

BAB IV. ANALISIS DAN PERANCANGAN SISTEM

4.1. Analisa Sistem

Pada penelitian ini, perlu dilakukan analisa untuk mengkaji lebih dalam tentang sistem yang akan dibuat serta untuk memperjelas keterhubungan satu komponen dengan komponen lainnya pada sistem. Analisa sistem dilakukan pada data yang akan digunakan pada sistem, metode yang digunakan pada sistem, dan kebutuhan fungsional dan non-fungsional sistem.

4.1.1. Analisa Data

Analisa data dilakukan pada penelitian ini untuk menganalisa data yang dibutuhkan pada tahap pelatihan dan pengujian menggunakan metode *backpropagation*. Data yang digunakan pada penelitian ini adalah data hasil kuesioner *Hamilton Anxiety Rating Scale* (HARS) yang terdiri dari 14 pertanyaan yang berkaitan dengan gejala kecemasan yang diderita. Data hasil kuesioner tersebut didapatkan dari penelitian sebelumnya yang dilakukan oleh Sakti pada tahun 2019. Adapun hasil rekapitulasi data yang digunakan pada penelitian ini sejumlah 120 baris data. Tabel 4.1 berikut menunjukkan sampel data hasil kuesioner HARS yang digunakan pada penelitian ini.

Tabel 4.1 Sampel Data Hasil Kuesioner HARS

No	X ₁	X ₂	X ₃	X ₄	...	X ₁₄	Hasil Diagnosa
1	Berat	Sedang	Tidak Ada	Berat	...	Ringan	<i>General Anxiety Disorder</i>
2	Berat	Sedang	Tidak Ada	Sedang	...	Ringan	<i>General Anxiety Disorder</i>
3	Ringan	Sedang	Tidak Ada	Sedang	...	Ringan	<i>General Anxiety Disorder</i>
4	Sedang	Ringan	Tidak Ada	Sedang	...	Ringan	<i>General Anxiety Disorder</i>
5	Sedang	Berat	Sedang	Sedang	...	Ringan	<i>Panic Disorder</i>

6	Sedang	Sedang	Sedang	Berat	...	Ringan	<i>Panic Disorder</i>
7	Berat	Berat	Sedang	Berat	...	Sedang	<i>Panic Disorder</i>
8	Sangat Berat	Berat	Sedang	Berat	...	Ringan	<i>Panic Disorder</i>
9	Sedang	Berat	Ringan	Sedang	...	Ringan	<i>Social Anxiety Disorder</i>
10	Berat	Sedang	Sedang	Berat	...	Berat	<i>Social Anxiety Disorder</i>
...
120	Sedang	Sedang	Sangat Berat	Sedang	...	Ringan	<i>Post Traumatic Stress Disorder</i>

Adapun simbol $X_0 - X_{13}$ pada tabel 4.1 melambangkan 14 variabel data masukan yang digunakan pada penelitian ini. Variabel data masukan berupa data gejala-gejala kecemasan. Pada penelitian ini, data gejala-gejala kecemasan didapatkan dari 14 pertanyaan yang terdapat pada kuesioner HARS. 14 pertanyaan pada kuesioner HARS tersebut mewakili tiap kelompok gejala kecemasan yang diderita pasien. Tabel 4.2 berikut menunjukkan variabel data masukan berupa gejala-gejala kecemasan yang digunakan untuk mendiagnosa jenis gangguan kecemasan yang diderita pasien.

Tabel 4.2 Variabel Data Masukan

Variabel	Keterangan
X_0	Perasaan cemas
X_1	Ketegangan
X_2	Ketakutan
X_3	Gangguan tidur
X_4	Gangguan kecerdasan
X_5	Perasaan depresi

X ₆	Gejala somatik/fisik (otot)
X ₇	Gejala somatik/fisik (sensorik)
X ₈	Gejala kardiovaskuler (jantung dan pembuluh darah)
X ₉	Gejala respiratori (pernafasan)
X ₁₀	Gejala gastrointestinal (pencernaan)
X ₁₁	Gejala urogenital (perkemihan dan kelamin)
X ₁₂	Gejala autoimun
X ₁₃	Tingkah laku/sikap

Apabila seluruh data yang dibutuhkan sudah terkumpul, maka data akan ditransformasi supaya bisa digunakan dalam perhitungan menggunakan metode *backpropagation*. Data yang ditransformasi adalah data jawaban pasien terhadap 14 pertanyaan yang terdapat di kuesioner HARS terkait gejala kecemasan yang dialami pasien. Pada tiap pertanyaan di kuesioner HARS terdapat 5 pilihan jawaban yang disediakan untuk menunjukkan tingkat keparahan dari tiap gejala kecemasan yang diderita. Kelima pilihan jawaban tersebut antara lain adalah tidak ada, ringan, sedang, berat, dan sangat berat. Masing-masing pilihan jawaban akan dikonversi menjadi skala angka dari 0 sampai 4. Transformasi nilai data gejala kecemasan dapat dilihat pada tabel 4.3.

Tabel 4.3 Transformasi Nilai Data Gejala Kecemasan

	Keterangan	Skala Nilai
Gejala Kecemasan	Tidak Ada	0
	Ringan	1
	Sedang	2
	Berat	3
	Sangat Berat	4

Selain data jawaban pasien pada kuesioner HARS, proses transformasi data juga dilakukan pada variabel data keluaran atau *output* jenis gangguan kecemasan yang diderita pasien berdasarkan jawaban yang diinputkan pasien pada tiap pertanyaan di kuesioner HARS. Terdapat 6 kelas gangguan kecemasan yang digunakan sebagai *output* dari penelitian ini. 6 kelas gangguan kecemasan tersebut

antara lain *general anxiety disorder*, *panic disorder*, *social anxiety disorder*, *specific phobia*, *obsessive compulsive disorder*, dan *post traumatic stress disorder*. Masing-masing jenis gangguan kecemasan tersebut akan dikonversi menjadi kelas 1 sampai kelas 6. Tabel 4.4 berikut menunjukkan transformasi nilai data *output* yang digunakan pada penelitian ini.

Tabel 4.4 Transformasi Nilai *Output*

Kelas	Keterangan
1	<i>General Anxiety Disorder</i>
2	<i>Panic Disorder</i>
3	<i>Social Anxiety Disorder</i>
4	<i>Specific Phobia</i>
5	<i>Obsessive Compulsive Disorder</i>
6	<i>Post Traumatic Stress Disorder</i>

Tabel 4.5 berikut menunjukkan contoh data hasil kuesioner HARS yang telah melalui proses transformasi data. Pada tabel 4.5, data yang ditransformasi merupakan data kuesioner HARS dari 6 pasien dengan hasil diagnosa gangguan kecemasan yang berbeda. Baris terbawah pada tabel ini menunjukkan kelas hasil diagnosa jenis gangguan kecemasan yang diderita masing-masing pasien. Hasil diagnosa tersebut didapatkan berdasarkan jawaban pasien pada pertanyaan yang terdapat di kuesioner HARS.

Tabel 4.5 Hasil Transformasi Data

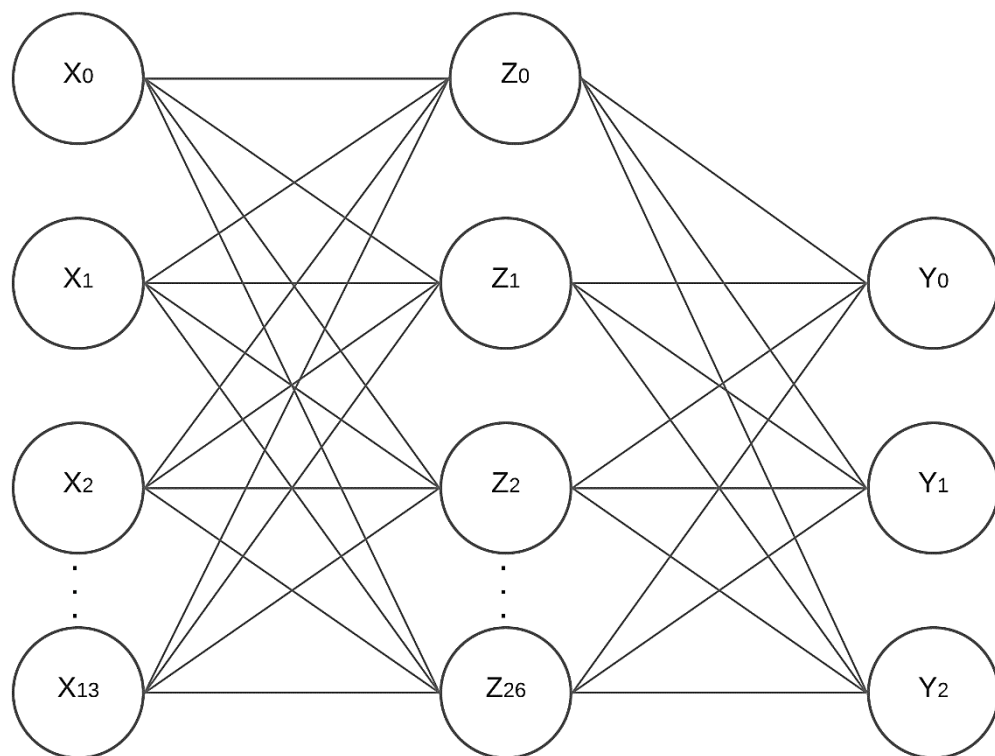
DATA	Pasien 1	Pasien 2	Pasien 3	Pasien 4	Pasien 5	Pasien 6
X ₀	3	2	2	2	3	2
X ₁	2	3	3	0	1	3
X ₂	0	2	1	2	3	4
X ₃	3	2	2	4	1	2
X ₄	3	3	1	0	0	0
X ₅	2	2	3	1	3	2
X ₆	0	2	2	0	0	2
X ₇	1	1	1	1	0	0

X ₈	2	1	2	4	0	3
X ₉	1	0	3	1	1	2
X ₁₀	2	2	1	0	0	0
X ₁₁	1	0	2	2	0	0
X ₁₂	2	3	2	2	1	2
X ₁₃	1	1	1	1	1	1
KELAS	1	2	3	4	5	6

Setelah data ditransformasi, selanjutnya data akan dibagi menjadi data latih dan data uji berdasarkan perbandingan yang telah ditentukan. Terdapat 3 perbandingan data yang digunakan pada penelitian ini, yaitu 90:10, 80:20, dan 70:30. Hasil perhitungan *backpropagation* pada ketiga perbandingan tersebut nantinya akan dibandingkan untuk mengetahui perbandingan mana yang memberikan hasil paling akurat dalam mendiagnosa jenis gangguan kecemasan.

4.1.2. Analisis Metode *Backpropagation*

Pada penelitian ini, proses diagnosa jenis gangguan kecemasan dilakukan menggunakan metode *backpropagation*. Proses perhitungan *backpropagation* dalam mendiagnosa jenis gangguan kecemasan dibagi menjadi 2 tahap, yaitu tahap pelatihan dan tahap pengujian. Tahap pelatihan dilakukan sebagai proses pembelajaran bagi sistem untuk mengenali pola data masukan, dalam hal ini pola data masukan yang dimaksud adalah data jawaban pasien pada kuesioner HARS dan data hasil diagnosa jenis gangguan kecemasan yang diderita pasien berdasarkan jawaban dari kuesioner HARS. Sedangkan tahap pengujian dilakukan untuk menguji akurasi sistem dalam mendiagnosa jenis gangguan kecemasan berdasarkan perhitungan menggunakan metode *backpropagation*. Pada tahap pengujian, hasil pembelajaran dari tahap pelatihan yang dilakukan sebelumnya dalam mengenali pola data masukan akan digunakan untuk menguji akurasi sistem. Gambar 4.1 menunjukkan arsitektur *backpropagation* yang digunakan pada penelitian ini.



Gambar 4.1 Arsitektur *Backpropagation*

Neuron pada lapisan input memiliki jumlah sesuai dengan input yang dimasukkan pada sistem, yaitu 14 variabel gejala kecemasan dari kuesioner HARS. Pada gambar 4.1, variabel data masukan disimbolkan dengan $X_0 - X_{13}$. Bobot yang menghubungkan antara lapisan input dengan lapisan tersembunyi atau *hidden layer* dilambangkan dengan $V_{00} - V_{1326}$.

Adapun penentuan jumlah *neuron* pada lapisan tersembunyi dalam arsitektur *backpropagation* tidak dibatasi pada aturan atau *range* tertentu, namun pada gambar 4.1, arsitektur *backpropagation* diilustrasikan dengan jumlah *neuron* pada lapisan tersembunyi sejumlah 27 *neuron*. *Neuron* pada lapisan tersembunyi dilambangkan dengan $Z_0 - Z_{26}$. Bobot yang menghubungkan *neuron* dari lapisan tersembunyi dengan *neuron* pada lapisan *output* adalah $W_{00} - W_{262}$.

Sementara jumlah *neuron* yang digunakan pada lapisan *output* terdiri 3 *neuron*. Tiap *neuron* pada lapisan *output* berisi satu digit bilangan biner sehingga pada lapisan *output* nantinya akan ada 3 digit bilangan biner, di mana kombinasi 3 digit bilangan biner akan menghasilkan maksimal 9 kombinasi bilangan biner. Oleh karena itu, keenam *output* jenis gangguan kecemasan akan dibuat ke dalam 3

neuron. Penggunaan kombinasi biner pada penelitian ini merujuk kepada penelitian yang dilakukan oleh Saputra pada tahun 2019, yang juga menggunakan metode *backpropagation momentum* dengan kombinasi biner dalam mengklasifikasikan jenis gangguan kecemasan (Saputra, 2019). Berdasarkan hasil penelitian tersebut, penggunaan kombinasi biner terbukti dapat digunakan untuk memetakan nilai sebuah kelas. Penentuan nilai kombinasi biner yang digunakan pada penelitian ini dilakukan tanpa mengikuti aturan khusus. Tabel berikut menunjukkan kombinasi biner untuk mewakili 6 kelas *output* jenis gangguan kecemasan. Pada tabel berikut, *neuron* pada lapisan *output* dilambangkan dengan Y_0 , Y_1 , dan Y_2 .

Tabel 4.6 Kombinasi Biner *Output* Jenis Gangguan Kecemasan

No	Kelas	Y_0	Y_1	Y_2	Keterangan
1	Kelas 1	0	0	0	<i>General Anxiety Disorder</i>
2	Kelas 2	0	0	1	<i>Panic Disorder</i>
3	Kelas 3	0	1	0	<i>Social Anxiety Disorder</i>
4	Kelas 4	1	0	1	<i>Specific Phobia</i>
5	Kelas 5	1	1	0	<i>Obsessive Compulsive Disorder</i>
6	Kelas 6	1	1	1	<i>Post Traumatic Stress Disorder</i>

Alasan penggunaan kombinasi biner pada lapisan *output* untuk perhitungan *backpropagation* dalam penelitian ini adalah dikarenakan fungsi aktivasi yang digunakan pada perhitungan menggunakan metode *backpropagation* adalah fungsi *sigmoid biner*. Fungsi aktivasi *sigmoid biner* memiliki rentang nilai antara 0 hingga 1, sehingga penggunaan kombinasi biner pada lapisan *output* merupakan pilihan yang lebih tepat untuk menghasilkan akurasi yang lebih baik dalam perhitungan menggunakan metode *backpropagation*. Fungsi aktivasi ini akan digunakan saat perambatan sinyal data antar lapisan.

Data yang digunakan pada saat tahap pelatihan adalah data latih. Tahap ini terdiri dari 3 fase. Fase pertama adalah fase perambatan maju, di mana pola masukan yang terdiri dari $X_0 - X_{13}$ dihitung maju dari lapisan input hingga lapisan *output* menggunakan fungsi *sigmoid biner*. Fase kedua adalah fase perambatan balik atau *backpropagation*. Pada fase ini, *output* jaringan akan dibandingkan dengan target *output* jaringan yang seharusnya dicapai. Fase ketiga adalah

perubahan bobot, di mana pada fase ini semua bobot akan dimodifikasi untuk menurunkan tingkat kesalahan atau *error*. Tahap pelatihan berguna dalam melakukan pembelajaran dengan metode *backpropagation* untuk mengenali pola dari inputan sehingga dapat digunakan untuk menghasilkan *output* jenis gangguan kecemasan sesuai target. Hasil pembelajaran dari tahap pelatihan nantinya akan digunakan pada tahap pengujian untuk mengukur akurasi sistem dalam mendiagnosa jenis gangguan kecemasan berdasarkan perhitungan menggunakan *backpropagation*.

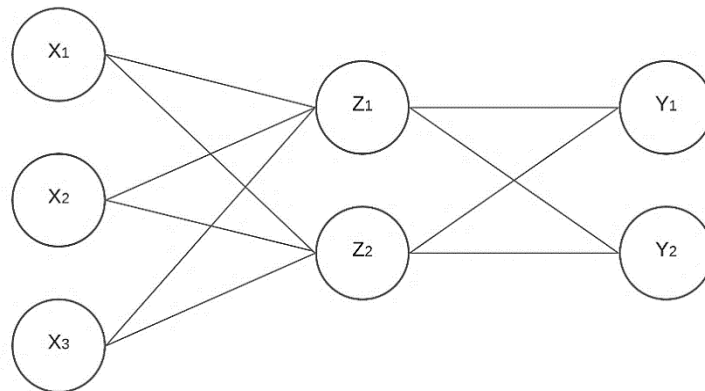
Adapun untuk tahap pengujian yang dilakukan pada sistem hanya terdiri dari satu fase, yaitu fase perambatan maju. Fase perambatan balik dan perubahan bobot tidak diperlukan pada tahap pengujian karena kedua fase tersebut digunakan dalam pembelajaran untuk mengenali pola inputan, sedangkan pada tahap pengujian tidak memerlukan proses pembelajaran lagi untuk mengenali pola inputan, sehingga data langsung diinputkan untuk mengetahui *output* yang dihasilkan berdasarkan hasil pembelajaran yang telah dilakukan sebelumnya dari tahap pelatihan. Pada tahap pengujian, data yang digunakan berupa data uji untuk menentukan tingkat akurasi sistem dalam mendiagnosa jenis gangguan kecemasan.

