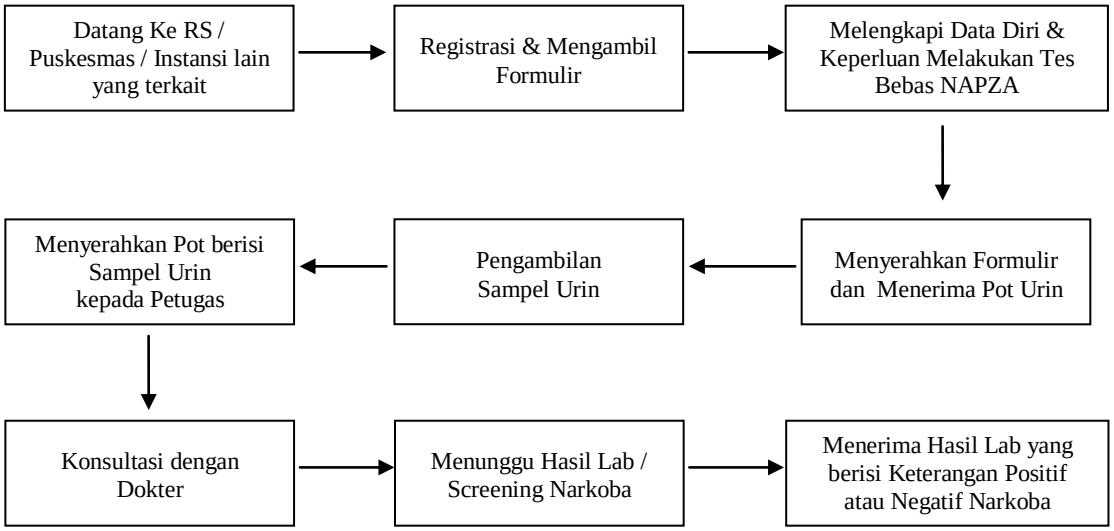
BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan kegiatan penguraian suatu sistem informasi yang utuh dan nyata ke dalam bagian-bagian atau komponen-komponen komputer yang bertujuan untuk mengidentifikasi serta mengevaluasi masalah-masalah yang muncul, hambatan-hambatan yang mungkin terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

3.1.1 Analisis Masalah

Sepanjang tahun 2013, BNN bersama Polisi sudah mengungkap banyak kasus penyelundupan narkoba baik melalui jalur darat, air maupun udara. Menemukan narkoba memang relatif lebih mudah dibandingkan mencari tahu siapa yang mengkonsumsinya. Para Polisi harus memeriksa setiap tersangka dengan sebuah tes. Tes narkoba biasanya dilakukan langsung di laboratorium yang ditunjuk khusus. Kebanyakan laboratorium melakukan pengujian dengan urine/air seni dari pasien untuk mendeteksi ada atau tidaknya narkotika dalam tubuh pecandu. Berikut ini alur pemeriksaan tes urin (urinalisis) yang disediakan oleh rumah sakit atau instansi kesehatan lainnya :

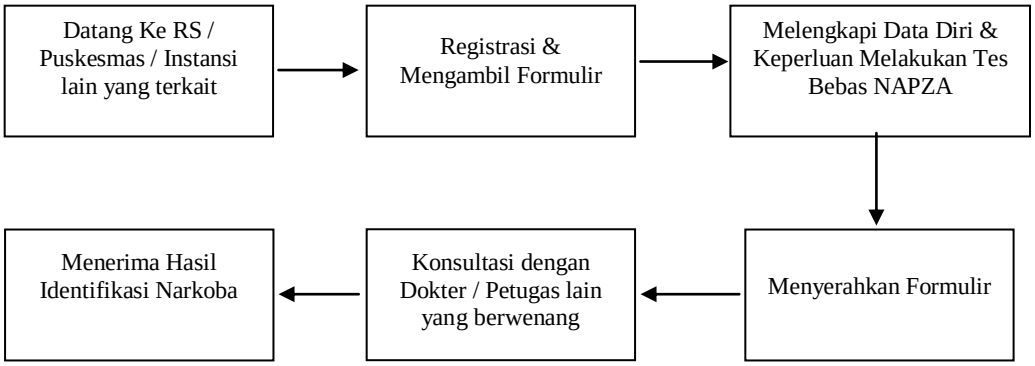


Gambar 3.1 Alur Pemeriksaan Tes Urin untuk Deteksi Narkoba

Tes narkoba di laboratorium relatif tidak butuh waktu lama. Beberapa menit di Klinik Umum untuk registrasi, kemudian diambil sampel urin di Laboratorium. Setelah dua jam, hasil pemeriksaan narkoba sudah selesai. Akan tetapi, jika rumah sakit sedang menangani banyak pasien, maka dibutuhkan waktu yang lebih lama dari yang diperkirakan untuk mendapatkan hasil skrining narkoba tersebut. Birokrasi juga menjadi salah satu faktor yang menyebabkan banyak pasien malas melakukan tes urin di laboratorium. Sehingga menyulitkan pemerintah untuk memantau kondisi dan menyarankan terapi rehabilitasi bagi pasien yang teridentifikasi narkoba.

3.1.2 Analisis Sistem Usulan

Sistem usulan yang dibangun dalam tugas akhir ini adalah sistem yang dapat mengidentifikasi jenis narkoba yang telah disalahgunakan berdasarkan efek samping yang diderita melalui sesi pertanyaan. Tujuan lain dari sistem ini adalah untuk mengetahui performansi *JST Backpropagation* dalam pengenalan efek samping dari masing-masing jenis narkoba. Berikut ini adalah alur pendeteksian jenis narkoba dengan menggunakan sistem usulan :



Gambar 3.2 Alur Pendeteksian Jenis Narkoba dengan Sistem Usulan

Masukan awal sistem usulan berupa representasi gejala atau efek psikis dan klinis yang ditimbulkan akibat penyalahgunaan narkoba. Sementara keluaran akhir sistem berupa representasi teridentifikasi menggunakan narkoba jenis Narkotika, Psikotropika ataupun Zat Adiktif lainnya, serta nilai akurasi algoritma *JST Backpropagation.*

1. Analisis Input

Data yang diolah adalah berasal dari data sampling yang berisi seluruh jenis narkoba dan turunannya beserta gejala yang terjadi pada setiap jenis narkoba seperti terlihat dalam tabel 2.1 Jenis Narkoba, Turunan dan Gejalanya pada bab 2. Mengacu pada tabel 2.1 Jenis Narkoba, Turunan dan Gejalanya, disusun menjadi variabel X_1 hingga X_{23} yang menjadi data masukan pada sistem aplikasi yang akan dirancang. Nilai diberikan pada setiap variabel, yaitu nilai 1 untuk penderita yang memiliki gejala tersebut dan nilai 0 untuk penderita yang tidak memiliki gejala. Seperti ditampilkan pada tabel 3.1 berikut ini.

Tabel 3.1 Variabel Input Gejala Efek Samping Penyalahguna Narkoba

VARIABEL INPUT	GEJALA EFEK SAMPING
X1	Sembelit
X2	Mual
X3	Hilang nafsu makan
X4	Keracunan
X5	Kerusakan Ginjal
X6	Kerusakan hati
X7	Kejang-kejang
X8	Paru-paru terganggu
X9	Suhu tubuh naik
X10	Nafas pendek
X11	Kecanduan
X12	Gangguan sex
X13	Daya ingat berkurang
X14	Gelisah
X15	Perasaan tertekan
X16	Perasaan sensitive
X17	Suka menyendiri
X18	Mudah marah
X19	Mudah lelah
X20	Mudah gugup
X21	Mental terganggu
X22	Halusinasi
X23	Paranoid

Terdapat batasan minimum gejala yang harus dipenuhi atau dipilih oleh pengguna sistem agar sistem usulan bisa mengenali pola dan mengidentifikasi jenis narkoba yang dikonsumsi oleh pasien. Minimum gejala yang bisa dikenali oleh sistem adalah 3 buah gejala berdasarkan referensi dari situs online untuk pengenalan pola identifikasi penyalahguna nikotin [10]. Jika pengguna memasukkan data input kurang dari 3 buah gejala maka sistem akan mengenali pola tersebut sebagai jenis narkoba tidak teridentifikasi.

