BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Pada bagian analisis sistem ini akan dibahas mengenai analisa-analisa masalah yang akan dilakukan terhadap sarjana psikologi, analisa pengguna dan juga kegunaan aplikasi yang akan dibangun.

3.1.1 Analisis Masalah

Setelah melakukan penelitian pada wawancara terhadap salah satu sarjana psikologi, saat melakukan pembacaan karakteristik seseorang menurut ilmu psikologi terdapat beberapa masalah yang dihadapi, yakni sebagai berikut:

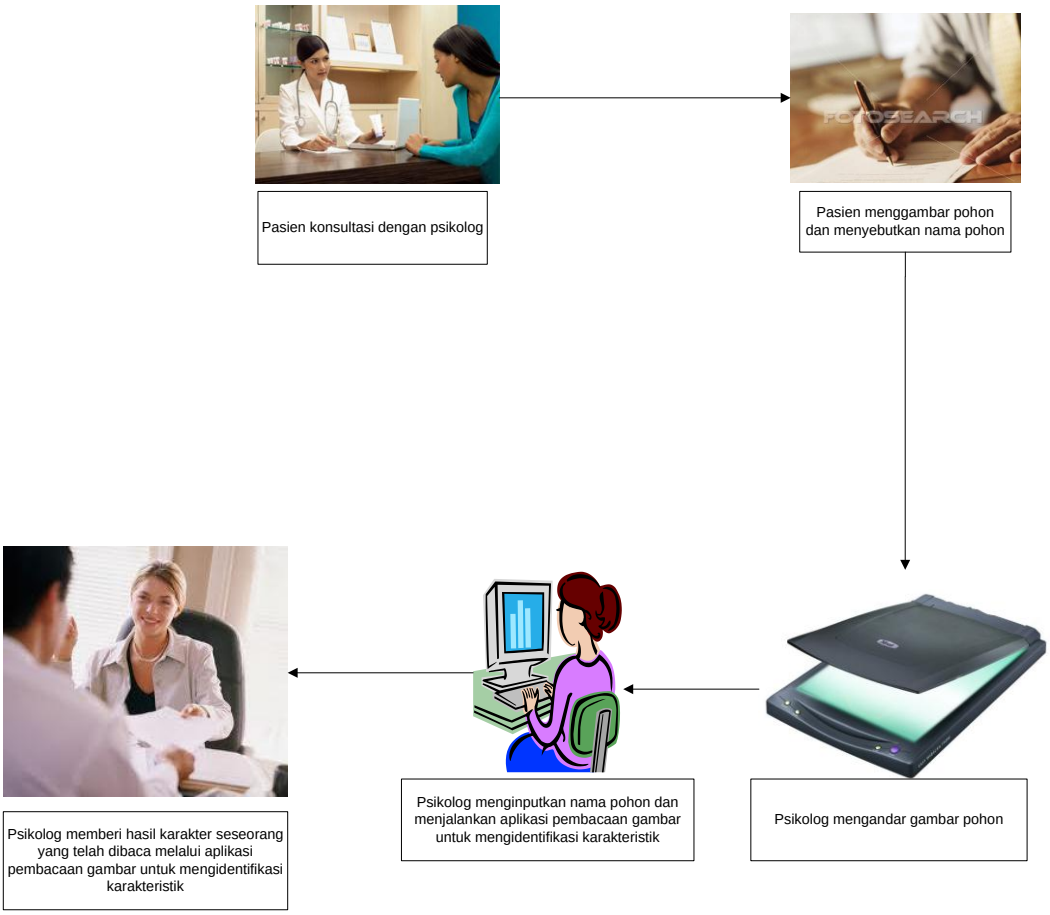
1. Sarjana psikologi harus memerlukan ingatan yang kuat untuk membaca setiap detail gambar yang sangat banyak untuk memahami karakteristik seseorang dari gambar tangannya.
2. Apabila sang psikolog memiliki banyak pasien, maka sarjana psikologi harus menghabiskan banyak waktu untuk melakukan identifikasi pola gambar pohon yang telah dibuat oleh pasien.

3.1.2 Gambaran Sistem

Gambaran sistem yang akan dibangun dapat dideskripsikan sebagai berikut:

1. Saat pasien datang, pasien akan berkonsultasi dahulu dengan psikolog.
2. Lalu psikolog akan memberikan perintah agar pasien menggambar pohon dan menyebutkan nama pohon yang digambar sesuai imajinasi pasien.
3. Kemudian psikolog memberikan tugas kepada tester untuk mengandar gambar pohon menggunakan *scanner*.
4. Tester bertugas untuk meresize gambar pohon yang telah digambar pasien sesuai dengan data pasien yang telah diinputkan dengan data pohon yang ada dan menjalankan aplikasi pembacaan gambar.

5. Setelah pembacaan gambar, tester memberikan hasil karakter pasien kepada psikolog dan psikolog memberi konsultasi serta solusi atas masalah pasien. Gambaran sistem yang telah dijelaskan dapat dilihat pada gambar 3.1.



Gambar 3.1 Gambaran Sistem

Hasil identifikasi berupa laporan sifat-sifat dari pasien. Contoh hasil identifikasi sebagai berikut:

1. Menyendiri dalam pergaulan
2. Merasa takut
3. Sok tahu
4. Ingin dikagumi
5. Pandangannya sempit
6. Sombong

3.1.3 Analisis Citra

Pada kasus pengidentifikasian karakter seseorang ini, citra pohon akan diubah menjadi citra keabuan dengan rumus: avg = (red + green + blue)/3. Kemudian hasil citra keabuan di *tresholding* dengan nilai ambang digunakan adalah nilai T = 128. Citra biner yang digunakan dengan nilai ambang yang apabila hasil nilai avg kurang dari 128, maka biner akan dianggap 0. Sebaliknya, apabila nilai avg lebih dari atau sama dengan 128 maka biner akan dianggap 1.

3.1.4 Analisis Metode

Metode-metode dalam pembangunan aplikasi pengidentifikasian karakter seseorang dengan pola gambar menggunakan metode *multi layer perception,* dengan algoritma pelatihan *backpropagation* serta algoritma nguyen widrow sebagai algoritma pembantu menginisialisasi bobot awal. Pada data yang akan dilatih (*training*) menggunakan semua algoritma yang telah dijabarkan seperti pada bab sebelumnya, sedangkan pada data tes hanya menggunakan algoritma propagasi balik yang inisialisasi awalnya didapatkan dari bobot-bobot baru hasil data *training*.

3.1.4.1 Analisis Algoritma Nguyen Widrow dan Backpropagation

Berikut adalah algoritma nguyen widrow dan backpropagation yang telah digunakan dalam kasus pengidentifikasian karakteristik seseorang dengan pola gambar pada data *training*.

1. Inisialisasi bobot (nguyen widrow)
   a. Inisialisasi bobot input ke hidden (Vij)

      Input: n = 10

      Hidden: p = (n/2)+1 = (10/2)+1= 6

      Output: m = 10

      $\beta = 0.7\sqrt[10]{6} = 0.837$

      buat V_{ij} (i = 1, 2, …, n; j = 1, 2, …,p) bilangan acak dalam interval [-0.5,0.5] seperti pada tabel 3.1

Tabel 3.1 Bilangan Acak $V_{ij} = V_{10x6}$

$V_{i(j)}$ / $V_{(i)j}$	1	2	3	4	5	6
1	0.5	-0.1	0.2	0.3	0.1	-0.2
2	0.1	-0.1	-0.1	0.2	0.3	-0.2
3	0.1	0.3	-0.1	0.4	0.1	0.3
4	0.1	-0.1	-0.1	0.3	0.3	0.1
5	0.5	0.1	-0.3	0.1	-0.2	0.1
6	0.5	0.5	0.2	-0.1	0.2	-0.3
7	-0.2	-0.5	0.3	0.4	-0.5	0.2
8	-0.1	-0.5	0.4	-0.2	0.2	0.5
9	-0.2	0.5	0.5	0.5	0.1	0.1
10	-0.1	0.5	0.1	-0.5	0.3	0.4

Hitung $\left|\left|V_j\right|\right| = \sqrt{{V_{1j}}^2 + \cdots + {V_{ij}}^2}$

$$\left|\left|V_1\right|\right| = \sqrt{\begin{matrix}{V_{11}}^2 + {V_{21}}^2 + {V_{31}}^2 + {V_{41}}^2 + {V_{51}}^2 + {V_{61}}^2 \\ +{V_{71}}^2 + {V_{81}}^2 + {V_{91}}^2 + {V_{101}}^2\end{matrix}}$$

$$\left|\left|V_1\right|\right| = \sqrt{\begin{matrix}0.5^2 + 0.1^2 + 0.1^2 + 0.1^2 + 0.5^2 + 0.5^2 \\ +(-0.2)^2 + (-0.1)^2 + (-0.2)^2 + (-0.1)^2\end{matrix}}$$

$\left|\left|V_1\right|\right| = 0.938083152$

Seperti pada salah satu perhitungan yang telah dibuat berikut akan ditampilkan seluruh perhitungan dari $\left|\left|V_j\right|\right|$ pada tabel 3.2

Tabel 3.2 Hasil Perhitungan $\left|\left|V_j\right|\right|$

$\left\|\left\|V_j\right\|\right\|$	
$\left\|\left\|V_1\right\|\right\|$	0.938083152
$\left\|\left\|V_2\right\|\right\|$	1.174734012
$\left\|\left\|V_3\right\|\right\|$	0.842614977
$\left\|\left\|V_4\right\|\right\|$	1.048808848
$\left\|\left\|V_5\right\|\right\|$	0.818535277
$\left\|\left\|V_6\right\|\right\|$	0.860232527

Bobot yang digunakan sebagai inisialisasi = $V_{ij} = \frac{\beta V_{ij}(lama)}{||V_j||}$,

$$V_{11} = \frac{\beta V_{11}(lama)}{||V_1||}$$
$$= \frac{0.837*0.5}{0.938083152}$$
$$= 0.446122$$

Perhitungan bobot yang digunakan dapat dilihat pada tabel 3.3

Tabel 3.3 Bobot Inisialisasi V_{ij}

$V_{i(j)}$ / $V_{(i)j}$	1	2	3	4	5	6
1	0.446122	-0.07125	0.198667	0.239414	0.102256	-0.1946
2	0.089224	-0.07125	-0.09933	0.15961	0.306767	-0.1946
3	0.089224	0.213751	-0.09933	0.319219	0.102256	0.291898
4	0.089224	-0.07125	-0.09933	0.239414	0.306767	0.097299
5	0.446122	0.07125	-0.298	0.079805	-0.20451	0.097299
6	0.446122	0.356251	0.198667	-0.0798	0.204512	-0.2919
7	-0.17845	-0.35625	0.298001	0.319219	-0.51128	0.194599
8	-0.08922	-0.35625	0.397334	-0.15961	0.204512	0.486496
9	-0.17845	0.356251	0.496668	0.399024	0.102256	0.097299
10	-0.08922	0.356251	0.099334	-0.39902	0.306767	0.389197

Inisialisasi bobot bias yang dipakai = V_{0j}= bil.acak dari [$-\beta, \beta$]

V_{0j} = [-0.837,0.837], pada tabel 3.4.

Tabel 3.4 Bobot Bias V_{0j}

V_{01}	V_{02}	V_{03}	V_{04}	V_{05}	V_{06}
0.2	0.1	-0.2	0.3	0.1	0.2

Inisialisasi bobot hidden ke output (W_{jk})

j = 1, 2, …, 6

k = 1, 2, …, 10

β untuk W = faktor skala=0.7 $\sqrt[j]{k}$. missal $\beta = 0.7\sqrt[6]{10} = 1.03$

buat W_{jk} bilangan acak dalam interval [-0.5,0.5]. seperti pada tabel 3.5.

Tabel 3.5 Bobot Hidden ke output $W_{jk} = W_{6x10}$

$W_{j(k)}$ / $W_{(j)k}$	1	2	3	4	5	6	7	8	9	10
1	0.5	0.1	0.1	0.1	0.5	0.5	-0.2	-0.1	-0.2	-0.1
2	-0.1	-0.1	0.3	-0.1	0.1	0.5	-0.5	-0.5	0.5	0.5
3	0.2	-0.1	-0.1	-0.1	-0.3	0.2	0.3	0.4	0.5	0.1
4	0.3	0.2	0.4	0.3	0.1	-0.1	0.4	-0.2	0.5	-0.5
5	0.1	0.3	0.1	0.3	-0.2	0.2	-0.5	0.2	0.1	0.3
6	-0.2	-0.2	0.3	0.1	0.1	-0.3	0.2	0.5	0.1	0.4

Hitung $\left||W_k|\right| = \sqrt{{W_{1k}}^2 + \cdots + {W_{jk}}^2}$

$$\left||W_1|\right| = \sqrt{{W_{11}}^2 + {W_{21}}^2 + {W_{31}}^2 + {W_{41}}^2 + {W_{51}}^2 + {W_{61}}^2}$$

$$\left||W_1|\right| = \sqrt{0.5^2 + (-0.1)^2 + 0.2^2 + 0.3^2 + 0.1^2 + (-0.2)^2}$$

$$\left||W_1|\right| = 0.663324958$$

Seperti pada salah satu perhitungan yang telah dibuat berikut akan ditampilkan seluruh perhitungan dari $\left||V_j|\right|$ pada tabel 3.6.

Tabel 3.6 Hasil Perhitungan $\left||W_k|\right|$

$\left||W_k|\right|$	
$\left||W_1|\right|$	0.663324958
$\left||W_2|\right|$	0.447213595
$\left||W_3|\right|$	0.608276253
$\left||W_4|\right|$	0.469041576
$\left||W_5|\right|$	0.640312424
$\left||W_6|\right|$	0.824621125
$\left||W_7|\right|$	0.911043358
$\left||W_8|\right|$	0.866025404
$\left||W_9|\right|$	0.9
$\left||W_{10}|\right|$	0.877496439

Bobot yang digunakan sebagai inisialisasi = $W_{jk} = \frac{\beta W_{jk}(lama)}{||W_k||}$

$W_{11} = \frac{\beta W_{11}(lama)}{||W_1||}$

$W_{11} = \frac{1.03*0.5}{0.663324958}$

$W_{11} = 0.776392$

Inisialisasi bobot W_{jk} yang digunakan dapat dilihat pada tabel 3.7.

Tabel 3.7 Inisialisasi Bobot W_{jk}

$W_{j(k)}$ / $W_{(j)k}$	1	2	3	4	5
1	0.776392	0.230315	0.169331	0.219597	0.804295
2	-0.15528	-0.23032	0.507993	-0.2196	0.160859
3	0.310557	-0.23032	-0.16933	-0.2196	-0.48258
4	0.465835	0.46063	0.677324	0.65879	0.160859
5	0.155278	0.690945	0.169331	0.65879	-0.32172
6	-0.31056	-0.46063	0.507993	0.219597	0.160859

$W_{j(k)}$ / $W_{(j)k}$	6	7	8	9	10
1	0.624529	-0.22611	-0.11893	-0.22889	-0.11738
2	0.624529	-0.56529	-0.59467	0.572222	0.586897
3	0.249812	0.339172	0.475737	0.572222	0.117379
4	-0.12491	0.452229	-0.23787	0.572222	-0.5869
5	0.249812	-0.56529	0.237868	0.114444	0.352138
6	-0.37472	0.226114	0.594671	0.114444	0.469518

Inisialisasi bobot bias yang digunakan: W_{0k} = bil.acak dari [-β, β]

W_{0k} = [-1.03, 1.03]. Digambarkan seperti tabel 3.8.

Tabel 3.8 Inisialisasi Bobot Bias W_{0k}

W_{01}	W_{02}	W_{03}	W_{04}	W_{05}	W_{06}	W_{07}	W_{08}	W_{09}	W_{010}
0.2	0.9	-0.8	0.7	0.2	0.8	1	0.7	-0.9	0.3

Tentukan *learning rate*. α = 0.2

Tentukan target error . Misal = 0.01

Tentukan $maks.epoch = \frac{jumlah\ bobot}{target\ error}$

$maks.epoch = \frac{60}{0.01} = 6000$

Tentukan input binner (binner yang didapatkan dari citra yang telah di konversi dari citra digital ke citra hitam putih kemudian menjadi citra binner). Input biner yang digunakan adalah citra dari data pohon. Input biner dapat dilihat pada tabel 3.9.

Tabel 3.9 Input Biner

x1	0
x2	0
x3	1
x4	1
x5	1
x6	0
x7	0
x8	0
x9	0
x10	0

Tentukan target binner yang didapatkan dari citra yang telah dikonvresi dari citra digital menjadi citra hitam putih kemudian menjadi citra biner. Target biner yang dimaksudkan berasal dari citra pohon yang sama dengan input biner, karena target biner ini yang akan digunakan pada data yang akan dilatih yaitu data master pohon. Target biner dapat dilihat pada tabel 3.10.

Tabel 3.10 Target Biner

T1	0
T2	0
T3	1
T4	1
T5	1
T6	0
T7	0
T8	0
T9	0
T10	0

b. Feed forward

Menghitung semua sinyal *input* dengan bobotnya:

$$z_in_j = V_{0j} + \sum_{i=1}^{n} X_i V_{ij} \quad ; n = 10$$

Berikut adalah salah satu hitungan z_{in_j} untuk j=1.

$$z_{in_1} = V_{01} + (X_1 * V_{11}) + (X_2 * V_{21}) + (X_3 * V_{31}) + (X_4 * V_{41}) + (X_5 * V_{51}) + (X_6 * V_{61}) + (X_7 * V_{71}) + (X_8 * V_{81}) + (X_9 * V_{91}) + (X_{10} * V_{101})$$

$$z_in_1 = 0.2 + (0 * 0.446122) + (0 * 0.089224) + (1 * 0.089224) + (1 * 0.089224) + (1 * 0.446122) + (0 * 0.446122) + \big(0 * (-0.17845)\big) + \big(0 * (-0.08922)\big) + \big(0 * (-0.17845)\big) + (0 * (-0.08922))$$

$$z_in_1 = 0.8246$$

Seperti pada salah satu perhitungan yang telah dibuat berikut akan ditampilkan seluruh perhitungan dari z_in_j pada tabel 3.11.

Tabel 3.11 Perhitungan z_in_1

z_in_j	
z_in_1	0.8246
z_in_2	0.4138
z_in_3	-0.297
z_in_4	0.8384
z_in_5	0.4045
z_in_6	0.6865

Menghitung nilai aktivasi setiap *unit hidden* sebagai *output unit hidden*

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

$$z_1 = \frac{1}{1+e^{-Z_in_1}}$$

$$z_1 = \frac{1}{1+e^{-0.8246}}$$

$$z_1 = 0.695206$$

Seperti pada salah satu perhitungan yang telah dibuat berikut akan ditampilkan seluruh perhitungan dari z_j pada tabel 3.12.

Tabel 3.12 Perhitungan z_j

z_j	
z_1	0.695206
z_2	0.601987
z_3	0.426372
z_4	0.698136
z_5	0.599771
z_6	0.665187

Unit *output* (z_j, j = 1,2,3,…,p)

Menghitung semua sinyal *input*nya dengan bobotnya.

Berikut adalah salah satu hitungan y_{in_k} untuk k=1.

$$y_{ink} = w_{0k} + \sum_{j=1}^{p} w_{jk} * z_j \;\; ; \; p = 6$$

$$y_{in1} = w_{01} + (w_{11} * z_1) + (w_{21} * z_2) + (w_{31} * z_3) + (w_{41} * z_4) + (w_{51} * z_5) + (w_{61} * z_6)$$

$$y_{in1} = 0.2 + (0.776392 * 0.695206) + (-0.15528 * 0.601987) + (0.310557 * 0.426372) + (0.465835 * 0.698136) + (0.155278 * 0.599771) + (-0.31056 * 0.665187)$$

$$y_{in1} = 0.990459$$

Seperti pada salah satu perhitungan yang telah dibuat berikut akan ditampilkan seluruh perhitungan dari y_{ink} pada tabel 3.13.

