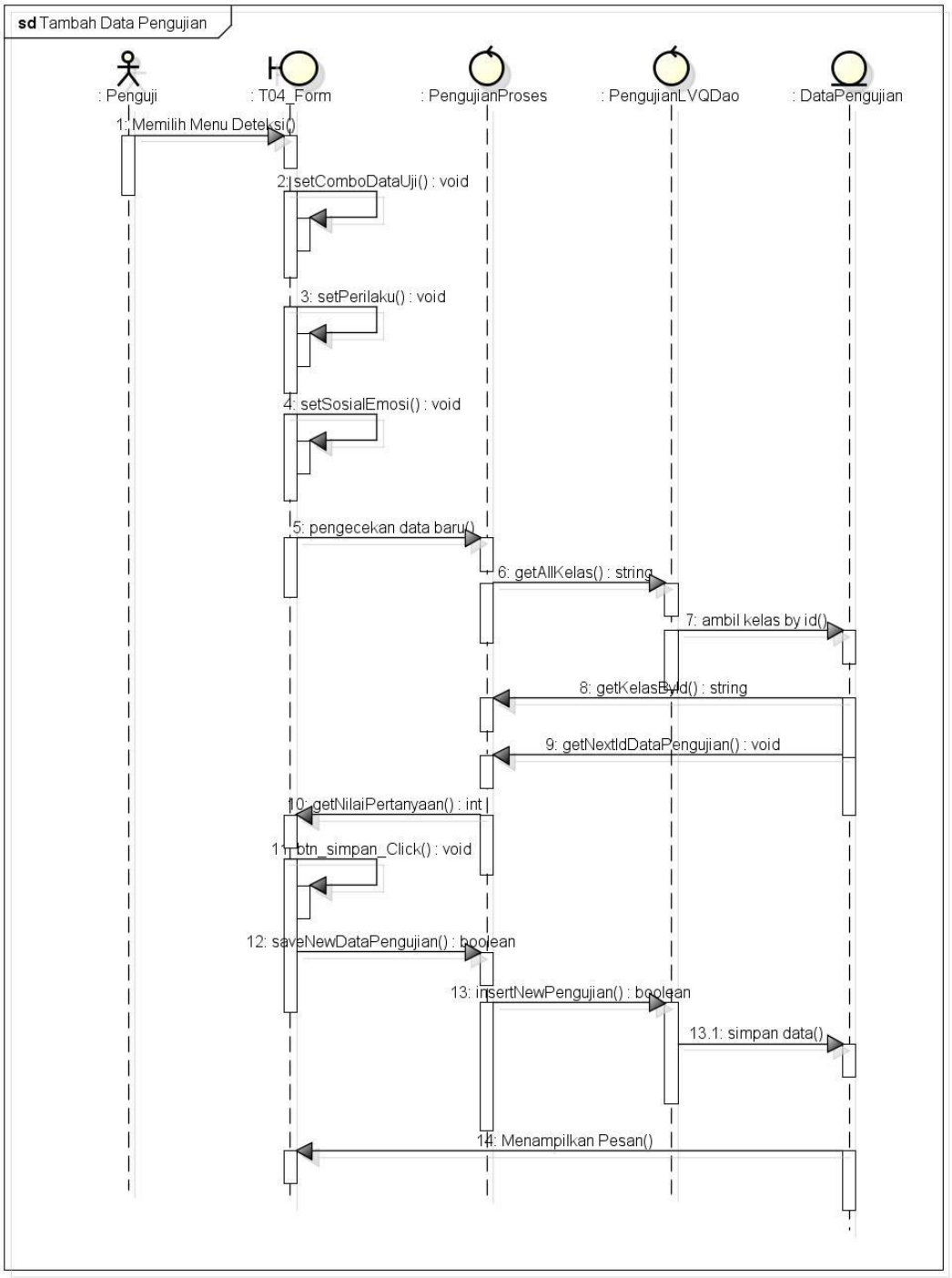
Gambar 3.17 *Sequence Diagram* Tampilkan Data Pembelajaran