2. Analisis Target Output

Target keluaran atau output yang dihasilkan sistem berupa representasi dari 3 jenis narkoba yaitu Narkotika, Psikotropika dan Zat Adiktif lainnya. Data sampling dijadikan sebagai bahan pelatihan untuk sistem yang dibangun, Sistem akan

mengidentifikasi jenis narkoba yang dikonsumsi pasien dengan target keluaran seperti berikut ini.

Tabel 3.2 Target output yang diharapkan

Jenis Narkoba	Output
Narkotika	0 0 1
Psikotropika	0 1 0
Zat Adiktif	1 0 0

3. Pemilihan Nilai Parameter JST

Penentuan nilai parameter bobot pada JST dipengaruhi pemilihan jumlah lapisan tersembunyi (*hidden layer*) dan nilai *learning rate* (laju pembelajaran) yang sesuai. Nilai *learning rate* yang tidak sesuai berdampak pada hasil klasifikasi yang kurang optimal. Apabila nilai *learning rate* terlalu besar, jaringan akan mencapai konvergen dalam waktu yang singkat tetapi *error* klasifikasi menjadi besar. Sebaliknya bila *learning rate* terlalu rendah, hasil klasifikasi memiliki akurasi yang baik tetapi proses *training* membutuhkan waktu yang lama.

Pada pembangunan jaringan *Backpropagation* yang akan digunakan dalam mendeteksi, hasil keputusan yang kurang tepat dapat diperbaiki dengan menggunakan nilai *learning rate* dan *hidden layer* secara *trial and error* untuk mendapatkan nilai bobot yang optimum agar MSE jaringan dapat diperbaiki.Langkah-langkah pemilihan jaringan yang optimum adalah sebagai berikut [2].

a. Proses pelatihan dilakukan terhadap data pelatihan dengan struktur jaringan yang memiliki bagian simpul tersembunyi berbeda akan diperoleh nilai *output* jaringan. Kemudian menghitung nilai MSE. Jaringan yang memiliki nilai MSE terendah dipilih sebagai jaringan yang optimum dan digunakan untuk pendeteksian.
b. Setelah proses pelatihan dilakukan proses pengujian dengan struktur jaringan yang memiliki bilangan simpul tersembunyi berbeda yang telah dilatih akan diperoleh nilai *output* jaringan. Nilai MSE dari masing-masing struktur jaringan dihitung. Proses pengujian digunakan untuk menguji prestasi pelatihan dan

sebagai pendukung bahwa jaringan terpilih sebagai jaringan yang tepat untuk sistem pendeteksi penyalahgunaan narkoba.

3.1.2.1 Analisis Kasus

Contoh kasus menggunakan data pasien yang menderita gejala sembelit (X1), hilang nafsu makan (X3), kejang-kejang (X7), kecanduan (X11), daya ingat berkurang (X13), gelisah (X14), mudah gugup (X20) serta paranoid (X23) sebagaimana terlihat pada tabel 3.3 berikut ini.

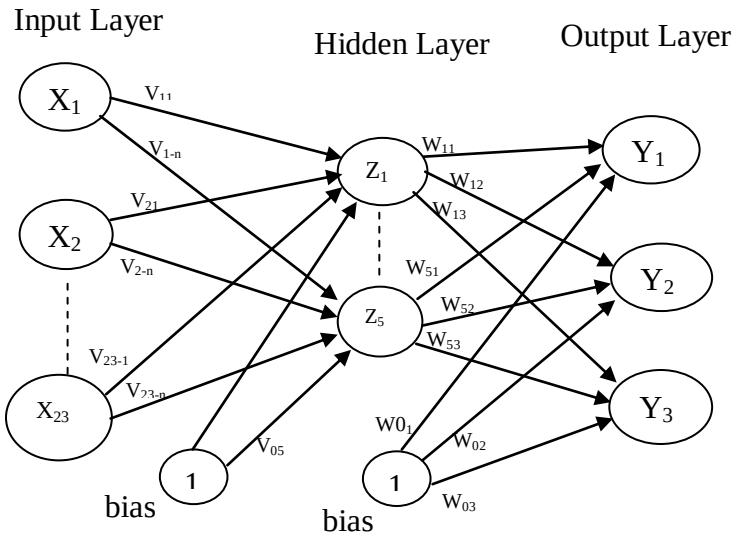
Tabel 3.3 Data Pasien dengan Gejala Efek Samping Penyalahgunaan Narkoba

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
1	0	0	1	0	0	1	0	0	0	1	0

X13	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23
1	1	0	0	0	0	0	1	0	0	1

Arsitektur Jaringan Syaraf Tiruan menggunakan metode *Backpropagation* (Gambar 3.3) terdiri dari:

1. Lapisan Input (X_i) dengan 23 neuron,
2. Lapisan Tersembunyi (Z_i) dengan 5 neuron, fungsi aktivasi yang digunakan pada setiap neuron pada lapisan ini adalah fungsi sigmoid ($z = \frac{1}{1+e^{-z_in}}$)
3. Lapisan Output (Y_i) dengan 3 neuron, fungsi aktivasi yang digunakan pada neuron di lapisan ini adalah fungsi sigmoid ($y = \frac{1}{1+e^{-y_in}}$)



Gambar 3.3 Arsitektur Jaringan Syaraf Tiruan metode Backpropagation

Keterangan :

X_i : Lapisan Input terdiri dari 23 *neuron*

V_{ij} : Bobot pada lapisan tersembunyi.

Z_j : Lapisan Tersembunyi terdiri dari 5 *neuron*

W_{ij} : Bobot pada lapisan keluaran.

V_{0j} : Bias pada lapisan tersembunyi

W_{0j} : Bias pada lapisan keluaran.

i,j,k : 1,2,3,…..,n.

n : Jumlah *neuron* dalam suatu lapisan.

Y_k : Lapisan Output terdiri dari 3 *neuron*

1 : Konstanta bias.

Bobot awal yang menghubungkan neuron-neuron pada lapisan input dan lapisan tersembunyi (V_{11}, $V_{1\text{-}n}$,V_{21}, $V_{2\text{-}n}$, …, $V_{23\text{-}n}$) dan bobot biasV_{01} dan V_{0n} dipilih secara acak. Demikian pula bobot awal yang menghubungkan neuron-neuron pada lapisan tersembunyi dan lapisan output (W_{11},W_{12},W_{13} , …, $W_{n\text{-}3}$) dan bobot bias W_0 juga dipilih secara acak.