Secara keseluruhan, proses perhitungan menggunakan metode *backpropagation* dimulai dengan menginput data latih dan data uji pada sistem. Setelah seluruh data latih dan data uji diinputkan pada sistem, sistem akan menerima dan memproses input *learning rate*, *momentum*, *epoch*, *target error*, jumlah *neuron* pada lapisan tersembunyi yang digunakan dalam perhitungan menggunakan *backpropagation*, dan perbandingan *split test* yang digunakan untuk membagi data latih dan data uji pada sistem. Selanjutnya sistem akan menginisialisasi nilai bobot dan bias awal secara acak. Sistem kemudian melakukan proses perhitungan pada fase perambatan maju, perambatan balik, dan perubahan bobot dan bias menggunakan parameter *momentum*. Ketiga fase tersebut dilakukan sebanyak maksimum *target error* dan *epoch* yang telah ditentukan. *Epoch* merupakan jumlah maksimal perulangan yang dapat dilakukan pada tahap pelatihan. Proses perulangan pada tahap pelatihan akan dihentikan bila sudah mencapai jumlah *epoch* yang telah ditentukan. Sedangkan *target error* adalah nilai *error* yang dijadikan target untuk menghitung kesalahan atau *error* yang dihasilkan

dari perhitungan menggunakan *backpropagation*. Perhitungan *error* dilakukan menggunakan rumus selisih antara target *output* dengan *output* yang dihasilkan. Jika perhitungan *error* menghasilkan nilai yang lebih kecil dari *target error*, maka perulangan akan dihentikan. Namun bila hasil perhitungan *error* lebih besar dari *target error*, maka *error* akan didistribusikan pada *output* dengan dihitung berdasarkan faktor *error* untuk selanjutnya digunakan pada fase perubahan bobot. Bobot dan bias akhir dari proses pelatihan kemudian akan disimpan untuk digunakan pada tahap pengujian. Setelah tahap pelatihan selesai, maka tahap pengujian akan dilakukan dengan menginisialisasi nilai bobot dan bias awal dengan bobot dan bias yang telah diperoleh sebelumnya dari tahap pelatihan. Selanjutnya sistem akan menjalankan fase perambatan maju hingga perhitungan *backpropagation* menghasilkan diagnosa jenis gangguan kecemasan pada tiap data uji. Selain digunakan pada tahap pengujian, bobot dan bias akhir dari tahap pelatihan juga digunakan saat proses tes individu atau *self-assessment* yang dilakukan oleh pasien pada sistem untuk mendiagnosa jenis gangguan kecemasan yang diderita. Proses tes individu ini memiliki tahapan yang sama dengan tahap pengujian, yaitu hanya terdiri dari fase perambatan maju untuk mendiagnosa jenis gangguan kecemasan yang diderita pasien.

4.1.3. Contoh Perhitungan Metode *Backpropagation*

Proses perhitungan menggunakan metode *backpropagation* dibagi menjadi 2 tahap, yaitu tahap pelatihan dan pengujian. Tahap pelatihan terdiri dari 3 fase yaitu fase perambatan maju, fase perambatan balik, dan fase perubahan bobot. Sedangkan pada tahap pengujian hanya terdiri dari satu fase, yaitu fase perambatan maju. Konfigurasi parameter yang digunakan pada contoh perhitungan manual sederhana ini terdiri dari *neuron input layer* sejumlah 3 *neuron*, *neuron hidden layer* sejumlah 2 *neuron*, *neuron output layer* sejumlah 2 *neuron*, *learning rate* senilai 0.2, dan *momentum* 0.2. Berikut ini adalah arsitektur jaringan saraf tiruan *backpropagation* yang akan digunakan dalam contoh perhitungan manual sederhana.



Gambar 4.2 Arsitektur *Backpropagation* Untuk Perhitungan Manual

Sebelum tahap pelatihan pada perhitungan *backpropagation* dilakukan, perlu dilakukan inisialisasi bobot dan bias secara acak. Berikut ini adalah tabel yang menampilkan rincian bobot awal pada *neuron* input.

Tabel 4.7 Rincian Bobot Awal pada *Neuron* Input

V_{11}	V_{12}	V_{21}	V_{22}	V_{31}	V_{32}
0.1	0.2	0.1	0.3	0.4	0.8

Berikut adalah tabel yang menampilkan rincian bobot awal pada *neuron hidden*.

Tabel 4.8 Rincian Bobot Awal pada *Neuron Hidden*

W_{11}	W_{12}	W_{21}	W_{22}
0.2	0.1	0.4	0.5

Berikut ini adalah data masukan atau data latih yang digunakan pada tahap pelatihan. Data masukan ini sudah ditransformasi ke dalam bentuk skala angka dan memiliki nilai $t_1 = 0$ dan $t_2 = 0$ untuk target diagnosa jenis gangguan kecemasan. Diasumsikan bahwa *output* yang dihasilkan pada perhitungan manual ini terdiri dari 4 kelas gangguan kecemasan, di mana untuk kelas gangguan kecemasan *General Anxiety Disorder* memiliki nilai target *output* 1 dan target *output* 2 masing-masing senilai 0. Dengan demikian, data masukan yang digunakan dalam contoh perhitungan manual ini termasuk ke dalam jenis gangguan kecemasan *General Anxiety Disorder*.

Tabel 4.9 Data Masukan

X_1	X_2	X_3
3	2	1

Perhitungan pada tahap pelatihan dimulai dari fase perambatan maju.

Fase 1: Perambatan Maju (*Feedforward Propagation*)

Tiap unit input menerima sinyal dan meneruskan sinyal ke *hidden layer*. Setiap unit *hidden layer* menjumlahkan bobot sinyal input menggunakan persamaan 2.6. Berikut adalah contoh perhitungan bobot sinyal input pada unit *hidden* ($v_{01} = 0.6, v_{02} = 0.3$).

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

$$z_{in_1} = v_{01} + \sum_{i=1}^n x_i v_{32}$$

$$z_{in_1} = 0.6 + 3 \times 0.1 + 2 \times 0.1 + 0.1 \times 0.4 = 1.14$$

$$z_{in_2} = 0.3 + 3 \times 0.2 + 2 \times 0.3 + 0.1 \times 0.8 = 1.58$$

Hasil perhitungan bobot sinyal input pada z_{in_1} sampai $z_{in_{14}}$ dapat dilihat pada tabel berikut.

Tabel 4.10 Hasil Perhitungan Bobot Sinyal Input

z_{in_1}	z_{in_2}
1.14	1.58

Selanjutnya akan dilakukan perhitungan sinyal *output* menggunakan fungsi aktivasi *sigmoid biner*. Sinyal *output* tersebut nantinya akan dikirimkan ke semua unit *output*. Berikut adalah contoh perhitungan sinyal *output* pada unit *hidden*.

$$z_j = \frac{1}{1 + e^{-z_{in_j}}}$$

$$z_1 = \frac{1}{1 + e^{-1.14}} = 0.757679$$

$$z_2 = \frac{1}{1 + e^{-1.58}} = 0.829204$$

Berikut ini adalah tabel yang menunjukkan hasil perhitungan sinyal *output* pada unit *hidden*.

Tabel 4.11 Hasil Perhitungan Sinyal *Output* pada Unit *Hidden*

z_1	z_2
0.757679	0.829204

Unit *output* selanjutnya akan menjumlahkan sinyal-sinyal input menggunakan persamaan 2.8 ($w_{01} = 0.3, w_{02} = 0.2$).

$$y_{in_k} = w_{0k} + \sum_{i=1}^p z_i w_{jk}$$

$$y_{in_1} = w_{01} + \sum_{i=1}^p z_i w_{21}$$

$$y_{in_1} = 0.3 + 0.757679 \times 0.2 + 0.829204 \times 0.4 = 0.783218$$

$$y_{in_2} = 0.2 + 0.757679 \times 0.1 + 0.829204 \times 0.5 = 0.69037$$

Berikut adalah tabel hasil penjumlahan sinyal-sinyal input oleh unit *output*.

Tabel 4.12 Hasil Penjumlahan Sinyal Input oleh Unit *Output*

y_{in_1}	y_{in_2}
0.783218	0.69037

Fungsi aktivasi *sigmoid biner* digunakan untuk menghitung sinyal *output* dari unit *output*. Berikut adalah contoh perhitungan sinyal *output* dari unit *output* menggunakan fungsi *sigmoid biner*.

$$y_1 = \frac{1}{1 + e^{-0.783218}} = 0.686373$$

$$y_2 = \frac{1}{1 + e^{-0.69037}} = 0.666049$$

Berikut adalah tabel hasil sinyal *output* dari unit *output*.

Tabel 4.13 Hasil Perhitungan Sinyal *Output* pada Unit *Output*

y_1	y_2
0.686373	0.666049

Fase 2: Perambatan Balik (*Backpropagation*)

Pada fase sebelumnya, unit *output* telah menerima pola target sesuai pola input. Selanjutnya pada fase *backpropagation*, perhitungan *error* dilakukan menggunakan persamaan 2.10, di mana t_k adalah target *output*.

$$\delta_k = (t_k - y_k) y_k (1 - y_k)$$

$$\delta_1 = (t_1 - y_1) y_1 (1 - y_1)$$

$$\delta_1 = (0 - 0.686373) \times 0.686373 \times (1 - 0.686373) = -0.14775$$

$$\delta_2 = (0 - 0.666049) \times 0.666049 \times (1 - 0.666049) = -0.14815$$

Perhitungan koreksi bobot dilakukan menggunakan persamaan 2.11. Nilai koreksi bobot didapatkan dari hasil perkalian *learning rate* dengan faktor kesalahan dan bobot pada *hidden layer*.

$$\Delta w_{jk} = \alpha \delta_k z_j$$

$$\Delta w_{11} = \alpha \delta_1 z_1$$

$$\Delta w_{11} = 0.2 \times (-0.14775) \times 0.757679 = -0.02239$$

$$\Delta w_{21} = 0.2 \times (-0.14775) \times 0.829204 = -0.0245$$

$$\Delta w_{12} = 0.2 \times (-0.14815) \times 0.757679 = -0.02245$$

$$\Delta w_{22} = 0.2 \times (-0.14815) \times 0.829204 = -0.02457$$

Berikut ini adalah tabel yang menunjukkan hasil perhitungan koreksi bobot.

Tabel 4.14 Hasil Perhitungan Koreksi Bobot pada Unit *Hidden*

Δw_{11}	Δw_{21}	Δw_{12}	Δw_{22}
-0.02239	-0.0245	-0.02245	-0.02457

Perhitungan koreksi bias dilakukan menggunakan persamaan 2.12. Nilai koreksi bias diperoleh dari hasil perkalian antara *learning rate* dengan faktor kesalahan.

$$\Delta w_{0k} = \alpha \delta_k$$

$$\Delta w_{01} = \alpha \delta_1$$

$$\Delta w_{01} = 0.2 \times (-0.14775) = -0.02955$$

$$\Delta w_{02} = 0.2 \times (-0.14815) = -0.02963$$

Perhitungan faktor *error* berdasarkan *error* pada *hidden layer* dilakukan menggunakan persamaan 2.13.

$$\delta_{in_j} = \delta_k w_{jk}$$

$$\delta_{in_1} = \delta_1 w_{11}$$

$$\delta_{in_1} = -0.14775 \times 0.2 = -0.02955$$

$$\delta_{in_2} = -0.14815 \times 0.5 = -0.07407$$

Berikut ini adalah tabel hasil faktor *error* pada *hidden layer*.

Tabel 4.15 Hasil Faktor *Error* pada *Hidden Layer*

δ_{in_1}	δ_{in_2}
-0.02955	-0.07407

Perhitungan informasi *error* dilakukan menggunakan persamaan 2.14.

$$\delta_j = \delta_{in} z_j (1 - z_j)$$

$$\delta_1 = \delta_{in_1} z_1 (1 - z_1)$$

$$\delta_1 = -0.02955 \times 0.757679 \times (1 - 0.757679) = -0.00543$$

$$\delta_2 = -0.07407 \times 0.829204 \times (1 - 0.829204) = -0.01049$$

Tabel 4.16 Hasil Perhitungan Informasi *Error*

δ_1	δ_2
-0.00543	-0.01049

Perhitungan koreksi bobot menggunakan persamaan 2.15 untuk memperbaiki nilai bobot. Nilai koreksi bobot didapatkan dari hasil perkalian antara *learning rate* dengan informasi *error* dan data masukan x_i . Pada contoh perhitungan berikut, hasil perhitungan telah dibulatkan.

$$\Delta v_{ij} = \alpha \delta_j x_i$$

$$\Delta v_{11} = \alpha \delta_1 x_1$$

$$\Delta v_{11} = 0.2 \times (-0.00543) \times 3 = -0.00326$$

$$\Delta v_{21} = 0.2 \times (-0.00543) \times 2 = -0.00217$$

$$\Delta v_{31} = 0.2 \times (-0.00543) \times 1 = -0.00011$$

$$\Delta v_{12} = 0.2 \times (-0.01049) \times 3 = -0.00629$$

$$\Delta v_{22} = 0.2 \times (-0.01049) \times 2 = -0.0042$$

$$\Delta v_{32} = 0.2 \times (-0.01049) \times 1 = -0.00021$$

Berikut adalah tabel yang menampilkan hasil perhitungan koreksi bobot.

Tabel 4.17 Hasil Koreksi Bobot

Δv_{11}	Δv_{21}	Δv_{31}	Δv_{12}	Δv_{22}	Δv_{32}
-0.00326	-0.00217	-0.00011	-0.00629	-0.0042	-0.00021

Perhitungan koreksi bias dilakukan menggunakan persamaan 2.16 untuk memperbaiki nilai bias. Nilai koreksi bias didapatkan dari hasil perkalian antara *learning rate* dengan informasi *error*.

$$\Delta v_{0j} = \alpha \delta_j$$

$$\Delta v_{01} = \alpha \delta_1$$

$$\Delta v_{01} = 0.2 \times (-0.00543) = -0.00109$$

$$\Delta v_{02} = 0.2 \times (-0.01049) = -0.0021$$

Berikut adalah tabel yang menampilkan hasil koreksi bias.

Tabel 4.18 Hasil Koreksi Bias

Δv_{01}	Δv_{02}
-0.00109	-0.0021

Fase 3: Perubahan Bobot

Perhitungan bobot baru pada *neuron* input dilakukan menggunakan persamaan 2.18. Perhitungan bobot baru menggunakan nilai *momentum* μ untuk memperbaiki nilai bobot dan bias.

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \mu \times \Delta v_{ij}$$

$$v_{11}(\text{baru}) = v_{11}(\text{lama}) + \mu \times \Delta v_{11}$$

$$v_{11}(\text{baru}) = 0.1 + 0.2 \times (-0.00326) = 0.099349$$

$$v_{21}(\text{baru}) = 0.1 + 0.2 \times (-0.00217) = 0.099566$$

$$v_{31}(\text{baru}) = 0.4 + 0.2 \times (-0.00011) = 0.399978$$

$$v_{12}(\text{baru}) = 0.2 + 0.2 \times (-0.00629) = 0.198741$$

$$v_{22}(\text{baru}) = 0.3 + 0.2 \times (-0.0042) = 0.299161$$

$$v_{32}(\text{baru}) = 0.8 + 0.2 \times (-0.00021) = 0.799958$$

Berikut adalah hasil perhitungan bobot baru pada *neuron* input menuju *hidden layer*.

Tabel 4.19 Hasil Perhitungan Bobot Baru pada Input Layer

v_{11}	v_{21}	v_{31}	v_{12}	v_{22}	v_{32}
0.099349	0.099566	0.399978	0.198741	0.299161	0.799958

Perhitungan bias baru pada input *layer* dilakukan menggunakan persamaan 2.18. Perhitungan bias baru menggunakan momentum μ untuk memperbaiki nilai bias pada input *layer*.

$$v_{0j}(\text{baru}) = v_{0j}(\text{lama}) + \mu \times \Delta v_{0j}$$

$$v_{01}(\text{baru}) = v_{01}(\text{lama}) + \mu \times \Delta v_{01}$$

$$v_{01}(\text{baru}) = 0.6 + 0.2 \times (-0.00109) = 0.599783$$

$$v_{02}(\text{baru}) = 0.3 + 0.2 \times (-0.0021) = 0.29958$$

Berikut adalah hasil perhitungan bias pada lapisan input.

Tabel 4.20 Hasil Perhitungan Bias Baru pada Input *Layer*

v_{01}	v_{02}
0.599783	0.29958

Perhitungan bobot baru pada *hidden layer* dilakukan menggunakan persamaan 2.17.

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \mu \times \Delta w_{jk}$$

$$w_{11}(\text{baru}) = 0.2 + 0.2 \times (-0.02239) = 0.195522$$

$$w_{21}(\text{baru}) = 0.4 + 0.2 \times (-0.0245) = 0.395099$$

$$w_{12}(\text{baru}) = 0.1 + 0.2 \times (-0.02245) = 0.09551$$

$$w_{22}(\text{baru}) = 0.5 + 0.2 \times (-0.02457) = 0.495086$$

Perhitungan bias baru pada *hidden layer* adalah sebagai berikut, dengan nilai $w_{01}(\text{lama})$ senilai 0.3, dan $w_{02}(\text{lama})$ 0.2.

$$\Delta w_{01} = 0.2 \times (-0.14775) = -0.02955$$

$$\Delta w_{02} = 0.2 \times (-0.14815) = -0.02963$$

$$w_{0j}(\text{baru}) = w_{0j}(\text{lama}) + \mu \times \Delta w_{0j}$$

$$w_{01}(\text{baru}) = 0.3 + 0.2 \times (-0.02955) = 0.29409$$

$$w_{02}(\text{baru}) = 0.2 + 0.2 \times (-0.02963) = 0.194074$$

Berikut adalah hasil perhitungan bias baru pada *hidden layer*.

Tabel 4.21 Hasil Perhitungan Bias Baru pada *Hidden Layer*

w_{01}	w_{02}
0.29409	0.194074

Setelah seluruh bobot dan bias yang digunakan dalam perhitungan *backpropagation* diperbarui pada fase perubahan bobot, maka selanjutnya akan dijalankan tahap pengujian. Pada tahap pengujian, data yang digunakan adalah data yang sama dengan data yang digunakan pada tahap pelatihan. Bobot dan bias yang digunakan pada tahap pengujian diperoleh dari fase perubahan bobot.

Fase 1: Perambatan Maju (*Feedforward Propagation*)

Hitung sinyal input pada *hidden layer* menggunakan persamaan 2.19.

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

$$z_in_1 = v_{01} + \sum_{i=1}^n x_i v_{i1}$$

$$z_in_1 = 0.599783 + 3 \times 0.099349 + 2 \times 0.099566 + 1 \times 0.399978 = 1.49694$$

$$z_in_2 = 0.29958 + 3 \times 0.198741 + 2 \times 0.299161 + 1 \times 0.799958 = 2.294083$$

Berikut adalah hasil perhitungan sinyal input pada *hidden layer*.

Tabel 4.22 Hasil Perhitungan Sinyal Input pada *Hidden Layer*

z_in_1	z_in_2
1.49694	2.294083

Gunakan fungsi *sigmoid biner* untuk menghitung sinyal *output* yang dihasilkan oleh *hidden layer*.

$$z_j = \frac{1}{1 + e^{-z_in_j}}$$

$$z_1 = \frac{1}{1 + e^{-1.49694}} = 0.817117$$

$$z_2 = \frac{1}{1 + e^{-2.294083}} = 0.908386$$

Berikut adalah tabel yang menunjukkan sinyal *output* dari *hidden layer*.

Tabel 4.23 Hasil Perhitungan Sinyal *Output* pada *Hidden Layer*

z_1	z_2
0.817117	0.908386

Selanjutnya tiap *unit output* akan menjumlahkan bobot sinyal input menggunakan persamaan 2.21.

$$y_in_k = w_{0k} + \sum_{i=1}^p z_i w_{ik}$$

$$y_in_1 = w_{01} + \sum_{i=1}^p z_i w_{i1}$$

$$y_in_1 = 0.29409 + 0.817117 \times 0.195522 + 0.908386 \times 0.395099 = 0.812757$$

$$y_in_2 = 0.194074 + 0.817117 \times 0.09551 + 0.908386 \times 0.495086 = 0.721846$$

Berikut adalah hasil perhitungan bobot sinyal input pada tiap unit *output*.