5. *Sequence Diagram* Tampilkan Data Pengujian



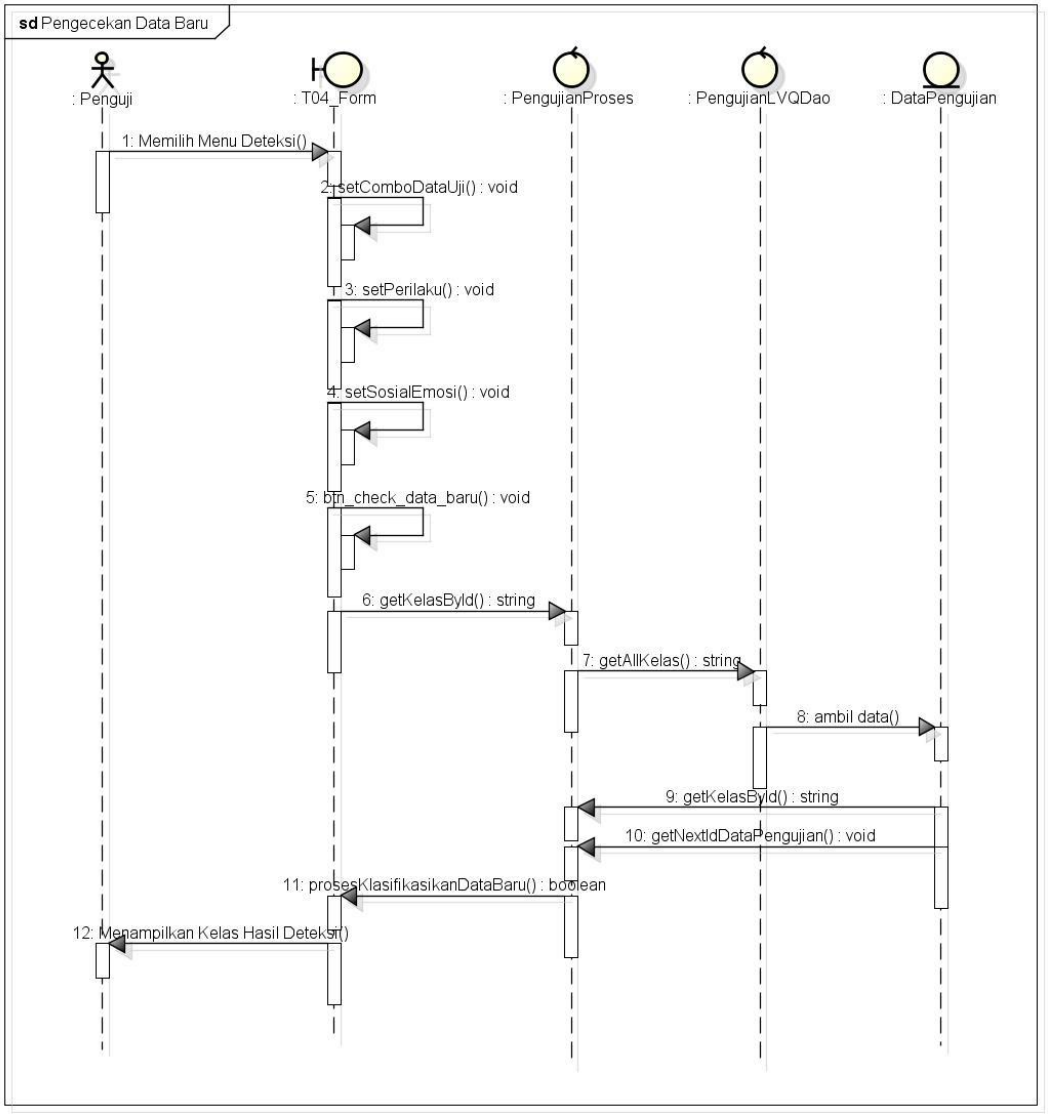
Gambar 3.18 *Sequence Diagram* Tampilkan Data Pengujian

6. *Sequence Diagram* Tambah Data Pengujian



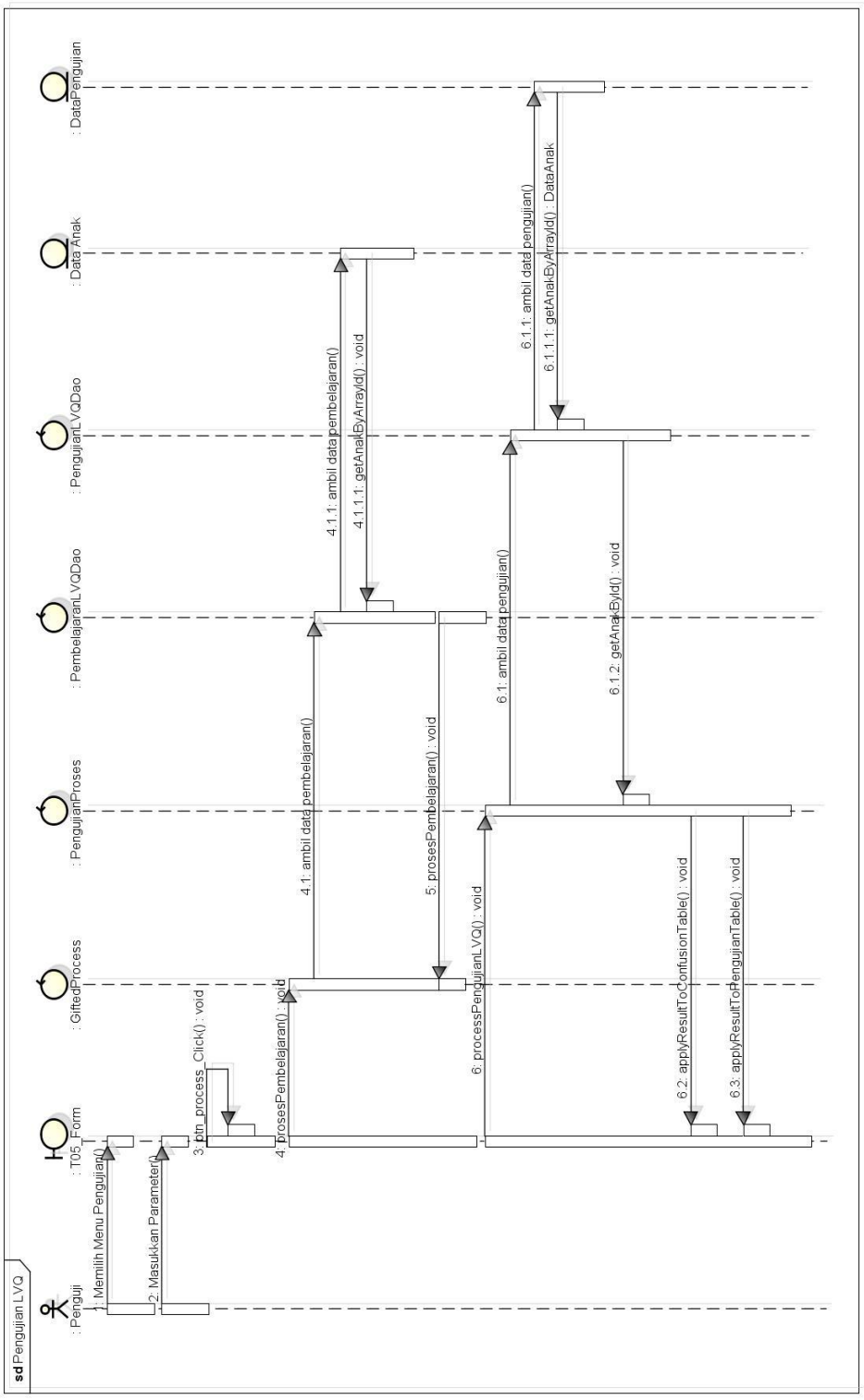
Gambar 3.19 *Sequence Diagram* Tambah Data Pengujian