Pada inisialisasi ditetapkan :

X = [1 0 0 1 0 0 1 0 0 0 1 0 1 1 0 0 0 0 0 1 0 0 1]

T = [0 0 1]

Learning Rate (α) = 0.1

Maksimum Epoch = 1000

Target Error = 0.01

Berikut ini perhitungan pelatihan menggunakan metode *Backpropagation:*

1. Inisialisasi bobot (ambil nilai random yang cukup kecil)

Bobot Awal Input ke *hidden layer* (V_{ij}) :

V_{11}=0.07855146,	V_{12}=0.733116,	V_{13}=0.788494,	V_{14} =0.329598,	V_{15}=0.654853
V_{21}=0.996165254,	V_{22} =0.340305,	V_{23}=0.035597,	V_{24}=0.85708,	V_{25}=0.585026
V_{31}= 0.852026034,	V_{32}=0.304204,	V_{33}= 0.46611,	V_{34}=0.109031,	V_{35}=0.147535
V_{41}= 0.948602419,	V_{42}= 0.95981,	V_{43}=0.640323,	V_{44}=0.711407	V_{45}=0.113696
V_{51}=0.712427873,	V_{52}=0.908327,	V_{53}=0.446046,	V_{54}=0.81902,	V_{55}=0.184678
V_{61}=0.294098899,	V_{62}=0.133795,	V_{63}=0.766229,	V_{64}=0.249095,	V_{65}=0.645821
V_{71}=0.908358139,	V_{72}=0.495677,	V_{73}=0.861076,	V_{74}=0.03086,	V_{75}=0.346567
V_{81}=0.189645965,	V_{82}=0.098198,	V_{83}=0.076191,	V_{84}=0.549092,	V_{85}=0.619027
V_{91}=0.872471354,	V_{92}=0.397678,	V_{93}=0.68711,	V_{94}=0.895432,	V_{95}=0.863345
$V_{10\text{-}1}$=0.4200949,	$V_{10\text{-}2}$=0.80614,	$V_{10\text{-}3}$=0.23028,	$V_{10\text{-}4}$=0.903489,	$V_{10\text{-}5}$=0.288819
$V_{11\text{-}1}$=0.832536648,	$V_{11\text{-}2}$=0.842948,	$V_{11\text{-}3}$=0.043922,	$V_{11\text{-}4}$=0.52974,	$V_{11\text{-}5}$=0.548577
$V_{12\text{-}1}$=0.350303442,	$V_{12\text{-}2}$=0.057243,	$V_{12\text{-}3}$=0.251268,	$V_{12\text{-}4}$=0.864594,	$V_{12\text{-}5}$=0.342873
$V_{13\text{-}1}$=0.018540204,	$V_{13\text{-}2}$=0.823601,	$V_{13\text{-}3}$=0.943009,	$V_{13\text{-}4}$=0.691028,	$V_{13\text{-}5}$=0.489989
$V_{14\text{-}1}$=0.382234413,	$V_{14\text{-}2}$=0.801777,	$V_{14\text{-}3}$=0.712296,	$V_{14\text{-}4}$=0.114852,	$V_{14\text{-}5}$=0.22946
$V_{15\text{-}1}$=0.126828286,	$V_{15\text{-}2}$=0.740072,	$V_{15\text{-}3}$=0.035502,	$V_{15\text{-}4}$=0.316769,	$V_{15\text{-}5}$=0.788995
$V_{16\text{-}1}$=0.255956097,	$V_{16\text{-}2}$=0.164026,	$V_{16\text{-}3}$=0.664831,	$V_{16\text{-}4}$=0.962364,	$V_{16\text{-}5}$=0.570106
$V_{17\text{-}1}$=0.095142454,	$V_{17\text{-}2}$=0.341009,	$V_{17\text{-}3}$=0.337385,	$V_{17\text{-}4}$=0.2337,	$V_{17\text{-}5}$=0.818475
$V_{18\text{-}1}$=0.608720185,	$V_{18\text{-}2}$=0.961752,	$V_{18\text{-}3}$=0.002232,	$V_{18\text{-}4}$=0.293125,	$V_{18\text{-}5}$=0.974924
$V_{19\text{-}1}$=0.562938948,	$V_{19\text{-}2}$=0.4277,	$V_{19\text{-}3}$=0.935351,	$V_{19j\text{-}4}$=0.828192,	$V_{19\text{-}5}$=0.555919
$V_{20\text{-}1}$=0.55451598,	$V_{20\text{-}2}$=0.868317,	$V_{20\text{-}3}$=0.607422,	$V_{20\text{-}4}$=0.27456,	$V_{20\text{-}5}$=0.495339
$V_{21\text{-}1}$=0.64520073,	$V_{21\text{-}2}$=0.11445,	$V_{21\text{-}3}$=0.549599,	$V_{21\text{-}4}$=0.443523,	$V_{21\text{-}5}$=0.757388
$V_{22\text{-}1}$=0.978990505,	$V_{22\text{-}2}$=0.633211,	$V_{22\text{-}3}$=0.755175,	$V_{22\text{-}4}$=0.198924,	$V_{22\text{-}5}$=0.330765
$V_{23\text{-}1}$=0.266969657,	$V_{23\text{-}2}$=0.66791,	$V_{23\text{-}3}$=0.358053,	$V_{23\text{-}4}$=0.042211,	$V_{23\text{-}5}$=0.626847

Bobot Awal Bias ke *hidden layer* (V_{0j}) :

V_{01}=0.114475484, V_{02} =0.868573, V_{03} =0.372911, V_{04} =0.758166, V_{05} =0.243426

Bobot awal *hidden layer* ke *output layer* (W_{jk}) :

W_{11} =0.007574319, W_{12} =0.456568, W_{13} =0.171757
W_{21} =0.167960543, W_{22} =0.534814, W_{23} =0.813095
W_{31} =0.260512707, W_{32} =0.386309, W_{33} =0.824383
W_{41} =0.59888563, W_{42} =0.029111, W_{43} =0.363525
W_{51} =0.435511666, W_{52} =0.626213, W_{53}=0.362942

Bobot awal bias ke *output layer* (W_{0j}) :

W_{01} =0.936221323, W_{02} =0.869177, W_{03} =0.113375

2. Selama kondisi berhenti bernilai salah, kerjakan:

Tahap Perambatan Maju (*Forward Propagation*)

a. Operasi pada *hidden layer* dengan persamaan (2.2)

$$Z_in_1 = V_{o1} + \sum_{i=1}^{23} x_i v_{i1} = 4.1047844$$
$$Z_in_2 = V_{o2} + \sum_{i=1}^{23} x_i v_{i2} = 7.06173001$$
$$Z_in_3 = V_{o3} + \sum_{i=1}^{23} x_i v_{i3} = 5.32750622$$
$$Z_in_4 = V_{o4} + \sum_{i=1}^{23} x_i v_{i4} = 3.48242073$$
$$Z_in_5 = V_{o5} + \sum_{i=1}^{23} x_i v_{i5} = 3.6329882$$

b. Fungsi aktivasi *sigmoid biner* pada *hidden layer* dengan persamaan (2.3)

$$Z_1 = \frac{1}{1+e^{-z_in1}} = 0.98377405$$
$$Z_2 = \frac{1}{1+e^{-z_in2}} = 0.99914344$$
$$Z_3 = \frac{1}{1+e^{-z_in3}} = 0.9951673$$
$$Z_4 = \frac{1}{1+e^{-z_in4}} = 0.97018343$$
$$Z_5 = \frac{1}{1+e^{-z_in5}} = 0.97424385$$

c. Operasi pada *output layer* dengan persamaan (2.4)

$Y_in_1 = W_{o1} + \sum_{i=1}^{5} Z_i W_{jk} =$ 2.37606662

$Y_in_2 = W_{o2} + \sum_{i=1}^{5} Z_i W_{jk} =$ 2.87546139

$Y_in_3 = W_{o3} + \sum_{i=1}^{5} Z_i W_{jk} =$ 2.62142384

d. Fungsi aktivasi *sigmoid biner* pada *output layer* dengan persamaan (2.5)

$Y_1 = \frac{1}{1+e^{-y_in1}} =$ 0.91498396

$Y_2 = \frac{1}{1+e^{-y_in2}} =$ 0.94661999

$Y_3 = \frac{1}{1+e^{-y_in3}} =$ 0.93222772

e. Cek *error* (iterasi berhenti bila |error|<0.01)