Tabel 4.24 Hasil Perhitungan Sinyal Input pada *Output Layer*

y_{in_1}	y_{in_2}
0.812757	0.721846

Gunakan fungsi aktivasi *sigmoid biner* untuk menghitung sinyal *output* pada *output layer*.

$$y_1 = \frac{1}{1 + e^{-0.812757}} = 0.692697$$

$$y_2 = \frac{1}{1 + e^{-0.721846}} = 0.673013$$

Berikut adalah hasil perhitungan sinyal *output* pada lapisan *output*.

Tabel 4.25 Hasil Perhitungan Sinyal *Output* pada *Output Layer*

y_1	y_2
0.692697	0.673013

Keterangan: Jika $y_k < 0.5$, maka nilai $Y_k = 0$

Jika $y_k \geq 0.5$, maka nilai $Y_k = 1$

Berdasarkan tabel di atas, seluruh nilai *output* memiliki nilai di atas 0.5.

Dengan demikian, seluruh sinyal *output* memiliki nilai 1.

4.1.4. Analisis Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak dapat dilihat pada tabel 4.26 berikut.

Tabel 4.26 Spesifikasi Perangkat Lunak

No.	Nama Perangkat Lunak
1	Sistem Operasi Windows 10
2	Teks <i>Editor</i> Visual Studio Code
3	XAMPP
4	<i>Browser</i> Microsoft Edge
5	PHP

4.1.5. Analisis Kebutuhan Perangkat Keras

Spesifikasi perangkat keras dapat dilihat pada tabel 4.27 berikut.

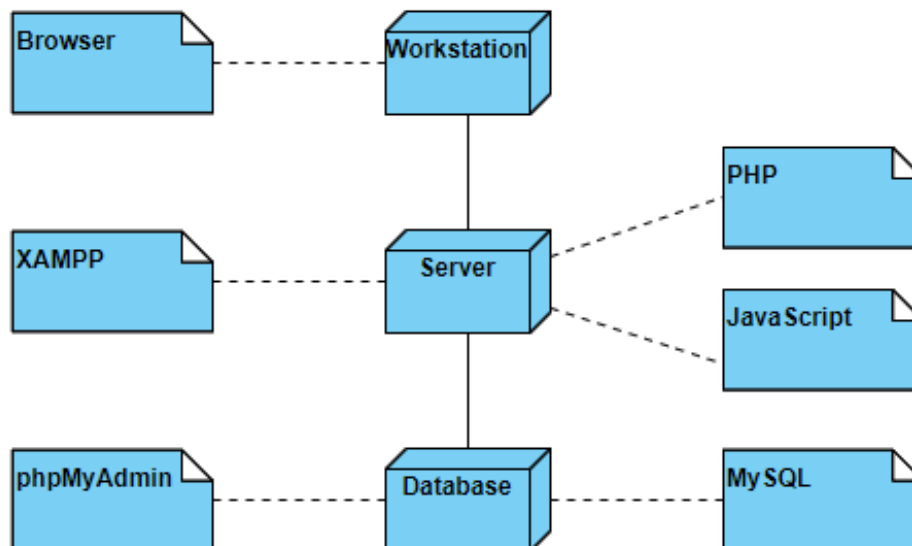
Tabel 4.27 Spesifikasi Perangkat Keras

No.	Nama Perangkat Keras
1	Processor Intel Celeron N4000
2	RAM 4GB
3	HDD 500GB

4.2. Perancangan Sistem

4.2.1. Arsitektur Sistem

Gambar 4.3 berikut menunjukkan arsitektur sistem pakar dalam bentuk *deployment diagram*. *Deployment diagram* merupakan *diagram* yang menggambarkan arsitektur fisik seperti *web server* dan semua perangkat lunak pendukung yang terdapat pada sistem pakar.



Gambar 4.3 Arsitektur Sistem

Pada gambar 4.3 dapat ditunjukkan bahwa arsitektur sistem pakar terdiri dari *workstation*, *server*, dan *database*. *Workstation* merupakan komputer *client*, di mana pada komputer *client* terdapat sebuah *browser* yang akan digunakan untuk

menjalankan aplikasi sistem pakar berbasis *web*. *Server* aplikasi sistem pakar berupa komponen-komponen penyusun *web* sistem pakar. Komponen-komponen penyusun tersebut antara lain terdiri dari bahasa pemrograman PHP dan Javascript dan paket perangkat lunak XAMPP yang digunakan sebagai layanan *web server* sistem pakar. Adapun *database* yang digunakan sebagai penyimpanan data pada sistem adalah MySQL dengan phpMyAdmin sebagai perangkat yang digunakan dalam pengelolaan *database*.

4.2.2. Use Case Diagram

Use case diagram adalah diagram yang mempresentasikan fungsi yang terdapat pada suatu sistem serta memvisualisasikan hubungan atau interaksi antara satu atau lebih aktor atau *user* dengan sistem. Adapun *user* atau pengguna yang menggunakan sistem terdiri dari 2 jenis *user*, yaitu pasien penderita gangguan kecemasan dan admin. Berikut ini adalah tabel *use case* pada sistem.

Tabel 4.28 Use Case Sistem

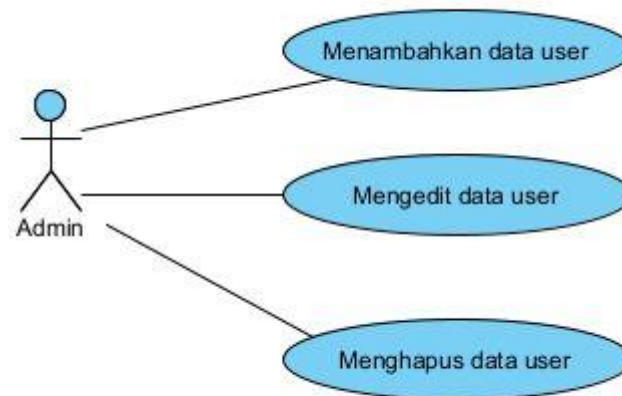
No	ID	Nama Use Case	Deskripsi
1	UC001	Melakukan registrasi	Pasien yang belum memiliki akun dapat melakukan pendaftaran atau <i>register</i> pada sistem. Data-data yang perlu diinputkan oleh pasien saat melakukan pendaftaran antara lain terdiri dari <i>username</i> , <i>password</i> , nama, alamat, jenis kelamin, dan email. Sementara admin yang belum memiliki akun tidak dapat melakukan pendaftaran pada sistem, admin perlu meminta bantuan dari admin lain untuk menambahkan akunnya ke sistem.
2	UC002	Melakukan <i>login</i>	Admin/pasien yang sudah memiliki akun dapat langsung melakukan <i>login</i> pada sistem dengan mengisi <i>username</i> dan <i>password</i> pada <i>form login</i> yang tersedia.

3	UC003	Melakukan <i>logout</i>	Admin/pasien yang sudah selesai menggunakan fitur-fitur pada sistem dapat melakukan <i>logout</i> .
4	UC004	Melihat daftar <i>user</i>	Admin dapat melihat daftar <i>user</i> baik <i>user</i> admin atau pasien yang terdapat dalam sistem.
5	UC005	Menambahkan data <i>user</i>	Admin dapat menambahkan data <i>user</i> baru ke dalam sistem.
6	UC006	Mengedit data <i>user</i>	Admin mengedit data <i>user</i> yang tersimpan dalam sistem.
7	UC007	Menghapus data <i>user</i>	Admin dapat menghapus data <i>user</i> dari sistem.
8	UC008	Melihat detail data <i>user</i>	Admin dapat melihat data <i>user</i> baik <i>user</i> admin atau pasien secara detail. Untuk <i>user</i> admin, detail data yang ditampilkan terdiri dari <i>username</i> , nama, jenis kelamin, email, dan alamat. Untuk <i>user</i> pasien, data yang ditampilkan berupa <i>username</i> , nama, jenis kelamin, alamat, email, dan hasil diagnosa jenis gangguan kecemasan yang diderita pasien beserta jawaban yang diinputkan oleh pasien pada masing-masing pertanyaan dari kuesioner HARS, bila pasien sudah melakukan <i>self-assessment</i> pada sistem.
9	UC009	Melihat daftar data latihan	Admin dapat melihat daftar data latihan yang tersimpan dalam sistem.
10	UC010	Menambahkan data latihan	Admin dapat menambahkan data latihan baru berupa jawaban kuesioner HARS dan hasil diagnosa ke dalam sistem.
11	UC011	Mengedit data latihan	Admin dapat mengedit data latihan pada sistem.

12	UC012	Menghapus data latihan	Admin dapat menghapus data latihan dari sistem.
13	UC013	Melihat detail data latihan	Admin dapat melihat detail data latihan yang terdiri dari jawaban kuesioner HARS dan hasil diagnosa.
14	UC014	Melihat daftar data uji	Admin dapat melihat daftar data uji yang terdapat dalam sistem.
15	UC015	Menambahkan data uji	Admin dapat menambahkan data uji baru ke dalam sistem berupa jawaban kuesioner HARS dan target diagnosa.
16	UC016	Mengedit data uji	Admin dapat mengedit data uji pada sistem.
17	UC017	Menghapus data uji	Admin dapat menghapus data uji dari sistem.
18	UC018	Melihat detail data uji	Admin dapat melihat detail data uji berupa jawaban kuesioner HARS dan target diagnosa.
19	UC019	Melakukan proses <i>backpropagation</i>	Sebelum melakukan proses <i>backpropagation</i> , admin perlu memastikan bahwa data latihan dan data uji telah diinputkan ke dalam sistem dan telah tersimpan dalam <i>database</i> . Selanjutnya admin dapat menginputkan data <i>learning rate</i> , <i>momentum</i> , <i>epoch</i> , <i>target error</i> , jumlah <i>neuron hidden layer</i> , dan perbandingan jumlah data latihan dan data uji yang akan digunakan pada sistem untuk melakukan proses <i>backpropagation</i> .
20	UC020	Melihat hasil pengujian <i>backpropagation</i>	Admin dapat melihat hasil diagnosa pada data uji dan melihat nilai akurasi yang dihasilkan sistem dalam melakukan pengujian <i>backpropagation</i> untuk mendiagnosa jenis gangguan kecemasan.

21	UC021	Melihat detail hasil pengujian <i>backpropagation</i>	Admin dapat melihat detail hasil pengujian <i>backpropagation</i> yang terdiri dari data jawaban kuesioner HARS, data target diagnosa, data hasil diagnosa, skor <i>neuron output 1</i> , skor <i>neuron output 2</i> , dan skor <i>neuron output 3</i> .
22	UC022	Melakukan <i>self-assessment</i>	Pasien menjawab pertanyaan dari kuesioner HARS pada sistem yang berhubungan dengan gejala kecemasan yang diderita, kemudian jawaban dari pasien akan dijadikan inputan dalam perhitungan menggunakan metode <i>backpropagation</i> untuk mendiagnosa jenis gangguan kecemasan yang diderita.
23	UC023	Melihat hasil diagnosa gangguan kecemasan	Setelah menjawab daftar pertanyaan kuesioner HARS, pasien dapat melihat hasil diagnosa gangguan kecemasan yang diderita.

Berikut ini adalah analisis *use case* pada sistem. Analisis *use case* dilakukan berdasarkan wewenang yang dimiliki oleh masing-masing aktor pada sistem. Pada analisis *use case* ini, *use case* yang akan dibahas adalah *use case* dari aktor admin. Aktor admin memiliki wewenang dalam melakukan manajemen data latih, manajemen data uji, dan manajemen data *user*. Dalam proses manajemen data latih, manajemen data uji, dan manajemen data *user*, admin dapat melakukan penambahan, edit, atau penghapusan data latih, data uji, atau data *user*. Proses penambahan, edit, atau penghapusan data tersebut dikelompokkan menjadi satu modul kegiatan manajemen yang dilakukan oleh aktor admin. Berikut adalah gambar dari *use case* manajemen data *user*.



Gambar 4.4 *Use Case Diagram* Manajemen Data User

Berdasarkan *use case* di atas, manajemen data *user* yang dilakukan oleh admin terdiri dari menambahkan data *user*, mengedit data *user*, dan menghapus data *user*. Tabel berikut menunjukkan skenario *use case* menambahkan data *user* pada modul manajemen data *user*.

Tabel 4.29 Skenario *Use Case* Menambahkan Data User

Nama <i>Use Case</i> : Menambahkan data <i>user</i>	<i>Use Case</i> ID: UC005
Aktor: Admin	
Deskripsi <i>Use Case</i> : Admin menambahkan data <i>user</i> baru ke dalam sistem yang terdiri dari <i>username</i> , <i>password</i> , nama, jenis kelamin, alamat, email, dan <i>role user</i> .	
Kondisi Sebelumnya: Admin berada di halaman <i>List User</i> .	
Pemicu: Admin mengklik tombol Tambah <i>User</i> yang terdapat di halaman <i>List User</i> .	
Alur Normal: Admin mengklik tombol Tambah <i>User</i> yang terdapat di <i>sidebar</i> halaman <i>List User</i> . Selanjutnya sistem akan mengarahkan ke halaman Tambah Data <i>User</i> dan menampilkan <i>form</i> untuk menambahkan data <i>user</i> baru. <i>Form</i> terdiri dari kolom <i>username</i> , <i>password</i> , konfirmasi <i>password</i> , nama, jenis kelamin, alamat, email,	

dan <i>role user</i> . Setelah mengisikan <i>form</i> , admin mengklik tombol Tambah <i>User</i> untuk men- <i>submit</i> data ke sistem.
Kondisi Berhasil: Sistem akan mengarahkan ke halaman <i>List User</i> dan menampilkan pesan bahwa data <i>user</i> berhasil ditambahkan.
Alur Alternatif: Sistem akan menampilkan pesan <i>error</i> pada halaman Tambah <i>User</i> yang menunjukkan bahwa data <i>user</i> tidak berhasil ditambahkan.

Tabel berikut menunjukkan skenario *use case* mengedit data *user* pada modul manajemen data *user*.

Tabel 4.30 Skenario *Use Case* Mengedit Data *User*

Nama <i>Use Case</i> : Mengedit data <i>user</i>	<i>Use Case</i> ID: UC006
Aktor: Admin	
Deskripsi <i>Use Case</i> : Admin dapat mengedit data <i>user</i> baik <i>user</i> admin atau pasien yang terdapat dalam sistem. Data <i>user</i> yang dapat diedit adalah data <i>username</i> , <i>password</i> , nama, jenis kelamin, alamat, dan email <i>user</i> .	
Kondisi Sebelumnya: Admin berada di halaman <i>List User</i> .	
Pemicu: Admin mengklik tombol Edit yang terdapat dalam tabel daftar <i>user</i> di halaman <i>List User</i> .	
Alur Normal: Admin mengklik tombol Edit <i>User</i> yang terdapat di salah satu baris data <i>user</i> pada tabel daftar <i>user</i> di halaman <i>List User</i> . Sistem selanjutnya mengarahkan ke halaman Edit <i>User</i> dan menampilkan <i>form</i> untuk mengedit data <i>user</i> . Setelah mengedit data <i>user</i> , admin mengklik tombol Simpan <i>User</i> untuk menyimpan perubahan yang telah dilakukan pada data <i>user</i> ke dalam sistem.	
Kondisi Berhasil:	

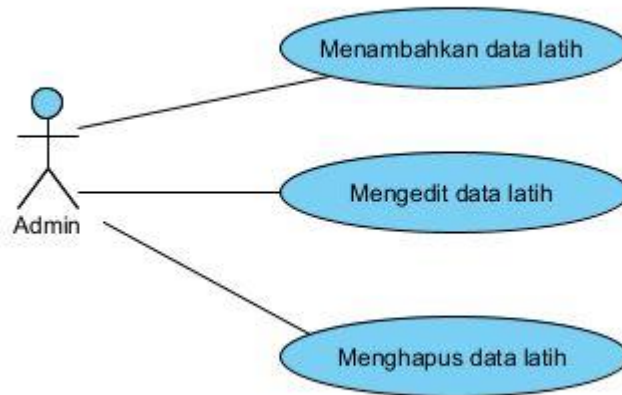
Sistem akan mengarahkan ke halaman <i>List User</i> dan menampilkan pesan bahwa data <i>user</i> berhasil diedit.
Alur Alternatif: Sistem akan menampilkan pesan <i>error</i> pada halaman <i>Edit User</i> yang menunjukkan bahwa data <i>user</i> gagal diedit.

Berikut adalah skenario *use case* menghapus data *user* pada modul manajemen data *user*.

Tabel 4.31 Skenario *Use Case* Menghapus Data *User*

Nama <i>Use Case</i> : Menghapus data <i>user</i>	<i>Use Case</i> ID: UC007
Aktor: Admin	
Deskripsi <i>Use Case</i> : Admin menghapus data <i>user</i> dari sistem.	
Kondisi Sebelumnya: Admin berada di halaman <i>List User</i> .	
Pemicu: Admin mengklik tombol Hapus pada tabel daftar <i>user</i> yang terdapat di halaman <i>List User</i> .	
Alur Normal: Admin mengklik tombol Hapus pada baris data yang ingin dihapus dari tabel daftar <i>user</i> pada halaman <i>List User</i> . Sistem selanjutnya akan menampilkan pesan konfirmasi penghapusan data <i>user</i> . Admin mengklik tombol OK untuk menghapus data <i>user</i> .	
Kondisi Berhasil: Sistem akan me- <i>refresh</i> halaman <i>List User</i> dan menampilkan pesan bahwa data <i>user</i> berhasil terhapus.	
Alur Alternatif: Sistem akan menampilkan pesan <i>error</i> yang menunjukkan bahwa data <i>user</i> gagal dihapus.	

Berikut adalah gambar yang menunjukkan *use case* manajemen data latih.



Gambar 4.5 *Use Case Diagram* Manajemen Data Latih

Pada gambar 4.5 di atas, modul manajemen data latih terdiri dari menambahkan data latih, mengedit data latih, dan menghapus data latih. Berikut adalah tabel yang menunjukkan skenario *use case* menambahkan data latih pada modul manajemen data latih.

Tabel 4.32 Skenario *Use Case* Menambahkan Data Latih

Nama <i>Use Case</i> : Menambahkan data latih	<i>Use Case</i> ID: UC010
Aktor: Admin	
Deskripsi <i>Use Case</i> : Admin menambahkan data latih baru ke dalam sistem yang berupa data jawaban kuesioner HARS dan data hasil diagnosa gangguan kecemasan.	
Kondisi Sebelumnya: Admin berada di halaman <i>List Data Training</i> .	
Pemicu: Admin mengklik tombol Tambah Data <i>Training</i> yang terdapat di halaman <i>List Data Training</i> .	
Alur Normal: Admin mengklik tombol Tambah Data <i>Training</i> yang terdapat di halaman <i>List Data Training</i> . Sistem akan mengarahkan ke halaman Tambah Data <i>Training</i> .	