Tabel 3.13 Perhitungan y_{ink}

y_{ink}	
yin1	0.990459
yin2	0.552856
yin3	1.463661
yin4	1.127962
yin5	0.676574
yin6	0.930019
yin7	-0.02579
yin8	0.334343
yin9	1.173582
yin10	0.635534

Menghitung nilai aktivasi setiap *unit otput* sebagai *output* jaringannya.

Berikut adalah salah satu hitungan y_k untuk k=1.

$$y_k = \frac{1}{1+e^{-y_in_k}}$$

$$y_1 = \frac{1}{1+e^{-0.990459}}$$

$$y_1 = 0.729179$$

Seperti pada salah satu perhitungan yang telah dibuat berikut akan ditampilkan seluruh perhitungan dari y_k pada tabel 3.14.

Tabel 3.14 Perhitungan y_k

	y_k
y_1	0.729179
y_2	0.634798
y_3	0.812092
y_4	0.755463
y_5	0.662974
y_6	0.717079
y_7	0.493552
y_8	0.582816
y_9	0.763792
y_{10}	0.653743

Hitung fungsi error kuadratis

$$err = \frac{1}{2}\sum_1^k (T_k - y_k)^2 \quad ; k = 10$$

$$err = \frac{1}{2}((T_1 - y_1)^2 + (T_2 - y_2)^2 + (T_3 - y_3)^2 + (T_4 - y_4)^2 + (T_5 - y_5)^2 + (T_6 - y_6)^2 + (T_7 - y_7)^2 + (T_8 - y_8)^2 + (T_9 - y_9)^2 + (T_{10} - y_{10})^2$$

$$err = \frac{1}{2}((0 - 0.729179)^2 + (0 - 0.634798)^2 + (1 - 0.812092)^2 + (1 - 0.755463)^2 + (1 - 0.662974)^2 + (0 - 0.717079)^2 + (0 - 0.493552)^2 + (0 - 0.582816)^2 + (0 - 0.763792)^2 + (0 - 0.653743)^2$$

$$err = \frac{1}{2} * 3.251592531$$

$$err = 1.625796266$$

c. Backpropagation

Hitung informasi errornya

$$\delta_k = (T_k - y_k) f'(y_{in_k})$$

$\delta_1 = (T_1 - y_1) * y_1 * (1 - y_1)$

$\delta_1 = (0 - 0.729179) * 0.729179 * (1 - 0.729179)$

$\delta_1 =$-0.104998891

Seperti pada salah satu perhitungan yang telah dibuat berikut akan ditampilkan seluruh perhitungan dari δ_k pada tabel 3.15.

Tabel 3.15 Perhitungan δ_k

δ_k	
δ_1	-0.104998891
δ_2	-0.093419981
δ_3	0.023286323
δ_4	0.034128438
δ_5	0.049925261
δ_6	-0.104319675
δ_7	-0.060888233
δ_8	-0.082588879
δ_9	-0.105249477
δ_{10}	-0.096743073

Meng*update* bobot hidden ke output

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

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

$\Delta w_{11} = 0.2 * (-0.104998891) * 0.695206$

$\Delta w_{11} = -0.014599169$

Seperti pada salah satu perhitungan yang telah dibuat berikut akan ditampilkan seluruh perhitungan dariΔw_{jk} pada tabel 3.16.

Tabel 3.16 Perhitungan Koreksi Bobot Hidden ke Output

Δw	1	2	3	4	5
1	-0.014599169	-0.012989224	0.003237758	0.004745258	0.006941667
2	-0.01264159	-0.01124752	0.002803612	0.004108974	0.00601087
3	-0.008953721	-0.007966336	0.001985728	0.002910283	0.004257349
4	-0.014660706	-0.013043975	0.003251405	0.00476526	0.006970927
5	-0.012595061	-0.011206122	0.002793293	0.00409385	0.005988746
6	-0.013968781	-0.012428353	0.003097952	0.004540359	0.006641928

Δw	6	7	8	9	10
1	-0.01450473	-0.008465971	-0.011483255	-0.014634011	-0.013451271
2	-0.012559814	-0.007330783	-0.009943484	-0.01267176	-0.011647611
3	-0.008895802	-0.00519221	-0.00704272	-0.00897509	-0.008249711
4	-0.014565869	-0.008501657	-0.011531658	-0.014695695	-0.013507969
5	-0.012513586	-0.007303801	-0.009906885	-0.01262512	-0.011604741
6	-0.01387842	-0.008100413	-0.010987411	-0.014002118	-0.012870448

Meng*update* bias

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

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

$\Delta w_{01} = 0.2 * (-0.104998891)$

$\Delta w_{01} = -0.020999778$

Seperti pada salah satu perhitungan yang telah dibuat berikut akan ditampilkan seluruh perhitungan dari Δw_{0k} pada tabel 3.17.

Tabel 3.17 Perhitungan Koreksi Bias

Δw_{01}	-0.020999778
Δw_{02}	-0.018683996
Δw_{03}	0.004657265
Δw_{04}	0.006825688
Δw_{05}	0.009985052
Δw_{06}	-0.020863935
Δw_{07}	-0.012177647
Δw_{08}	-0.016517776
Δw_{09}	-0.021049895
Δw_{010}	-0.019348615

Menghitung semua koreksi error

$\delta_{in_j} = \sum_{k=1}^{m} \delta_k w_{jk}$; $m = 10$

$\delta_{in_1} = (\delta_1 * w_{11}) + (\delta_2 * w_{12}) + (\delta_3 * w_{13}) + (\delta_4 * w_{14}) + (\delta_5 * w_{15}) +$
$(\delta_6 * w_{16}) + (\delta_7 * w_{17}) + (\delta_8 * w_{18}) + (\delta_9 * w_{19}) + (\delta_{10} * w_{110})$

$\delta_{in_1} = \big((-0.104998891) * 0.776392\big) + \big((-0.093419981) *$
$0.230315\big) + (0.023286323 * 0.169331) + (0.034128438 *$
$0.219597) + (0.049925261 * 0.804295) + \big((-0.104319675) *$
$0.624529\big) + \big((-0.060888233) * (-0.22611)\big) +$
$\big((-0.082588879) * (-0.11893)\big) + ((-0.105249477 *$
$(-0.22889)) + ((-0.096743073) * (-0.11738))$

$\delta_{in_1} = -0.057558335$

Seperti pada salah satu perhitungan yang telah dibuat berikut akan ditampilkan seluruh perhitungan dari δ_{in_j} pada tabel 3.18.

Tabel 3.18 Perhitungan Delta Input

δ_{in_1}	-0.057558335
δ_{in_2}	0.687353473
δ_{in_3}	-0.204206571
δ_{in_4}	-0.043965287
δ_{in_5}	-0.127885837
δ_{in_6}	0.021736583

Menghitung nilai aktivasi koreksi error

$\delta_j = \delta_{in_j} z_j$

$\delta_1 = \delta_{in_1} z_1$

$\delta_1 = -0.057558335 * 0.695206$

δ_1 =-0.027818589

Seperti pada salah satu perhitungan yang telah dibuat berikut akan ditampilkan seluruh perhitungan dari pengaktifan informasi *error* pada tabel 3.19.

Tabel 3.19 Perhitungan Pengaktifan Informasi Error

δ_1	-0.027818589
δ_2	0.249088757
δ_3	-0.037123375
δ_4	-0.021428427
δ_5	-0.046003788
δ_6	0.009617869

Hitung koreksi bobot input ke hidden

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

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

$\Delta V_{11} = 0.2 * (-0.027818589) * 0$

$\Delta V_{11} = 0$

Seperti pada salah satu perhitungan yang telah dibuat berikut akan ditampilkan seluruh perhitungan dari ΔV_{ij} pada tabel 3.20.

Tabel 3.20 Perhitungan Koreksi Bobot Input ke Hidden

$\Delta V_{i(j)}$ / $\Delta V_{(i)j}$	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	-0.005563718	0.049817751	-0.007424675	-0.004285685	-0.009200758	0.001923574
4	-0.005563718	0.049817751	-0.007424675	-0.004285685	-0.009200758	0.001923574
5	-0.005563718	0.049817751	-0.007424675	-0.004285685	-0.009200758	0.001923574
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0

Hitung koreksi bias

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

$\Delta V_{01} = 0.2 * (-0.027818589)$

$\Delta V_{01} =$ -0.005563718

Seperti pada salah satu perhitungan yang telah dibuat berikut akan ditampilkan seluruh perhitungan dari ΔV_{0j} pada tabel 3.21.

Tabel 3.21 Perhitungan Koreksi Bias

ΔV_{01}	ΔV_{02}	ΔV_{03}	ΔV_{04}	ΔV_{05}	ΔV_{06}
-0.005563718	0.049817751	-0.007424675	-0.004285685	-0.009200758	0.001923574

Meng*update* bias dan bobotnya menurut yk

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

$$w_{11}(baru) = w_{11}(lama) + \Delta w_{11}$$

$$= 0.776392 - 0.014599169$$

$$=0.761792543$$

$$w_{0k}(baru) = w_{0k}(lama) + \Delta w_{0k}$$

$$w_{01}(baru) = w_{01}(lama) + \Delta w_{01}$$

$$= 0.2 - 0.020999778$$

$$= 0.179000222$$

Seperti pada salah satu perhitungan yang telah dibuat berikut akan ditampilkan seluruh perhitungan dari $w_{jk}(baru)$ dan $w_{0k}(baru)$ pada tabel 3.22.

Tabel 3.22 Perhitungan Perbaikan Bias dan Bobotnya Menurut y_k

$w_{j(k)}$ / $w_{(j)k}$	1	2	3	4	5
1	0.761792543	0.21732578	0.172568715	0.224341996	0.81123654
2	-0.16791993	-0.2415625	0.510796483	-0.215487764	0.16686984
3	0.301602964	-0.2382813	-0.167345229	-0.216686455	-0.47831958
4	0.451174321	0.44758603	0.680575233	0.663555473	0.1678299
5	0.142683281	0.67973888	0.17212425	0.662884064	-0.3157292
6	-0.32452547	-0.4730584	0.511090823	0.224137097	0.1675009
w_{0k}	0.179000222	0.881316	-0.795342735	0.706825688	0.20998505

	6	7	8	9	10
1	0.610025	-0.2345803	-0.13042	-0.2435229	-0.13083066
2	0.611969	-0.5726167	-0.60461	0.55955046	0.57524936
3	0.240916	0.33397935	0.468694	0.56324713	0.10912968
4	-0.13947	0.44372709	-0.2494	0.55752653	-0.60040494
5	0.237298	-0.5725897	0.227961	0.10181932	0.34053344
6	-0.3886	0.21801396	0.583683	0.10044233	0.45664713
w_{0k}	0.779136	0.98782235	0.683482	-0.9210499	0.28065139

Meng*update* bias dan bobotnya menurut zj

$$V_{ij}(baru) = V_{ij}(lama) + \Delta V_{ij}$$
$$= 0.446122 + 0$$
$$= 0.446122$$

Seperti pada salah satu perhitungan yang telah dibuat berikut akan ditampilkan seluruh perhitungan dari $V_{ij}(baru)$ pada tabel 3.23.

Tabel 3.23 Perhitungan Perbaikan Bias dan Bobotnya Menurut z_j

$V_{i(j)}$ / $V_{(i)j}$	1	2	3	4	5	6
1	0.446122	-0.07125	0.198667	0.239414	0.102255	-0.1946
2	0.089224	-0.07125	-0.09933	0.159609	0.306767	-0.1946
3	0.083660	0.263568	-0.10675	0.314933	0.093055	0.293821
4	0.083660	-0.02143	-0.10675	0.235128	0.297566	0.099223
5	0.440558	0.121067	-0.30542	0.075519	-0.21371	0.099223
6	0.446122	0.356250	0.198667	-0.07980	0.204511	-0.2919
7	-0.17844	-0.35625	0.298000	0.319219	-0.51127	0.194599
8	-0.08922	-0.35625	0.397334	-0.15960	0.204511	0.486496
9	-0.17844	0.356250	0.496668	0.399024	0.102255	0.097299
10	-0.08922	0.356250	0.099333	-0.39902	0.306767	0.389197
V_{0j}	0.194436	0.149817	-0.20742	0.295714	0.090799	0.201924

Test kondisi, apabila kuadratis *error* lebih besar dari 0.01 maka ulang perhitungan *feedforward* sampai *backpropagation*. Atau apabila *epoch* kurang dari 6000 maka ulang kembali perhitungan *feedforward* sampai *backpropagation*. Dan apabila *epoch* sudah lebih dari 6000 maka bobot terakhir yang digunakan untuk inisialisasi bobot pada data *testing*. Apabila salah satu kondisi sudah ada

yang terpenuhi maka test kondisi berhenti. Pada kasus ini, *error* belum mencapai 0.01 dan *epoch* kurang dari 6000, akan tetapi akan diinisialisasikan telah mencapai untuk melanjutkan perhitungan data *testing*.

Berikut adalah perhitungan data *testing* yang inisialisasi bobot-bobot awal didapat dari bobot update pada data *training*.

Perhitungan bobot yang digunakan dapat dilihat pada tabel 3.24

Tabel 3.24 Bobot Inisialisasi V_{ij}

$V_{i(j)}$ / $V_{(i)j}$	1	2	3	4	5	6
1	0.446122499	-0.0712502	0.19866725	0.239414456	0.10225582	-0.1946
2	0.0892245	-0.0712502	-0.099333625	0.159609637	0.30676747	-0.1946
3	0.066613641	0.26356827	-0.124764392	0.304505243	0.07403185	0.306654
4	0.066613641	-0.0214324	-0.124764392	0.224700424	0.2785435	0.112055
5	0.42351164	0.12106792	-0.323431642	0.065090786	-0.23273562	0.112055
6	0.446122499	0.35625086	0.19866725	-0.079804819	0.20451165	-0.2919
7	-0.178449	-0.3562509	0.298000874	0.319219275	-0.51127912	0.194599
8	-0.0892245	-0.3562509	0.397334499	-0.159609637	0.20451165	0.486496
9	-0.178449	0.35625086	0.496668124	0.399024094	0.10225582	0.097299
10	-0.0892245	0.35625086	0.099333625	-0.399024094	0.30676747	0.389197

Inisialisasi bobot bias yang digunakan pada tabel 3.25.

Tabel 3.25 Bobot Bias V_{0j}

V_{01}	V_{02}	V_{03}	V_{04}	V_{05}	V_{06}
0.194436282	0.14981775	-0.207424675	0.295714315	0.09079924	0.201924

Inisialisasi bobot W_{jk} yang digunakan dapat dilihat pada tabel 3.26.

Tabel 3.26 Inisialisasi Bobot W_{jk}

$W_{j(k)}$ / $W_{(j)k}$	1	2	3	4	5
1	0.761792543	0.21732578	0.172568715	0.224341996	0.81123654
2	-0.16791993	-0.2415625	0.510796483	-0.215487764	0.16686984
3	0.301602964	-0.2382813	-0.167345229	-0.216686455	-0.47831958
4	0.451174321	0.44758603	0.680575233	0.663555473	0.1678299
5	0.142683281	0.67973888	0.17212425	0.662884064	-0.3157292
6	-0.32452547	-0.4730584	0.511090823	0.224137097	0.1675009

	6	7	8	9	10
1	0.610025	-0.2345803	-0.13042	-0.2435229	-0.13083066
2	0.611969	-0.5726167	-0.60461	0.55955046	0.57524936
3	0.240916	0.33397935	0.468694	0.56324713	0.10912968
4	-0.13947	0.44372709	-0.2494	0.55752653	-0.60040494
5	0.237298	-0.5725897	0.227961	0.10181932	0.34053344
6	-0.3886	0.21801396	0.583683	0.10044233	0.45664713

Inisialisasi bobot bias yang digunakan dapat dilihat pada tabel 3.27.

Tabel 3.27 Inisialisasi Bobot Bias W_{0k}

W_{01}	W_{02}	W_{03}	W_{04}	W_{05}
0.179000222	0.881316	-0.795342735	0.706825688	0.20998505

W_{06}	W_{07}	W_{08}	W_{09}	W_{010}
0.779136	0.98782235	0.683482	-0.9210499	0.28065139

Tentukan input binner (binner yang didapatkan dari citra yang telah di konversi dari citra digital ke citra hitam putih kemudian menjadi citra binner). Input biner yang digunakan adalah citra dari gambar pohon yang dibuat oleh pasien. Input biner dapat dilihat pada tabel 3.28.

Tabel 3.28 Input Biner

x1	0
x2	0
x3	1
x4	1
x5	1
x6	1
x7	1
x8	0
x9	0
x10	1

Target biner yang digunakan pada perhitungan data testing berasal dari target biner yang ada pada data *training* data pohon. Target biner dapat dilihat pada tabel 3.29.

Tabel 3.29 Target Biner

T1	0
T2	0
T3	1
T4	1
T5	1
T6	0
T7	0
T8	0
T9	0
T10	0

a. Feed forward

Menghitung semua sinyal *input* dengan bobotnya. Seperti perhitungan sebelumnya pada *feedforward* data *training*, maka ditampilkan seluruh perhitungan dari z_in_j pada tabel 3.30.

Tabel 3.30 Perhitungan z_in_1

z_in_j	
z_in_1	0.980765627
z_in_2	0.869272379
z_in_3	-0.130365075
z_in_4	0.761686171
z_in_5	0.26770862
z_in_6	0.986088484

Menghitung nilai aktivasi setiap *unit hidden* sebagai *output unit hidden.* Seperti pada perhitungan yang sebelumnya, berikut akan ditampilkan seluruh perhitungan dari z_j pada tabel 3.31.

Tabel 3.31 Perhitungan z_j

z_1	0.727260107
z_2	0.704594273
z_3	0.46745481
z_4	0.681719708
z_5	0.566530288
z_6	0.728314633

Unit *output* (z_j, j = 1,2,3,…,p)

Menghitung semua sinyal *input*nya dengan bobotnya.Seperti pada salah satu perhitungan yang sebelumnya, berikut akan ditampilkan seluruh perhitungan dari y_{ink} pada tabel 3.32.

Tabel 3.32 Perhitungan y_{ink}

yin1	0.90774406
yin2	1.1034646
yin3	0.54554763
yin4	1.60800317
yin5	1.11116254
yin6	1.68031355
yin7	1.88899984
yin8	1.58465971
yin9	-0.0198724
yin10	1.18182887

Menghitung nilai aktivasi setiap *unit otput* sebagai *output* jaringannya. Nilai aktivasi inilah yang digunakan sebagai nilai testing. Seperti pada perhitungan yang sebelumnya, berikut akan ditampilkan seluruh perhitungan dari y_k pada tabel 3.33.

Tabel 3.33 Perhitungan y_k

y_1	0.712538307
y_2	0.750908705
y_3	0.63310199
y_4	0.833133969
y_5	0.752345781
y_6	0.842946046
y_7	0.868641451
y_8	0.829863434
y_9	0.495032061
y_{10}	0.76527648

Akan dibuat permisalan $y_k < 0.8$ maka $y_k = 0$; dan apabila $y_k \geq 0.8$ maka $y_k = 1$. Seperti pada tabel 3.34.