7. *Sequence Diagram* Pengecekan Data Baru



Gambar 3.20 *Sequence Diagram* Pengecekan Data Baru

8. *Sequence Diagram* Pengujian LVQ

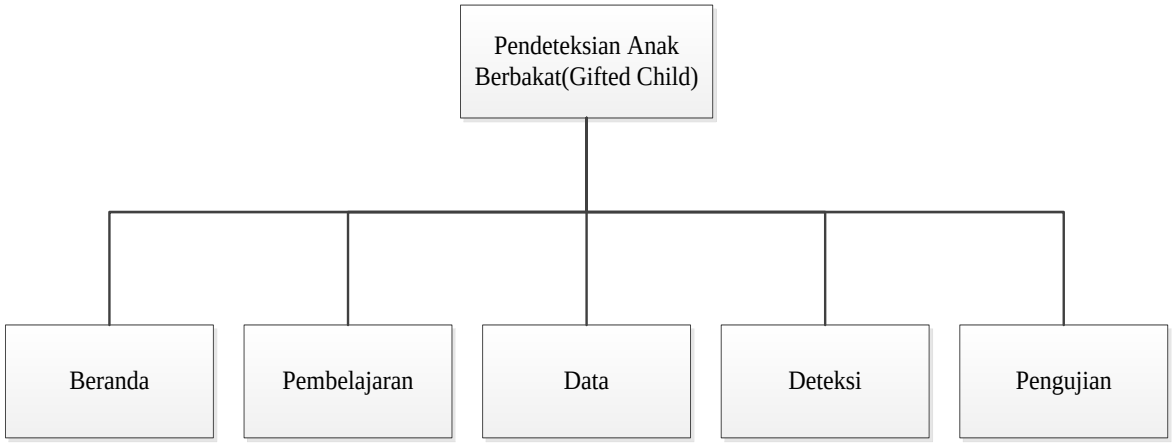


Gambar 3.21 *Sequence Diagram* Pengujian LVQ

Perancangan adalah penggambaran dan perencanaan dari beberapa elemen sistem berdasarkan analisis sistem yang telah dilakukan sebelumnya. Tahapan perancangan dari simulasi pendeteksian anak berbakat(*gifted child*) meliputi perancangan struktur menu, perancangan antarmuka dan perancangan jaringan semantik.

3.7 Perancangan Struktur Menu

Perancangan struktur menu merupakan gambaran untuk alur menu perancangan aplikasi yang akan dibangun. Perancangan struktur menu dapat memberikan gambaran mengenai struktur program dari aplikasi kita. Menu - menu yang tersedia akan aktif sesuai dengan prosedurnya. Perancangan struktur menu simulasi pendeteksian anak berbakat(*gifted child*) akan dijelaskan pada gambar 3.22.

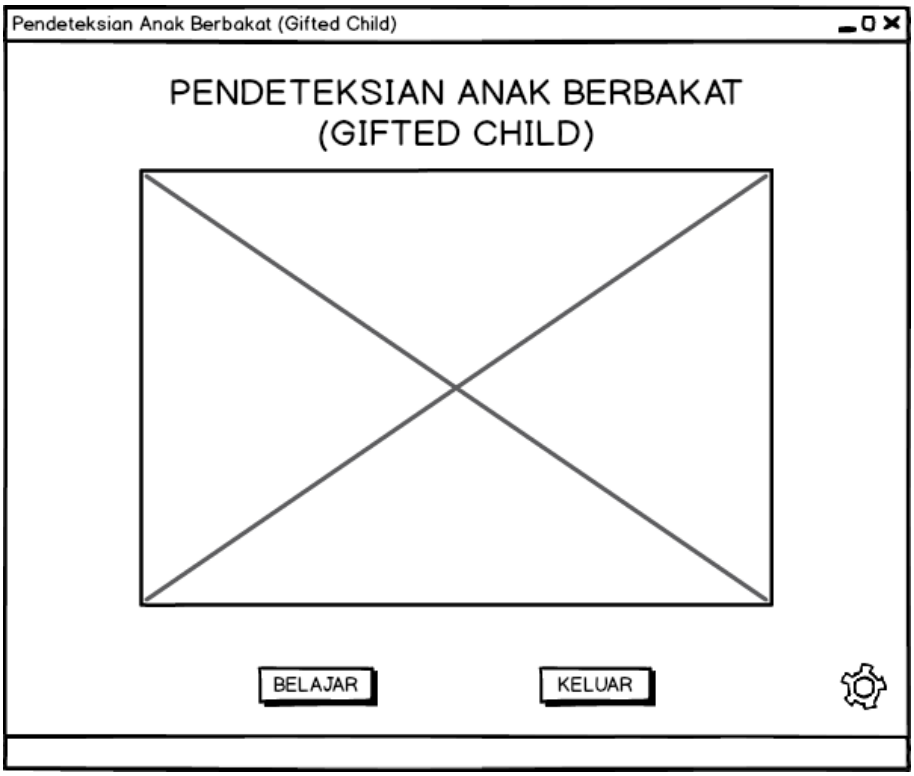


Gambar 3.22 Struktur Menu Simulasi Pendeteksian Anak Berbakat(*Gifted Child*)

3.8 Perancangan Antarmuka

Perancangan antarmuka merupakan gambaran dari rancangan tampilan aplikasi yang akan dibangun. Perancangan antarmuka simulasi pendeteksian anak berbakat(*gifted child*) akan dijelaskan pada gambar 3.23 sampai dengan gambar 3.37.