<p>Pada halaman Tambah Data <i>Training</i>, admin dapat mengisi data jawaban dari 14 pertanyaan dari kuesioner HARS dan data hasil diagnosa gangguan kecemasan. Admin selanjutnya dapat mengklik tombol Tambah Data <i>Training</i> untuk menambahkan data latih baru ke dalam sistem.</p>
<p>Kondisi Berhasil: Sistem akan mengarahkan ke halaman <i>List Data Training</i> dan menampilkan pesan bahwa data latih berhasil ditambahkan.</p>
<p>Alur Alternatif: Sistem akan menampilkan pesan <i>error</i> pada halaman Tambah Data <i>Training</i> yang menunjukkan bahwa data latih tidak berhasil ditambahkan.</p>

Berikut adalah skenario *use case* mengedit data latih pada modul manajemen data latih.

Tabel 4.33 Skenario *Use Case* Mengedit Data Latih

Nama <i>Use Case</i> : Mengedit data latih	<i>Use Case</i> ID: UC011
Aktor: Admin	
Deskripsi <i>Use Case</i> : Admin dapat mengedit data latih yang terdapat dalam sistem.	
Kondisi Sebelumnya: Admin berada di halaman <i>List Data Training</i> .	
Pemicu: Admin mengklik tombol Edit yang terdapat dalam tabel daftar data latih di halaman <i>List Data Training</i> .	
Alur Normal: Admin mengklik tombol Edit yang terdapat di salah satu baris data latih pada tabel daftar data latih di halaman <i>List Data Training</i> . Sistem selanjutnya mengarahkan ke halaman Edit Data <i>Training</i> . Admin dapat mengedit data jawaban kuesioner HARS dan data hasil diagnosa gangguan kecemasan pada halaman Edit Data <i>Training</i> . Setelah mengedit data latih, admin mengklik tombol Simpan Data <i>Training</i> untuk menyimpan perubahan yang telah dilakukan ke dalam sistem.	
Kondisi Berhasil:	

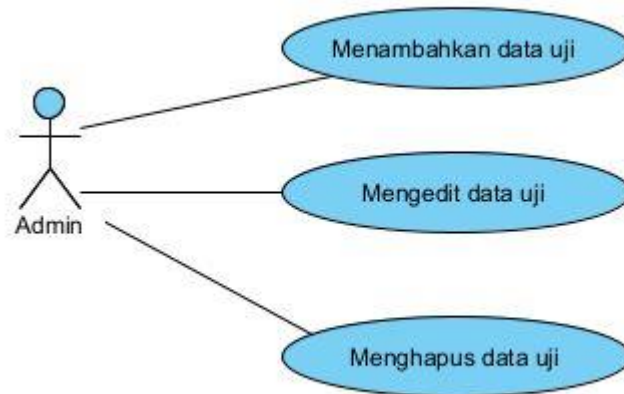
Sistem akan mengarahkan ke halaman <i>List Data Training</i> dan menampilkan pesan bahwa data <i>training</i> berhasil diedit.
Alur Alternatif: Sistem akan menampilkan pesan <i>error</i> pada halaman Edit Data <i>Training</i> yang menunjukkan bahwa data <i>training</i> tidak berhasil diedit.

Berikut adalah skenario *use case* menghapus data latih pada modul manajemen data latih.

Tabel 4.34 Skenario *Use Case* Menghapus Data Latih

Nama <i>Use Case</i> : Menghapus data latih	<i>Use Case</i> ID: UC012
Aktor: Admin	
Deskripsi <i>Use Case</i> : Admin menghapus data latih dari sistem.	
Kondisi Sebelumnya: Admin berada di halaman <i>List Data Training</i> .	
Pemicu: Admin mengklik tombol Hapus pada tabel daftar data latih yang terdapat di halaman <i>List Data Training</i> .	
Alur Normal: Admin mengklik tombol Hapus pada baris data yang ingin dihapus dari tabel daftar data latih pada halaman <i>List Data Training</i> . Sistem selanjutnya akan menampilkan pesan konfirmasi penghapusan data <i>user</i> . Admin mengklik tombol OK untuk menghapus data latih.	
Kondisi Berhasil: Sistem akan me- <i>refresh</i> halaman <i>List Data Training</i> dan menampilkan pesan bahwa data latih berhasil terhapus.	
Alur Alternatif: Sistem akan menampilkan pesan <i>error</i> yang menunjukkan bahwa data latih gagal dihapus.	

Berikut adalah *use case* manajemen data uji.



Gambar 4.6 *Use Case Diagram* Manajemen Data Uji

Pada gambar 4.6 di atas, modul manajemen data uji terdiri dari menambahkan data uji, mengedit data uji, dan menghapus data uji. Tabel berikut menunjukkan skenario *use case* menambahkan data uji pada modul manajemen data uji.

Tabel 4.35 Skenario *Use Case* Menambahkan Data Uji

Nama <i>Use Case</i> : Menambahkan data uji	<i>Use Case</i> ID: UC015
Aktor: Admin	
Deskripsi <i>Use Case</i> : Admin menambahkan data uji baru ke dalam sistem yang terdiri dari data jawaban kuesioner HARS dan data target diagnosa gangguan kecemasan.	
Kondisi Sebelumnya: Admin berada di halaman <i>List Data Testing</i> .	
Pemicu: Admin mengklik tombol Tambah Data <i>Testing</i> yang terdapat di halaman <i>List Data Testing</i> .	
Alur Normal: Admin mengklik tombol Tambah Data <i>Testing</i> yang terdapat di halaman <i>List Data Testing</i> . Sistem mengarahkan ke halaman Tambah Data <i>Testing</i> . Admin mengisikan data jawaban dari 14 pertanyaan dari kuesioner HARS dan data target	

<p>diagnosa gangguan kecemasan pada halaman Tambah Data <i>Testing</i>. Admin selanjutnya mengklik tombol Tambah Data <i>Testing</i> untuk menambahkan data uji baru ke dalam sistem.</p>
<p>Kondisi Berhasil: Sistem akan mengarahkan ke halaman <i>List Data Testing</i> dan menampilkan pesan bahwa data uji berhasil ditambahkan.</p>
<p>Alur Alternatif: Sistem akan menampilkan pesan <i>error</i> pada halaman Tambah Data <i>Testing</i> yang menunjukkan bahwa data uji gagal ditambahkan.</p>

Berikut adalah skenario *use case* mengedit data uji pada modul manajemen data uji.

Tabel 4.36 Skenario *Use Case* Mengedit Data Uji

Nama <i>Use Case</i> : Mengedit data uji	<i>Use Case</i> ID: UC016
Aktor: Admin	
<p>Deskripsi <i>Use Case</i>: Admin dapat mengedit data uji yang terdapat dalam sistem.</p>	
<p>Kondisi Sebelumnya: Admin berada di halaman <i>List Data Testing</i>.</p>	
<p>Pemicu: Admin mengklik tombol Edit yang terdapat dalam tabel daftar data uji di halaman <i>List Data Testing</i>.</p>	
<p>Alur Normal: Admin mengklik tombol Edit yang terdapat di salah satu baris data uji pada tabel daftar data uji di halaman <i>List Data Testing</i>. Sistem selanjutnya mengarahkan ke halaman Edit Data <i>Testing</i>. Admin dapat mengedit data jawaban kuesioner HARS dan data target diagnosa gangguan kecemasan pada halaman Edit Data <i>Testing</i>. Setelah mengedit data uji, admin mengklik tombol Simpan Data <i>Testing</i> untuk menyimpan perubahan yang telah dilakukan ke sistem.</p>	
<p>Kondisi Berhasil: Sistem akan mengarahkan ke halaman <i>List Data Testing</i> dan menampilkan pesan bahwa data <i>testing</i> berhasil diedit.</p>	

Alur Alternatif:

Sistem akan menampilkan pesan *error* pada halaman Edit Data *Testing* yang menunjukkan bahwa data *testing* tidak berhasil diedit.

Berikut adalah skenario *use case* menghapus data uji pada modul manajemen data uji.

Tabel 4.37 Skenario *Use Case* Menghapus Data Uji

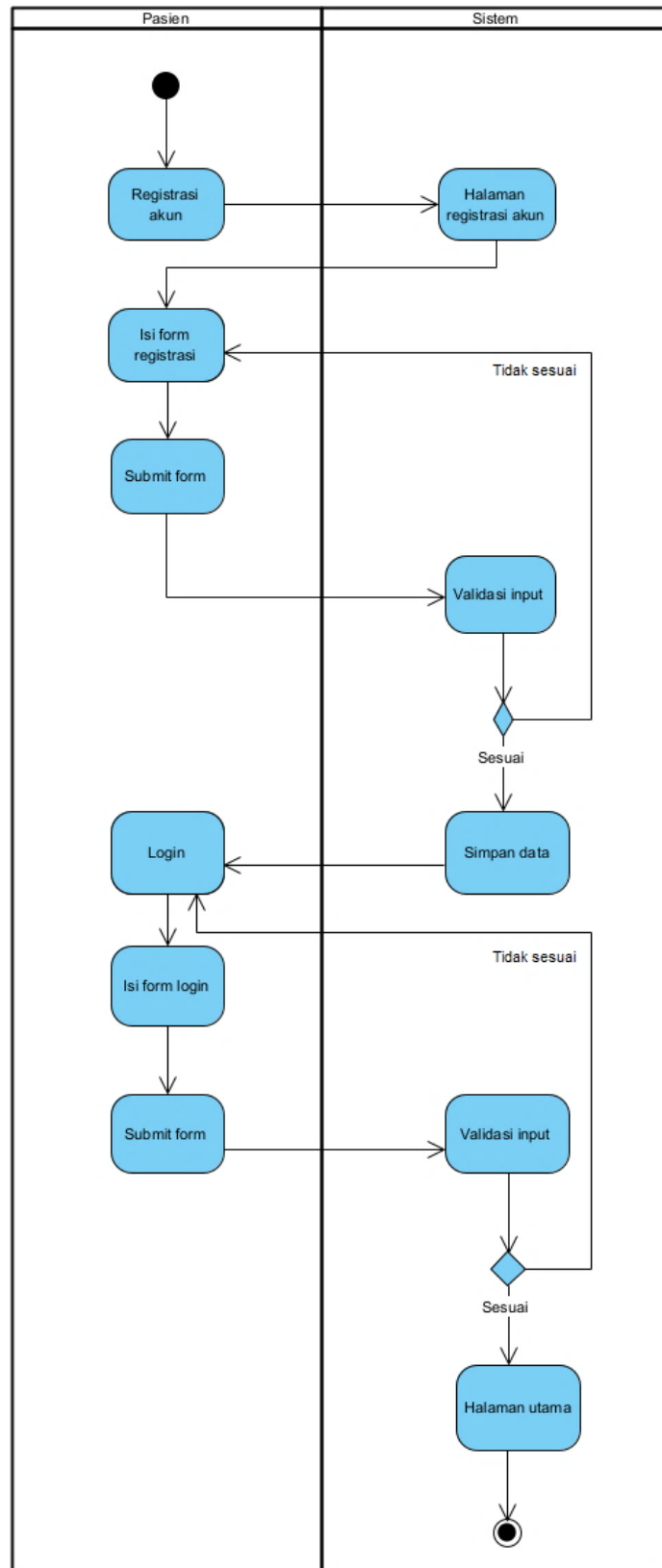
Nama <i>Use Case</i> : Menghapus data uji	<i>Use Case</i> ID: UC017
Aktor: Admin	
Deskripsi <i>Use Case</i> : Admin dapat menghapus data uji dari sistem.	
Kondisi Sebelumnya: Admin berada di halaman <i>List Data Testing</i> .	
Pemicu: Admin mengklik tombol Hapus pada tabel daftar data uji yang terdapat di halaman <i>List Data Testing</i> .	
Alur Normal: Admin mengklik tombol Hapus pada baris data yang ingin dihapus dari tabel daftar data uji pada halaman <i>List Data Testing</i> . Sistem selanjutnya akan menampilkan pesan konfirmasi penghapusan data <i>user</i> . Admin mengklik tombol OK untuk menghapus data uji.	
Kondisi Berhasil: Sistem akan me- <i>refresh</i> halaman <i>List Data Testing</i> dan menampilkan pesan bahwa data uji berhasil terhapus.	
Alur Alternatif: Sistem akan menampilkan pesan <i>error</i> yang menunjukkan bahwa data uji gagal dihapus.	

4.2.3. Activity Diagram

Activity diagram adalah diagram yang menggambarkan bagaimana aktivitas dari suatu sistem dijalankan. Adapun *activity diagram* pada sistem pakar antara lain sebagai berikut:

a. *Activity Diagram* Registrasi Pasien

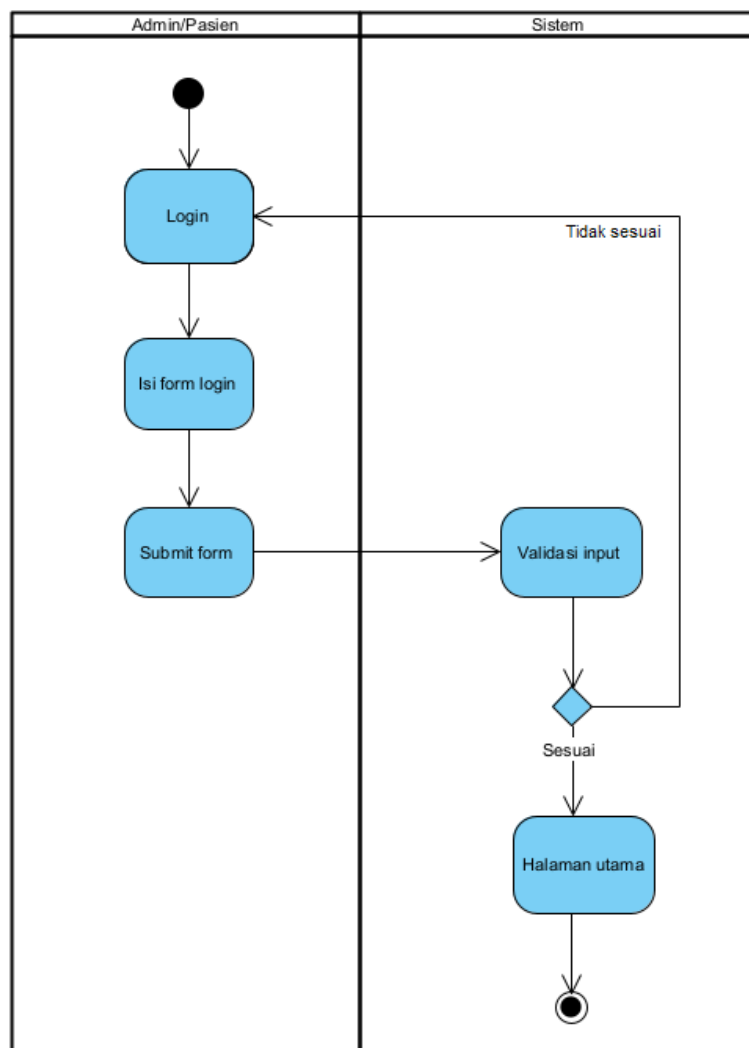
Activity diagram ini menggambarkan proses saat pasien melakukan registrasi akun pada sistem. Pasien mengklik tombol registrasi akun, lalu selanjutnya sistem akan menampilkan halaman registrasi akun. Pasien kemudian dapat mengisi data pada *form* registrasi yang terdiri dari *username*, *password*, nama, jenis kelamin, alamat, dan email. Pasien selanjutnya dapat men-*submit form* registrasi. Sistem akan melakukan validasi input data yang telah dimasukkan oleh pasien pada *form* registrasi. Bila input data yang dimasukkan sesuai, maka data akan disimpan ke dalam *database* dan otomatis sistem akan mengarahkan pasien ke halaman *login*. Namun bila input data yang dimasukkan tidak sesuai, maka pasien perlu melakukan registrasi ulang. Setelah registrasi yang dilakukan berhasil, pasien dapat melakukan *login* dengan memasukkan *username* dan *password* pada *form login*, lalu pasien men-*submit form login*. Sistem kemudian melakukan proses validasi untuk memastikan *username* dan *password* yang dimasukkan cocok sesuai dengan data yang ada pada *database*. Jika sesuai, maka sistem otomatis akan mengarahkan *user* masuk ke dalam sistem. Jika tidak sesuai, maka sistem akan mengarahkan pasien kembali ke halaman *login* supaya pasien melakukan *login* ulang. Gambar 4.7 berikut merupakan gambar *activity diagram* untuk registrasi pasien:



Gambar 4.7 Activity Diagram Registrasi Akun

b. *Activity Diagram Login*

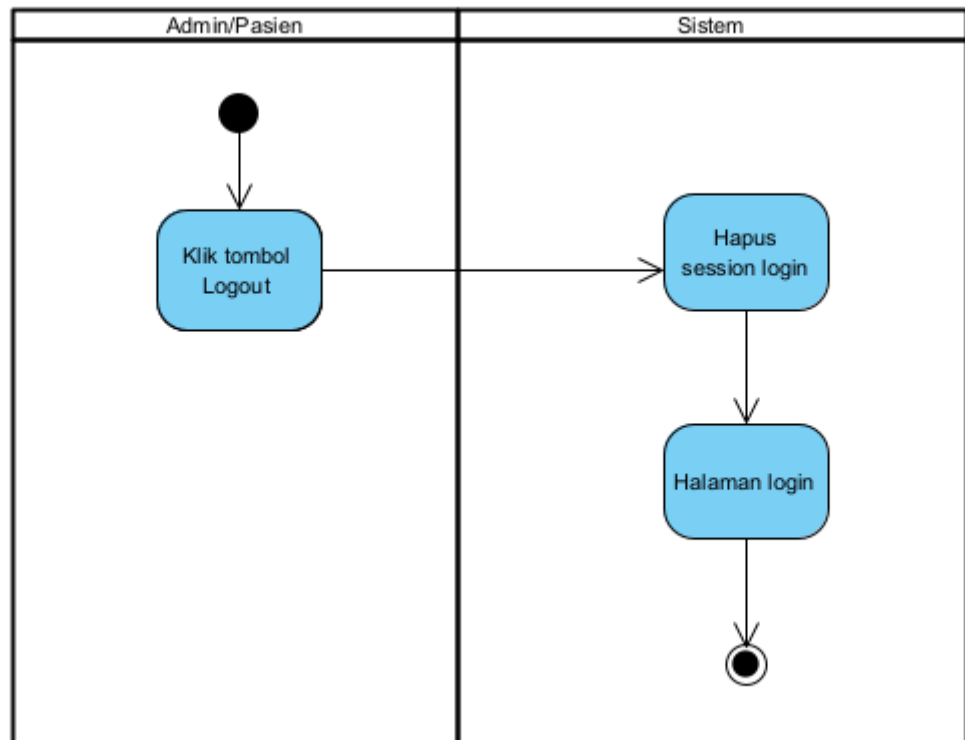
Activity diagram ini menggambarkan langkah-langkah proses *login* pada sistem pakar berbasis web. *User* sistem baik admin atau pasien akan masuk ke halaman *login*, kemudian *user* memasukkan *username* dan *password* di *form login*. Setelah men-*submit form login*, data *username* dan *password* yang telah dimasukkan akan divalidasi berdasarkan data yang ada di *database*. Jika data *username* dan *password* tidak cocok, maka sistem akan mengarahkan *user* kembali ke halaman *login* sehingga *user* dapat melakukan *login* ulang. Namun jika *username* dan *password* yang dimasukkan cocok, maka *login* berhasil dan *user* akan diarahkan masuk ke halaman utama sistem.