Tabel 3.34 Permisalan nilai y_k

y_1	0
y_2	0
y_3	0
y_4	1
y_5	0
y_6	1
y_7	1
y_8	1
y_9	0
y_{10}	0

Hitung fungsi error kuadratis

$$err = \frac{1}{2}\sum_{k=10}(T_k - y_k)^2$$

$$err = \frac{1}{2}((T_1 - y_1)^2 + (T_2 - y_2)^2 + (T_3 - y_3)^2 + (T_4 - y_4)^2 + (T_5 - y_5)^2 + (T_6 - y_6)^2 + (T_7 - y_7)^2 + (T_8 - y_8)^2 + (T_9 - y_9)^2 + (T_{10} - y_{10})^2$$

$$err = \frac{1}{2}((0 - 0)^2 + (0 - 0)^2 + (1 - 0)^2 + (1 - 1)^2 + (1 - 0)^2 + (0 - 1)^2 + (0 - 1)^2 + (0 - 1)^2 + (0 - 0)^2 + (0 - 0)^2$$

$$err = \frac{1}{2} * 5$$

err =2.5

Setelah perhitungan algoritma *backpropagation* pada data *training* maka bobot-bobot yang akan di-*update* akan dijadikan sebagai inisialisasi pada bobot awal di data testing. Dari hasil *feedforward* yang didapatkan maka diberikan validasi, apabila y_k>=0.8 maka y_k dianggap 1, dan apabila y_k<0.8 maka y_k dianggap 0. Kemudian, hitung *error*. *Error* pada kasus ini didapatkan hasil perbandingan ketidakmiripan gambar master pohon dengan gambar pohon pasien sebesar 2.5%

3.1.5 Analisis dan Kebutuhan Non-Fungsional

3.1.5.1 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar bisa saling berinteraksi diantara keduanya. Perangkat lunak yang dibutuhkan untuk membangun aplikasi pengidentifikasian karakteristik seseorang ini, terdapat pada:

1. Sistem Operasi Windows 7
2. Netbeans 7.2
3. Jdk 1.6
4. MySQL 5.0
5. SQLyog 8.8.2

3.1.5.2 Analisis Kebutuhan Perangkat Keras

Komputer terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi-instruksi kepada perangkat keras untuk melakukan suatu tugas tertentu, sehingga dapat menjalankan suatu sistem di dalamnya. Pada aplikasi pengidentifikasian karakteristik seseorang ini, menggunakan spesifikasi perangkat keras, yang dapat dilihat sebagai berikut:

1. Prosessor Intel Core i5
2. Monitor LCD 14”
3. Harddisk 160 GB
4. RAM 4 GB

5. Keyboard
6. Mouse
7. Scanner
8. Printer

3.1.6 Analisis dan Kebutuhan Fungsional

Analisis kebutuhan fungsional bertujuan menganalisis proses yang akan diterapkan dalam sistem dan menjelaskan kebutuhan yang diperlukan agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan proses bisnis dari pengidentifikasian karakteristik seseorang.

Analisis fungsional dimodelkan dengan menggunakan *UML* (*Unified Modeling Language*). Tahapan pemodelan dalam analisis tersebut antara lain mengidentifikasi aktor, pembuatan *use case* diagram, *use case scenario*, *activity* diagram, *sequence* diagram, *class* diagram, dan *state* diagram.

3.1.6.1 Identifikasi Aktor

Tahap pertama yang dilakukan dalam melakukan analisis berorientasi objek menggunakan *UML* adalah menentukan aktor atau pengguna sistem. Kata aktor dalam konteks *UML*, menampilkan peran (*roles*) yang pengguna (atau sesuatu di luar sistem yang dikembangkan yang dapat berupa perangkat keras, *end user*, sistem yang lain, dan sebagainya).

Identifikasi aktor adalah Aktor yang berperan dalam menjalankan sistem, aktor yang berperan dapat dilihat pada tabel 3.35.

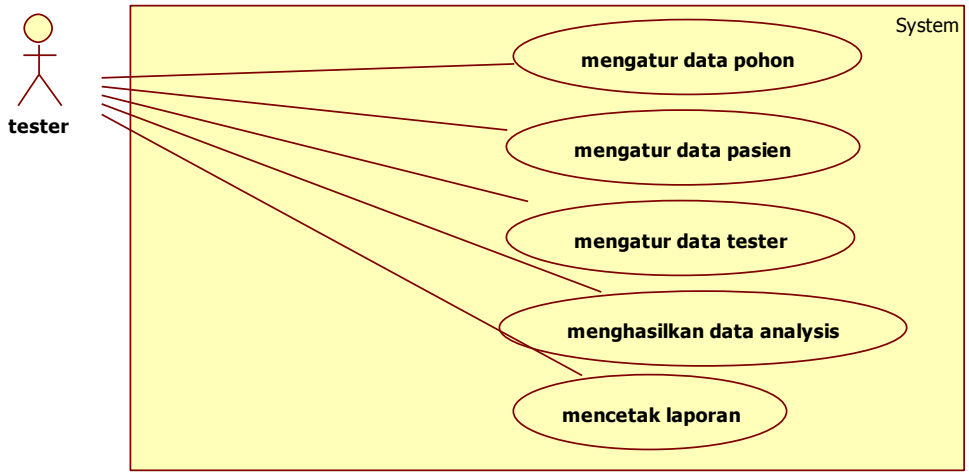
Tabel 3.35 Identifikasi Aktor

No.	Aktor	Deskripsi
1.	Tester	Merupakan pengguna dari aplikasi pengidentifikasian karakter seseorang

3.1.6.2 Use Case Diagram

Use case diagram digunakan untuk mendeskripsikan interaksi antara pengguna dengan sistem, dengan memberikan sebuah narasi tentang bagaimana sistem tersebut digunakan. Pembuatan diagram *use case* dimulai dengan

menjabarkan skenario. Adapun identifikasi dari diagram *use case* pada aplikasi pengidentifikasian karakteristik dapat dilihat pada gambar 3.2.



Gambar 3.2 Use Case Diagram

3.1.6.3 Definisi Use Case

Definisi *Use Case* berfungsi untuk menjelaskan proses yang terdapat pada setiap *Use Case*. Dijelaskan pada tabel 3.36.

Tabel 3.36 Definisi Use Case

No	Use Case	Deskripsi
1	Mengatur Data Pohon	Proses untuk menambah kode pohon, menambah, mengubah, dan menghapus: kategori, deskripsi serta indikasi dari gambar pohon yang telah diandar.
2	Mengatur Data Pasien	Proses untuk menambah id pasien, menambah, mengubah, dan menghapus: nama pasien, jenis kelamin pasien, usia, pendidikan terakhir, foto pasien.
3	Mengatur Data Tester	Proses untuk menambah id tester, menambah, mengubah, dan menghapus: nama tester, alamat, tanggal lahir, telepon, jenis kelamin tester.
4	Mengatur Data Analysis	Proses untuk menambah kode test, menampilkan id pasien, id tester, tanggal test dan gambar pohon.
5	Mencetak Laporan	Proses untuk mencetak hasil dari data test.

3.1.6.4 Skenario Use Case

Skenario *Use case* berisi mengenai skenario (*flow of event*) untuk beberapa *use case* utama, yang menggambarkan urutan interaksi aktor dengan *use case* tersebut dari awal sampai akhir. *Use case* skenario yang terdapat pada pembangunan aplikasi pengidentifikasian karakter seseorang dengan pola gambar meliputi pengaturan data pohon, pengaturan data pasien, pengaturan data tester, dan pengaturan data test.

1. Skenario mengatur data pohon

Tabel 3.37 Skenario mengatur data pohon

Identifikasi	
Nomor	1
Nama	Mengatur data pohon
Tujuan	Mengatur data pohon
Deskripsi	Proses untuk mengatur kode pohon, kategori, deskripsi serta indikasi dari gambar pohon yang telah diandar.
Aktor	Tester
Skenario Utama	
Kondisi Awal	Menampilkan halaman awal identifikasi karakteristik seseorang dengan pola gambar.
Aksi Aktor	Reaksi Sistem
1. Memilih tombol data pohon	
	2. Menampilkan halaman data pohon
3. Jika tester memilih tombol master pohon baru	
	4. Menampilkan kolom data pohon
5. Memilih tombol save	
	6. Memvalidasi data kosong: a. Jika kosong, menampilkan pesan data harus terisi dan kembali pada kolom data pohon b. Jika terisi, menampilkan pesan data berhasil disimpan dan kembali pada halaman data pohon
7. Jika tester memilih tombol ubah pada data yang akan	

diubah	
	8. Menampilkan kolom data pohon yang akan diubah
9. Jika tester memilih tombol update	
	10. Memvalidasi data kosong: a. Jika iya, menampilkan pesan data harus terisi dan kembali pada halaman tambah data pohon b. Jika tidak, menampilkan pesan data berhasil disimpan dan kembali pada halaman data pohon
11. Jika tester memilih tombol filter	
	12. Menampilkan kolom filter data pohon
13. Tester memilih tombol ok	
	14. Memvalidasi berhasil mencari data: a. Jika iya, menampilkan data yang berhasil dicari b. Jika tidak, menampilkan data kosong
15. Jika tester memilih tombol cancel padakolom filter data pohon	
	16. Menampilkan halaman data pohon
17. Jika tester memilih tombol hapus pada data yang akan dihapus	
	18. Memvalidasi data yang akan dihapus: a. Jika iya, menghapus data pohon b. Jika tidak, menampilkan halaman data pohon
Kondisi Akhir	Menampilkan halaman data pohon

2. Skenario mengatur data pasien

Tabel 3.38 Skenario Mengatur Data Pasien

Identifikasi	
Nomor	2
Nama	Mengatur data pasien
Tujuan	Mengatur data pasien

Deskripsi	Proses untuk mengatur id pasien, nama pasien, jenis kelamin pasien, usia, pendidikan terakhir, foto pasien
Aktor	Tester
Skenario Utama	
Kondisi Awal	Menampilkan halaman awal identifikasi karakteristik seseorang dengan pola gambar.

Aksi Aktor	Reaksi Sistem
1. Memilih tombol data pasien	
	2. Menampilkan halaman data pasien
3. Jika tester memilih tombol pasien baru	
	4. Menampilkan kolom data pasien
5. Memilih tombol save	
	6. Memvalidasi data kosong: a. Jika iya, menampilkan pesan data harus terisi dan kembali pada kolom data pasien b. Jika tidak, menampilkan pesan data berhasil disimpan dan kembali pada halaman data pohon
7. Jika tester memilih tombol ubah pada data yang akan diubah	
	8. Menampilkan kolom data pasien yang akan diubah
9. Jika tester memilih tombol update	
	10. Memvalidasi data kosong: a. Jika kosong, menampilkan pesan data harus terisi dan kembali pada kolom data pasien b. Jika terisi, menampilkan pesan data berhasil disimpan dan kembali pada halaman data pasien
11. Jika tester memilih tombol filter	
	12. Menampilkan kolom filter data pasien
13. Tester memilih tombol ok	
	14. Memvalidasi berhasil mencari data: a. Jika iya, menampilkan data yang berhasil dicari b. Jika tidak, menampilkan data kosong

15. Jika tester memilih tombol cancel pada kolom filter data pasien	
	16. Menampilkan halaman data pasien
17. Jika tester memilih tombol hapus pada data yang akan dihapus	
	18. Memvalidasi data yang akan dihapus: a. Jika iya, menghapus data pasien b. Jika tidak, menampilkan halaman data pasien
Kondisi Akhir	Menampilkan halaman data pasien

3. Skenario mengatur data tester

Tabel 3.39 Skenario Mengatur Data Tester

Identifikasi	
Nomor	3
Nama	Mengatur data tester
Tujuan	Mengatur data tester
Deskripsi	Proses untuk mengatur data id tester, nama tester, alamat, tanggal lahir, telepon, jenis kelamin tester
Aktor	Tester
Skenario Utama	
Kondisi Awal	Menampilkan halaman awal identifikasi karakteristik seseorang dengan pola gambar.
Aksi Aktor	Reaksi Sistem
1. Memilih tombol data tester	
	2. Menampilkan halaman data tester
3. Jika tester memilih tombol tambah data tester atau tombol ubah data tester	
	4. Menampilkan kolom data tester
5. Memilih tombol save	
	6. Memvalidasi data kosong: a. Jika iya, menampilkan pesan data harus terisi dan kembali pada kolom data tester b. Jika tidak, menampilkan pesan data

	berhasil disimpan dan kembali pada halaman data tester
7. Jika tester memilih tombol ubah pada data yang akan diubah	
	8. Menampilkan kolom data tester yang akan diubah
9. Jika tester memilih tombol update	
	10. Memvalidasi data kosong: a. Jika kosong, menampilkan pesan data harus terisi dan kembali pada kolom data tester b. Jika terisi, menampilkan pesan data berhasil disimpan dan kembali pada halaman data tester
11. Jika tester memilih tombol filter	
	12. Menampilkan kolom filter data tester
13. Tester memilih tombol ok	
	14. Memvalidasi berhasil mencari data: a. Jika iya, menampilkan data yang berhasil dicari b. Jika tidak, menampilkan data kosong
15. Jika tester memilih tombol cancel pada kolom filter data tester	
	16. Menampilkan halaman data tester
17. Jika tester memilih tombol hapus pada data yang akan dihapus	
	18. Memvalidasi data yang akan dihapus: a. Jika iya, menghapus data tester b. Jika tidak, menampilkan halaman data tester
Kondisi Akhir	Menampilkan halaman data tester

4. Skenario menghasilkan data analysis

Tabel 3.40 Skenario Menghasilkan Data Analysis

Identifikasi	
Nomor	4

Nama	Menghasilkan data analysis
Tujuan	Menghasilkan data analysis
Deskripsi	Proses untuk menghasilkan data analysis
Aktor	Tester
Skenario Utama	
Kondisi Awal	Menampilkan halaman awal identifikasi karakteristik seseorang dengan pola gambar.

Aksi Aktor	Reaksi Sistem
1. Memilih tombol data analysis	
	2. Menampilkan halaman data analysis
3. Jika tester memilih tombol analysis baru	
	4. Menampilkan kolom data analysis yang kosong
5. Memilih tester, pasien dan gambar pohon yang akan dianalysis	
6. Memilih tombol analisis	
	7. Mengenerate data analysis dan menyimpan data dalam database
	8. Menampikan halaman data analysis yang telah terisi
9. Jika tester memilih tombol kembali	
	10. Menampilkan halaman data pohon
11. Jika tester memilih tombol filter	
	12. Menampilkan kolom filter
13. Jika tester memilih tombol ok	
	14. Memvalidasi berhasil mencari data: a. Jika iya, menampilkan data yang berhasil dicari b. Jika tidak, menampilkan data kosong
15. Jika memilih tombol cancel	
	16. Menampilkan halaman data analysis
17. Memilih tombol hapus pada data analysis yang akan	

dihapus	
	19. Memvalidasi data yang akan dihapus: a. Jika iya, menghapus data tester b. Jika tidak, menampilkan halaman data tester
Kondisi Akhir	Menampilkan halaman data analysis

5. Skenario mencetak laporan

Tabel 3.41 Skenario Mencetak Laporan

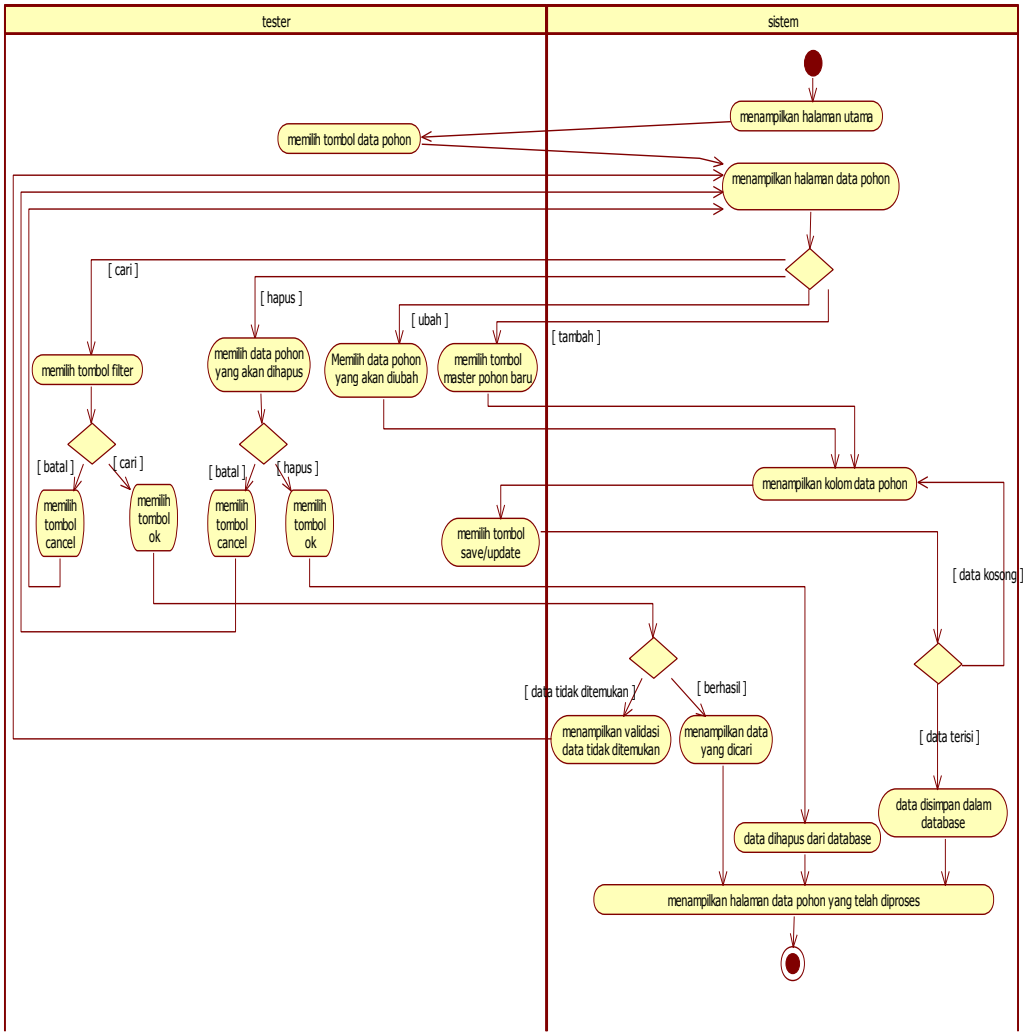
Identifikasi	
Nomor	6
Nama	Mencetak laporan
Tujuan	Mencetak laporan data test
Deskripsi	Proses untuk mencetak data analysis untuk dikonsultasikan tester kepada pasien
Aktor	Tester
Skenario Utama	
Kondisi Awal	Menampilkan halaman awal identifikasi karakteristik seseorang dengan pola gambar.
Aksi Aktor	Reaksi Sistem
1. Memilih tombol print	
	2. Menampilkan halaman data cetak
3. Jika tester memilih tombol filter	
	4. Menampilkan halaman data cetak
5. Jika tester memilih tombol cetak pada data analysis yang telah dipilih	
	6. Menampilkan preview laporan yang akan dicetak
7. Memilih tombol kembali	
	8. Menampilkan laporan data test yang akan dicetak
Kondisi Akhir	Menampilkan halaman data cetak

3.1.6.5 Activity Diagram

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dari urutan aktivitas dalam suatu proses yang mengacu pada *use case diagram*. *Activity diagram* pada aplikasi ini dibuat menjadi 5 bagian seperti pada use case diagram yang telah dibuat.