1. **T01**

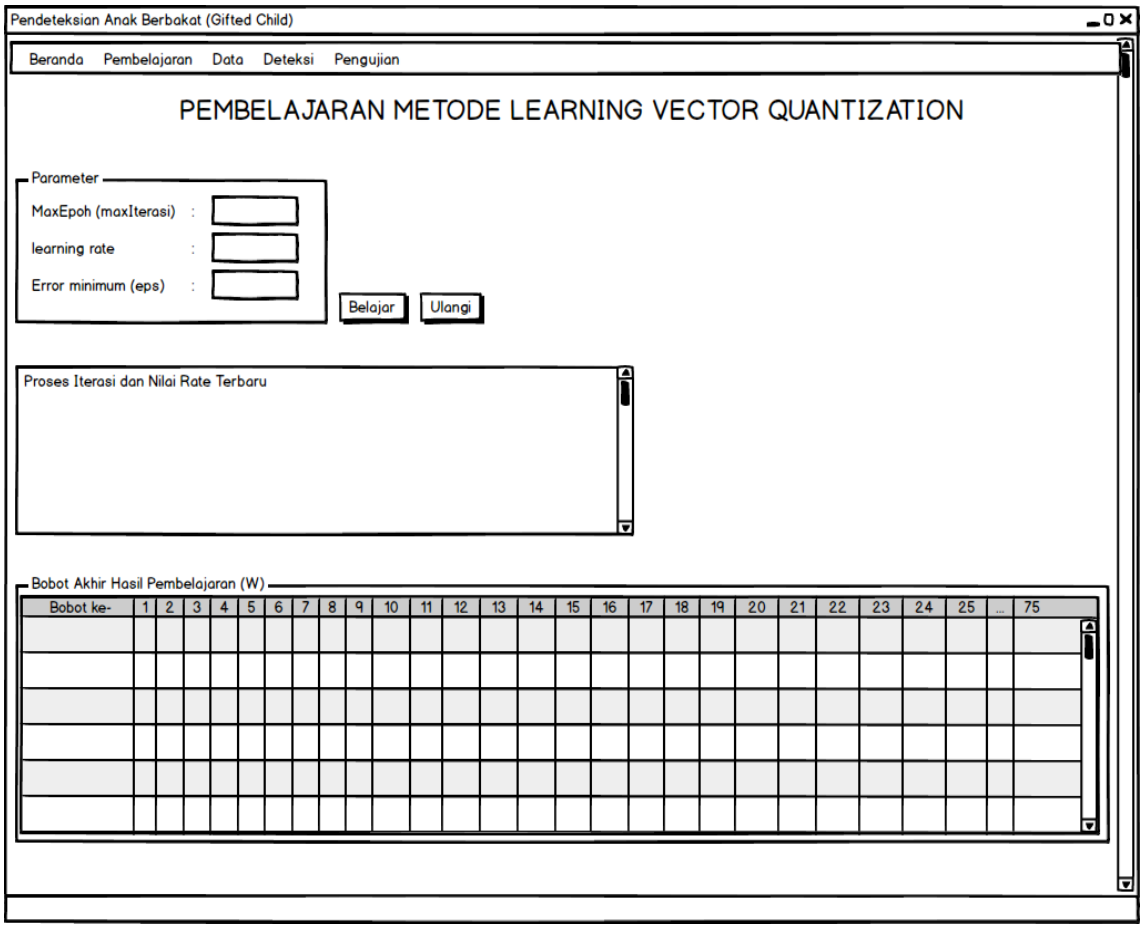


Gambar 3.23 Perancangan Antarmuka Tampilan Awal

Keterangan :

1	T01	: Tampilan awal simulasi
2	Klik tombol Belajar	: Menuju ke T02 jika sudah terkoneksi dengan database atau menuju ke M01 jika belum terkoneksi dengan database
3	Klik tombol Keluar	: Menuju ke M02
4	Klik *icon setting*	: Menuju ke T06