Gambar 4.8 *Activity Diagram Login*

c. *Activity Diagram Logout*

Activity diagram ini menggambarkan proses *logout* yang dilakukan *user* pada sistem. *User* akan mengklik tombol *logout* untuk *logout* dari sistem. Sistem kemudian akan menghapus sesi *login user* dan menampilkan kembali halaman *login*. Gambar 4.9 berikut menunjukkan *activity diagram* *logout*:

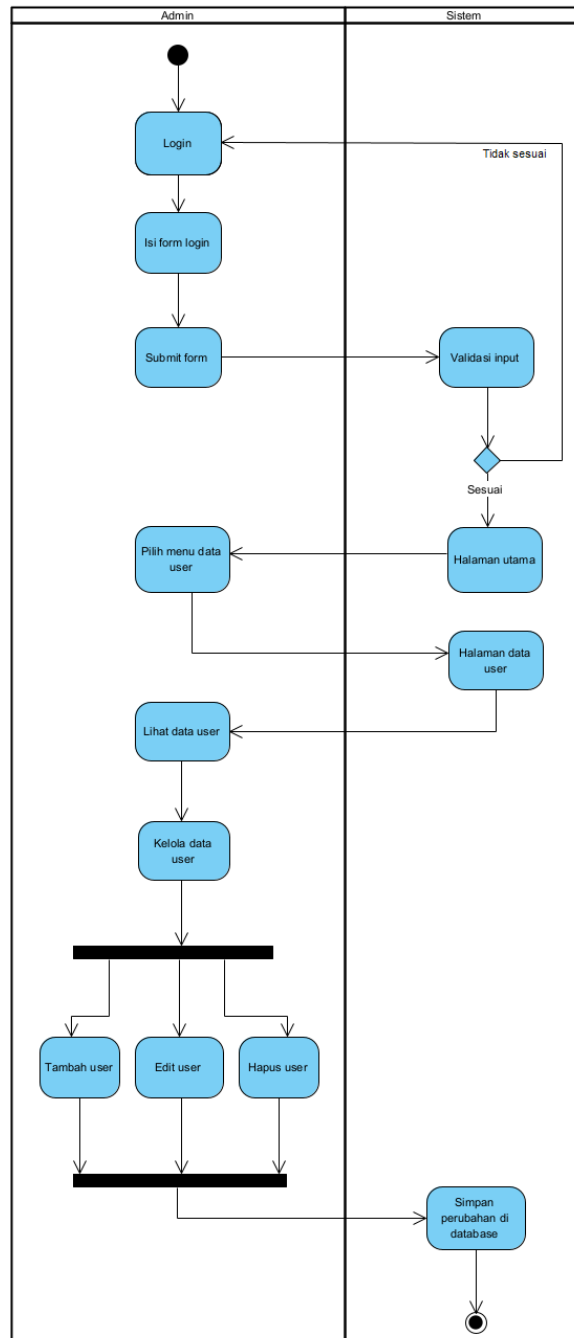


Gambar 4.9 *Activity Diagram Logout*

d. *Activity Diagram Manajemen Data User*

Manajemen data *user* pada sistem dilakukan oleh admin. Mula-mula admin akan melakukan *login*. Setelah *login*, admin akan diarahkan masuk ke halaman utama sistem, yaitu halaman proses pelatihan dan pengujian *backpropagation*. Admin memilih menu *user*, kemudian sistem akan menampilkan halaman *user*. Admin dapat melihat daftar *user* pada halaman *user*. Selanjutnya admin dapat mengelola data *user*. Pengelolaan data *user* yang dilakukan oleh admin dapat berupa penambahan, penghapusan, atau edit data *user*. Admin dapat mengedit data *user* yang terdiri dari nama, jenis kelamin, alamat, email, dan *password user*, namun admin tidak dapat

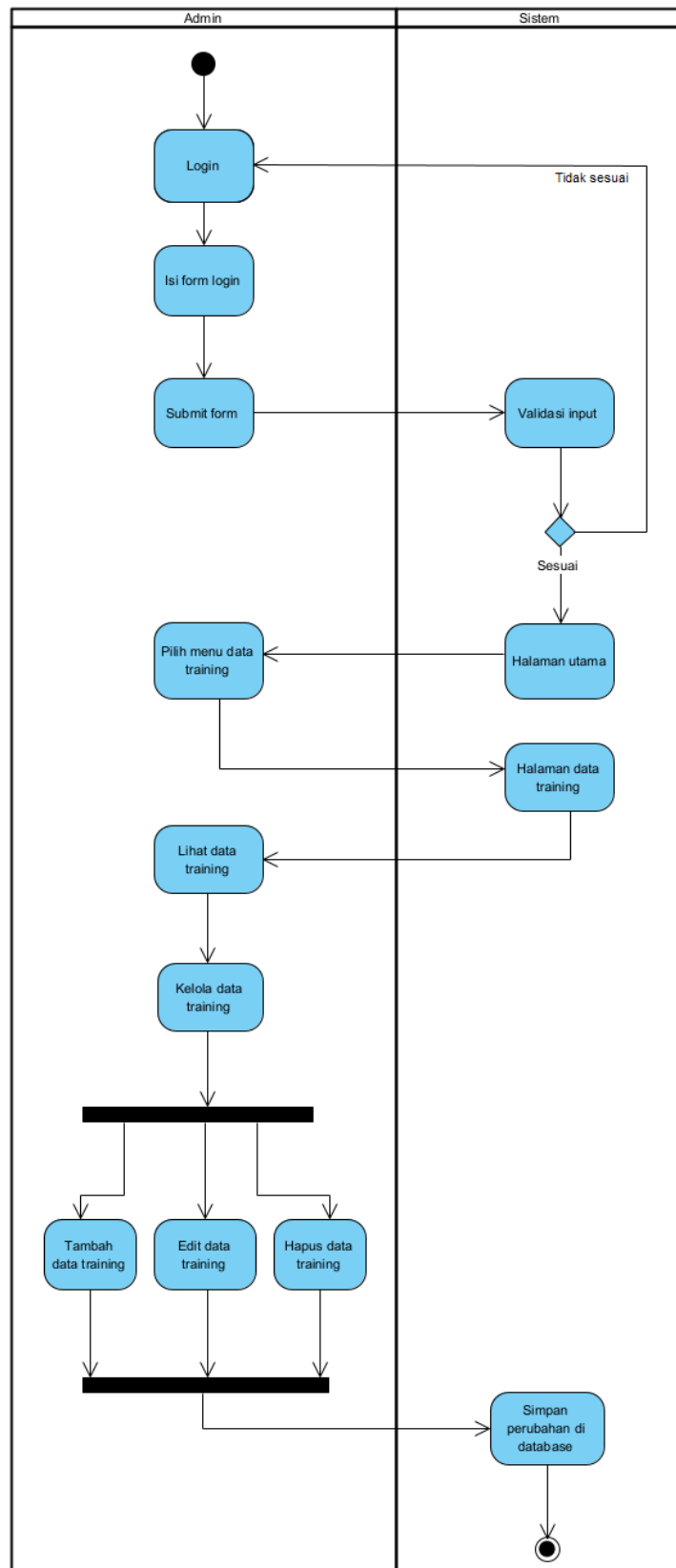
mengedit data jawaban pasien terhadap serangkaian pertanyaan yang digunakan untuk mendiagnosa jenis gangguan kecemasan yang diderita dan hasil diagnosa jenis gangguan kecemasan pasien berdasarkan perhitungan menggunakan metode *backpropagation*. Seluruh perubahan yang dilakukan oleh admin pada data *user* kemudian akan disimpan pada *database*. Gambar 4.10 berikut menunjukkan *activity diagram* manajemen data *user*:



Gambar 4.10 Activity Diagram Manajemen Data User

e. *Activity Diagram* Manajemen Data Latih

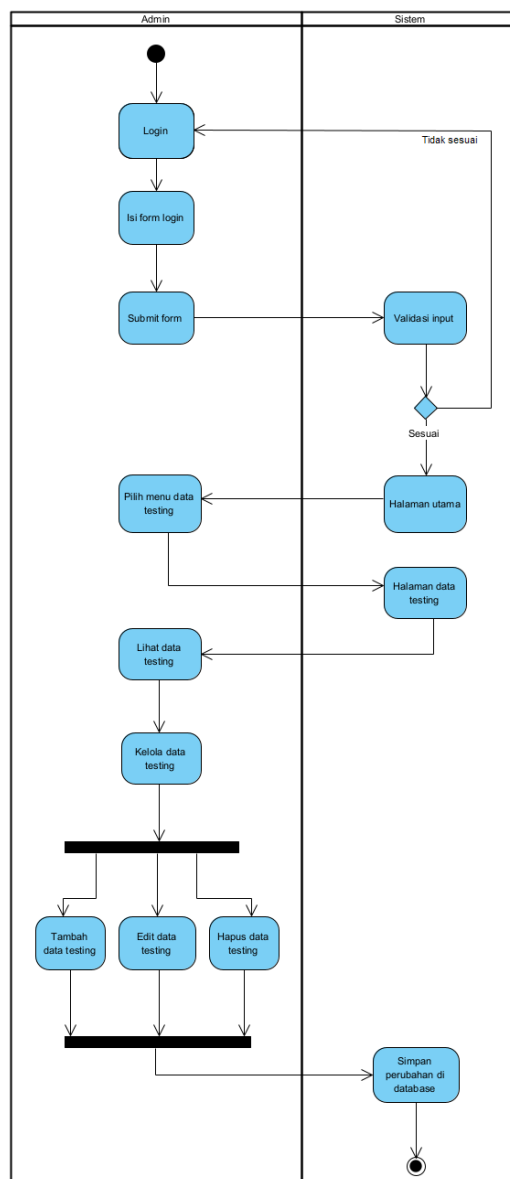
Admin dapat melakukan manajemen data latih pada sistem dengan menambahkan, mengedit, atau menghapus data latih. Admin akan melakukan *login*, lalu setelah *login* berhasil, admin akan diarahkan masuk ke halaman utama sistem yaitu halaman proses pelatihan dan pengujian *backpropagation*. Admin memilih menu data *training* dan selanjutnya sistem akan mengarahkan ke halaman data *training*. Untuk menambahkan data latih baru, maka admin dapat mengklik tombol tambah data *training*. Admin juga dapat mengedit data latih yang terdapat pada halaman data *training*. Data yang dapat diedit adalah data nilai masing-masing variabel gejala kecemasan dan data hasil diagnosa gangguan kecemasan. Selain mengedit data latih, admin juga dapat menghapus data latih pada halaman data *training*. Gambar 4.11 berikut menunjukkan *activity diagram* pada proses manajemen data latih yang dilakukan admin:



Gambar 4.11 Activity Diagram Manajemen Data Latih

f. *Activity Diagram* Manajemen Data Uji

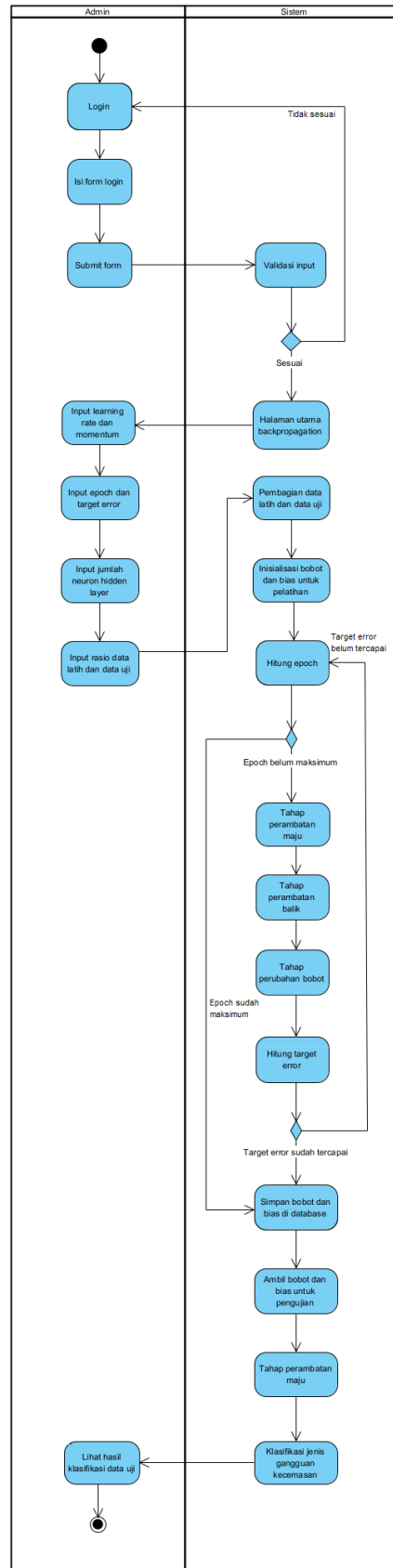
Activity diagram berikut menggambarkan proses saat admin melakukan manajemen data uji pada sistem. Admin melakukan *login*. Selanjutnya admin dapat memilih menu data *testing*. Admin dapat menambahkan data uji dengan mengklik tombol tambah data *testing* pada halaman data *testing*. Admin juga dapat mengubah atau mengedit nilai pada masing-masing variabel gangguan kecemasan dan hasil diagnosa gangguan kecemasan pada tiap data uji. Selain itu admin dapat menghapus data uji pada halaman data *testing*.



Gambar 4.12 *Activity Diagram* Manajemen Data Uji

g. *Activity Diagram* Proses Pelatihan dan Pengujian *Backpropagation*

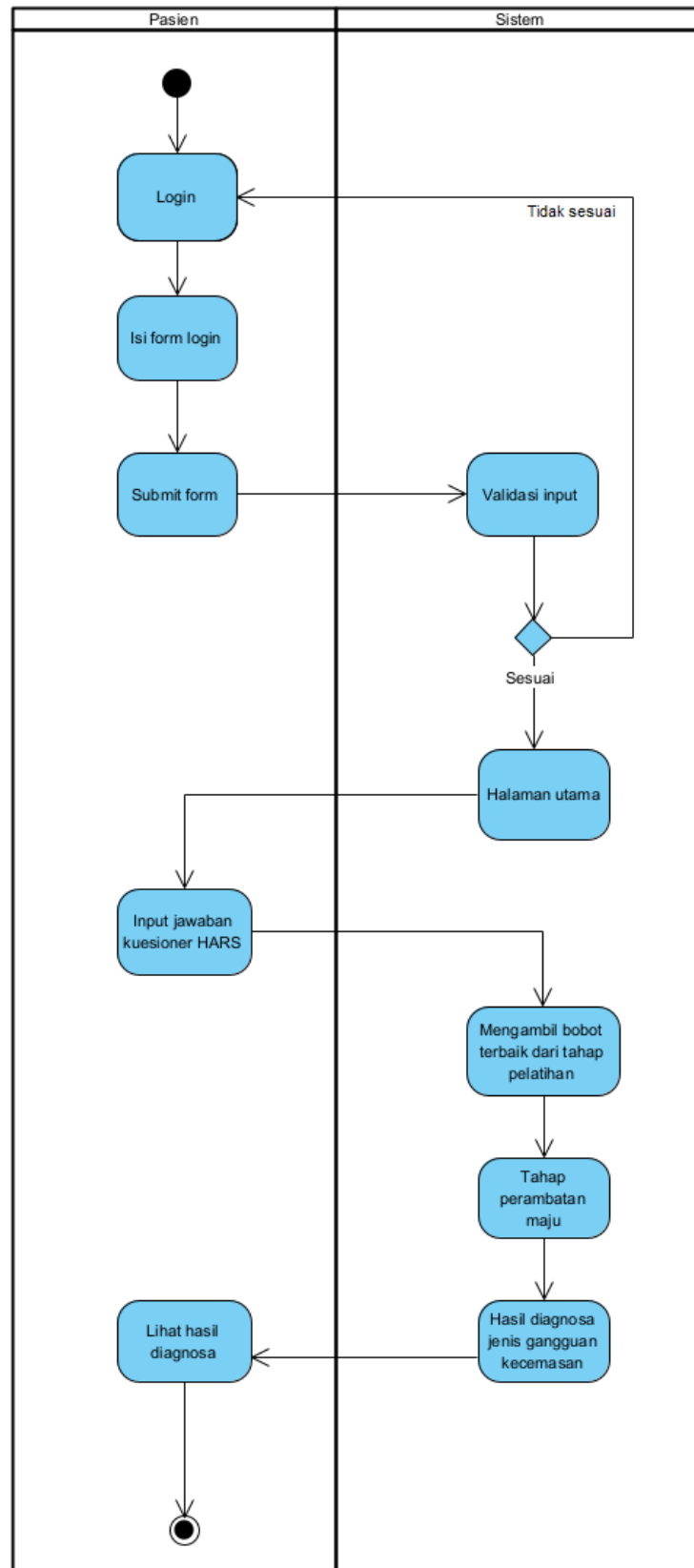
Activity diagram pada gambar 4.13 berikut menunjukkan proses pelatihan dan pengujian pada sistem menggunakan metode *backpropagation*. Proses pelatihan dilakukan sebagai pembelajaran bagi sistem untuk mengenali pola inputan dari data sehingga sistem dapat digunakan untuk mendiagnosa jenis gangguan kecemasan berdasarkan pola data yang diinputkan. Sementara proses pengujian diperlukan untuk mengetahui seberapa besar akurasi yang dihasilkan oleh sistem berdasarkan hasil pembelajaran dari proses pelatihan untuk mendiagnosa jenis gangguan kecemasan menggunakan *backpropagation*. Admin melakukan *login*, selanjutnya sistem akan mengarahkan admin menuju halaman proses pelatihan dan pengujian *backpropagation*. Pada halaman pengujian *backpropagation*, admin dapat menginputkan *learning rate*, *momentum*, *target error*, *epoch*, jumlah *neuron hidden layer*, dan perbandingan data yang digunakan dalam proses pengujian menggunakan *backpropagation*. Selama pelatihan menggunakan *backpropagation*, sistem akan menjalankan proses pembelajaran pada data latih melalui 3 tahap, yaitu tahap perambatan maju, tahap perambatan balik, dan tahap perubahan bobot. Ketiga tahap tersebut akan terus dijalankan hingga proses pelatihan mencapai nilai *epoch* atau *target error* yang telah diinputkan. Selanjutnya sistem akan menggunakan hasil pembelajaran dari proses pelatihan untuk mendiagnosa jenis gangguan kecemasan pada data uji. Setelah admin melakukan proses pengujian *backpropagation*, halaman pelatihan dan pengujian *backpropagation* akan menampilkan tabel data uji dan hasil akurasi pengujian menggunakan *backpropagation*.