1. Activity diagram mengatur data pohon

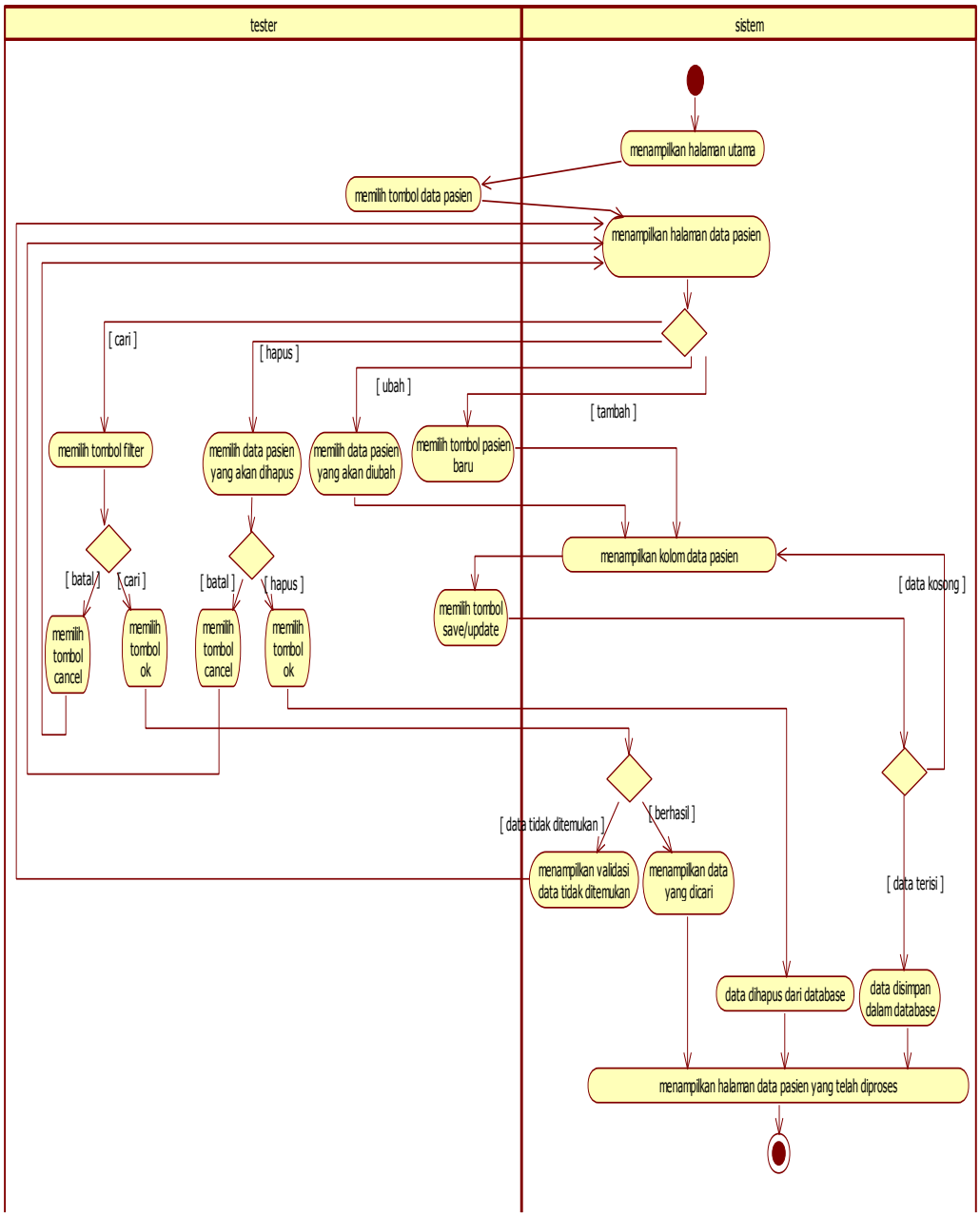
   Activity diagram mengatur data pohon mendeskripsikan aliran kerja dari urutan aktivitas dalam mengatur data pohon, seperti tambah master data pohon, ubah master data pohon, hapus master data pohon yang mengacu pada *use case* diagram. Dapat dilihat seperti pada gambar 3.3.



Gambar 3.3 Activity diagram mengatur data pohon

2. Activity diagram mengatur data pasien

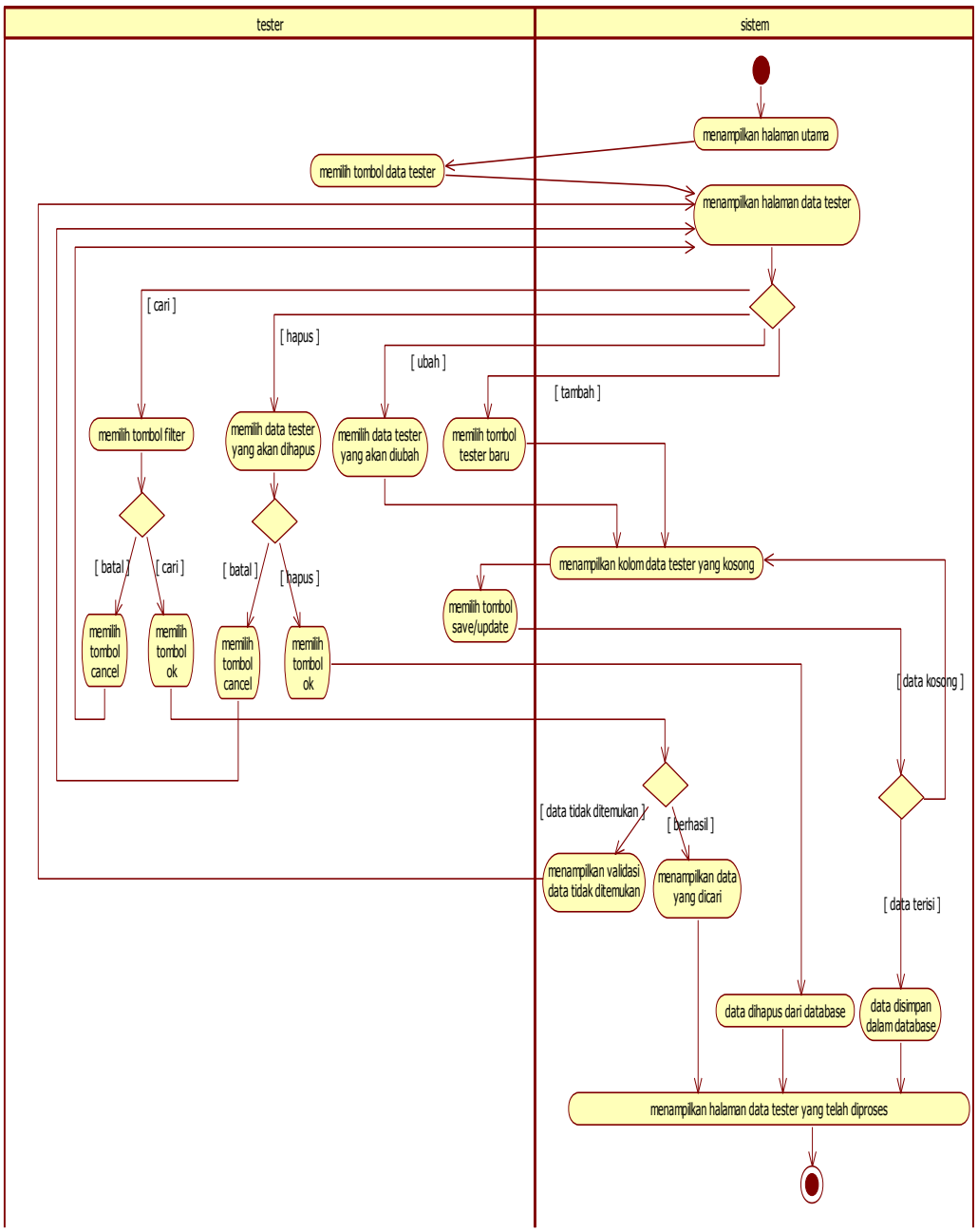
Activity diagram mengatur data pasien mendeskripsikan aliran kerja dari urutan aktivitas dalam mengatur data pasien, seperti tambah data pasien, ubah data pasien, hapus data pasien yang mengacu pada *use case* diagram. Dapat dilihat seperti pada gambar 3.4.



Gambar 3.4 Activity diagram mengatur data pasien

3. Activity diagram mengatur data tester

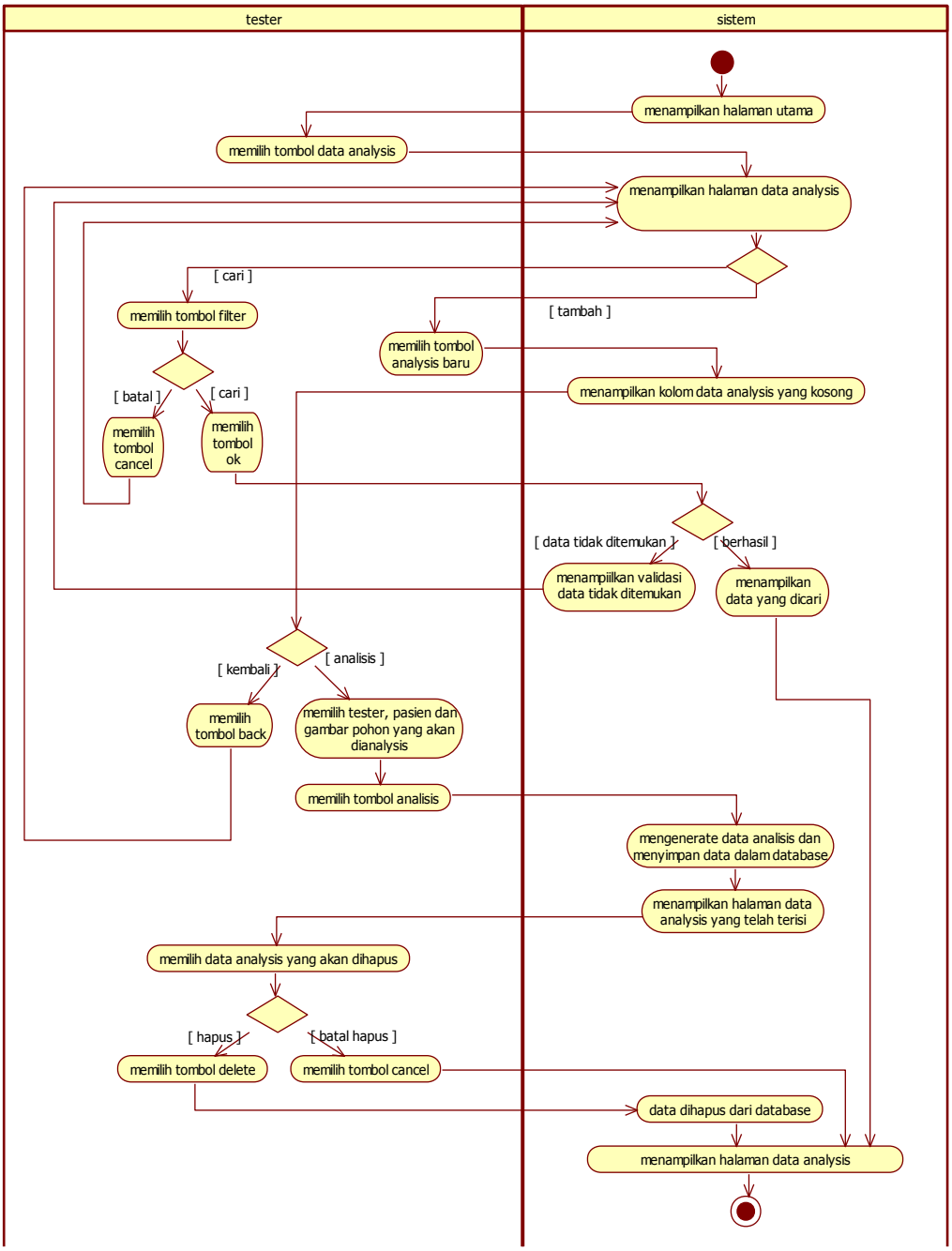
Activity diagram mengatur data tester mendeskripsikan aliran kerja dari urutan aktivitas dalam mengatur data tester, seperti tambah data tester, ubah data tester, hapus data tester yang mengacu pada use case diagram. Dapat dilihat seperti pada gambar 3.5.



Gambar 3.5 Activity diagram mengatur data tester

4. Activity diagram menghasilkan data analysis

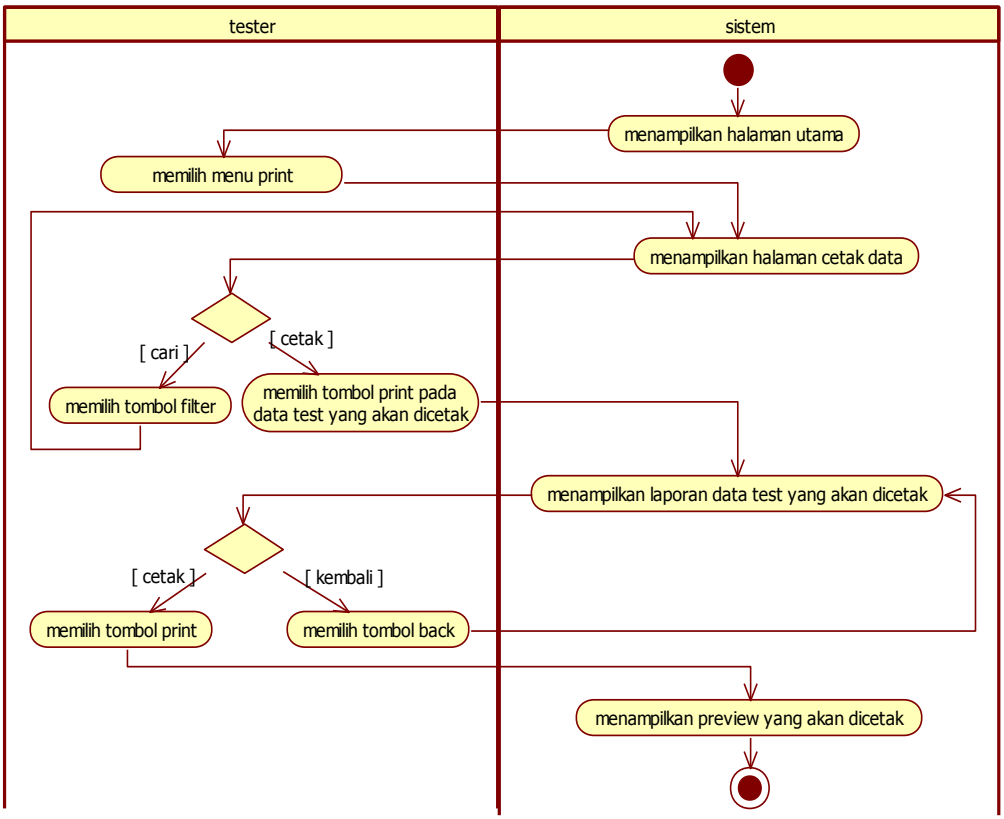
Activity diagram menghasilkan data analysis yang mendeskripsikan aliran kerja dari urutan aktivitas dalam menghasilkan data analysis yang mengenerate hasil analisis dari dari data pasien dan gambar pasien tersebut. Dapat dilihat seperti pada gambar 3.6.



Gambar 3.6 Activity diagram menghasilkan data analysis

5. Activity diagram mencetak laporan

Activity diagram mencetak laporan mendeskripsikan aliran kerja dari urutan aktivitas dalam mencetak laporan yang mengacu pada use case diagram. Dapat dilihat seperti pada gambar 3.7.

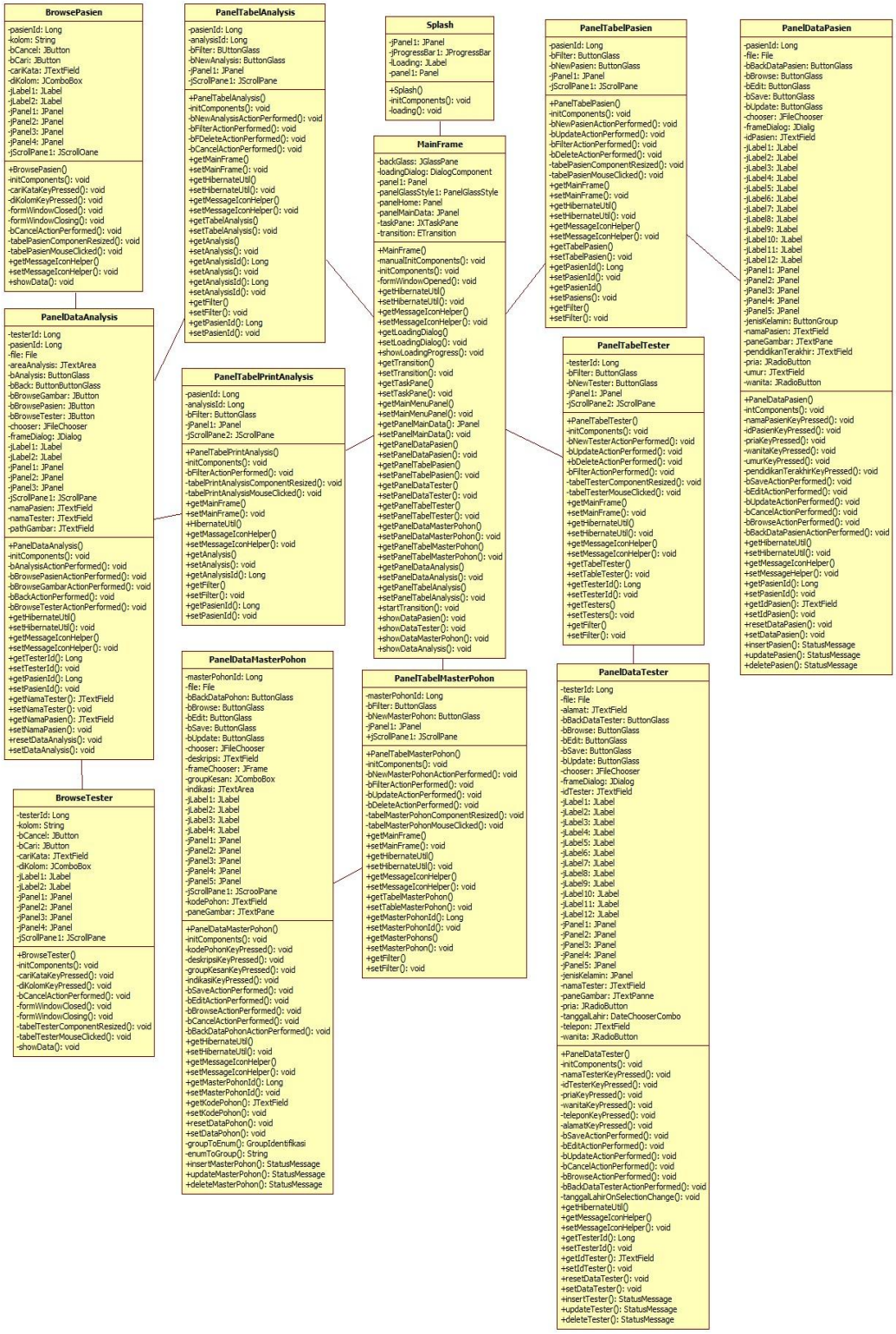


Gambar 3.7 Activity diagram mencetak laporan

3.1.6.6 Class Diagram

Class adalah sebuah spesifikasi yang jika diinstasiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/property) suatu system, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Pada aplikasi pengidentifikasian karakteristik seseorang melalui pola gambar, digambarkan pada class diagram seperti pada gambar 3.8.