2. T02

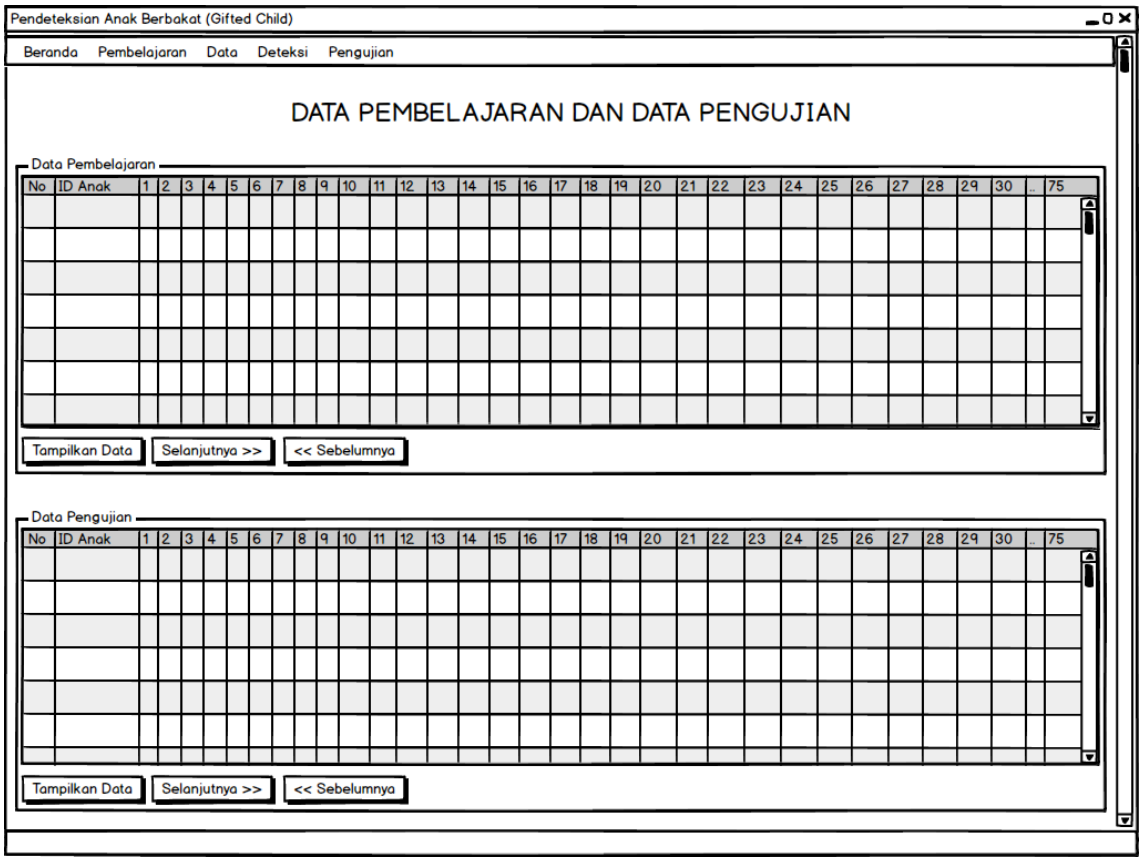


Gambar 3.24 Perancangan Antarmuka Form Pembelajaran

Keterangan :

1	T02	: Halaman untuk melakukan pembelajaran
2	Menu Beranda	: Menuju ke T01
3	Menu Pembelajaran	: Menuju ke T02
4	Menu Data	: Menuju ke T03
5	Menu Deteksi	: Menuju ke T06
6	Menu Pengujian	: Menuju ke T05
7	MaxEpoh	: Maksimal iterasi atau perulangan yang akan dilakukan
8	*Learning rate*	: Mengatur laju pembelajaran
9	*Error* minimum	: Menentukan keakuratan data
10	Tombol Belajar	: Untuk melakukan proses pembelajaran dan jika parameter ada yang tidak di isi maka akan tampil pesan M03
11	Tombol Ulangi	: Untuk membersihkan form
12	Textbox	: Menampilkan banyaknya epoh atau iterasi dan nilai *learning rate*
13	Data Grid Bobot akhir	: Menampilkan bobot akhir hasil pembelajaran

3. T03

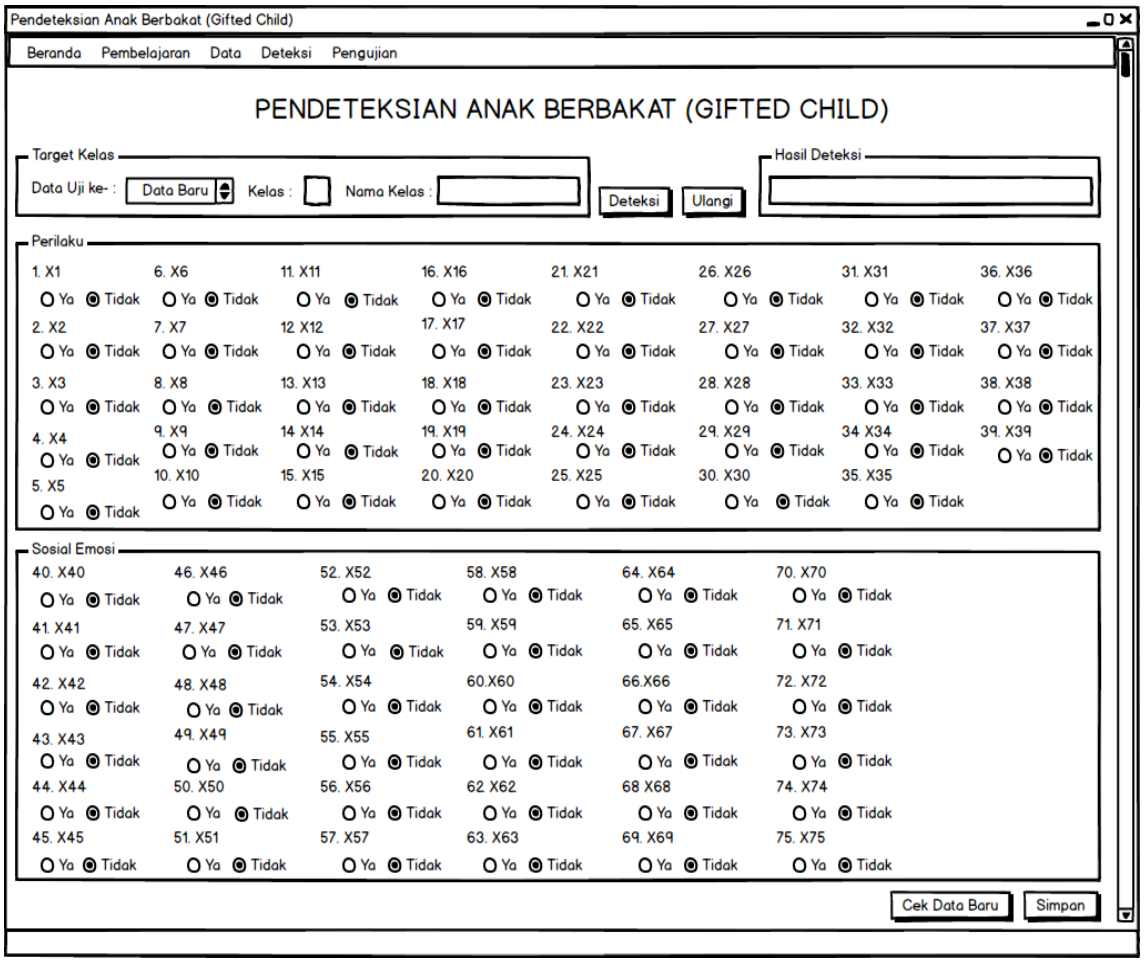


Gambar 3.25 Perancangan Antarmuka Form Data

Keterangan :