Gambar 4.13 Activity Diagram Proses Pelatihan dan Pengujian *Backpropagation*

h. *Activity Diagram Self-Assessment* Pasien

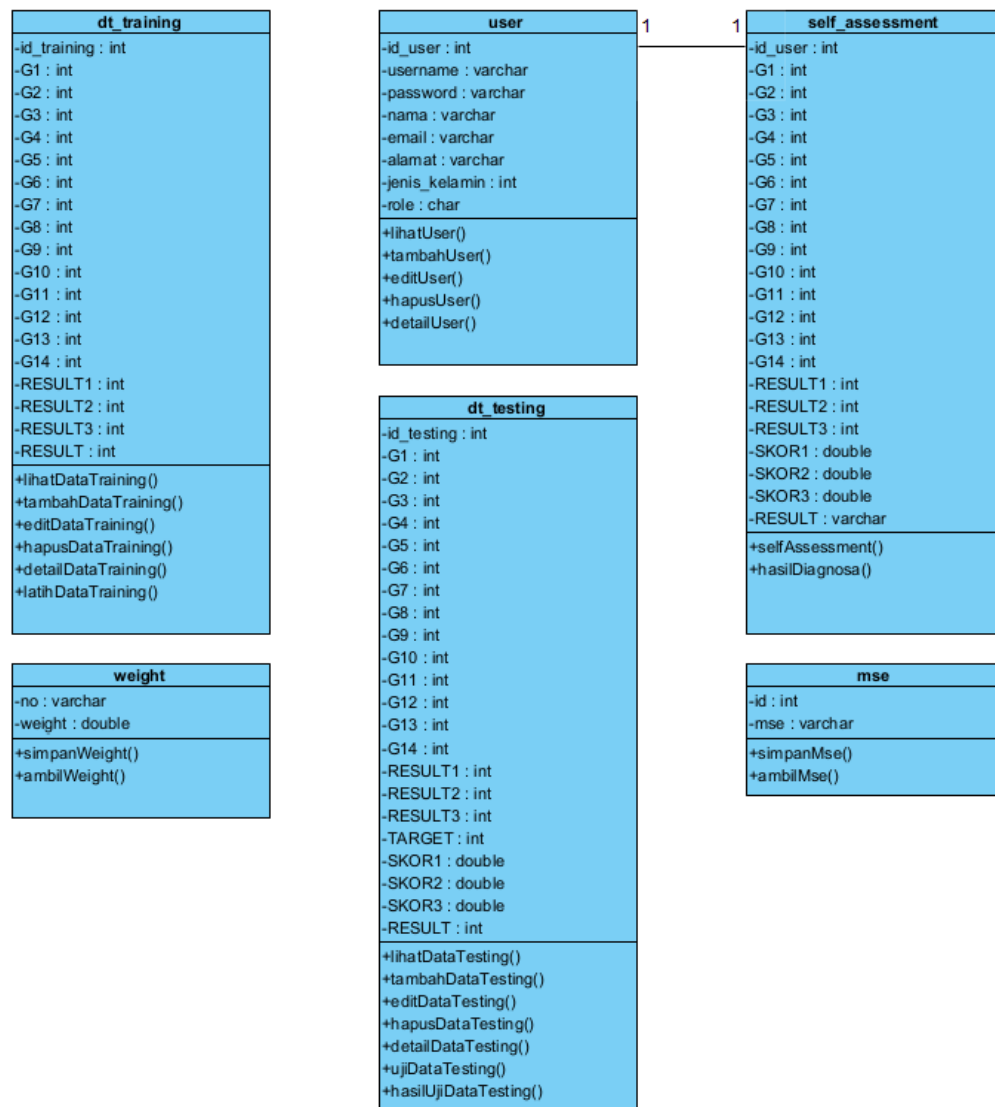
Pasien dapat melakukan tes secara individu pada sistem pakar untuk mengetahui jenis gangguan kecemasan yang diderita. Tes individu ini berdasarkan *self-assessment* atau penilaian secara pribadi dari diri pasien. Pasien mula-mula melakukan *login*, lalu sistem akan mengarahkan pasien ke halaman *self-assessment* untuk mulai mendiagnosa jenis gangguan kecemasan yang diderita. Pada halaman *self-assessment*, sistem akan menampilkan 14 pertanyaan dari kuesioner HARS yang mewakili tiap gejala kecemasan yang diderita pasien. Pasien menjawab dan men-*submit* tiap pertanyaan yang tersedia pada halaman *self-assessment*, kemudian sistem akan melakukan perhitungan menggunakan metode *backpropagation* pada input jawaban yang dimasukkan pasien untuk menentukan jenis gangguan kecemasan yang diderita pasien. Selama perhitungan menggunakan *backpropagation*, terdapat fase perambatan maju di mana pola data masukan akan dihitung maju dari lapisan input hingga lapisan *output* menggunakan bobot yang tersimpan di *database*. Setelah proses perhitungan selesai, sistem akan menampilkan *output* jenis gangguan kecemasan yang diderita pasien. Gambar 4.14 berikut menunjukkan *activity diagram self-assessment* yang dilakukan oleh pasien:



Gambar 4.14 Activity Diagram Self-Assessment Pasien

4.2.4. Class Diagram

Class diagram adalah diagram yang digunakan untuk memvisualisasikan struktur dalam suatu sistem dengan memodelkan atribut, objek, dan kelas serta hubungan antar kelas. Gambar berikut adalah gambar *class diagram* pada sistem pakar diagnosa gangguan kecemasan menggunakan metode *backpropagation*.



Gambar 4.15 *Class Diagram* Sistem

Terdapat 6 kelas pada *class diagram* di atas, yaitu *dt_training*, *user*, *self_assessment*, *weight*, *dt_testing*, dan *mse*. Pada kelas *dt_training*, terdapat atribut *id_training* yang merupakan atribut yang digunakan sebagai nomor id pada data *training* atau data latih. Sementara atribut *G1*, *G2*, *G3*, *G4*, *G5*, *G6*, *G7*, *G8*, *G9*, *G10*, *G11*, *G12*, *G13*, dan *G14* pada kelas *dt_training* merupakan atribut nilai

gejala kecemasan pada data latih berdasarkan 14 pertanyaan yang terdapat pada kuesioner HARS. Atribut RESULT1, RESULT2, dan RESULT3 berisi data kombinasi biner dari 3 *neuron output* yang mewakili kelas gangguan kecemasan dari masing-masing data latih. Atribut RESULT berisi nilai kelas *output* jenis gangguan kecemasan dari masing-masing data latih. Operasi atau *method* yang dapat dijalankan pada kelas *dt_training* antara lain adalah *lihatDataTraining()* untuk melihat data latih, *tambahDataTraining()* untuk menambahkan data latih baru ke dalam sistem, *editDataTraining()* untuk mengedit data latih, *hapusDataTraining()* untuk menghapus data latih, *detailDataTraining()* untuk menampilkan data latih secara detail, dan *latihDataTraining()* untuk menjalankan tahap pelatihan menggunakan metode *backpropagation* pada data latih.

Adapun kelas *user* memiliki atribut *id_user* yang berisi nomor id data *user*. Atribut lain yang terdapat pada kelas *user* adalah *username*, *password*, *nama*, *email*, *alamat*, *jenis_kelamin*, dan *role*. Atribut *username*, *password*, *nama*, *email*, *alamat*, dan *jenis_kelamin* masing-masing digunakan untuk menyimpan data *username*, *password*, *nama*, *email*, *alamat*, dan *jenis kelamin user*. Sementara atribut *role* digunakan untuk menyimpan data *role user*, yang mana *role user* dibagi menjadi 2 jenis yaitu *admin* dan *pasien*. Pada kelas *user* terdapat 4 operasi yang dapat dilakukan, yaitu operasi *lihatUser()* untuk melihat data *user*, *tambahUser()* untuk menambahkan data *user* baru, *editUser()* untuk mengedit data *user*, *hapusUser()* untuk menghapus data *user*, dan *detailUser()* untuk melihat detail data *user*.

Pada kelas *self_assessment* memiliki satu atribut yang sama dengan kelas *user*, yaitu atribut *id_user*. Atribut *id_user* pada kelas *self_assessment* digunakan untuk menyimpan data id *user* yang telah melakukan proses *self-assessment*. Kelas *user* dan *self_assessment* memiliki hubungan *one-to-one* melalui atribut *id_user*. Kedua kelas tersebut memiliki hubungan *one-to-one* karena tiap *user* hanya dapat melakukan satu kali *self-assessment* dan satu data *self-assessment* hanya dapat dimiliki oleh satu *user*. Atribut lain yang terdapat pada kelas *self_assessment* adalah atribut G1, G2, G3, G4, G5, G6, G7, G8, G9, G10, G11, G12, G13, dan G14 yang berisi nilai jawaban pasien pada kuesioner HARS. Atribut RESULT1, RESULT2, dan RESULT3 menunjukkan kombinasi biner dari nilai 3 *neuron output* selama proses *self-assessment* menggunakan metode *backpropagation*. Atribut SKOR1,

SKOR2, dan SKOR3 menunjukkan skor yang dihasilkan pada masing-masing *neuron output* dari proses *self-assessment* menggunakan metode *backpropagation*. Atribut RESULT berisi kelas hasil diagnosa gangguan kecemasan pasien. Operasi yang dapat dilakukan pada kelas ini adalah selfAssessment() untuk mendiagnosa jenis gangguan kecemasan yang diderita pasien dan hasilDiagnosa() untuk menampilkan hasil diagnosa gangguan kecemasan pasien pada sistem.

Kelas *weight* terdiri dari 2 atribut, yaitu *no* yang menunjukkan nomor dari bobot dan *weight* yang berisi nilai bobot. Operasi yang dapat dilakukan pada kelas *weight* adalah simpanWeight() untuk menyimpan data bobot serta operasi ambilWeight() untuk mengambil data bobot sehingga bobot dapat digunakan saat proses pelatihan dan *self-assessment*.

Kelas *dt_testing* memiliki atribut yang hampir sama dengan kelas *dt_training*, hanya saja pada kelas *dt_testing*, atribut yang digunakan untuk menyimpan nomor id data uji bernama *id_testing*. Pada kelas *dt_testing* juga terdapat penambahan atribut TARGET yang berisi target hasil diagnosa jenis gangguan kecemasan pada tiap data uji dan atribut SKOR1, SKOR2, dan SKOR3 yang menunjukkan skor yang dihasilkan pada masing-masing *neuron output* dari pengujian *backpropagation*. Operasi yang dapat dilakukan pada kelas *dt_testing* adalah lihatDataTesting() untuk melihat data uji, tambahDataTesting() untuk menambah data uji, editDataTesting() untuk mengedit data uji, hapusDataTesting() untuk menghapus data uji, detailDataTesting() untuk melihat data uji secara lebih detail, ujiDataTesting() untuk menjalankan tahap pengujian menggunakan metode *backpropagation* pada data uji, dan hasilUjiDataTesting() untuk menampilkan akurasi perhitungan sistem dalam mendiagnosa jenis gangguan kecemasan berdasarkan hasil tahap pengujian.

Kelas *mse* terdiri dari 2 atribut dari *id* yang berisi nomor id dari data nilai *Mean Square Error* (MSE) dan atribut *mse* yang berisi nilai MSE dari hasil perhitungan *error backpropagation*. Operasi yang dapat dilakukan pada kelas *mse* adalah simpanMse() untuk menyimpan data MSE di dalam *database* dan ambilMse() untuk mengambil data nilai MSE.

4.2.5. Perancangan Struktur *Database*

Database yang digunakan pada sistem merupakan *database* MySQL dengan nama sistem-pakar. Terdapat 6 tabel pada *database* sistem-pakar. Keenam tabel tersebut antara lain adalah *dt_training*, *dt_testing*, *user*, *self_assessment*, *mse*, dan *weight*. Keenam tabel tersebut digunakan untuk menyimpan data yang diperlukan pada sistem pakar untuk mendiagnosa jenis gangguan kecemasan menggunakan metode *backpropagation*. Berikut ini adalah deskripsi dari masing-masing tabel yang terdapat pada *database* sistem-pakar.

1. Tabel *dt_training*

Tabel ini merupakan tabel yang digunakan untuk menyimpan data latihan yang dibutuhkan untuk proses pelatihan pada sistem. *Primary key* pada tabel ini adalah *id_training* yang menyimpan id dari masing-masing data latihan. Adapun *primary key* merupakan kolom yang mengidentifikasi tiap data pada tabel dan nilai pada *primary key* harus bersifat unik atau berbeda antara satu dengan lainnya. Berikut ini adalah tabel yang menunjukkan struktur dari tabel *dt_training*.

Tabel 4.38 Struktur Tabel *dt_training*

No	Nama field	Type and length	Deskripsi
1	<i>id_training</i>	int(11)	Nilai id dari tiap data latihan
2	G1	int(1)	Nilai jawaban pasien untuk kelompok gejala perasaan cemas pada kuesioner HARS
3	G2	int(1)	Nilai jawaban pasien untuk kelompok gejala ketegangan pada kuesioner HARS
4	G3	int(1)	Nilai jawaban pasien untuk kelompok gejala ketakutan pada kuesioner HARS
5	G4	int(1)	Nilai jawaban pasien untuk kelompok gejala gangguan tidur pada kuesioner HARS
6	G5	int(1)	Nilai jawaban pasien untuk kelompok gejala gangguan kecerdasan pada kuesioner HARS
7	G6	int(1)	Nilai jawaban pasien untuk kelompok gejala perasaan depresi pada kuesioner HARS

8	G7	int(1)	Nilai jawaban pasien untuk kelompok gejala somatik/fisik (otot) pada kuesioner HARS
9	G8	int(1)	Nilai jawaban pasien untuk kelompok gejala somatik/fisik (sensorik) pada kuesioner HARS
10	G9	int(1)	Nilai jawaban pasien untuk kelompok gejala kardiovaskuler (jantung dan pembuluh darah) pada kuesioner HARS
11	G10	int(1)	Nilai jawaban pasien untuk kelompok gejala respiratori (pernafasan) pada kuesioner HARS
12	G11	int(1)	Nilai jawaban pasien untuk kelompok gejala gastrointestinal (pencernaan) pada kuesioner HARS
13	G12	int(1)	Nilai jawaban pasien untuk kelompok gejala urogenital (perkemihan dan kelamin) pada kuesioner HARS
14	G13	int(1)	Nilai jawaban pasien untuk kelompok gejala autoimun pada kuesioner HARS
15	G14	int(1)	Nilai jawaban pasien untuk kelompok gejala tingkah laku/sikap pada kuesioner HARS
16	RESULT1	int(1)	Nilai biner <i>neuron output</i> pertama pada lapisan <i>output</i>
17	RESULT2	int(1)	Nilai biner <i>neuron output</i> kedua pada lapisan <i>output</i>
18	RESULT3	int(1)	Nilai biner pada <i>neuron output</i> ketiga pada lapisan <i>output</i>
19	RESULT	int(1)	Nilai kelas gangguan kecemasan

2. Tabel dt_testing

Tabel ini merupakan tabel yang digunakan untuk menyimpan data uji yang digunakan untuk proses pengujian pada sistem. *Primary key* pada tabel ini adalah *id_testing* yang menyimpan id dari masing-masing data uji. Berikut ini adalah tabel yang menunjukkan struktur dari tabel *dt_testing*.

Tabel 4.39 Struktur Tabel dt_testing

No	Nama field	Type and length	Deskripsi
1	id_testing	int(11)	Nilai id dari tiap data latih
2	G1	int(1)	Nilai jawaban pasien untuk kelompok gejala perasaan cemas pada kuesioner HARS
3	G2	int(1)	Nilai jawaban pasien untuk kelompok gejala ketegangan pada kuesioner HARS
4	G3	int(1)	Nilai jawaban pasien untuk kelompok gejala ketakutan pada kuesioner HARS
5	G4	int(1)	Nilai jawaban pasien untuk kelompok gejala gangguan tidur pada kuesioner HARS
6	G5	int(1)	Nilai jawaban pasien untuk kelompok gejala gangguan kecerdasan pada kuesioner HARS
7	G6	int(1)	Nilai jawaban pasien untuk kelompok gejala perasaan depresi pada kuesioner HARS
8	G7	int(1)	Nilai jawaban pasien untuk kelompok gejala somatik/fisik (otot) pada kuesioner HARS
9	G8	int(1)	Nilai jawaban pasien untuk kelompok gejala somatik/fisik (sensorik) pada kuesioner HARS
10	G9	int(1)	Nilai jawaban pasien untuk kelompok gejala kardiovaskuler (jantung dan pembuluh darah) pada kuesioner HARS
11	G10	int(1)	Nilai jawaban pasien untuk kelompok gejala respiratori (pernafasan) pada kuesioner HARS
12	G11	int(1)	Nilai jawaban pasien untuk kelompok gejala gastrointestinal (pencernaan) pada kuesioner HARS
13	G12	int(1)	Nilai jawaban pasien untuk kelompok gejala urogenital (perkemihan dan kelamin) pada kuesioner HARS

14	G13	int(1)	Nilai jawaban pasien untuk kelompok gejala autoimun pada kuesioner HARS
15	G14	int(1)	Nilai jawaban pasien untuk kelompok gejala tingkah laku/sikap pada kuesioner HARS
16	RESULT1	int(1)	Nilai biner <i>neuron output</i> pertama pada lapisan <i>output</i>
17	RESULT2	int(1)	Nilai biner <i>neuron output</i> kedua pada lapisan <i>output</i>
18	RESULT3	int(1)	Nilai biner pada <i>neuron output</i> ketiga pada lapisan <i>output</i>
19	TARGET	int(1)	Nilai kelas target gangguan kecemasan
20	SKOR1	double	Skor <i>neuron output</i> pertama di lapisan <i>output</i> berdasarkan hasil pengujian <i>backpropagation</i>
21	SKOR2	double	Skor <i>neuron output</i> kedua di lapisan <i>output</i> berdasarkan hasil pengujian <i>backpropagation</i>
22	SKOR3	double	Skor <i>neuron output</i> ketiga di lapisan <i>output</i> berdasarkan hasil pengujian <i>backpropagation</i>
23	RESULT	int(1)	<i>Output</i> kelas gangguan kecemasan yang diderita pasien

3. Tabel *user*

Tabel ini menyimpan data *user*. *Primary key* dari tabel ini adalah *id_user* yang menyimpan id dari tiap data *user*. Berikut ini adalah struktur tabel *user*.