$error_1 =$ -0.914984

jumlah kuadrat $error_1 =$ 0.837196

$error_2 =$ -0.94662

jumlah kuadrat $error_2 =$ 0.896089

$error_3 =$ - 0.9322277

jumlah kuadrat $error_3 =$ 0.869049

Tahap Perambatan Balik (*Backpropagation*)

a. Setiap unit output menerima pola target yang sesuai dengan pola input pelatihan, kemudian hitung *error* dengan dengan persamaan (2.6)

$\delta_1 = (T_1 - y) * (\frac{1}{1+e^{-y_in1}}) * [1 - (\frac{1}{1+e^{-y_in1}})] =$ -0.0711751

$\delta_2 = (T_2 - y) * (\frac{1}{1+e^{-y_in2}}) * [1 - (\frac{1}{1+e^{-y_in2}})] =$ -0.0478333

$\delta_3 = (T_3 - y) * (\frac{1}{1+e^{-y_in3}}) * [1 - (\frac{1}{1+e^{-y_in3}})] =$ 0.0042818

b. Hitung koreksi bobot dengan persamaan (2.8)

$\Delta W_{11} = \alpha * \delta_1 * Z_1 =$ -0.00700202
$\Delta W_{12} = \alpha * \delta_2 * Z_1 =$ -0.00470571
$\Delta W_{13} = \alpha * \delta_3 * Z_1 =$ 0.000421232
$\Delta W_{21} = \alpha * \delta_1 * Z_2 =$ -0.00711141
$\Delta W_{22} = \alpha * \delta_2 * Z_2 =$ -0.00477923
$\Delta W_{23} = \alpha * \delta_3 * Z_2 =$ 0.000427813
$\Delta W_{31} = \alpha * \delta_1 * Z_3 =$ -0.00708311
$\Delta W_{32} = \alpha * \delta_2 * Z_3 =$ -0.00476021
$\Delta W_{33} = \alpha * \delta_3 * Z_3 =$ 0.000426111
$\Delta W_{41} = \alpha * \delta_1 * Z_4 =$ -0.00690529
$\Delta W_{42} = \alpha * \delta_2 * Z_4 =$ -0.0046407
$\Delta W_{43} = \alpha * \delta_3 * Z_4 =$ 0.000415413
$\Delta W_{51} = \alpha * \delta_1 * Z_5 =$ -0.00693419
$\Delta W_{52} = \alpha * \delta_2 * Z_5 =$ -0.00466013
$\Delta W_{53} = \alpha * \delta_3 * Z_5 =$ 0.000417152

$\Delta W_{01} = \alpha * \delta_1 =$ -0.00711751
$\Delta W_{02} = \alpha * \delta_2 =$ -0.00478
$\Delta W_{03} = \alpha * \delta_3 =$ 0.000428

c. Setiap unit tersembunyi menjumlahkan delta input-nya dengan persamaan (2.10)

$\delta_{in1} = \sum_{k=1}^{3} \delta_k W_{1k} =$ -0.02164

$\delta_{in2} = \sum_{k=1}^{3} \delta_k W_{2k} =$ -0.03405

$\delta_{in3} = \sum_{k=1}^{3} \delta_k W_{3k} =$ -0.03349

$\delta_{in4} = \sum_{k=1}^{3} \delta_k W_{4k} =$ -0.04246

$\delta_{in5} = \sum_{k=1}^{3} \delta_k W_{5k} =$ -0.0594

d. Hitung informasi *error* dengan persamaan (2.11)

$$\delta_1 = \delta_{in1} * (\frac{1}{1+e^{-z_in1}}) * [1 - (\frac{1}{1+e^{-z_in1}})] = \text{-0.00428704}$$

$$\delta_2 = \delta_{in2} * (\frac{1}{1+e^{-z_in2}}) * [1 - (\frac{1}{1+e^{-z_in2}})] = \text{-0.00669827}$$

$$\delta_3 = \delta_{in3} * (\frac{1}{1+e^{-z_in3}}) * [1 - (\frac{1}{1+e^{-z_in3}})] = \text{-0.00659934}$$

$$\delta_4 = \delta_{in4} * (\frac{1}{1+e^{-z_in4}}) * [1 - (\frac{1}{1+e^{-z_in4}})] = \text{-0.0084628}$$

$$\delta_5 = \delta_{in5} * (\frac{1}{1+e^{-z_in5}}) * [1 - (\frac{1}{1+e^{-z_in5}})] = \text{-0.0118165}$$

e. Hitung koreksi bobot dengan dengan persamaan (2.12)

$\Delta V_{jk} = \alpha * \delta_j x_i$

ΔV_{11} =-0.00043, ΔV_{12} =-0.00067, ΔV_{13} =-0.00066, ΔV_{14} =-0.00085, ΔV_{15} =-0.00118165
ΔV_{21} =0, ΔV_{22} =0, ΔV_{23} =0, ΔV_{24} =0, ΔV_{25} =0
ΔV_{31} =0, ΔV_{32} =0, ΔV_{33} =0, ΔV_{34} =0, ΔV_{35} =0
ΔV_{41} =-0.00043, ΔV_{42} =-0.00067 ΔV_{43} =-0.00066, ΔV_{44} =-0.00085, ΔV_{45} =-0.00118165
ΔV_{51} =0, ΔV_{52} =,0 ΔV_{53} =0, ΔV_{54} =0, ΔV_{55} =0
ΔV_{61} =0, ΔV_{62} =0, ΔV_{63} =0, ΔV_{64} =0, ΔV_{65} =0
ΔV_{71} =-0.00043, ΔV_{72} =-0.00067, ΔV_{73} =-0.00066, ΔV_{74} =-0.00085, ΔV_{75} =-0.00118165
ΔV_{81} =0, ΔV_{82} =0, ΔV_{83} =0, ΔV_{84} =0, ΔV_{85} =0
ΔV_{91} =0, ΔV_{92} =0, ΔV_{93} =0, ΔV_{94} =0, ΔV_{95} =0
ΔV_{10-1} =0, ΔV_{10-2} =0, ΔV_{10-3} =0, ΔV_{10-4} =0, ΔV_{10-5} =0
ΔV_{11-1} =-0.00043, ΔV_{11-2} =-0.00067, ΔV_{11-3} =-0.00066, ΔV_{11-4} =-0.00085, ΔV_{11-5} =-0.00118165
ΔV_{12-1} =0, ΔV_{12-2} =0, ΔV_{12-3} =0, ΔV_{12-4} =0, ΔV_{12-5} =0
ΔV_{13-1} =-0.00043, ΔV_{13-2} =-0.00067, ΔV_{13-3} =-0.00066, ΔV_{13-4} =-0.00085, ΔV_{13-5} =-0.00118165
ΔV_{14-1} =-0.00043, ΔV_{14-2} =-0.00067, ΔV_{14-3} =-0.00066, ΔV_{14-4} =-0.00085, ΔV_{14-5} =-0.00118165
ΔV_{15-1} =0, ΔV_{15-2} =0, ΔV_{15-3} =0, ΔV_{15-4} =0, ΔV_{15-5} =0
ΔV_{16-1} =0, ΔV_{16-2} =0, ΔV_{16-3} =0, ΔV_{16-4} =0, ΔV_{16-5} =0
ΔV_{17-1} =0, ΔV_{17-2} =0, ΔV_{17-3} =0, ΔV_{17-4} =0, ΔV_{17-5} =0
ΔV_{18-1} =0, ΔV_{18-2} =0, ΔV_{18-3} =0, ΔV_{18-4} =0, ΔV_{18-5} =0
ΔV_{19-1} =0, ΔV_{19-2} =0, ΔV_{19-3} =0, ΔV_{19-4} =0, ΔV_{19-5} =0
ΔV_{20-1} =-0.00043, ΔV_{20-2} =-0.00067, ΔV_{20-3} =-0.00066, ΔV_{20-4} =-0.00085, ΔV_{20-5} =-0.00118165
ΔV_{21-1} =0, ΔV_{21-2} =0, ΔV_{21-3} =0, ΔV_{21-4} =0, ΔV_{21-5} =0
ΔV_{22-1} =0, ΔV_{22-2} =0, ΔV_{22-3} =0, ΔV_{22-4} =0, ΔV_{22-5} =0
ΔV_{23-1} =-0.00043, ΔV_{23-2} =-0.00067, ΔV_{23-3} =-0.00066, ΔV_{23-4} =-0.00085, ΔV_{23-5} =-0.00118165

f. Hitung koreksi bias dengan dengan persamaan (2.13)