1	T03	: Halaman untuk menampilkan data pembelajaran dan data pengujian
2	Menu Beranda	: Menuju ke T01
3	Menu Pembelajaran	: Menuju ke T02
4	Menu Data	: Menuju ke T03
5	Menu Deteksi	: Menuju ke T04
6	Menu Pengujian	: Menuju ke T05
7	Data Grid Pembelajaran	: Menampilkan data pembelajaran
8	Data Grid Pengujian	: Menampilkan data pengujian
9	Tombol Tampilkan data	: Untuk menampilkan data pembelajaran atau data pengujian
10	Tombol Selanjutnya	: Untuk menampilkan data pada halaman berikutnya
11	Tombol Sebelumnya	:Untuk menampilkan data pada halaman sebelumnya

4. T04



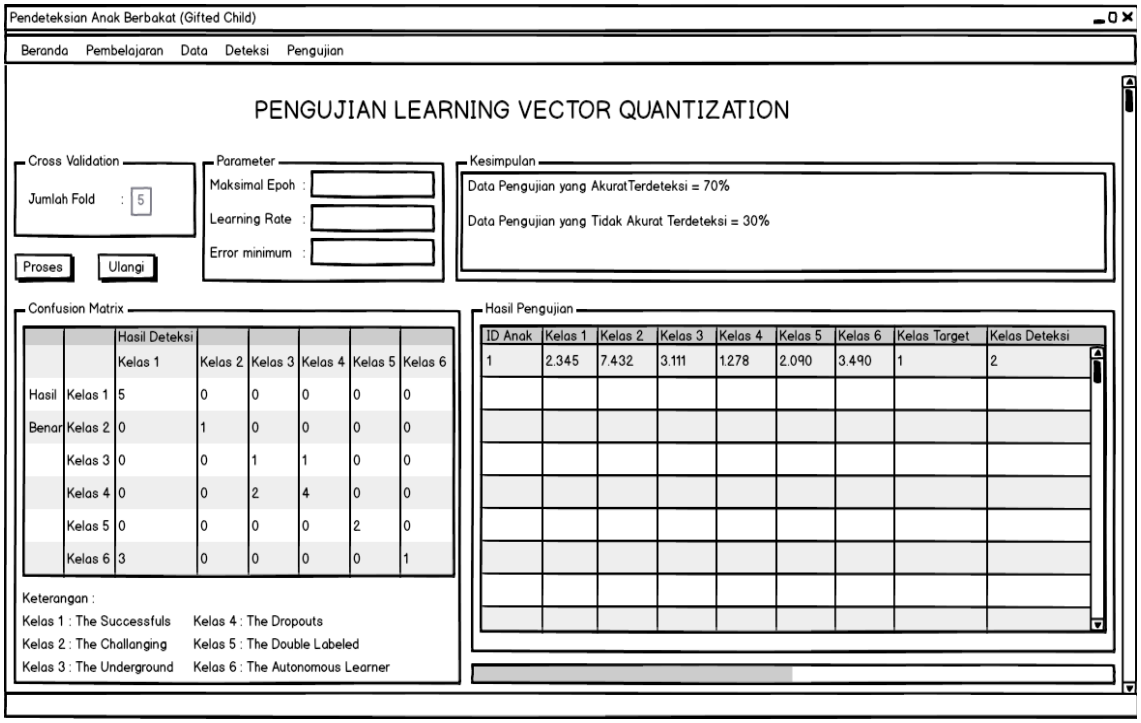
Gambar 3.26 Perancangan Antarmuka Form Deteksi

Keterangan :

1	T04	: Halaman untuk melakukan deteksi
2	Menu Beranda	: Menuju ke T01
3	Menu Pembelajaran	: Menuju ke T02
4	Menu Data	: Menuju ke T03
5	Menu Deteksi	: Menuju ke T04
6	Menu Pengujian	: Menuju ke T05
7	Data Uji ke-	: Menunjukkan data ke berapa yang akan diuji
8	Data Baru	: Untuk menambah data pengujian baru
9	Kelas	: Menunjukkan kode target kelas dari data yang diuji
10	Nama Kelas	: Menampilkan nama target kelas dari data yang diuji
11	Hasil Deteksi	: Menampilkan target kelas setelah di lakukan proses deteksi
12	Tombol Deteksi	: Untuk melakukan proses deteksi

13	Tombol Ulangi	:	Untuk membersihkan form
14	Grup Perilaku	:	Data inputan berdasarkan perilaku
15	Grup Sosial Emosi	:	Data inputan berdasarkan sosial emosi
16	Tombol Cek Data Baru	:	Untuk mengetahui kelas dan nama kelas target, jika berhasil maka akan muncul pesan M05 sedangkan jika gagal maka akan muncul pesan M04 atau M06.
17	Tombol Simpan Data Baru	:	Untuk menyimpan data pengujian baru ke database, jika berhasil maka akan muncul pesan M07.