Tabel 4.40 Struktur Tabel *user*

No	Nama <i>field</i>	Type and length	Deskripsi
1	<i>id_user</i>	int(5)	Nilai id dari masing-masing <i>user</i>
2	<i>username</i>	varchar(16)	<i>Username</i> dari tiap <i>user</i>
3	<i>password</i>	varchar(50)	<i>Password</i> dari tiap <i>user</i>
4	nama	varchar(70)	Nama <i>user</i>
5	email	varchar(50)	Email <i>user</i>

6	alamat	varchar(100)	Alamat <i>user</i>
7	jenis_kelamin	int(1)	Jenis kelamin <i>user</i>
8	role	char(1)	<i>Role user</i> (admin/pasien)

4. Tabel self_assessment

Tabel ini merupakan tabel yang berfungsi menyimpan data jawaban pasien pada kuesioner HARS dan hasil diagnosa gangguan kecemasan yang diderita pasien berdasarkan perhitungan menggunakan metode *backpropagation*. *Primary key* dari tabel ini adalah *id_user*. Atribut *id_user* pada tabel *self_assessment* merupakan *foreign key* dari tabel *user*. *Foreign key* adalah kolom suatu tabel yang nilainya mengacu ke kolom dari tabel lain. Tabel *self_assessment* mengambil *foreign key* dari tabel *user* karena untuk menyimpan data *id user* pasien yang telah melakukan *self-assessment* pada tabel *self_assessment* maka perlu mengambil data *id_user* pasien yang terdapat di tabel *user*. Berikut adalah tabel yang menunjukkan struktur dari tabel *self_assessment*.

Tabel 4.41 Struktur Tabel *self_assessment*

No	Nama field	Type and length	Deskripsi
1	<i>id_user</i>	int(5)	Nilai <i>id user</i> pasien yang telah melakukan <i>self-assessment</i>
2	G1	int(1)	Nilai jawaban pasien untuk kelompok gejala perasaan cemas pada kuesioner HARS
3	G2	int(1)	Nilai jawaban pasien untuk kelompok gejala ketegangan pada kuesioner HARS
4	G3	int(1)	Nilai jawaban pasien untuk kelompok gejala ketakutan pada kuesioner HARS
5	G4	int(1)	Nilai jawaban pasien untuk kelompok gejala gangguan tidur pada kuesioner HARS
6	G5	int(1)	Nilai jawaban pasien untuk kelompok gejala gangguan kecerdasan pada kuesioner HARS

7	G6	int(1)	Nilai jawaban pasien untuk kelompok gejala perasaan depresi pada kuesioner HARS
8	G7	int(1)	Nilai jawaban pasien untuk kelompok gejala somatik/fisik (otot) pada kuesioner HARS
9	G8	int(1)	Nilai jawaban pasien untuk kelompok gejala somatik/fisik (sensorik) pada kuesioner HARS
10	G9	int(1)	Nilai jawaban pasien untuk kelompok gejala kardiovaskuler (jantung dan pembuluh darah) pada kuesioner HARS
11	G10	int(1)	Nilai jawaban pasien untuk kelompok gejala respiratori (pernafasan) pada kuesioner HARS
12	G11	int(1)	Nilai jawaban pasien untuk kelompok gejala gastrointestinal (pencernaan) pada kuesioner HARS
13	G12	int(1)	Nilai jawaban pasien untuk kelompok gejala urogenital (perkemihan dan kelamin) pada kuesioner HARS
14	G13	int(1)	Nilai jawaban pasien untuk kelompok gejala autoimun pada kuesioner HARS
15	G14	int(1)	Nilai jawaban pasien untuk kelompok gejala tingkah laku/sikap pada kuesioner HARS
16	RESULT1	int(1)	Nilai biner pada <i>neuron output</i> pertama di lapisan <i>output</i>
17	RESULT2	int(1)	Nilai biner pada <i>neuron output</i> kedua di lapisan <i>output</i>
18	RESULT3	int(1)	Nilai biner pada <i>neuron output</i> ketiga di lapisan <i>output</i>
19	SKOR1	double	Skor hasil perhitungan <i>backpropagation</i> pada <i>neuron output</i> pertama di lapisan <i>output</i>
20	SKOR2	double	Skor hasil perhitungan <i>backpropagation</i> pada <i>neuron output</i> kedua di lapisan <i>output</i>

21	SKOR3	double	Skor hasil perhitungan <i>backpropagation</i> pada <i>neuron output</i> ketiga di lapisan <i>output</i>
22	RESULT	varchar(100)	<i>Output</i> jenis gangguan kecemasan yang diderita pasien

5. Tabel mse

Tabel ini merupakan tabel yang menyimpan data *Mean Square Error* (MSE), nilai *error* yang dihasilkan pada proses perhitungan menggunakan *backpropagation*. *Primary key* pada tabel mse adalah id yang menyimpan id dari masing-masing data mse. Berikut adalah struktur dari tabel mse.

Tabel 4.42 Struktur Tabel mse

No	Nama field	Type and length	Deskripsi
1	id	int(5)	Nilai id dari masing-masing data MSE
2	mse	varchar(100)	Nilai MSE dalam bentuk <i>string/varchar</i>

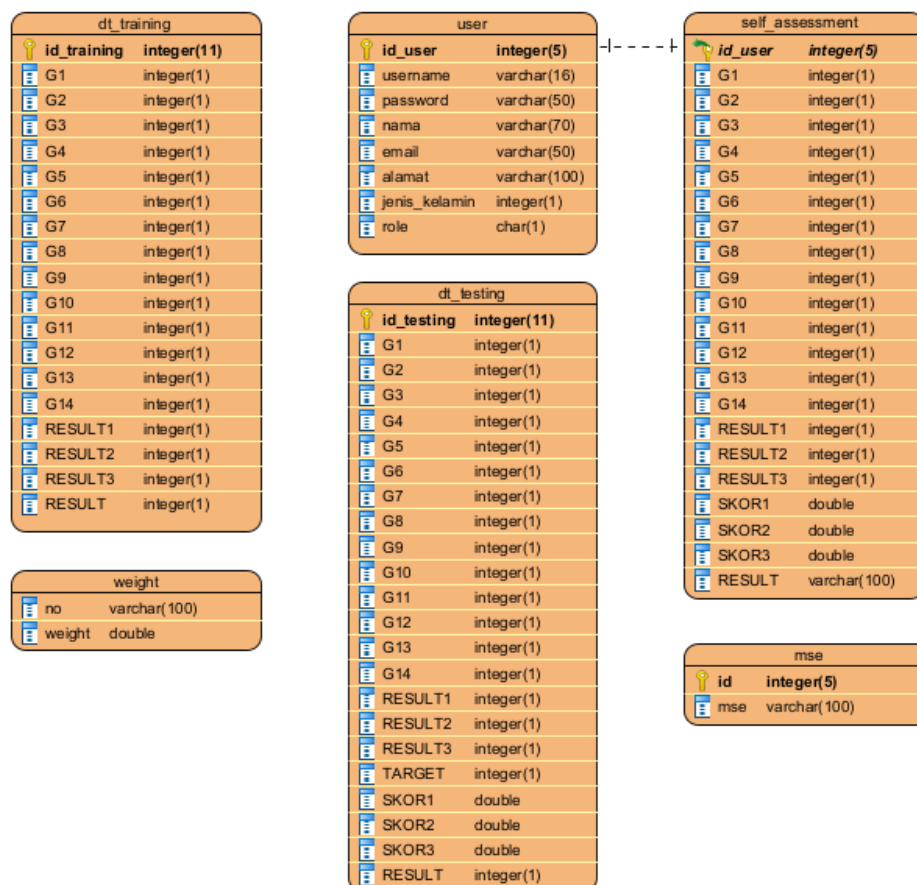
6. Tabel weight

Tabel ini merupakan tabel yang berfungsi menyimpan data bobot yang dihasilkan pada proses pelatihan menggunakan metode *backpropagation*. *Primary key* dari tabel ini adalah no yang berisi data nomor yang menunjukkan *neuron* keberapa saja yang saling dihubungkan dengan *weight* pada tabel. Berikut ini adalah struktur dari tabel *weight*.

Tabel 4.43 Struktur Tabel *weight*

No	Nama field	Type and length	Deskripsi
1	no	varchar(100)	Nomor yang menunjukkan <i>neuron</i> keberapa saja yang saling dihubungkan dengan <i>weight</i> yang terdapat pada tabel
2	weight	double	Nilai <i>weight</i> /bobot dari hasil pelatihan <i>backpropagation</i>

Berikut gambar *conceptual data model* (CDM) dari struktur *database* yang digunakan pada penelitian ini. CDM adalah diagram yang menggambarkan seluruh informasi yang terdapat dalam suatu *database* secara konseptual. Berdasarkan diagram CDM berikut, tabel *self_assessment* dan tabel *user* mempunyai relasi *one-to-one* satu sama lain karena *user* hanya dapat melakukan *self-assessment* sebanyak satu kali pada sistem. Tabel *self_assessment* memiliki *foreign key* sekaligus *primary key* berupa *id_user* yang menghubungkan ke tabel *user*.



Gambar 4.16 *Conceptual Data Model* (CDM) *Database*

4.2.6. Perancangan Antarmuka

Antarmuka sistem berfungsi sebagai sarana interaksi antara pengguna dengan sistem. Antarmuka sistem perlu dirancang untuk mempermudah proses implementasi sistem. Perancangan antarmuka sistem pada penelitian ini berupa rancangan halaman-halaman web sistem pakar.

4.2.6.1. Halaman *Login*

Perancangan halaman *login* pada web sistem pakar terdapat pada gambar berikut. Halaman *login* berisi *form* untuk *login* yang terdiri dari kolom *username* dan *password* yang perlu diisi oleh *user*. Di bawah *form login* terdapat tombol *Login* untuk masuk ke dalam sistem. Jika *user* belum memiliki akun, maka *user* dapat mengklik tombol *Registrasi* untuk mendaftar ke dalam sistem.



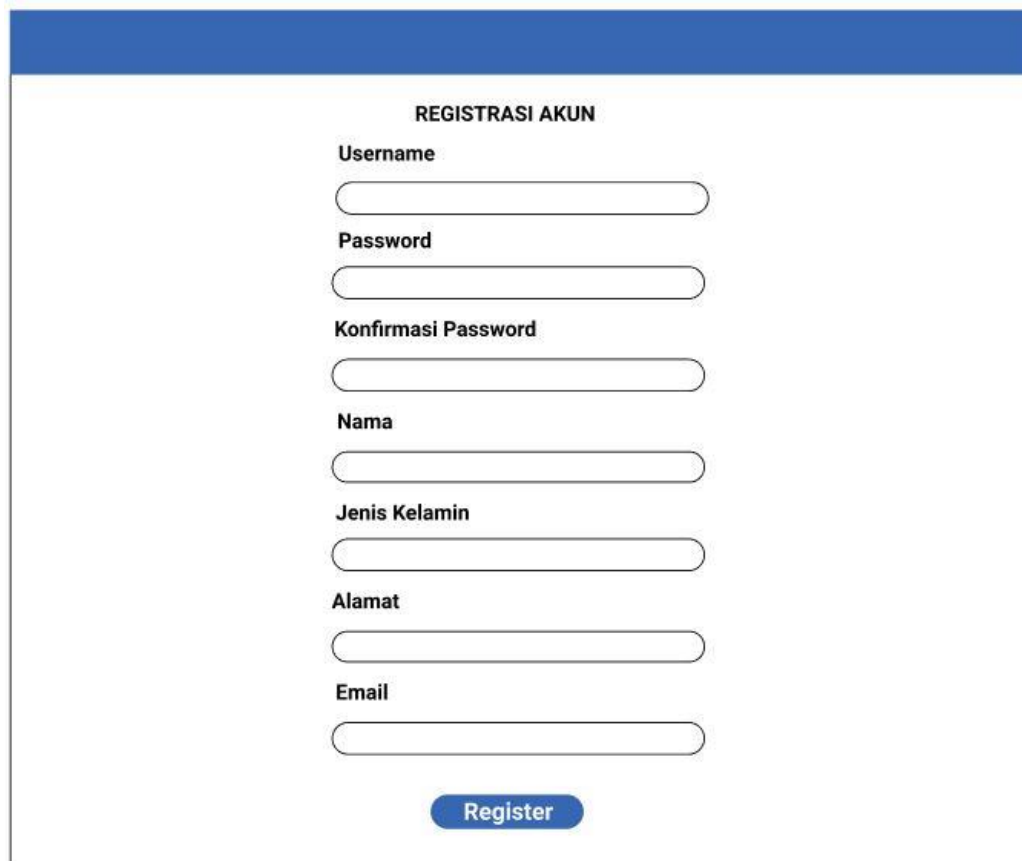
The image shows a web interface for a system titled "SISTEM PAKAR DIAGNOSA GANGGUAN KECEMASAN MENGGUNAKAN METODE BACKPROPAGATION". At the top right, there is a blue header bar with a white button labeled "Registrasi". The main content area is white and contains the following elements:

- Centered text: "SISTEM PAKAR DIAGNOSA GANGGUAN KECEMASAN" and "MENGGUNAKAN METODE BACKPROPAGATION".
- Section header: "LOGIN".
- Form fields: "Username" and "Password", each followed by a white input box with rounded ends.
- Button: A blue button labeled "Login" centered below the input fields.

Gambar 4.17 Halaman *Login*

4.2.6.2. Halaman Registrasi Akun

Halaman registrasi akun berisi *form* registrasi yang terdiri dari kolom *username*, *password*, konfirmasi *password*, nama, jenis kelamin, alamat, dan email. Di bawah *form* registrasi terdapat tombol *Registrasi* untuk mendaftarkan akun ke dalam sistem. Data *user* yang didaftarkan melalui halaman registrasi akan otomatis terdaftar sebagai *user* pasien.



The image shows a registration form with the following fields and a button:

- REGISTRASI AKUN**
- Username**:
- Password**:
- Konfirmasi Password**:
- Nama**:
- Jenis Kelamin**:
- Alamat**:
- Email**:
- Register**:

Gambar 4.18 Halaman Registrasi

4.2.6.3. Halaman *List Data Training*

Halaman *list data training* menampilkan daftar data *training* dalam bentuk tabel. Pada tabel data *training* terdiri dari lima kolom. Kolom pertama berisi nomor dari data *training* dan kolom kedua terdiri dari hasil diagnosa pada data *training*. Sementara kolom ketiga, keempat, dan kelima pada tabel masing-masing berisi tombol Detail, Edit, dan Hapus. Bila tombol Detail diklik, maka sistem menampilkan detail dari data *training*. Adapun tombol Edit dan Hapus masing-masing berfungsi untuk mengedit dan menghapus data *training*. Di bagian atas tabel data *training* terdapat tombol Tambah Data *Training* untuk menambahkan data *training* baru ke dalam sistem. Pada bagian paling atas halaman *list data training* terdapat menu *user*, *data training*, *data testing*, dan *backpropagation*.

User Data Training Data Testing Backpropagation				
LIST DATA TRAINING				
Tambah Data Training				
No	Hasil Diagnosa	Detail	Edit	Hapus
1	General Anxiety Disorder	Detail	Edit	Hapus
2	Panic Disorder	Detail	Edit	Hapus
3	Social Anxiety Disorder	Detail	Edit	Hapus
4	Specific Phobia	Detail	Edit	Hapus
5	Obsessive Compulsive	Detail	Edit	Hapus
6	Post Traumatic Stress	Detail	Edit	Hapus

Gambar 4.19 Halaman *List Data Training*

4.2.6.4. Halaman Tambah Data *Training*

Halaman tambah data *training* berisi 14 pertanyaan dari kuesioner HARS beserta pilihan jawaban dari masing-masing pertanyaan tersebut. Pada bagian bawah dari 14 pertanyaan dan pilihan jawaban kuesioner HARS terdapat pilihan hasil diagnosa data *training*. Setelah dilakukan pengisian kuesioner HARS dan hasil diagnosa data *training*, maka data *training* dapat diinputkan ke dalam sistem dengan mengklik tombol Tambah Data *Training*.

TAMBAH DATA TRAINING

Pertanyaan 1
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 2
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 3
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 4
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 14
 Tidak ada Ringan Sedang Berat Sangat berat

Hasil Diagnosa

General Anxiety Disorder
 Panic Disorder
 Social Anxiety Disorder
 Specific Phobia
 Obsessive Compulsive Disorder
 Post Traumatic Stress Disorder

Tambah Data Training

Gambar 4.20 Halaman Tambah Data *Training*

4.2.6.5. Halaman Edit Data *Training*

Jawaban dari pertanyaan yang terdapat di kuesioner HARS serta data hasil diagnosa pada data *training* dapat diubah melalui halaman edit data *training*. Untuk menyimpan perubahan data *training* dapat menggunakan tombol Simpan Data *Training*.

EDIT DATA TRAINING

Pertanyaan 1
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 2
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 3
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 4
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 14
 Tidak ada Ringan Sedang Berat Sangat berat

Hasil Diagnosa

General Anxiety Disorder
 Panic Disorder
 Social Anxiety Disorder
 Specific Phobia
 Obsessive Compulsive Disorder
 Post Traumatic Stress Disorder

[Simpan Data Training](#)

Gambar 4.21 Halaman Edit Data *Training*

4.2.6.6. Halaman Detail Data *Training*

Halaman detail data *training* menampilkan detail data *training* dalam bentuk tabel yang terdiri dari 3 kolom. Kolom pertama berisi kode dari kelompok gejala berdasarkan kuesioner HARS. Kolom kedua berisi data kelompok gejala dari kuesioner HARS. Kolom tiga berisi jawaban atau tingkat keparahan pada masing-masing kelompok gejala. Di bawah tabel detail data *training* terdapat hasil diagnosa data *training* beserta tombol Edit untuk mengedit data *training* dan tombol Hapus untuk menghapus data *training*.

DETAIL DATA TRAINING		
Kode	Gejala	Keterangan
G1	Kelompok gejala 1	BERAT
G2	Kelompok gejala 2	RINGAN
G3	Kelompok gejala 3	RINGAN
G14	Kelompok gejala 14	BERAT

HASIL DIAGNOSA
General Anxiety Disorder

[Edit](#) [Hapus](#)

Gambar 4.22 Halaman Detail Data *Training*4.2.6.7. Halaman *List Data Testing*

Halaman *list data testing* menampilkan daftar data *testing* dalam bentuk tabel yang terdiri dari lima kolom. Kolom pertama berisi nomor data *testing* dan kolom kedua berisi target diagnosa data *testing*. Sementara kolom ketiga, keempat, dan kelima masing-masing berisi tombol Detail untuk melihat detail data *testing*, tombol Edit untuk mengubah data *testing*, dan tombol Hapus untuk menghapus data *testing*. Di bagian atas tabel *list data testing* terdapat tombol untuk menambahkan data *testing* baru.

User Data Training Data Testing Backpropagation				
LIST DATA TESTING				
Tambah Data Testing				
No	Target Diagnosa	Detail	Edit	Hapus
1	General Anxiety Disorder	Detail	Edit	Hapus
2	Panic Disorder	Detail	Edit	Hapus
3	Social Anxiety Disorder	Detail	Edit	Hapus
4	Specific Phobia	Detail	Edit	Hapus
5	Obsessive Compulsive	Detail	Edit	Hapus
6	Post Traumatic Stress	Detail	Edit	Hapus

Gambar 4.23 Halaman List Data *Testing*

4.2.6.8. Halaman Tambah Data *Testing*

Penambahan data *testing* baru pada sistem dapat dilakukan melalui halaman tambah data *testing*. Pada halaman tambah data *testing* terdapat daftar pertanyaan dari kuesioner HARS beserta pilihan jawaban dari masing-masing pertanyaan tersebut. Di bagian bawah daftar pertanyaan terdapat pilihan target diagnosa data *testing*. Setelah mengisikan seluruh jawaban dari 14 pertanyaan dari kuesioner HARS dan memilih target diagnosa pada data *testing*, maka data *testing* baru dapat disimpan ke dalam sistem dengan mengklik tombol Tambah Data *Testing*.