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

ΔV_{01} =-0.00043, ΔV_{02} =-0.00067, ΔV_{03} =-0.00066, ΔV_{04} =-0.00085, ΔV_{05} =-0.00118

g. Hitung perubahan bobot dan bias dengan persamaan (2.14) dan (2.15)

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

$V_{11(baru)}$=0.078123, $V_{12(baru)}$=0.732447, $V_{13(baru)}$=0.787834, $V_{14(baru)}$=0.328751, $V_{15(baru)}$=0.65367113
$V_{21(baru)}$=0.996165, $V_{22(baru)}$=0.340305, $V_{23(baru)}$=0.035597, $V_{24(baru)}$=0.85708, $V_{25(baru)}$=0.585026488
$V_{31(baru)}$=0.852026, $V_{32(baru)}$=0.304204, $V_{33(baru)}$=0.46611, $V_{34(baru)}$=0.109031, $V_{35(baru)}$=0.147535182
$V_{41(baru)}$=0.948174, $V_{42(baru)}$=0.95914, $V_{43(baru)}$=0.639663, $V_{44(baru)}$=0.71056, $V_{45(baru)}$=0.112514186
$V_{51(baru)}$=0.712428, $V_{52(baru)}$=0.908327, $V_{53(baru)}$=0.446046, $V_{54(baru)}$=0.81902, $V_{55(baru)}$=0.184677847
$V_{61(baru)}$=0.294099, $V_{62(baru)}$=0.133795, $V_{63(baru)}$=0.766229, $V_{64(baru)}$=0.249095, $V_{65(baru)}$=0.645820891
$V_{71(baru)}$=0.907929, $V_{72(baru)}$=0.495007, $V_{73(baru)}$=0.860416, $V_{74(baru)}$=0.030013, $V_{75(baru)}$=0.345384884
$V_{81(baru)}$=0.189646, $V_{82(baru)}$=0.098198, $V_{83(baru)}$=0.076191, $V_{84(baru)}$=0.549092, $V_{85(baru)}$=0.61902733
$V_{91(baru)}$=0.872471, $V_{92(baru)}$=0.397678, $V_{93(baru)}$=0.68711, $V_{94(baru)}$=0.895432, $V_{95(baru)}$=0.8633448
$V_{10\text{-}1(baru)}$=0.420095, $V_{10\text{-}2(baru)}$=0.80614, $V_{10\text{-}3(baru)}$=0.23028, $V_{10\text{-}4(baru)}$=0.903489, $V_{10\text{-}5(baru)}$=0.28881874
$V_{11\text{-}1(baru)}$=0.832108, $V_{11\text{-}2(baru)}$=0.842278, $V_{11\text{-}3(baru)}$=0.043262, $V_{11\text{-}4(baru)}$=0.528893, $V_{11\text{-}5(baru)}$=0.547395064
$V_{12\text{-}1(baru)}$=0.350303, $V_{12\text{-}2(baru)}$=0.057243, $V_{12\text{-}3(baru)}$=0.251268, $V_{12\text{-}4(baru)}$=0.864594, $V_{12\text{-}5(baru)}$=0.342872919
$V_{13\text{-}1(baru)}$=0.018112, $V_{13\text{-}2(baru)}$=0.822931, $V_{13\text{-}3(baru)}$=0.942349, $V_{13\text{-}4(baru)}$=0.690181, $V_{13\text{-}5(baru)}$=0.488807324
$V_{14\text{-}1(baru)}$=0.381806, $V_{14\text{-}2(baru)}$=0.801108, $V_{14\text{-}3(baru)}$=0.711636, $V_{14\text{-}4(baru)}$=0.114006, $V_{14\text{-}5(baru)}$=0.228278128
$V_{15\text{-}1(baru)}$=0.126828, $V_{15\text{-}2(baru)}$=0.740072, $V_{15\text{-}3(baru)}$=0.035502, $V_{15\text{-}4(baru)}$=0.316769, $V_{15\text{-}5(baru)}$=0.788995277
$V_{16\text{-}1(baru)}$=0.255956, $V_{16\text{-}2(baru)}$=0.164026, $V_{16\text{-}3(baru)}$=0.664831, $V_{16\text{-}4(baru)}$=0.962364, $V_{16\text{-}5(baru)}$=0.570105693
$V_{17\text{-}1(baru)}$=0.095142, $V_{17\text{-}2(baru)}$=0.341009, $V_{17\text{-}3(baru)}$=0.337385, $V_{17\text{-}4(baru)}$=0.2337, $V_{17\text{-}5(baru)}$=0.818475325
$V_{18\text{-}1(baru)}$=0.60872, $V_{18\text{-}2(baru)}$=0.961752, $V_{18\text{-}3(baru)}$=0.002232, $V_{18\text{-}4(baru)}$=0.293125, $V_{18\text{-}5(baru)}$=0.974924017
$V_{19\text{-}1(baru)}$=0.562939, $V_{19\text{-}2(baru)}$=0.4277, $V_{19\text{-}3(baru)}$=0.935351, $V_{19\text{-}4(baru)}$=0.828192, $V_{19\text{-}5(baru)}$=0.555919137
$V_{20\text{-}1(baru)}$=0.554087, $V_{20\text{-}2(baru)}$=0.867647, $V_{20\text{-}3(baru)}$=0.606762, $V_{20\text{-}4(baru)}$=0.273714, $V_{20\text{-}5(baru)}$=0.494157786
$V_{21\text{-}1(baru)}$=0.645201, $V_{21\text{-}2(baru)}$=0.11445, $V_{21\text{-}3(baru)}$=0.549599, $V_{21\text{-}4(baru)}$=0.443523, $V_{21\text{-}5(baru)}$=0.757387678
$V_{22\text{-}1(baru)}$=0.978991, $V_{22\text{-}2(baru)}$=0.633211, $V_{22\text{-}3(baru)}$=0.755175, $V_{22\text{-}4(baru)}$=0.198924, $V_{22\text{-}5(baru)}$=0.330764566
$V_{23\text{-}1(baru)}$=0.266541, $V_{23\text{-}2(baru)}$=0.66724, $V_{23\text{-}3(baru)}$=0.357393, $V_{23\text{-}4(baru)}$=0.041365, $V_{23\text{-}5(baru)}$=0.625664872

$V_{0j}(\text{baru}) = V_{0j}(\text{lama}) + \Delta V_{0j}$

$V_{01(baru)}$=0.114047, $V_{02(baru)}$=0.867903, $V_{03(baru)}$=0.372251, $V_{04\ (baru)}$=0.75732, $V_{05\ (baru)}$=0.242244