5. T05



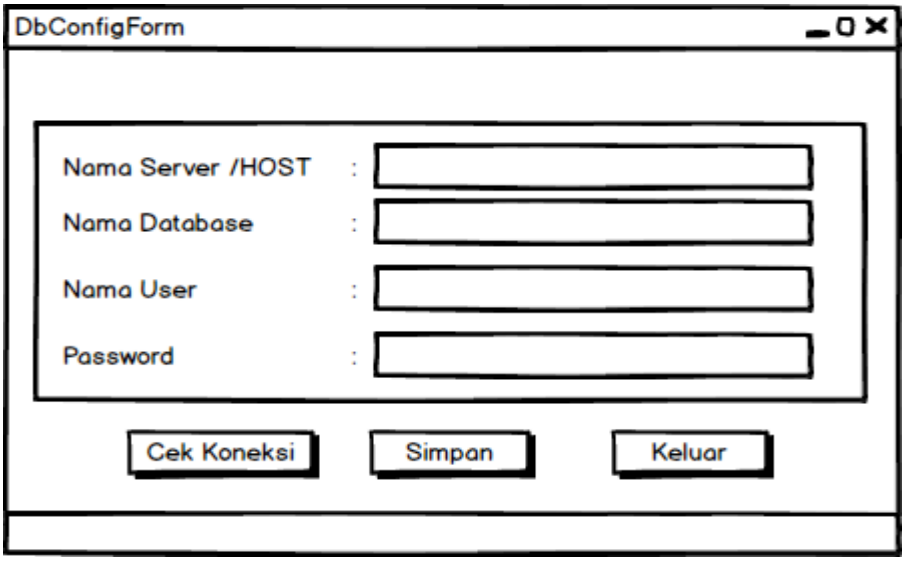
Gambar 3.27 Perancangan Antarmuka Form Pengujian

Keterangan :

1	T05	: Halaman untuk melakukan proses pengujian
2	Menu Beranda	: Menuju ke T01
3	Menu Pembelajaran	: Menuju ke T02
4	Menu Data	: Menuju ke T03
5	Menu Deteksi	: Menuju ke T04
6	Menu Pengujian	: Menuju ke T05
7	Jumlah *Fold*	: Jumlah pembagian data yang akan dilakukan
8	MaxEpoh	: Maksimal iterasi yang akan dilakukan
9	*Learning rate*	: Mengatur laju pembelajaran
10	*Error* minimum	: Menentukan batasan minimum *error*

11	Grup Kesimpulan	: Menampilkan hasil kesimpulan dari pengujian
12	Tombol Proses	: Melakukan proses *confusion matrix*
13	Tombol Ulangi	: Membersihkan form
14	Data Grid *Confusion Matrix*	: Menampilkan hasil proses dari *confusion matrix*
15	Data Grid Hasil Pengujian	: Menampilkan hasil perhitungan pengujian

6. T06



Gambar 3.28 Perancangan Antarmuka Form *Setting Database*

Keterangan :

1	Tombol Cek Koneksi	: Menampilkan pesan M08 jika koneksi database berhasil atau M09 jika koneksi database gagal
2	Tombol Simpan	: Menuju ke T01
3	Tombol Keluar	: Menuju ke T01

7. M01

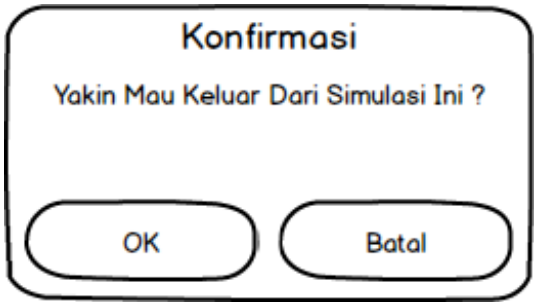


Gambar 3.29 Perancangan Pesan *Setting Database*

Keterangan :

1. Tombol OK : Menuju ke T01

8. M02

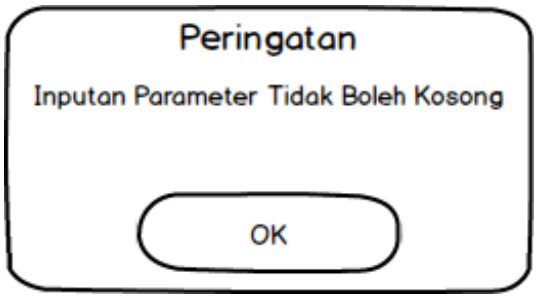


Gambar 3.30 Perancangan Pesan Keluar Simulasi

Keterangan :

1. Tombol OK : Keluar dari simulasi
2. Tombol Batal : Menuju ke T01

9. M03

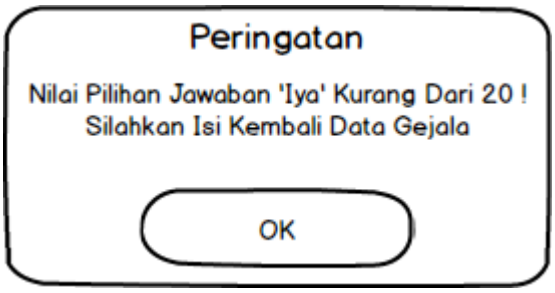


Gambar 3.31 Perancangan Pesan Kesalahan Inputan

Keterangan :

1. Tombol OK : Menuju ke T02

10. M04

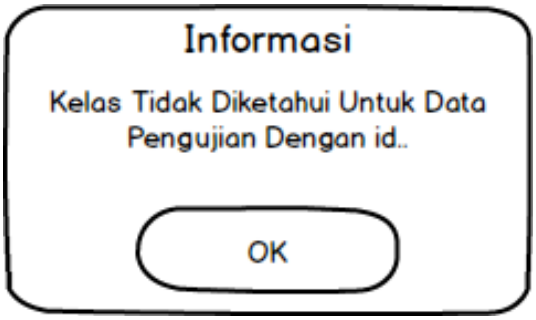


Gambar 3.32 Perancangan Pesan Kesalahan Cek Data Baru

Keterangan :