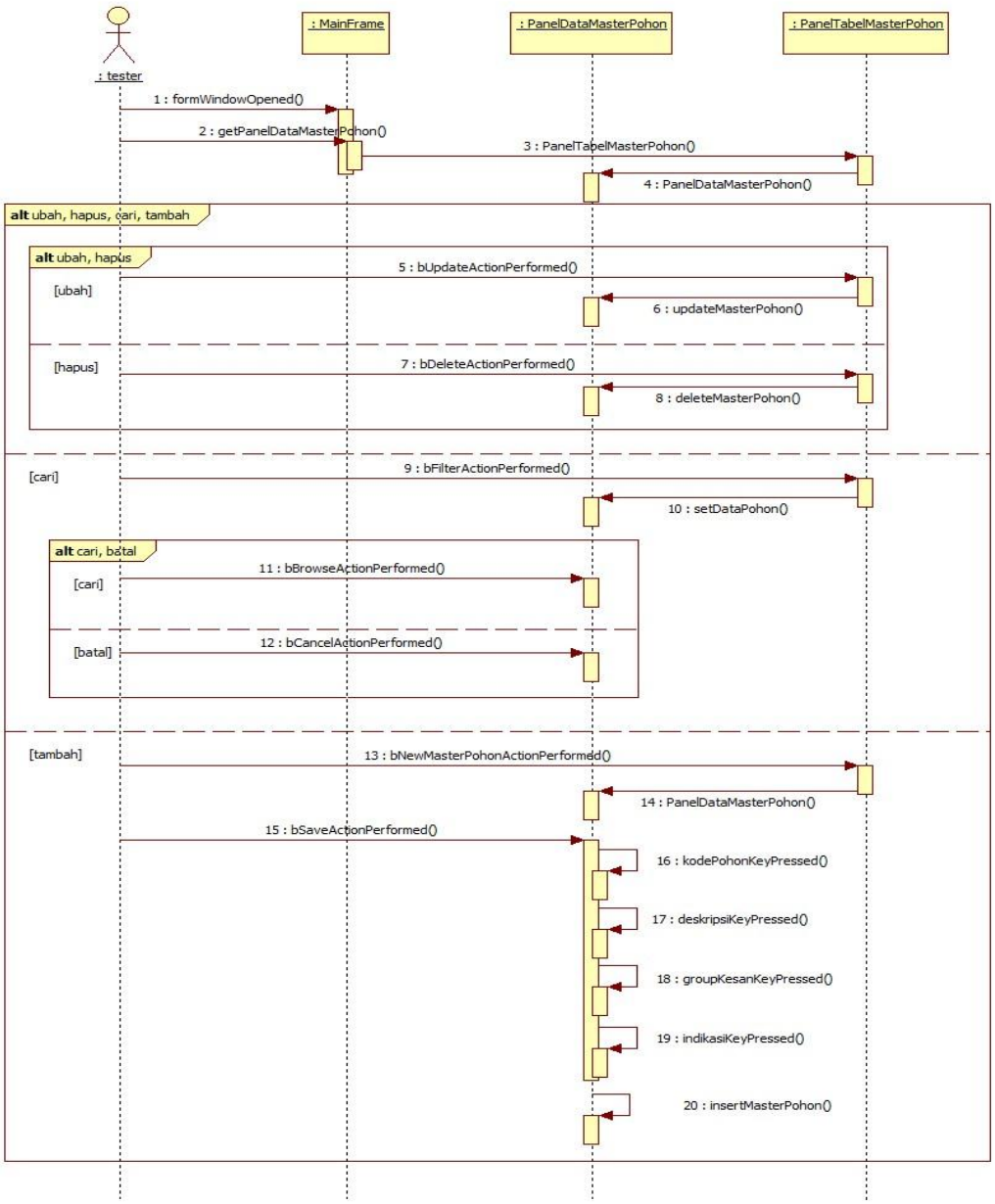


Gambar 3.8 Class Diagram

3.1.6.7 Sequence Diagram

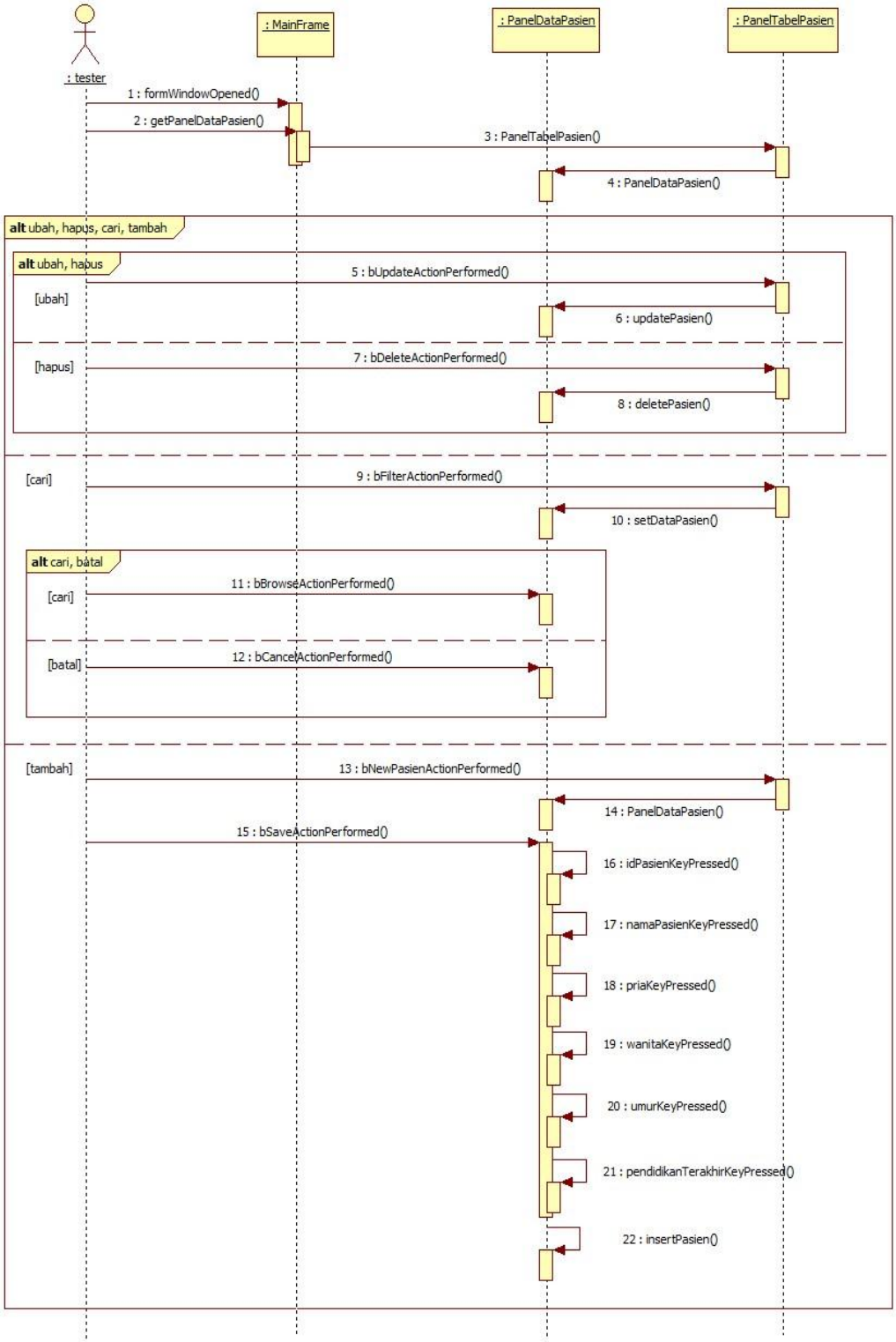
Sequence diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian untuk menghasilkan *output* tertentu. *Sequence* diagram diawali dari apa yang me-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan.

1. Sequence data pohon



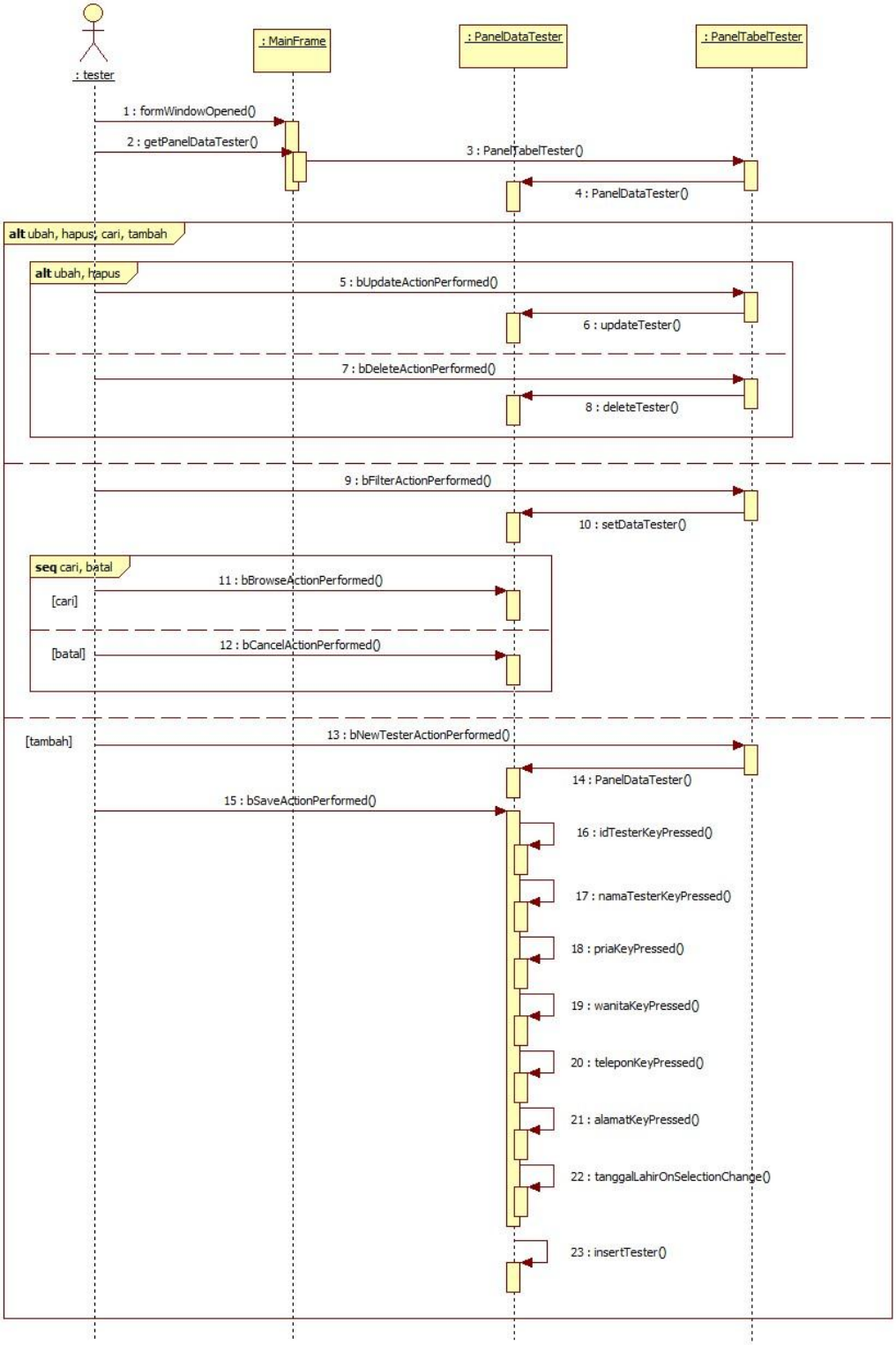
Gambar 3.9 Sequence data pohon

2. Sequence data pasien



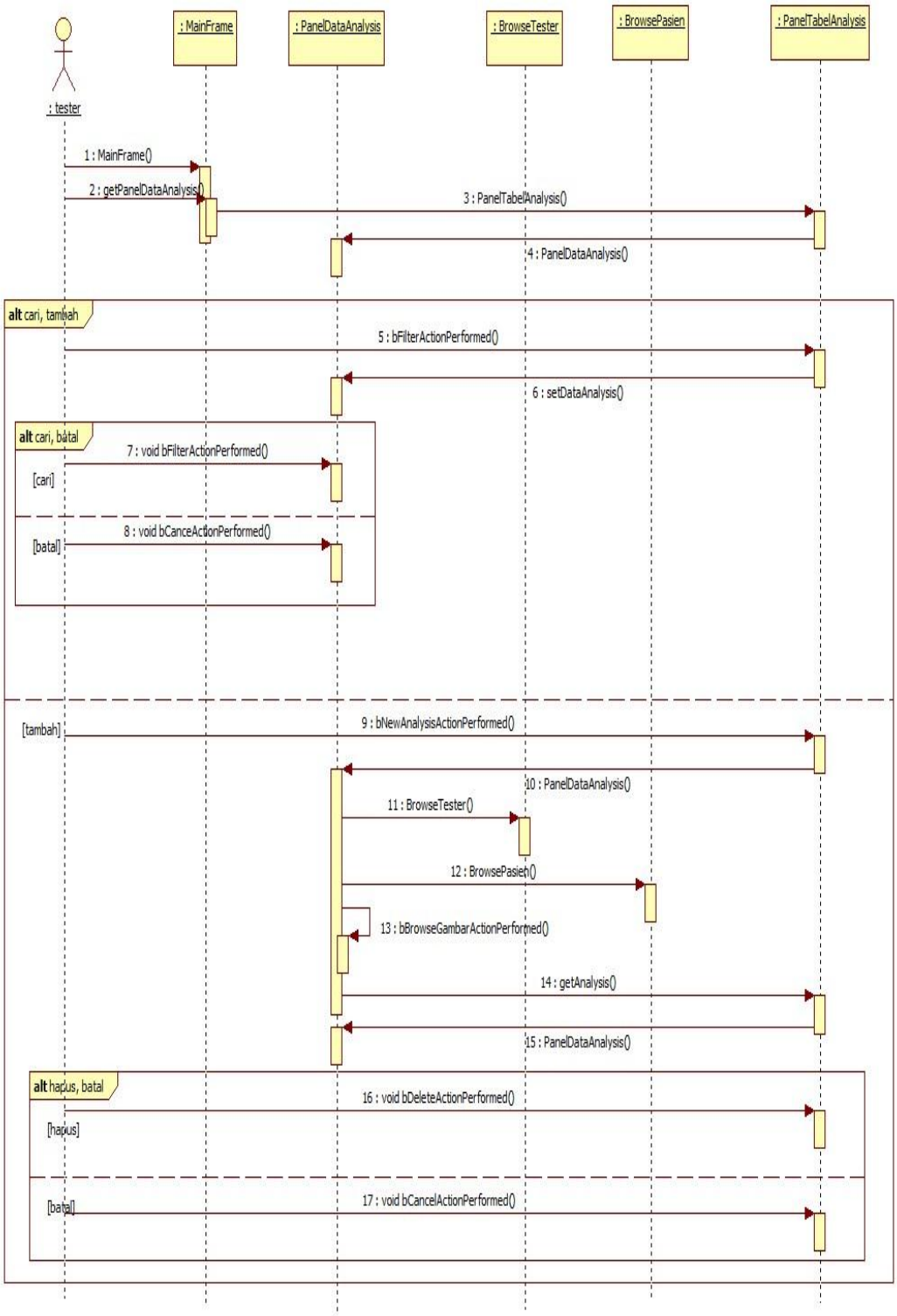
Gambar 3.10 Sequence data pasien

3. Sequence data tester



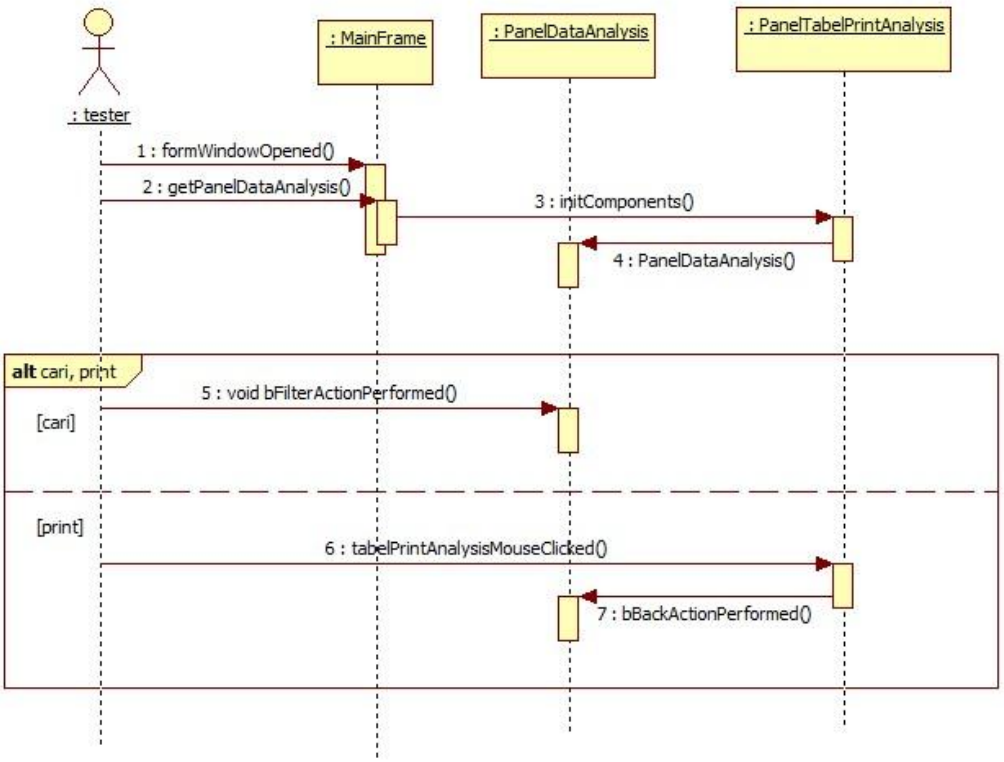
Gambar 3.11 Sequence Data Tester

4. Sequence data analysis



Gambar 3.12 Sequence Data Analysis

5. Sequence cetak laporan



Gambar 3.13 Sequence Cetak Laporan

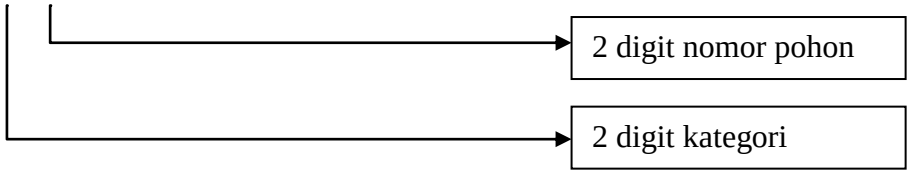
3.2 Perancangan Sistem

3.2.1 Perancangan Pengkodean

Perancangan pengkodean merupakan perancangan yang bertujuan untuk mengidentifikasi dan mengklasifikasi data agar dapat lebih mudah dalam proses pengolahan data dalam sistem. Berikut merupakan perancangan pengkodean data pohon yang disarankan agar sistem berjalan lebih baik :

Pada aplikasi pengidentifikasian karakteristik seseorang memiliki data pohon yang diberikan kode nomor urut pohon. Kode pohon terdiri dari 4 digit.

Format : XX-99



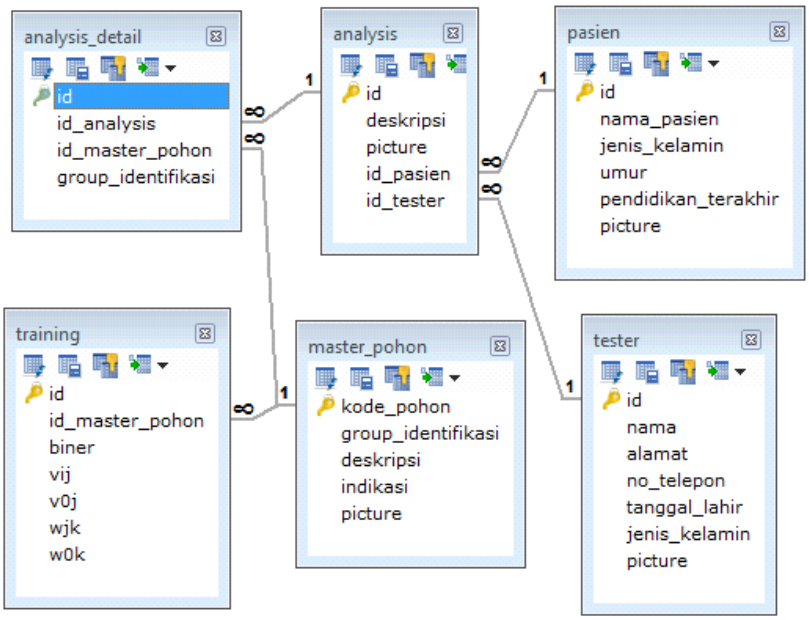
Contoh : KU01, KU menunjukkan kategori kesan umum dan 01 adalah nomor urut pohon.

3.2.2 Perancangan Database

Perancangan database merupakan perancangan kumpulan data yang terhubung dan tersimpan secara bersama-sama.

3.2.2.1 Skema Relasi

Skema relasi menggambarkan suatu hubungan antar tabel yang sudah ada dalam keadaan normal. Perancangan tabel relasi dalam membangun sebuah perangkat lunak aplikasi ini dapat dilihat pada gambar 3.14.



Gambar 3.14 Skema Relasi

3.2.2.2 Perancangan Struktur Tabel

Berikut diberikan struktur tabel yang terdapat pada aplikasi pengidentifikasian karakteristik seseorang, yang terdiri dari tabel analysis, tabel detail analysis, tabel master pohon, tabel tester, tabel pasien.