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

$W_{11(baru)}$=0.000572, $W_{12(baru)}$=0.451862398, $W_{11(baru)}$=0.172178

$W_{21(baru)}$=0.160849, $W_{22(baru)}$=0.530034967, $W_{23(baru)}$=0.813523

$W_{31(baru)}$=0.25343, $W_{32(baru)}$=0.381548806, $W_{33(baru)}$=0.82481

$W_{41(baru)}$=0.59198, $W_{42(baru)}$=0.024469798, $W_{43(baru)}$=0.36394

$W_{51(baru)}$=0.428577, $W_{52(baru)}$=0.621552736, $W_{53(baru)}$=0.363359

$$W_{0j}(\text{baru}) = W_{0j}(\text{lama}) + \Delta W_{0j}$$

$W_{01(baru)}$=0.929104, $W_{02(baru)}$=0.864393455, $W_{03(baru)}$=0.113803

Pada data kedua, juga dilakukan operasi-operasi yang sama dengan menggunakan bobot-bobot akhir hasil pengolahan data pertama sebagai bobot awalnya. Proses ini dilakukan secara berulang sampai pada maksimum epoch (1000) atau kuadrat error < target error (0.01). Misal pada akhir pelatihan *Backpropagation*, diperoleh :

Bobot akhir input ke *hidden layer* (V_{ij}) :

V_{11}= 0.88444, V_{12}=-0.22605, V_{13}=0.47554, V_{14}=1.6429, V_{15}=1.536

V_{21}= 0.18166, V_{22}=0.68351, V_{23}=0.29806, V_{24}=1.4187, V_{25}=1.1442

V_{31}=-0.25971, V_{32}=1.9308, V_{33}=1.0766, V_{34}=0.14171, V_{35}=0.080294

V_{41}=1.649, V_{42}=0.5834, V_{43}=0.61843, V_{44}=-1.6112, V_{45}=-1.8788

V_{51}=-1.9319, V_{52}=2.406, V_{53}=2.0674, V_{54}=-1.4113, V_{55}=-2.2298

V_{61}=1.914, V_{62}=-0.04316, V_{63}=0.87491, V_{64}=-1.5868, V_{65}=-1.5542

V_{71}=-1.8938, V_{72}=1.4196, V_{73}=0.87357, V_{74}=1.7042, V_{75}=0.6312

V_{81}=0.73953, V_{82}=0.51322, V_{83}=0.08222, V_{84}=1.3861, V_{85}=1.258

V_{91}=-1.0797, V_{92}=1.6791, V_{93}=1.3922, V_{94}=0.26126, V_{95}=-0.28927

$V_{10\text{-}1}$=-0.87211, $V_{10\text{-}2}$=1.5974, $V_{10\text{-}3}$=0.76715, $V_{10\text{-}4}$=0.70155, $V_{10\text{-}5}$=0.31055

$V_{11\text{-}1}$=1.5447, $V_{11\text{-}2}$=-0.1262, $V_{11\text{-}3}$=0.75466, $V_{11\text{-}4}$=-0.94324, $V_{11\text{-}5}$=-0.87843

$V_{12\text{-}1}$=0.94294, $V_{12\text{-}2}$=0.62101, $V_{12\text{-}3}$=-0.07511, $V_{12\text{-}4}$=1.2947, $V_{12\text{-}5}$=1.0268

$V_{13\text{-}1}$=-1.3224, $V_{13\text{-}2}$=0.88726, $V_{13\text{-}3}$=0.53987, $V_{13\text{-}4}$=1.6719, $V_{13\text{-}5}$=1.238

$V_{14\text{-}1}$=0.55984, $V_{14\text{-}2}$=-0.13256, $V_{14\text{-}3}$=0.21533, $V_{14\text{-}4}$=2.3959, $V_{14\text{-}5}$=2.1921

$V_{15\text{-}1}$=1.015, $V_{15\text{-}2}$=0.66759, $V_{15\text{-}3}$=0.005331, $V_{15\text{-}4}$=1.2415, $V_{15\text{-}5}$=0.83659

$V_{16\text{-}1}$=-1.1269, $V_{16\text{-}2}$=2.6004, $V_{16\text{-}3}$=1.5323, $V_{16\text{-}4}$=-0.79507, $V_{16\text{-}5}$=-1.3883

$V_{17\text{-}1}$=-2.9134, $V_{17\text{-}2}$=3.5874, $V_{17\text{-}3}$=2.3692, $V_{17\text{-}4}$=-0.84018, $V_{17\text{-}5}$=-1.5826

$V_{18\text{-}1}$=2.3835, $V_{18\text{-}2}$=0.10961, $V_{18\text{-}3}$=0.71249, $V_{18\text{-}4}$=-1.7729, $V_{18\text{-}5}$=-1.844

$V_{19\text{-}1}$=2.9582, $V_{19\text{-}2}$=-0.30206, $V_{19\text{-}3}$=-0.12377, $V_{19\text{-}4}$=-0.16696, $V_{19\text{-}5}$=0.098903

$V_{20\text{-}1}$=1.9509, $V_{20\text{-}2}$=0.090264, $V_{20\text{-}3}$=0.4515, $V_{20\text{-}4}$=-1.2712, $V_{20\text{-}5}$=-0.7519
$V_{21\text{-}1}$=-0.02677, $V_{21\text{-}2}$=0.39201, $V_{21\text{-}3}$=0.73494, $V_{21\text{-}4}$=0.45553, $V_{21\text{-}5}$=1.1426
$V_{22\text{-}1}$=-2.9379, $V_{22\text{-}2}$=2.6578, $V_{22\text{-}3}$=2.6986, $V_{22\text{-}4}$=0.038712, $V_{22\text{-}5}$=-0.92192
$V_{23\text{-}1}$=0.31494, $V_{23\text{-}2}$=0.41028, $V_{23\text{-}3}$=-0.14382, $V_{23\text{-}4}$=1.5393, $V_{23\text{-}5}$=1.8257

Bobot akhir bias ke *hidden layer* (V_{0j}) :

V_{01}= 0.1, V_{02}= 0.2, V_{03}= 0.1, V_{04}= 0.3, V_{05}= 0.1

Bobot akhir *hidden layer* ke *output layer* (W_{jk}) :

W_{11} =4.572, W_{12} =-4.6913, W_{13} =-0.056739
W_{21} =-1.7893, W_{22} =1.5398, W_{23} =-1.6333
W_{31} =-0.95479, W_{32} =1.062, W_{33} =-1.7039
W_{41} =-2.5482, W_{42} =-2.326, W_{43} =3.1142
W_{51} =-2.0125, W_{52} =-2.6524, W_{53} =3.5165

Bobot awal akhir ke *output layer* (W_{0j}):