1. Tombol OK : Menuju ke T04

11. M05

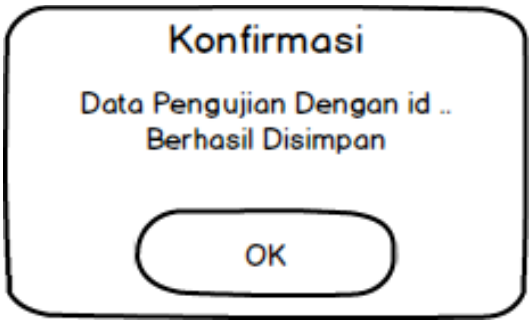


Gambar 3.33 Perancangan Pesan Kelas Tak Terdeteksi

Keterangan :

1. Tombol OK : Menuju ke T04

12. M06

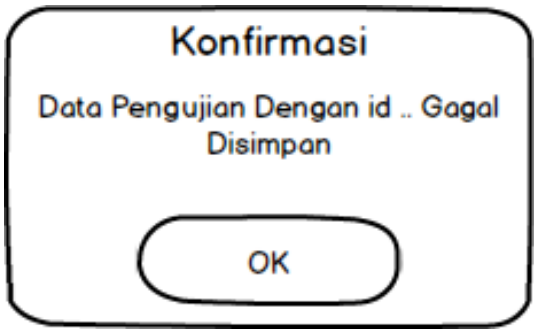


Gambar 3.34 Perancangan Pesan Tambah Data Pengujian Berhasil

Keterangan :

1. Tombol OK : Menuju ke T04

13. M07



Gambar 3.35 Perancangan Pesan Tambah Data Pengujian Gagal

Keterangan :

1. Tombol OK : Menuju ke T04

14. M08



Gambar 3.36 Perancangan Pesan *Setting Database* Berhasil

Keterangan :

1. Tombol OK : Menuju ke T06

15. M09



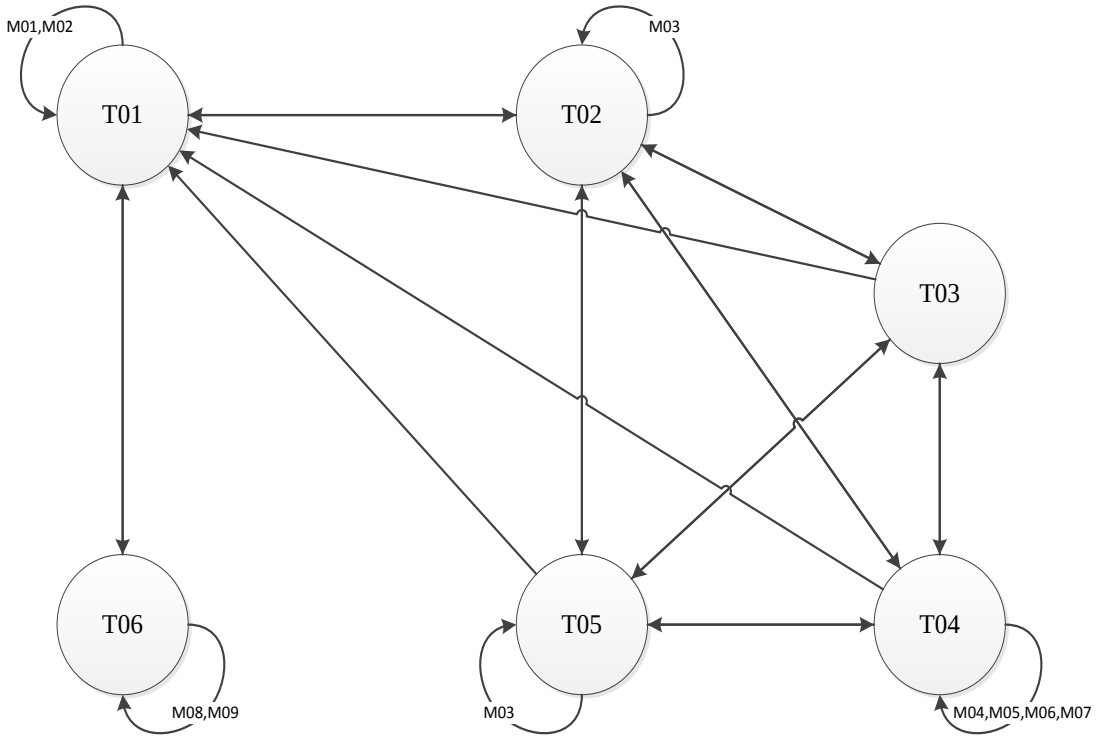
Gambar 3.37 Perancangan Pesan *Setting Database* Gagal

Keterangan :

1. Tombol OK : Menuju ke T06

3.9 Perancangan Jaringan Semantik

Jaringan semantik adalah gambaran pengetahuan grafis yang menunjukkan hubungan antara berbagai objek. Jaringan semantik terdiri dari lingkaran - lingkaran yang menunjukan objek dan informasi tentang objek - objek tertentu. Dalam perancangan jaringan semantik akan membahas mengenai perancangan form yang akan dibuat serta memperlihatkan hubungan navigasi antara form-form tersebut dan pesan - pesan yang akan muncul pada setiap bagian form. Perancangan jaringan semantik simulasi pendeteksian anak berbakat(*gifted child*) dapat dilihat pada gambar 3.38.



Gambar 3.38 Perancangan Jaringan Semantik

{Halaman ini dikosongkan}