1. Tabel Analysis

Tabel 3.42 Data Analysis

Nama Entitas: analysis		
Deskripsi Entitas: tabel untuk menyimpan data analisis		
Atribut	Tipe	Keterangan
id	bigint(20) NOT NULL	PK, Unique
deskripsi	varchar(500) NULL	
picture	blob NULL	
id_pasien	bigint(20) NULL	FK, Ref pasien(id)
id_tester	bigint(20) NULL	FK, Ref tester(id)

2. Tabel Detail Analysis

Tabel 3.43 Data Detail Analysis

Nama Entitas: analysis_detail		
Deskripsi Entitas: tabel untuk menyimpan data detail analisis		
Atribut	Tipe	Keterangan
id	bigint(20) NOT NULL	PK, Unique
id_analysis	bigint(20) NULL	FK, Ref analysis_detail(id)
id_master_pohon	varchar(20) NULL	FK, Ref Pasien(id)
group_identifikasi	varchar(25) NULL	

3. Tabel Master Pohon

Tabel 3.44 Data Master Pohon

Nama Entitas: master_pohon		
Deskripsi Entitas: tabel untuk menyimpan data master pohon		
Atribut	Tipe	Keterangan
kode_pohon	varchar(20) NULL	PK, Unique
group_identifikasi	varchar(30) NULL	
deskripsi	varchar(250) NULL	
indikasi	varchar(500) NULL	
picture	longblob NULL	

4. Tabel Tester

Tabel 3.45 Data Tester

Nama Entitas: tester		
Deskripsi Entitas: tabel untuk menyimpan data tester		
Atribut	Tipe	Keterangan
id	bigint(20) NOT NULL	PK, Unique
nama	varchar(30) NULL	

Nama Entitas: tester		
Deskripsi Entitas: tabel untuk menyimpan data tester		
alamat	varchar(80) NULL	
telepon	varchar(13) NULL	
tanggal_lahir	date NULL	
jenis_kelamin	varchar(10) NULL	
picture	longblob NULL	

5. Tabel Pasien

Tabel 3.46 Data Pasien

Nama Entitas: pasien		
Deskripsi Entitas: tabel untuk menyimpan data pasien		
Atribut	Tipe	Keterangan
id	bigint(20) NOT NULL	PK, Unique
nama_pasien	varchar(35) NULL	
jenis_kelamin	varchar(25) NULL	
umur	int(11) NULL	
pendidikan_terakhir	varchar(50) NULL	
picture	blob NULL	

6. Tabel Training

Tabel 3.47 Data Training

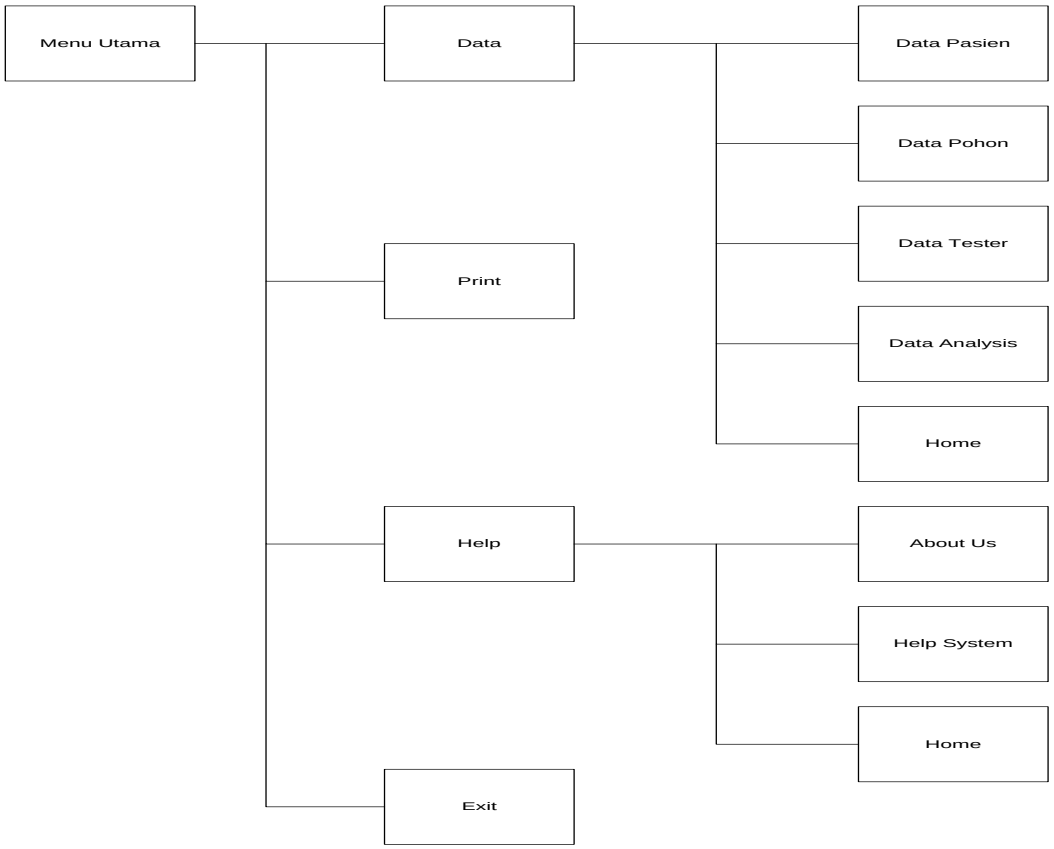
Nama Entitas: training		
Deskripsi Entitas: tabel untuk menyimpan data training		
Atribut	Tipe	Keterangan
id	bigint(20) NOT NULL	PK, Unique
id_master_pohon	varchar(20) NULL	FK, Ref master_pohon (kode_pohon)
biner	Int(11) NULL	
vij	decimal(20,6) NULL	
v0j	decimal(20,6) NULL	
Wjk	decimal(20,6) NULL	
w0k	decimal(20,6) NULL	

3.2.3 Perancangan Menu

3.2.3.1 Perancangan Struktur Menu

Dalam perancangan aplikasi dibutuhkan struktur menu yang berisikan menu dan submenu-submenu yang berfungsi untuk memudahkan user dalam

menggunakan aplikasi tersebut. Berikut ini adalah gambar struktur menu pada aplikasi pengidentifikasian karakteristik seseorang menggunakan pola gambar, seperti pada gambar 3.15.



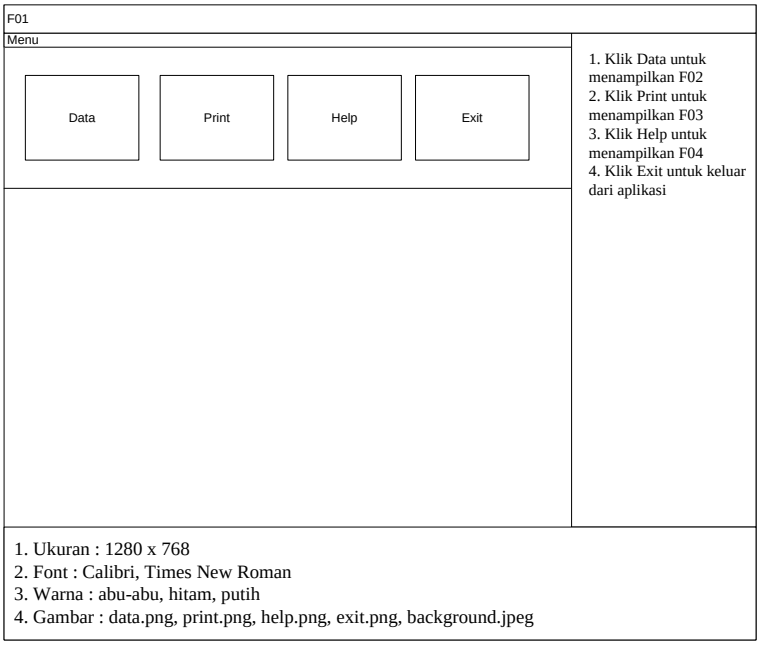
Gambar 3.15 Perancangan StrukturMenu

3.2.3.2 Perancangan Antar Muka

Antar muka merupakan tampilan dari suatu perangkat lunak yang berperan sebagai media komunikasi yang digunakan sebagai sarana berdialog antara perangkat lunak dengan *user*. Perancangan antar muka perangkat lunak pada aplikasi pengindentifikasian karakter seseorang ini dapat digambarkan seperti berikut:

1. Perancangan Antar Muka Halaman Utama

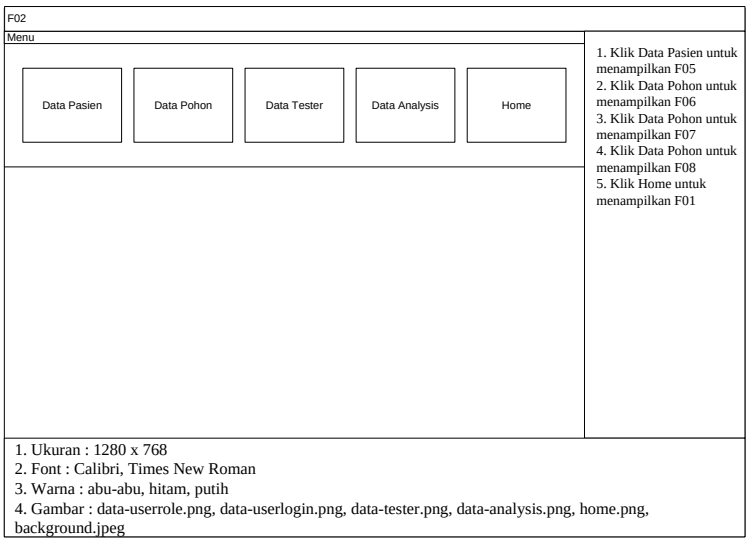
   Perancarangan antar muka halaman utama menampilkan menu-menu utama yang terdapat pada aplikasi. Seperti pada gambar 3.16.



Gambar 3.16 Perancangan Antar Muka Halaman Utama

2. Perancangan Antar Muka Menu Data

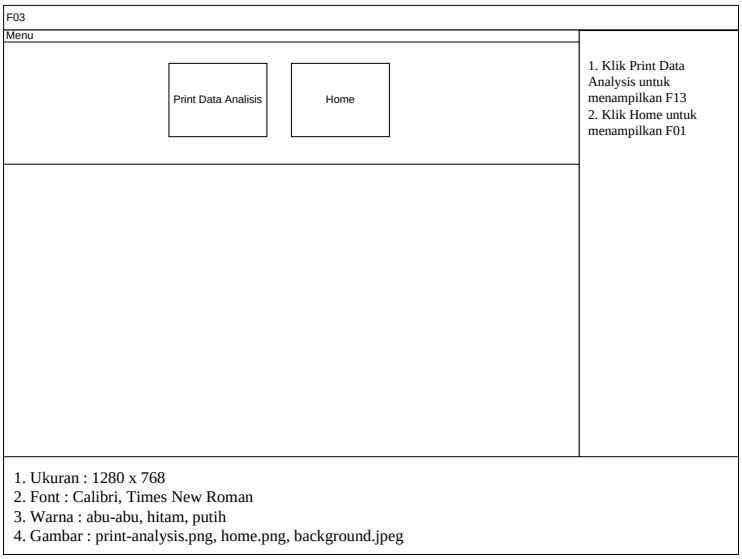
Perancangan antar muka menu data menampilkan halaman yang berisikan data-data yang akan tambahkan, diedit, maupun dihapus. Seperti pada gambar 3.17.



Gambar 3.17 Perancangan Antar Muka Menu Data

3. Perancangan Antar Muka Menu Print

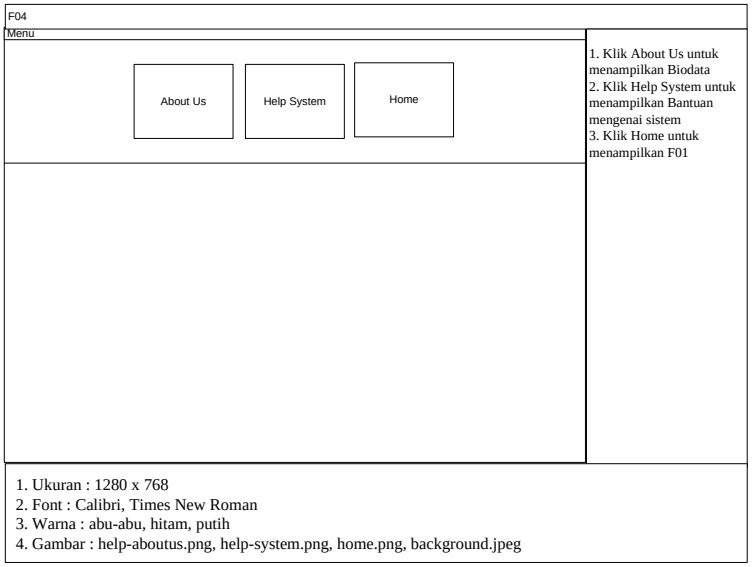
   Perancangan antar muka menu print menampilkan menu print data analysis seperti pada gambar 3.18.



Gambar 3.18 Perancangan Antar Muka Menu Print

4. Perancangan Antar Muka Menu Help

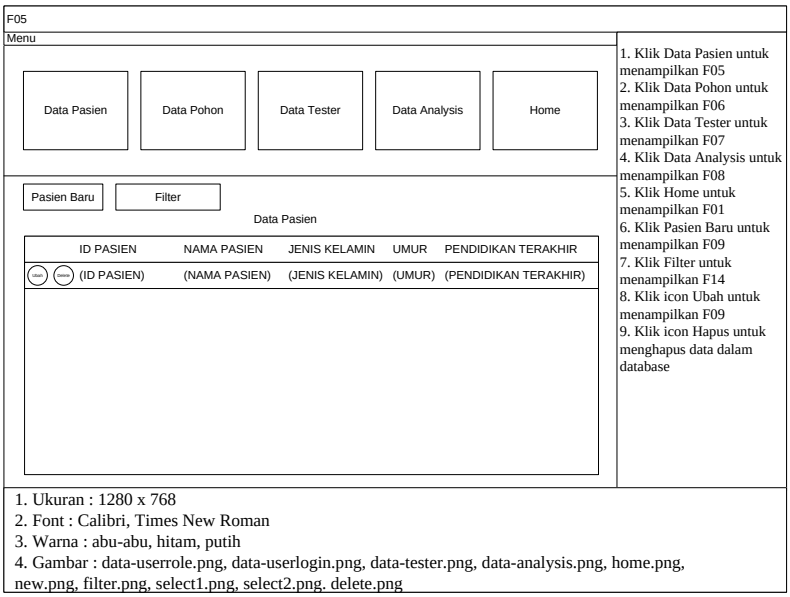
   Perancangan antar muka menu help menampilkan menu identitas, menu help system. Seperti pada gambar 3.19.



Gambar 3.19 Perancangan Antar Muka Menu Help

5. Perancangan Antar Muka Data Pasien

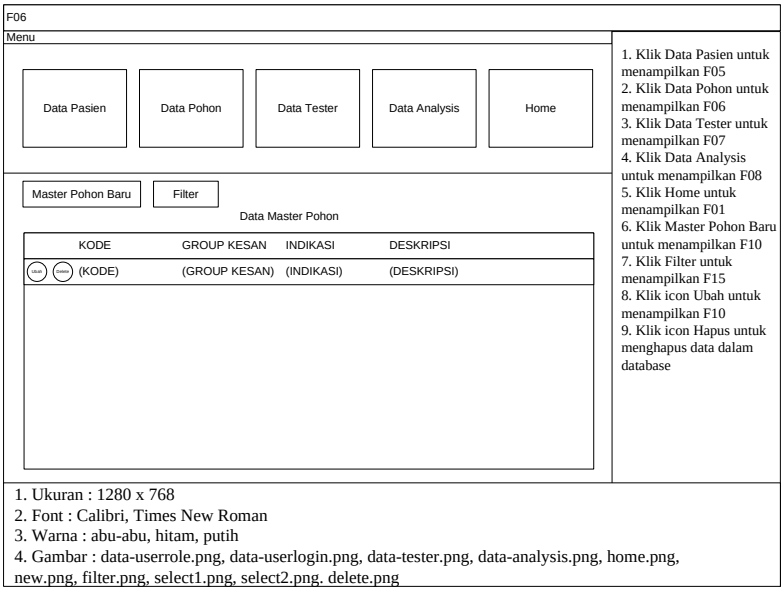
Perancangan antar muka data pasien menampilkan data-data informasi mengenai pasien. Seperti pada gambar 3.20.



Gambar 3.20 Perancangan Antar Muka Data Pasien

6. Perancangan Antar Muka Data Pohon

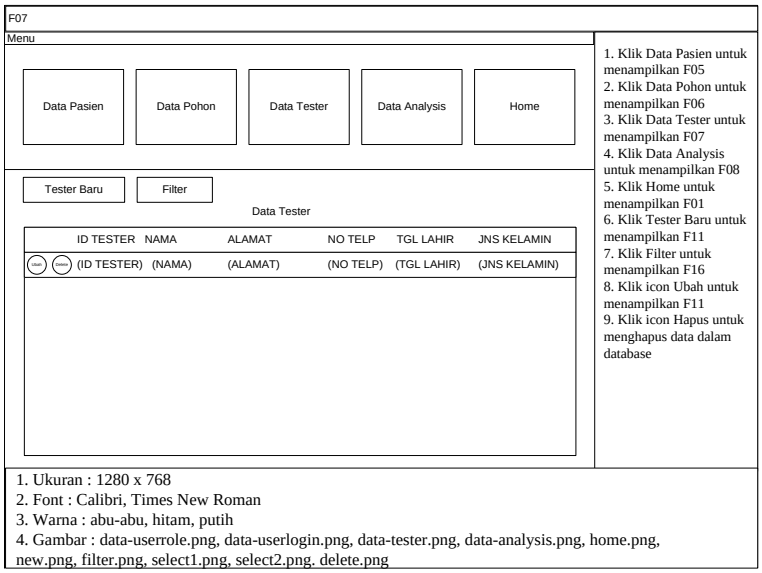
Perancangan antar muka data pohon menampilkan data-data informasi mengenai master pohon. Seperti pada gambar 3.21.



Gambar 3.21 Perancangan Antar Muka Data Pohon

7. Perancangan Antar Muka Data Tester

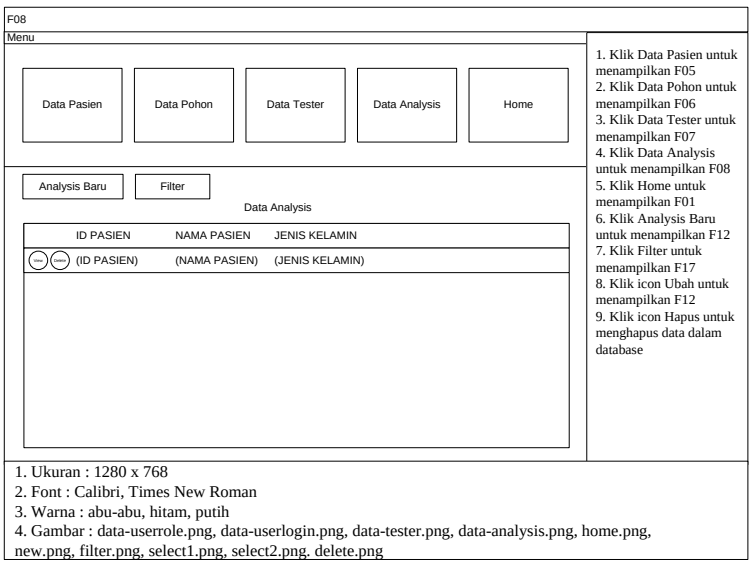
Perancangan antar muka data tester menampilkan data-data yang berisikan informasi mengenai tester. Seperti pada gambar 3.22.