W_{01} =0.2, W_{02} =0.2, W_{03} =0.1

Setelah proses pelatihan selesai, *Backpropagation* dapat digunakan untuk pengujian. Berikut ini pengujian algoritma metode *Backpropagation.*

a. Operasi pada *Hidden Layer* dengan persamaan (2.2)

$$Z_in_1 = V_{o1} + \sum_{i=1}^{23} x_i v_{i1} = 4.58883$$
$$Z_in_2 = V_{o2} + \sum_{i=1}^{23} x_i v_{i2} = 3.24078$$
$$Z_in_3 = V_{o3} + \sum_{i=1}^{23} x_i v_{i3} = 5.012386$$
$$Z_in_4 = V_{o4} + \sum_{i=1}^{23} x_i v_{i4} = 3.818425$$
$$Z_in_5 = V_{o5} + \sum_{i=1}^{23} x_i v_{i5} = 4.051489$$

b. Fungsi aktivasi *sigmoid biner* pada *hidden layer* dengan persamaan (2.3)

$$Z_1 = \frac{1}{1+e^{-z_in1}} = 0.98993754$$
$$Z_2 = \frac{1}{1+e^{-z_in2}} = 0.96234039$$
$$Z_3 = \frac{1}{1+e^{-z_in3}} = 0.99338899$$
$$Z_4 = \frac{1}{1+e^{-z_in4}} = 0.97850962$$
$$Z_5 = \frac{1}{1+e^{-z_in5}} = 0.98290101$$

c. Operasi pada *output layer* dengan persamaan (2.4)

$$Y_in_1 = W_{o1} + \sum_{i=1}^{5} Z_i W_{jk} = \text{-2.5439129}$$
$$Y_in_2 = W_{o2} + \sum_{i=1}^{5} Z_i W_{jk} = \text{-2.518422}$$
$$Y_in_3 = W_{o3} + \sum_{i=1}^{5} Z_i W_{jk} = 3.09228314$$

d. Fungsi aktivasi *sigmoid biner* pada *output layer* dengan persamaan (2.5)

$$Y_1 = \frac{1}{1+e^{-y_in1}} = 0.07283649$$
$$Y_2 = \frac{1}{1+e^{-y_in2}} = 0.07457678$$
$$Y_3 = \frac{1}{1+e^{-y_in3}} = 0.95657331$$

Kemudian hasil dari fungsi aktivasi dibandingkan dengan nilai ambang tertentu. Misalkan diambil nilai ambang (*threshold*) = 0.5, artinya jika nilai $y_k \geq 0.5$ maka output yang diberikan adalah 1, namun jika nilai $y_k < 0.5$ output yang diberikan adalah 0. Dengan demikian hasil output dari x = [1 0 0 1 0 0 1 0 0 0 1 0 1 1 0 0 0 0 0 1 0 0 1] adalah y = [0 0 1] (karena y_1 = 0.07283649<0.5 , y_2 = 0.07457678<0.5 , y_3 = 0.95657331>0.5). Sesuai dengan target yang diharapkan yaitu [0 0 1].

3.1.3 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah langkah dimana seorang pembangun perangkat lunak (*software developer*) menganalisis sumber daya yang akan digunakan dan menggunakan perangkat lunak yang dibangun. Perangkat keras dan perangkat lunak yang dimiliki harus sesuai dengan kebutuhan atau permintaan, sehingga dapat ditentukan kompabilitas aplikasi yang dibangun terhadap sumber daya yang ada.

Analisis kebutuhan non fungsional yang dilakukan dibagi dalam tiga tahap, yaitu:

1. Analisis perangkat keras (hardware).
2. Analisis perangkat lunak (software).
3. Analisis pengguna (user).

3.1.3.1 Analisis Kebutuhan Perangkat Keras

Perangkat keras yang digunakan untuk membangun sistem pendeteksi penyalahgunaan narkoba memiliki spesifikasi sebagai berikut:

1. Prosesor Intel Core 2 Duo 2.00 GHz.
2. Memori 1.00 GB DDR2.
3. Hardisk 300 GB.s
4. Monitor 14.1”

3.1.3.2 Analisis Kebutuhan Perangkat Lunak

Komponen perangkat lunak (*software*) pendukung dibutuhkan dalam membangun sebuah aplikasi. Berikut ini perangkat lunak yang digunakan saat ini:

1. Sistem operasi Windows XP Profesional.
2. Bahasa Pemrograman Java dengan Editor NetBeans IDE 6.9.1
3. CASE Tools menggunakan Rational Rose E.E
4. Basis Data menggunakan MySQL Database 5.1.30-community

3.1.3.3 Analisis Kebutuhan Pengguna

Analisis pengguna merupakan uraian mengenai siapa saja yang akan menggunakan sistem dan terlibat dalam pengolahan data beserta karakteristiknya sehingga dapat diketahui tingkat pengalaman dan pemahaman pengguna terhadap sistem.

Adapun user yang akan menggunakan sistem pendeteksi penyalahgunaan narkoba adalah sebagai berikut.

Tabel 3.4 Analisa Pengguna

Pengguna	Tanggung Jawab	Tingkat Pendidikan	Tingkat Keterampilan
Anggota Penyidik Satreskrim Narkoba	Mengelola data dan menjalankan proses identifikasi narkoba dengan sistem.	Minimal Bintara Kepolisian	Bisa menggunakan komputer, memiliki keahlian khusus di bidang narkoba.

3.1.4 Analisis Data/File

Analisis data adalah kegiatan menganalisis data yang akan diolah dan disimpan dalam database. Dalam analisis ini direpresentasikan darimana data berasal dan atribut dari data tersebut. Berikut adalah atribut dari entitas mseEpoch, tKategori, tNarkoba, dan dataPecandu.

Tabel 3.5 Atribut pada Entitas mseEpoch

No	Atribut	Keterangan
1	epoch	Lama iterasi untuk proses pelatihan jaringan
2	MSE	Nilai *Mean Square Error* untuk setiap iterasi kelipatan 100 hingga epoch ke-1000

Tabel 3.6 Atribut pada Entitas tKategori

No	Atribut	Keterangan
1	jenis	Jenis Narkoba (Narkotika, Psikotropika dan Zat Adiktif Lain)
2	kode	Kode untuk setiap jenis narkoba

Tabel 3.7 Atribut pada Entitas tNarkoba

No	Atribut	Keterangan
1	turunan	Turunan dari jenis narkoba seperti ganja yang merupakan turunan dari narkotika
2	pengertian	Pengertian tentang turunan narkoba tersebut
3	gejalaEfekSamping	Gejala dan efek samping yang diderita ketika mengkonsumsi turunan tersebut
4	gambar	Gambar dari turunan tesebut
5	keterangan	Jenis Narkoba

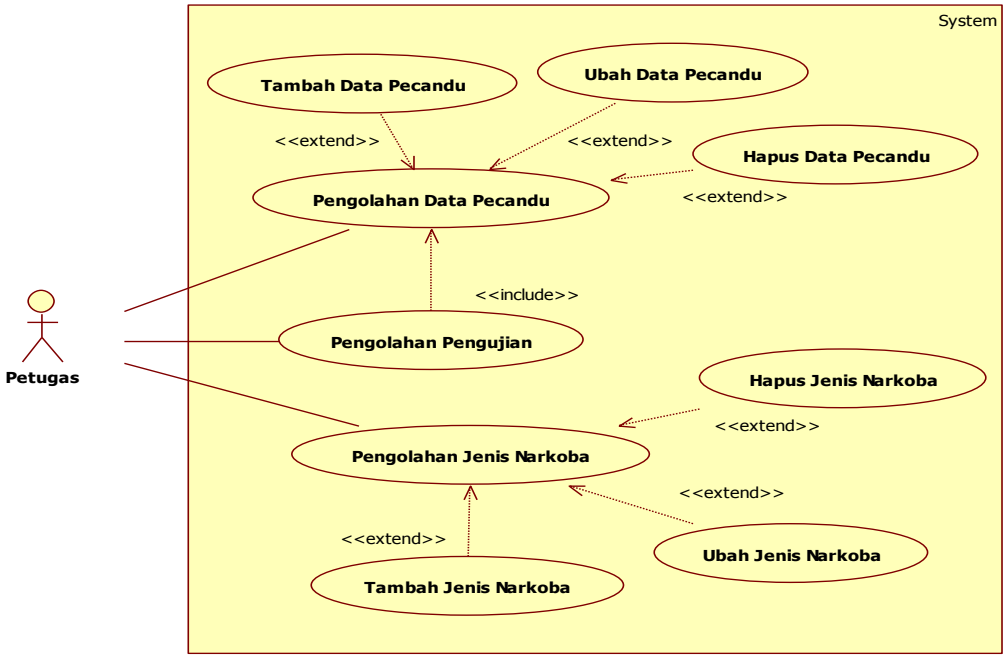
Tabel 3.8 Atribut pada Entitas DataPecandu