TAMBAH DATA TESTING

Pertanyaan 1
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 2
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 3
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 4
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 14
 Tidak ada Ringan Sedang Berat Sangat berat

Target Diagnosa

General Anxiety Disorder
 Panic Disorder
 Social Anxiety Disorder
 Specific Phobia
 Obsessive Compulsive Disorder
 Post Traumatic Stress Disorder

Tambah Data Testing

Gambar 4.24 Halaman Tambah Data *Testing*

4.2.6.9. Halaman Edit Data *Testing*

Pada halaman edit data *testing* dapat dilakukan proses perubahan data *testing* dengan mengubah data jawaban dari pertanyaan kuesioner HARS dan data target diagnosa data *testing*. Untuk menyimpan perubahan data *testing* dapat dilakukan dengan mengklik tombol Simpan Data *Testing*.

EDIT DATA TESTING

Pertanyaan 1
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 2
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 3
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 4
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 14
 Tidak ada Ringan Sedang Berat Sangat berat

Target Diagnosa

General Anxiety Disorder
 Panic Disorder
 Social Anxiety Disorder
 Specific Phobia
 Obsessive Compulsive Disorder
 Post Traumatic Stress Disorder

[Simpan Data Testing](#)

Gambar 4.25 Halaman Edit Data *Testing*

4.2.6.10. Halaman Detail Data *Testing*

Halaman detail data *testing* menampilkan detail data *testing* dalam bentuk tabel yang terdiri dari 3 kolom. Kolom pertama berisikan kode dari kelompok gejala dari kuesioner HARS. Kolom kedua berisi data kelompok gejala. Kolom ketiga berisi tingkat keparahan dari masing-masing kelompok gejala yang diinputkan pada data *testing*. Di bawah tabel detail data *testing* terdapat target diagnosa data *testing* dan tombol Edit untuk mengedit data *testing* serta tombol Hapus untuk menghapus data *testing*.

DETAIL DATA TESTING		
Kode	Gejala	Keterangan
G1	Kelompok gejala 1	BERAT
G2	Kelompok gejala 2	RINGAN
G3	Kelompok gejala 3	RINGAN
G14	Kelompok gejala 14	BERAT

TARGET DIAGNOSA
General Anxiety Disorder

[Edit](#) [Hapus](#)

Gambar 4.26 Halaman Detail Data *Testing*

4.2.6.11. Halaman *List Data User*

Halaman *list data user* menampilkan data *user* dalam bentuk tabel yang terdiri dari lima kolom. Kolom pertama berisi *username* dari masing-masing *user* dan kolom kedua berisi nama dari masing-masing *user*. Kolom ketiga, keempat, dan kelima pada tabel masing-masing berisi tombol Detail untuk melihat detail data *user*, tombol Edit untuk mengedit data *user*, dan tombol Hapus untuk menghapus data *user*. Pada bagian atas tabel *list data user* terdapat tombol untuk menambahkan *user* baru ke dalam sistem.

User Data Training Data Testing Backpropagation				
LIST DATA USER				
Tambah Data User				
Username	Nama	Detail	Edit	Hapus
admin	admin	Detail	Edit	Hapus
pasien	pasien	Detail	Edit	Hapus

Gambar 4.27 Halaman List Data *User*

4.2.6.12. Halaman Tambah *User*

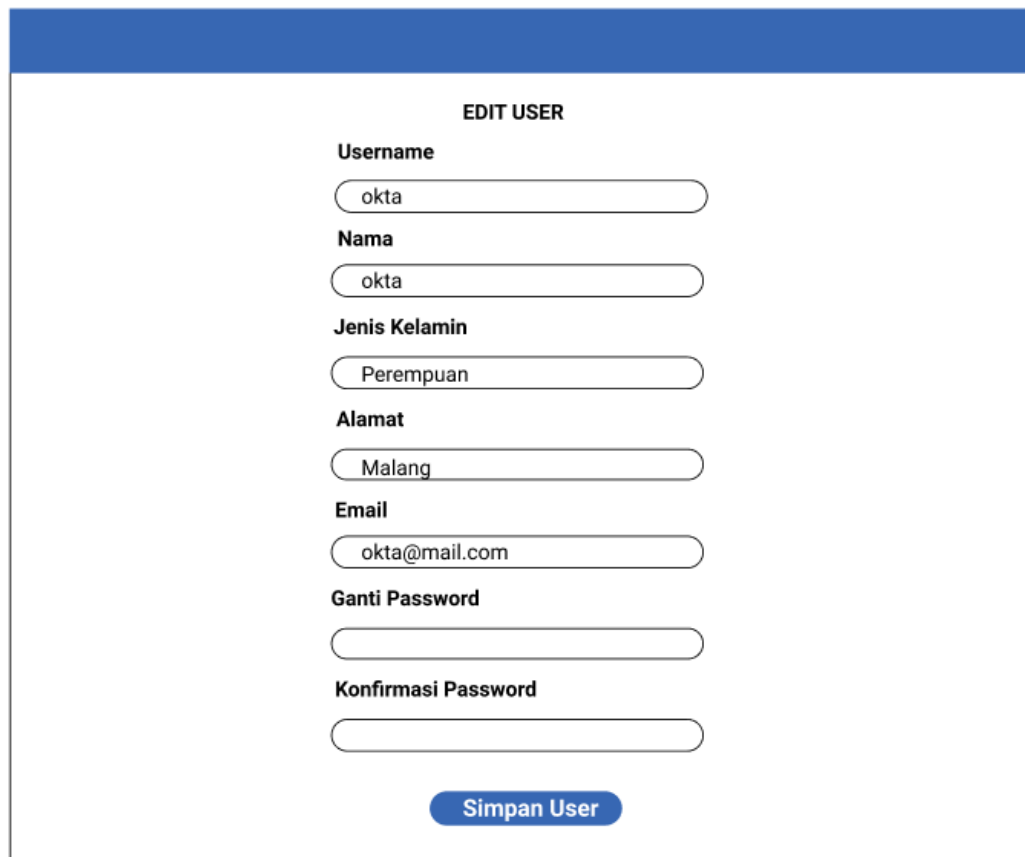
Pada halaman tambah user terdapat *form* yang terdiri dari *username*, *password*, konfirmasi *password*, nama, jenis kelamin, alamat, email, dan *role* dari *user*. Setelah melakukan pengisian data pada *form* tambah *user*, data *user* selanjutnya dapat disimpan ke dalam sistem dengan mengklik tombol Tambah *User*.

TAMBAH USER	
Username	<input type="text"/>
Password	<input type="password"/>
Konfirmasi Password	<input type="password"/>
Nama	<input type="text"/>
Jenis Kelamin	<input type="text"/>
Alamat	<input type="text"/>
Email	<input type="text"/>
Role	<input type="text"/>
Tambah User	

Gambar 4.28 Halaman Tambah *User*

4.2.6.13. Halaman Edit *User*

Proses perubahan data *user* dapat dilakukan melalui halaman edit *user*. Pada halaman edit *user* terdapat *form* untuk mengedit data *user*. Setelah mengedit data *user*, data *user* dapat disimpan ke dalam sistem dengan mengklik tombol *Simpan User*.



The image shows a web form titled "EDIT USER". It contains several input fields with the following labels and values:

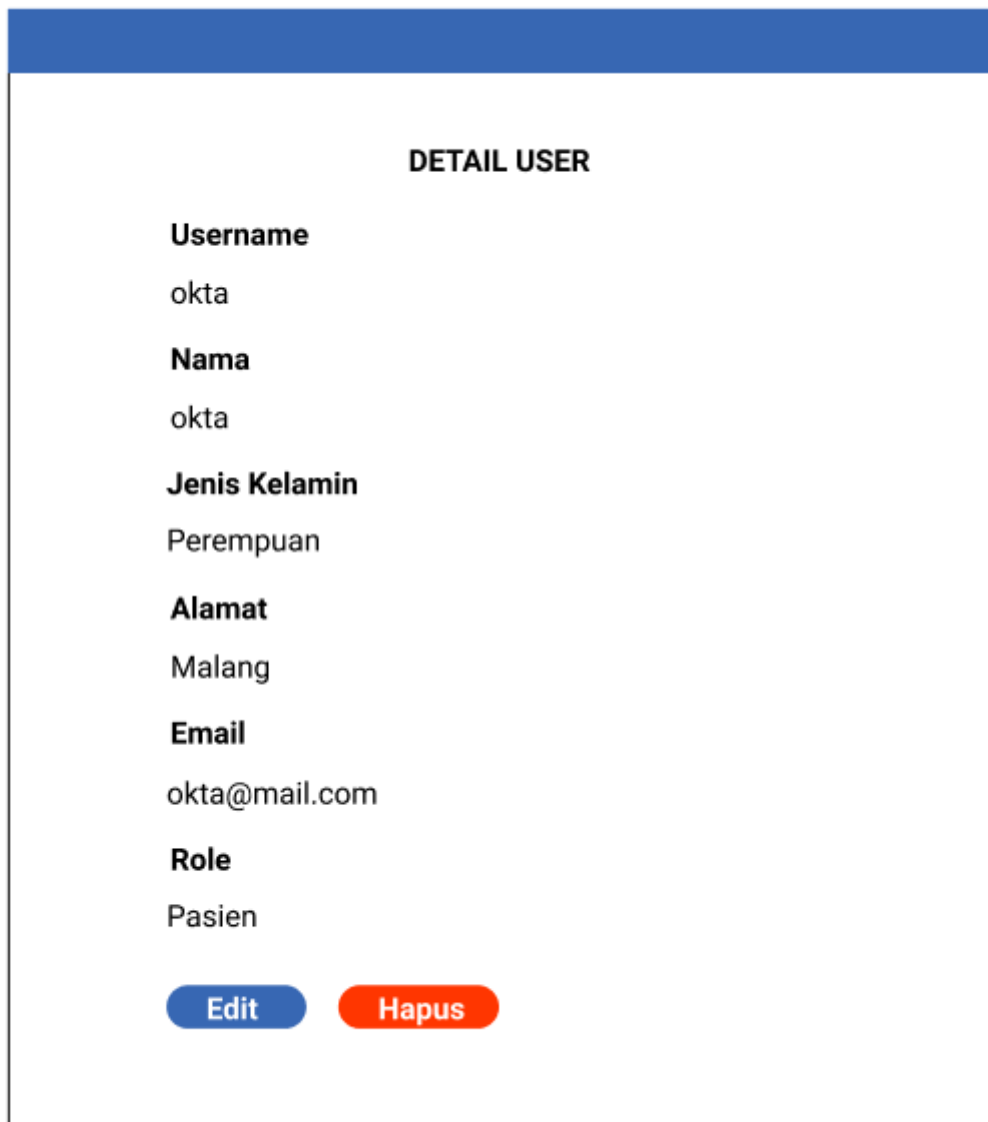
- Username**: okta
- Nama**: okta
- Jenis Kelamin**: Perempuan
- Alamat**: Malang
- Email**: okta@mail.com
- Ganti Password**: (empty field)
- Konfirmasi Password**: (empty field)

At the bottom of the form is a blue button labeled "Simpan User".

Gambar 4.29 Halaman Edit *User*

4.2.6.14. Halaman Detail *User*

Halaman detail *user* menampilkan data detail *user* yang terdiri dari *username*, nama, jenis kelamin, alamat, email, dan *role* user. Di bagian bawah detail data *user* terdapat tombol Edit untuk mengedit data *user* dan tombol Hapus untuk menghapus data *user*.



The image shows a web interface for user details. At the top, there is a blue header bar. Below it, the title 'DETAIL USER' is centered. The user information is listed in a vertical stack, with each field name in bold and its value below it. At the bottom, there are two buttons: 'Edit' in a blue rounded rectangle and 'Hapus' in a red rounded rectangle.

Username	okta
Nama	okta
Jenis Kelamin	Perempuan
Alamat	Malang
Email	okta@mail.com
Role	Pasien

[Edit](#) [Hapus](#)

Gambar 4.30 Halaman Detail *User*

4.2.6.15. Halaman Proses *Backpropagation*

Pada halaman proses *backpropagation* terdapat *form* untuk mengisi data *learning rate*, *momentum*, *target error*, *epoch*, *neuron* pada *hidden layer*, *neuron* pada *output layer*, dan *split data* atau jumlah pembagian data *training* dan data *testing* yang digunakan dalam perhitungan *backpropagation*. Di bawah *form* terdapat tombol *Process* untuk memproses data yang telah diinputkan melalui *form* dan tombol *Reset* untuk menghapus seluruh data yang diinputkan pada *form*.

User Data Training Data Testing Backpropagation

PROSES BACKPROPAGATION

Learning rate

Momentum

Target error

Epoch

Neuron Input

Neuron Hidden

Neuron Output

Split Data

Gambar 4.31 Halaman Proses *Backpropagation*

4.2.6.16. Halaman Hasil Proses *Backpropagation*

Halaman hasil proses *backpropagation* menampilkan tabel hasil pengujian *backpropagation* yang telah dilakukan pada sistem. Tabel hasil pengujian tersebut terdiri dari tiga kolom. Kolom pertama berisi nomor data *testing*, kolom kedua berisi target diagnosa data *testing*, dan kolom ketiga berisi hasil diagnosa pada data *testing* berdasarkan hasil perhitungan *backpropagation*.

HASIL PROSES BACKPROPAGATION		
Data Testing	Target Diagnosa	Hasil Diagnosa
1	General Anxiety Disorder	General Anxiety Disorder
2	Panic Disorder	Panic Disorder
3	Social Anxiety Disorder	Social Anxiety Disorder
4	Specific Phobia	Specific Phobia
5	Obsessive Compulsive	Obsessive Compulsive
6	Post Traumatic Stress	Post Traumatic Stress

Gambar 4.32 Halaman Hasil Proses *Backpropagation*

4.2.6.17. Halaman *Self-Assessment* Pasien

Halaman *self-assessment* pasien berisi daftar pertanyaan dari kuesioner HARS beserta pilihan jawaban pada masing-masing pertanyaan tersebut. Setelah mengisi seluruh jawaban kuesioner HARS yang terdapat pada halaman *self-assessment* pasien, proses diagnosa jenis gangguan kecemasan yang diderita pasien dapat dilakukan dengan mengklik tombol Diagnosa.

SELF-ASSESSMENT PASIEN

Pertanyaan 1
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 2
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 3
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 4
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 5
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 6
 Tidak ada Ringan Sedang Berat Sangat berat

Pertanyaan 7
 Tidak ada Ringan Sedang Berat Sangat berat

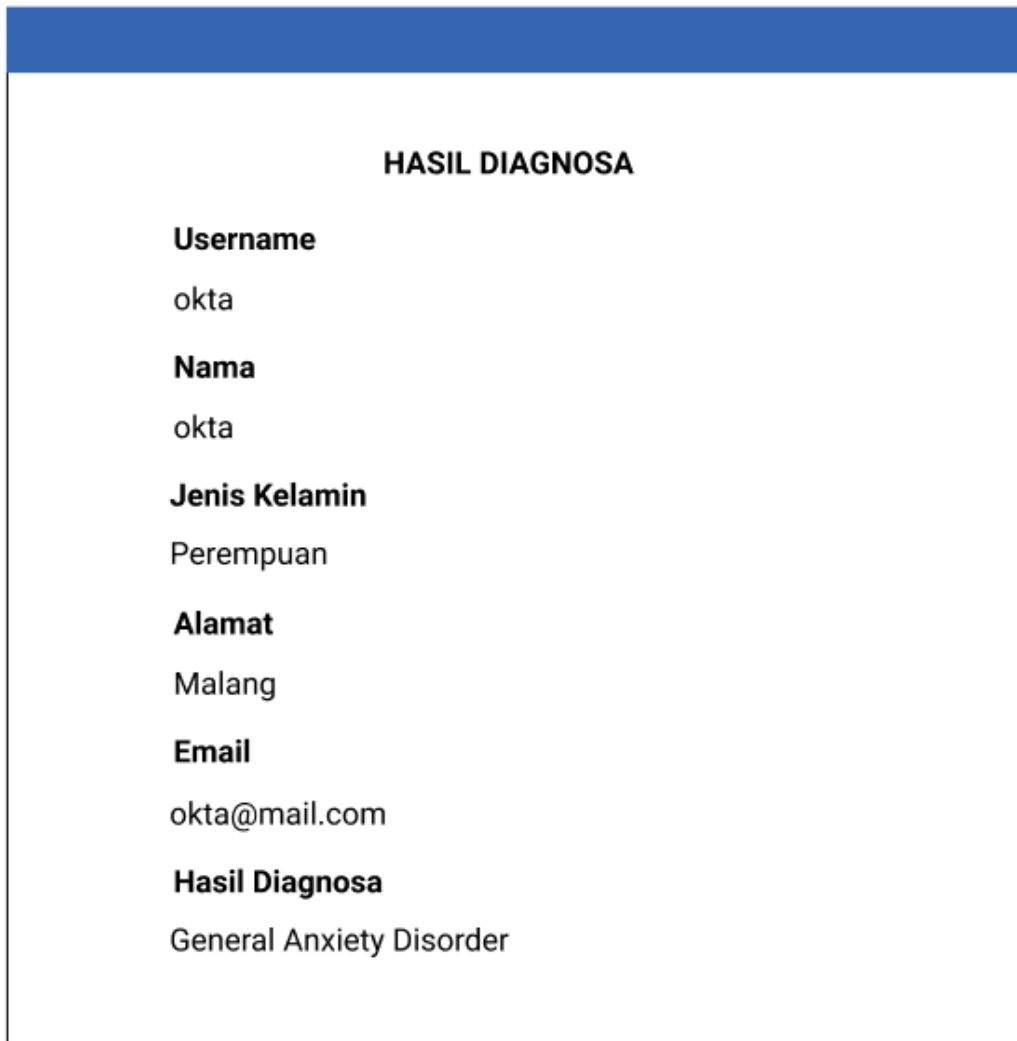
Pertanyaan 14
 Tidak ada Ringan Sedang Berat Sangat berat

Diagnosa

Gambar 4.33 Halaman *Self-Assessment* Pasien

4.2.6.18. Halaman Hasil Diagnosa

Halaman hasil diagnosa menampilkan data hasil diagnosa jenis gangguan kecemasan yang diderita pasien berdasarkan inputan jawaban dari *self-assessment* pasien. Pada halaman ini menampilkan data *username*, nama, jenis kelamin, alamat, email, dan hasil diagnosa gangguan kecemasan yang diderita pasien.



HASIL DIAGNOSA

Username
okta

Nama
okta

Jenis Kelamin
Perempuan

Alamat
Malang

Email
okta@mail.com

Hasil Diagnosa
General Anxiety Disorder

Gambar 4.34 Halaman Hasil Diagnosa