Gambar 3.22 Perancangan Antar Muka Data Tester

8. Perancangan Antar Muka Data Analysis

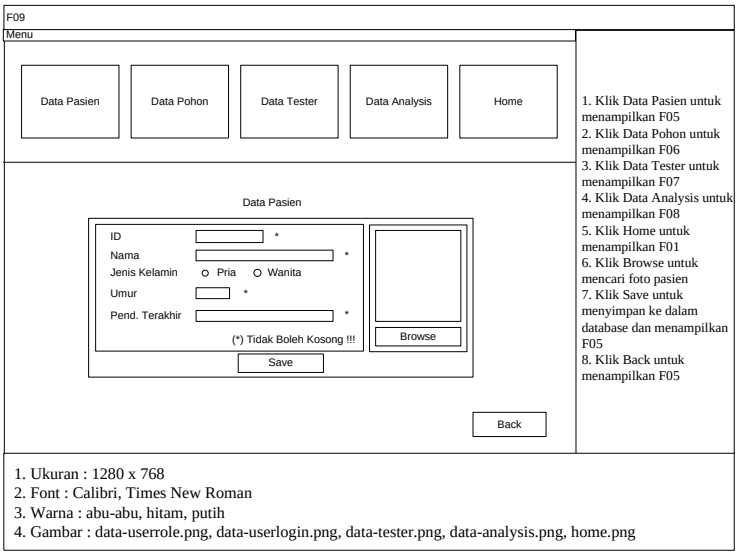
Perancangan antar muka data analysis menampilkan data-data yang berisikan mengenai analysis dari gambar pasien. Seperti pada gambar 3.23.



Gambar 3.23 Perancangan Antar Muka Data Analysis

9. Perancangan Antar Muka Isi Data Pasien

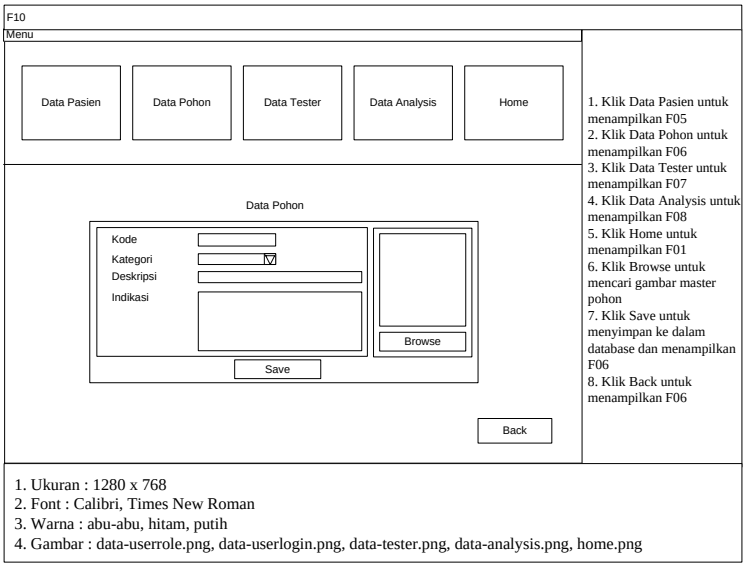
Perancangan antar muka isi data pasien menampilkan data setiap pasien yang ingin ditambahkan, maupun diubah. Seperti pada gambar 3.24.



Gambar 3.24 Perancangan Antar Muka Tambah Data Pasien

10. Perancangan Antar Muka Isi Data Pohon

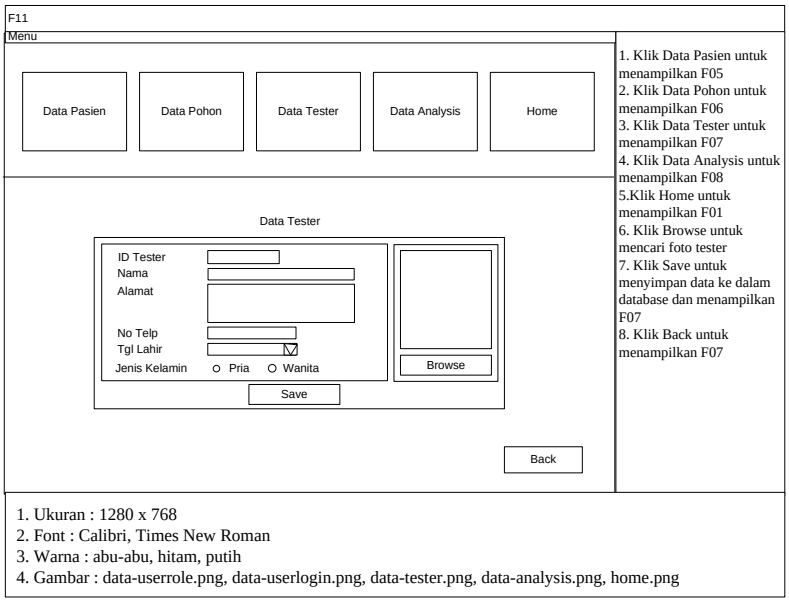
Perancangan antar muka isi data pohon menampilkan setiap data master pohon yang ingin ditambahkan maupun diubah. Seperti pada gambar 3.25.



Gambar 3.25 Perancangan Antar Muka Tambah Data Pohon

11. Perancangan Antar Muka Isi Data Tester

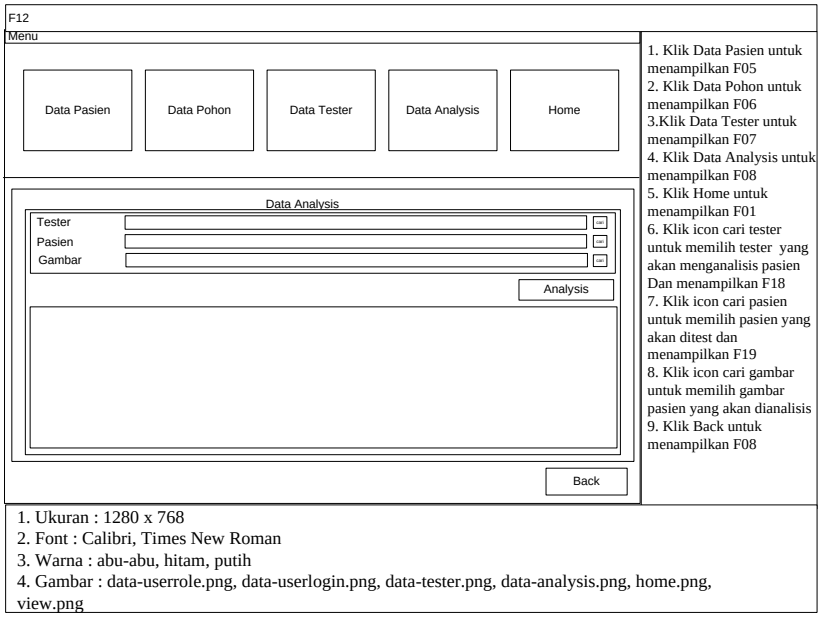
Perancangan antar muka isi data tester menampilkan setiap data tester yang akan ditambahkan maupun diubah. Seperti pada gambar 3.26.



Gambar 3.26 Perancangan Antar Muka Tambah Data Tester

12. Perancangan Antar Muka Isi Data Analysis

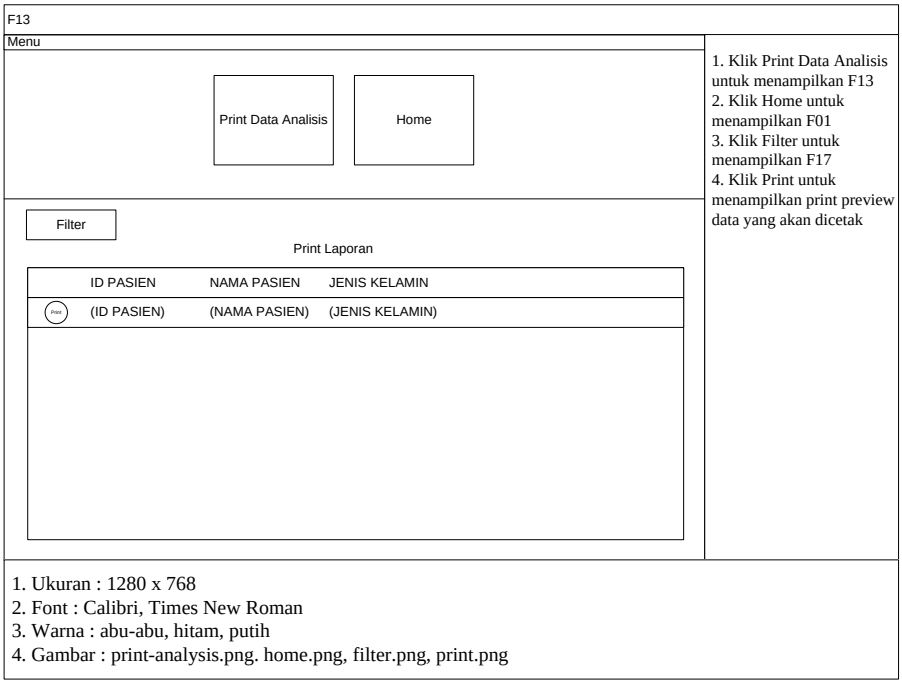
Perancangan antar muka isi data analysis menampilkan setiap data analysis yang akan ditambahkan maupun diubah. Seperti pada gambar 3.27.



Gambar 3.27 Perancangan Antar Muka Tambah Data Analysis

13. Perancangan Antar Muka Print Data Analisis

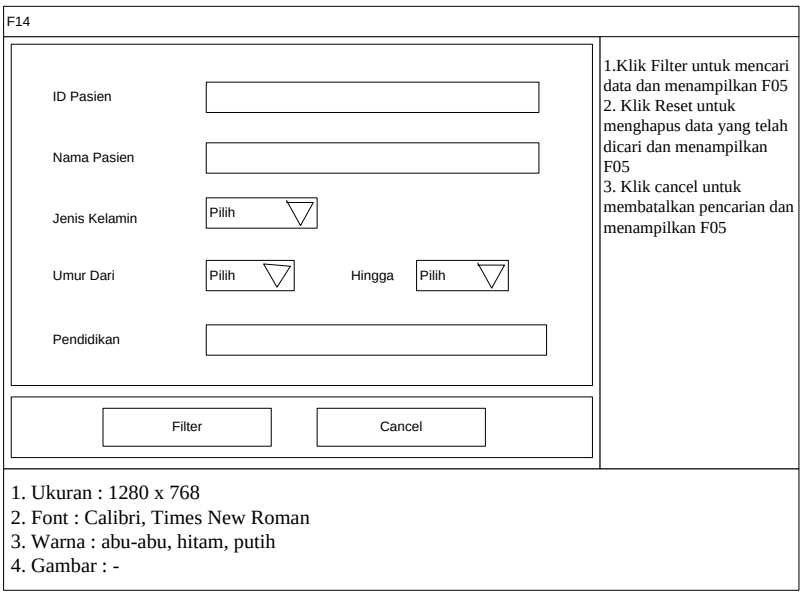
Perancangan antar muka print data analysis menampilkan daftar data analysis yang akan dicetak. Seperti pada gambar 3.28.



Gambar 3.28 Perancangan Antar Muka Print Data Analisis

14. Perancangan Antar Muka Filter Data Pasien

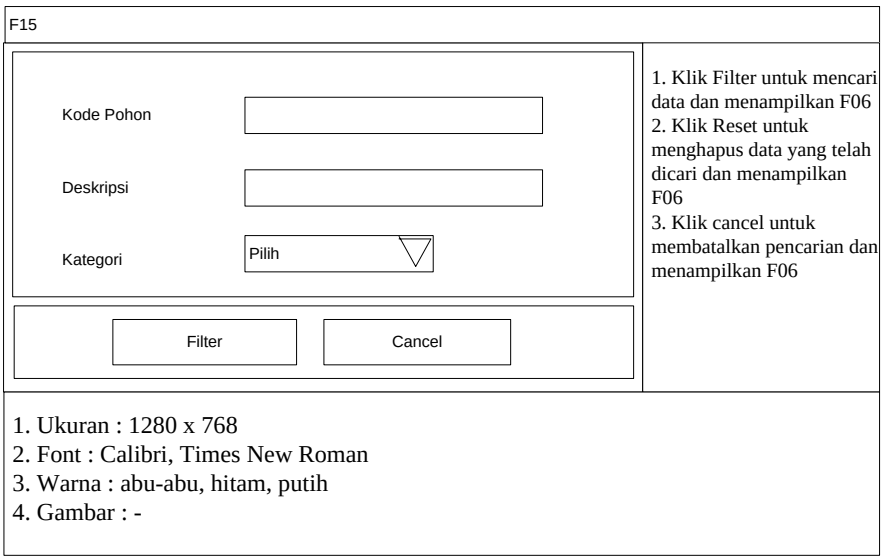
Perancangan antar muka filter data pasien menampilkan kolom data pasien yang akan dicari berdasarkan data yang akan dicari. Seperti pada gambar 3.29.



Gambar 3.29 Perancangan Antar Muka Filter Data Pasien

15. Perancangan Antar Muka Filter Data Pohon

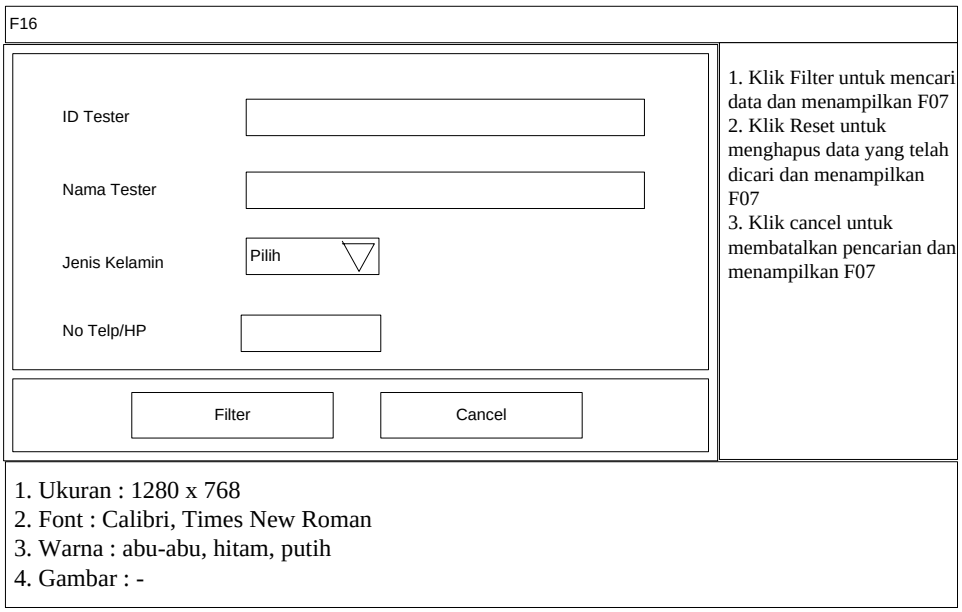
Perancangan antar muka filter data pohon menampilkan kolom data pohon yang akan dicari berdasarkan data yang akan dicari. Seperti pada gambar 3.30.



Gambar 3.30 Perancangan Antar Muka Filter Data Pohon

16. Perancangan Antar Muka Filter Data Tester

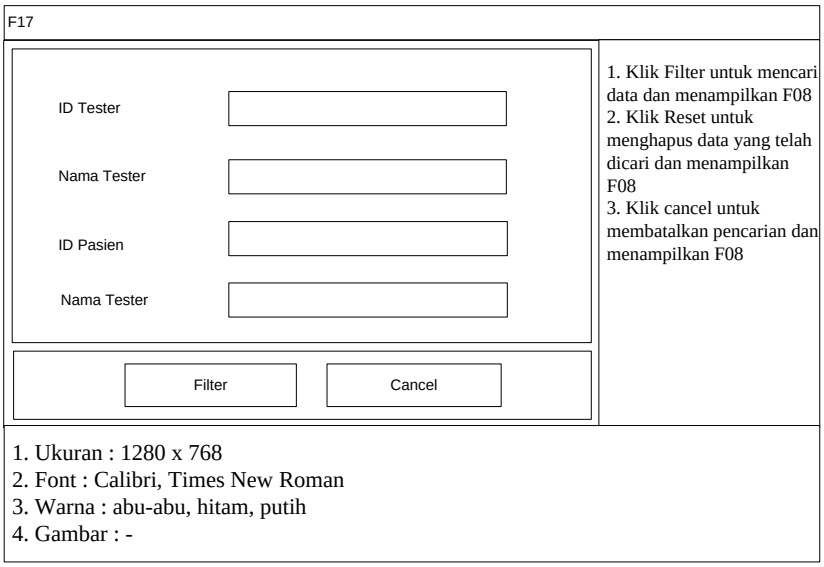
Perancangan antar muka filter data tester menampilkan kolom data tester yang akan dicari berdasarkan data yang akan dicari. Seperti pada gambar 3.31.



Gambar 3.31 Perancangan Antar Muka Filter Data Tester

17. Perancangan Antar Muka Filter Data Analysis

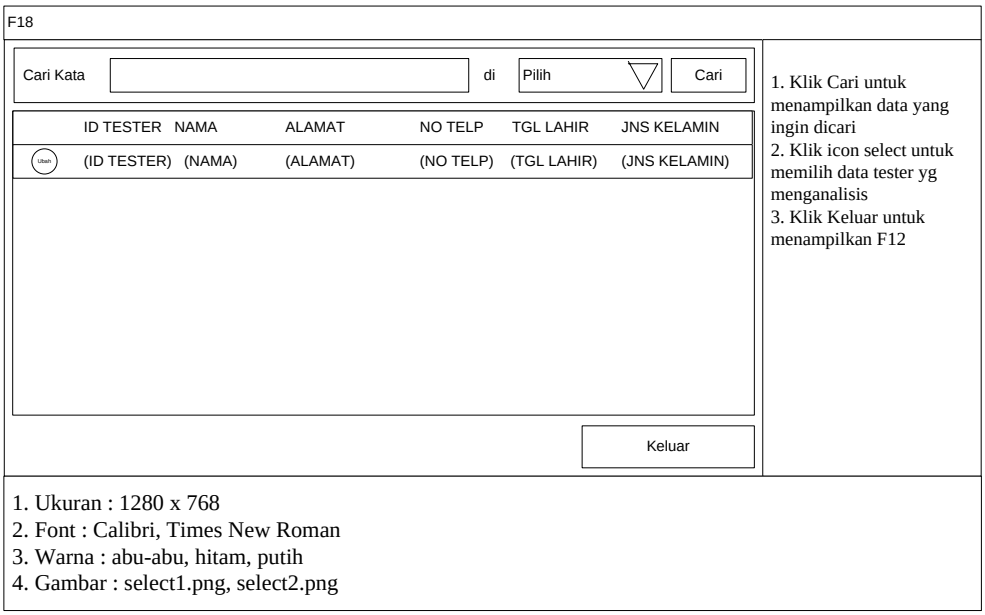
Perancangan antar muka filter data analysis menampilkan data analysis yang akan dicari berdasarkan data yang akan dicari. Seperti pada gambar 3.32.



Gambar 3.32 Perancangan Antar Muka Filter Data Analysis

18. Perancangan Antar Muka Cari Data Tester

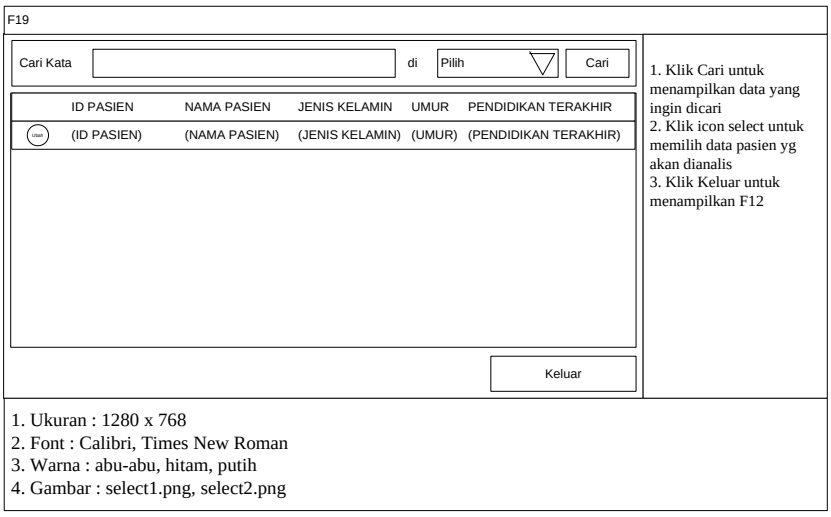
Perancangan antar muka cari data tester menampilkan data tester yang akan dicari berdasarkan data yang dipilih. Seperti pada gambar 3.33.



Gambar 3.33 Perancangan Antar Muka Cari Data Tester

19. Perancangan Antar Muka Cari Data Pasien

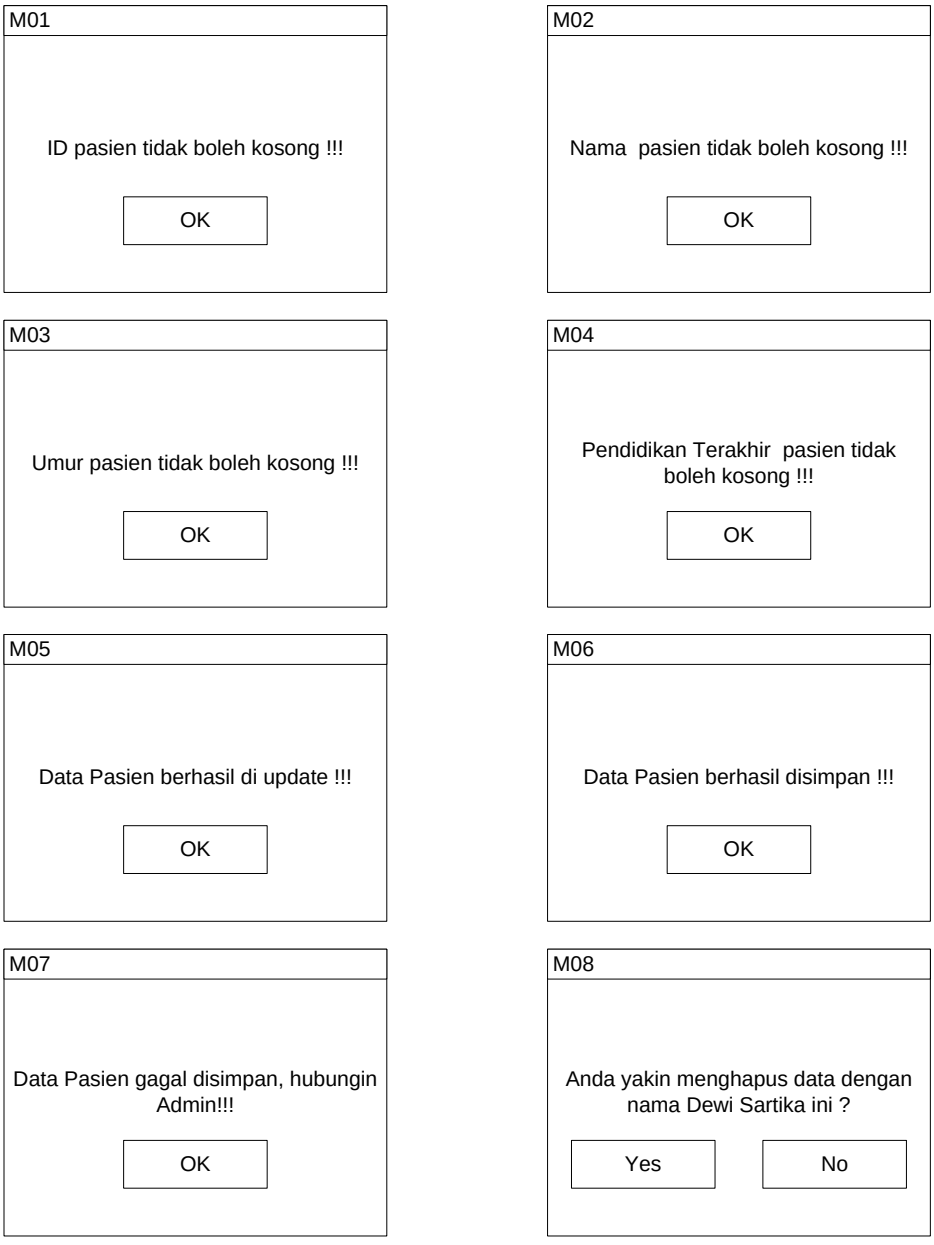
Perancangan antar muka cari data pasien menampilkan data pasien yang akan dicari berdasarkan data yang dipilih. Seperti pada gambar 3.34.

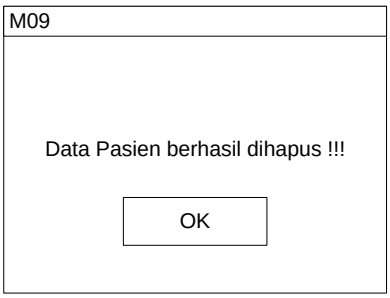


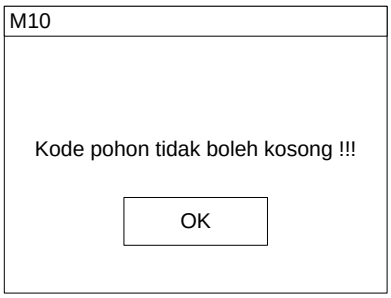
Gambar 3.34 Perancangan Antar Muka Cari Data Pasien

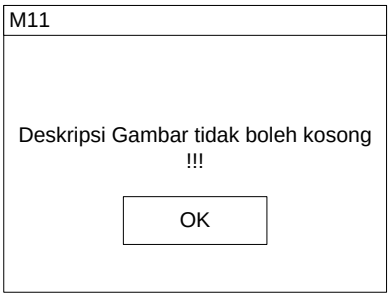
3.2.4 Perancangan Pesan

Perancangan pesan menampilkan kesalahan-kesalahan atau validasi dalam setiap penggunaan menu-menu yang terdapat pada aplikasi.

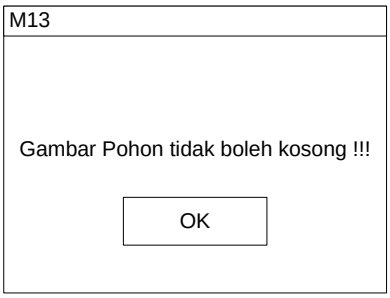


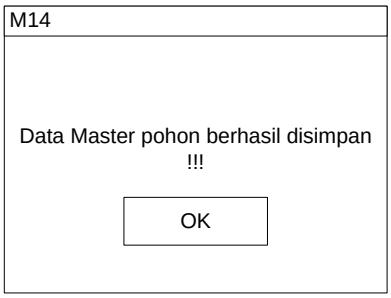

M09
Data Pasien berhasil dihapus !!!
OK

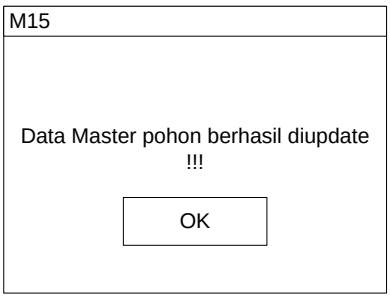

M10
Kode pohon tidak boleh kosong !!!
OK

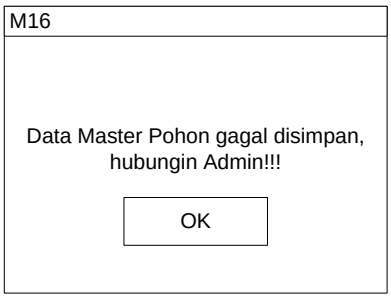

M11
Deskripsi Gambar tidak boleh kosong !!!
OK

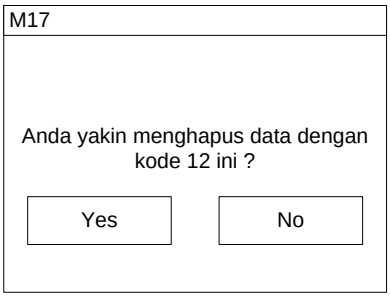

M12
Indikasi tidak boleh kosong !!!
OK

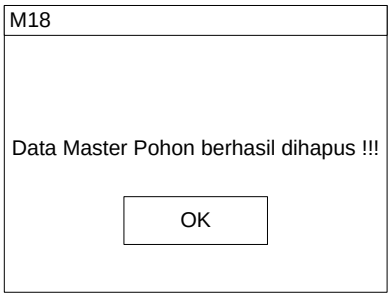

M13
Gambar Pohon tidak boleh kosong !!!
OK

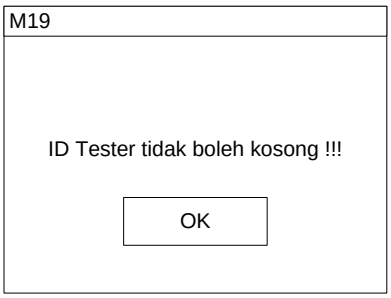

M14
Data Master pohon berhasil disimpan !!!
OK

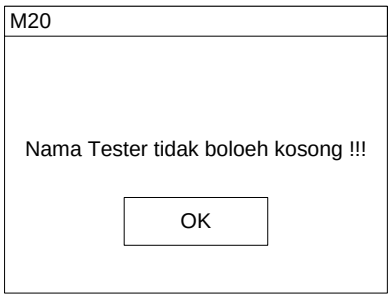

M15
Data Master pohon berhasil diupdate !!!
OK

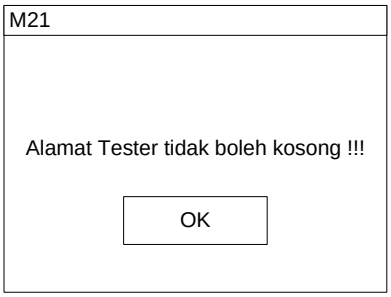

M16
Data Master Pohon gagal disimpan, hubungin Admin!!!
OK

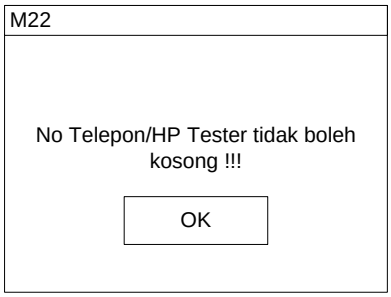

M17
Anda yakin menghapus data dengan kode 12 ini ?
Yes
No

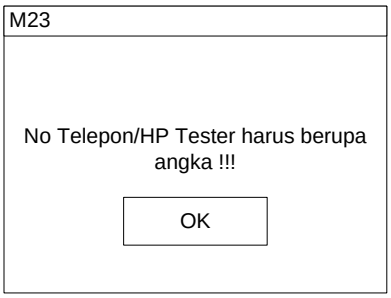

M18
Data Master Pohon berhasil dihapus !!!
OK

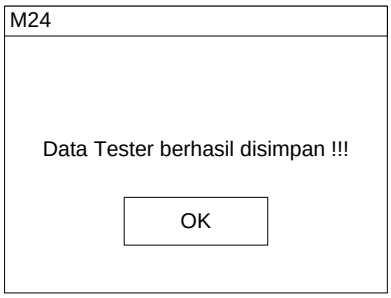

M19
ID Tester tidak boleh kosong !!!
OK

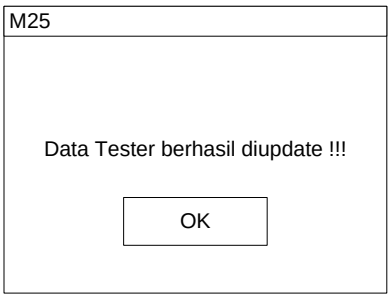

M20
Nama Tester tidak boloeh kosong !!!
OK

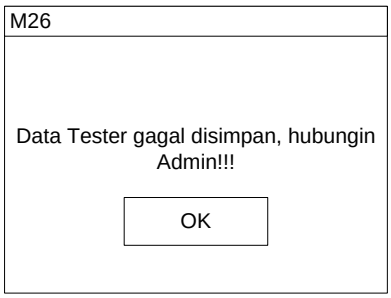

M21
Alamat Tester tidak boleh kosong !!!
OK

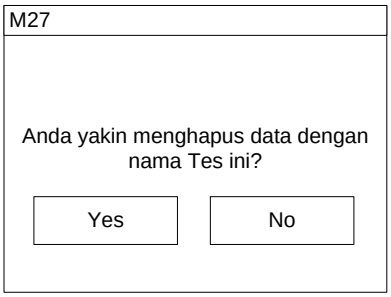

M22
No Telepon/HP Tester tidak boleh kosong !!!
OK


M23
No Telepon/HP Tester harus berupa angka !!!
OK

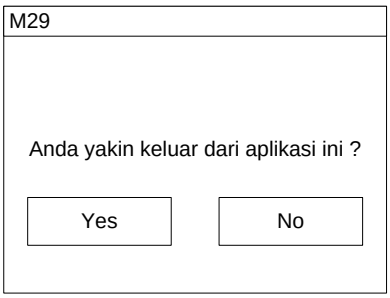

M24
Data Tester berhasil disimpan !!!
OK


M25
Data Tester berhasil diupdate !!!
OK


M26
Data Tester gagal disimpan, hubungin Admin!!!
OK


M27
Anda yakin menghapus data dengan nama Tes ini?
Yes
No

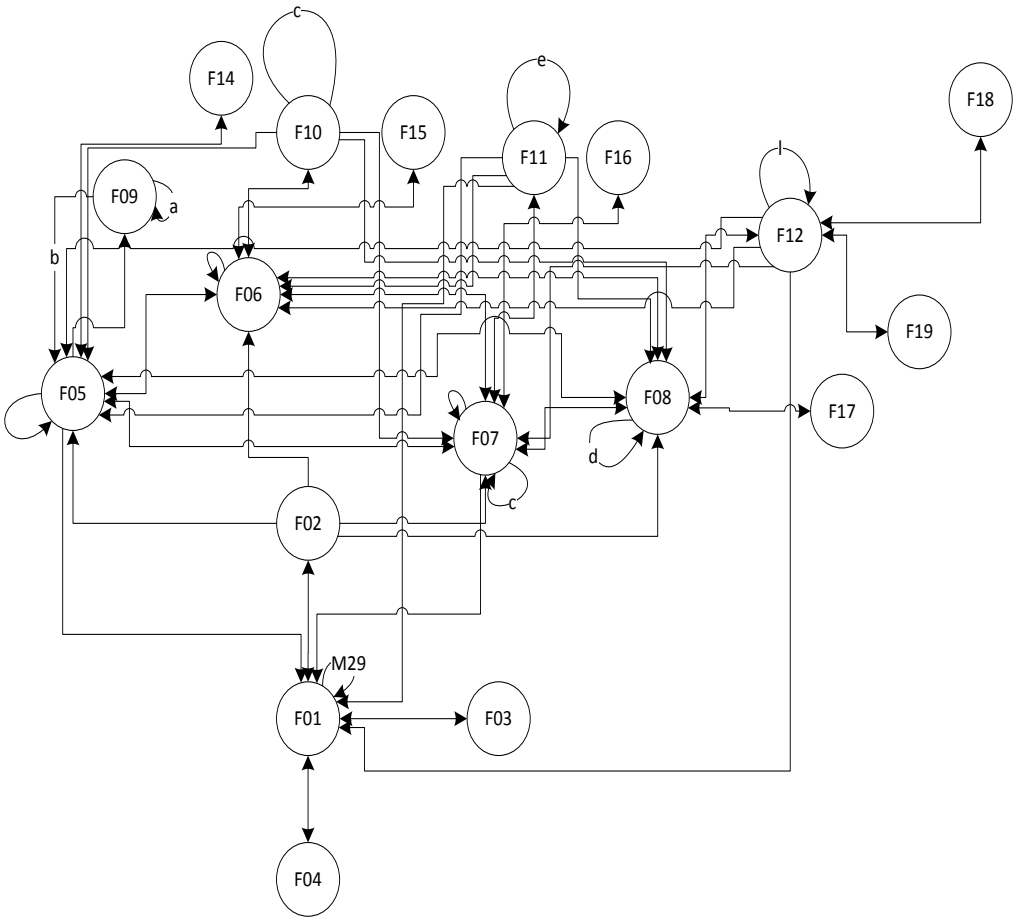

M28
Data Tester berhasil dihapus !!!
OK



Gambar 3.35 Perancangan Pesan

3.2.5 Jaringan Semantik

Jaringan semantik menggambarkan keterhubungan nagivasi menu dari satu halaman ke halaman lainnya. Jaringan semantik pada aplikasi pengidentifikasian karakterstik seseorang dapat dilihat pada gambar 3.36



Gambar 3.36 Jaringan Semantik

Keterangan :

a = M01, M02, M03, M04, M07, M08

b = M06, M09

c = M10, M11, M12, M13, M16, M17

d = M14, M15, M18

e = M19, M20, M21, M22, M23,M26

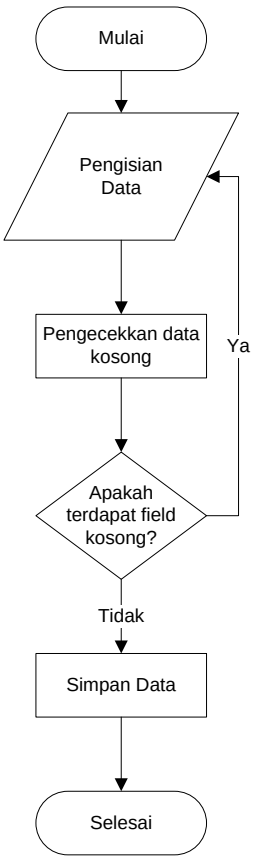
f = M24, M25, M27, M28

3.2.6 Perancangan Method

Sebagai langkah terakhir dalam perancangan yaitu merancang method yang akan diimplementasikan ke dalam sistem.

1. Perancangan Method Tambah Data

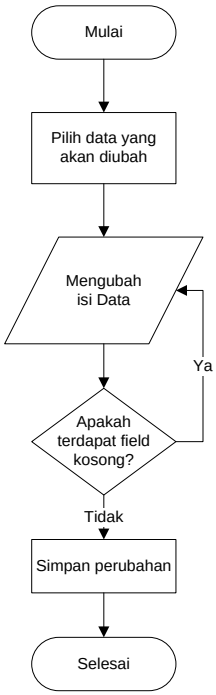
   Adapun perancangan method tambah data pada aplikasi pengidentifikasian karakteristik seseorang dengan pola gambar adalah seperti pada gambar 3.37.



Gambar 3.37 Perancangan Method Tambah Data

2. Perancangan Method Ubah Data

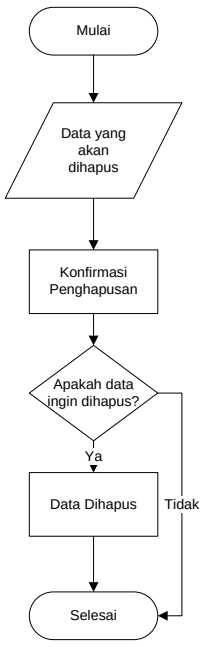
   Perancangan method ubah data dapat dilihat seperti gambar 3.38.



Gambar 3.38 Perancangan Method Ubah Data

3. Perancangan Method Hapus Data

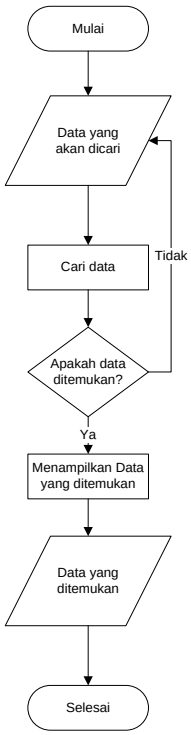
   Perancangan method hapus data dapat dilihat seperti gambar 3.39.



Gambar 3.39 Perancangan Method Hapus Data

4. Perancangan Method Cari Data

Perancangan method cari data dapat dilihat seperti gambar 3.40.



Gambar 3.40 Perancangan Method Cari Data