No	Atribut	Keterangan
1	idPasien	Nomor id pasien
2	Nama	Nama pasien
3	Usia	Usia pasien
4	x1	Variabel Input Gejala Efek Samping Penyalahgunaan Narkoba
5	x2	
6	x3	
7	x4	
8	x5	
9	x6	
10	x7	
11	x8	
12	x9	
13	x10	
14	x11	
15	x12	
16	x13	
17	x14	
18	x15	
19	x16	
20	x17	
21	x18	
22	x19	
23	x20	
24	x21	
25	x22	
26	x23	
27	Identifikasi	Hasil identifikasi jenis narkoba yang dikonsumsi

3.1.5 Analisis Kebutuhan Fungsional

3.1.5.1 Use Case Diagram

Use case merupakan gambaran umum dari perancangan sistem yang akan dibuat. Pada *use case* Sistem Pendeteksi Penyalahgunaan Narkoba hanya memiliki satu *user*.



Gambar 3.4 Use Case Diagram

1. Identifikasi *Use Case*

Identifikasi *Use Case* adalah *Use Case* yang bekerja pada sistem aplikasi dan *Use Case* yang bekerja pada sistem adalah sebagai berikut.

Tabel 3.9 Deskripsi Use Case

No	*Use Case*	Deskripsi
UC-01	Pengolahan Data Pecandu	Fungsionalitas untuk mengelola data pecandu.
UC-02	Pengolahan Pengujian	Fungsionalitas untuk mengidentifikasi pasien baru berdasarkan hasil pelatihan sebelumnya.
UC-03	Pengolahan Jenis Narkoba	Fungsionalitas untuk mengelola data jenis narkoba beserta turunan dan gejala efek samping yang dimiliki tiap jenisnya.
UC-04	Tambah Data Pecandu	Fungsionalitas untuk menambah data pecandu.
UC-05	Ubah Data Pecandu	Fungsionalitas untuk mengubah data pecandu yang ada.
UC-06	Hapus Data Pecandu	Fungsionalitas untuk menghapus data pecandu yang ada.
UC-07	Tambah Jenis Narkoba	Fungsionalitas untuk menambah data turunan narkoba dan gejala efek samping.
UC-08	Ubah Jenis Narkoba	Fungsionalitas untuk mengubah data turunan narkoba yang ada.
UC-09	Hapus Jenis Narkoba	Fungsionalitas untuk menghapus data turunan narkoba yang ada.

2. Identifikasi Aktor

Aktor yang berperan dalam menjalankan sistem dapat dilihat pada tabel dibawah ini.

Tabel 3.10 Deskripsi Aktor

No	Aktor	Deksripsi
P-01	Petugas atau Anggota Penyidik SatReskrim Narkoba	Merupakan aktor yang menggunakan sistem pendeteksi penyalahgunaan narkoba.

3. Skenario *Use Case*

Skenario setiap bagian pada *use case* menunjukkan proses apa yang terjadi pada setiap bagian didalam use case tersebut, dimana user memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh sistem kepada user setelah user memberikan perintah pada setiap bagian-bagian *use case*. Berikut adalah skenario *use case* pada sistem pendeteksi penyalahgunaan narkoba.

Tabel 3.11 Skenario Use Case Pengolahan Data Pecandu

Nama Use Case	Pengolahan Data Pecandu
Deskripsi	Pengguna dapat mengelola data pecandu.
Aktor	Petugas
Kondisi Awal	Sumber data input tersedia
Kondisi Akhir	Sistem menyimpan perubahan data pecandu
Extend	Tambah Data Pecandu, Ubah Data Pecandu, Hapus Data Pecandu
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Pengguna berada pada tampilan beranda sistem pendeteksi penyalahgunaan narkoba	
2. Pengguna memilih menu data pecandu	3. Menampilkan halaman menu data pecandu.

Tabel 3.12 Skenario Use Case Tambah Data Pecandu

Nama Use Case	Tambah Data Pecandu
Deskripsi	Pengguna dapat menambah data pecandu yang baru.
Aktor	Petugas
Kondisi Awal	Data pecandu baru belum ada
Kondisi Akhir	Sistem menyimpan data pecandu yang baru
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih tombol tambah data pecandu baru	2. Menampilkan halaman tambah data pecandu.
3. Mengisi data pecandu yang baru (data tidak harus berurutan)	
4. Memilih tombol simpan	5. Melakukan validasi data.
	6. Menyimpan data baru ke database.
	7. Menampilkan pesan "Data berhasil disimpan"
Eksepsi	
3. Jika data tidak valid: a. Jika data belum lengkap, menampilkan pesan "Lengkapi data terlebih dahulu", kemudian kembali ke tahap 3.	

Tabel 3.13 Skenario Use Case Ubah Data Pecandu

Nama Use Case	Ubah Data Pecandu
Deskripsi	Pengguna dapat mengubah data pecandu.
Aktor	Petugas
Kondisi Awal	Data pecandu tersedia dan data tidak sesuai
Kondisi Akhir	Sistem menyimpan perubahan data pecandu.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih data pecandu yang ingin diubah	
2. Memilih tombol ubah data pecandu	3. Menampilkan halaman ubah data pecandu.
4. Mengganti data dengan data pecandu baru (data tidak harus berurutan)	
5. Memilih tombol ubah	6. Melakukan validasi data.
	7. Menyimpan perubahan data baru ke database.
	8. Menampilkan pesan "Data berhasil diubah"
Eksepsi	
6. Jika data tidak valid: a. Jika data belum lengkap, menampilkan pesan "Lengkapi data terlebih dahulu", kemudian kembali ke tahap 3.	

Tabel 3.14 Skenario Use Case Hapus Data Pecandu

Nama Use Case	Hapus Data Pecandu
Deskripsi	Pengguna dapat menghapus data pecandu.
Aktor	Petugas
Kondisi Awal	Data pecandu tersedia
Kondisi Akhir	Sistem menghapus data pecandu.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih data pecandu yang akan dihapus	2. Menampilkan konfirmasi "Apakah yakin akan menghapus data?"
3. Memilih tombol YA	4. Data pecandu dihapus.
	5. Menampilkan pesan "Data Berhasil dihapus".
Alternatif	
4. Jika pengguna memilih tombol TIDAK, data pecandu tidak dihapus dan kembali ke halaman data pecandu.	

Tabel 3.15 Skenario Use Case Pengolahan Pengujian

Nama Use Case	Pengolahan Pengujian
Deskripsi	Pengguna dapat menguji data pecandu untuk mendapatkan hasil identifikasi.
Aktor	Petugas
Kondisi Awal	Data pelatihan *Backpropagation* tersedia, data identifikasi belum tersedia
Kondisi Akhir	Menampilkan hasil pengujian *Backpropagation*, menyimpan hasil pengujian *Backpropagation*, dan menampilkan grafik hasil pengujian *Backpropagation*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu pengolahan pengujian	2. Menampilkan halaman pengolahan pengujian
3. Mengisi data pecandu	
4. Menentukan target identifikasi	
5. Memilih tombol proses	6. Melakukan validasi data.
	7. Menghitung proses pendeteksian dengan algoritma JST *Backpropagation* berdasarkan parameter hasil pelatihan jaringan.
	8. Menampilkan identifikasi hasil pengujian *Backpropagation.*
Alternatif	
9. Memilih tombol *Backpropagation*	10. Menampilkan form tabel hasil pengujian algoritma *Backpropagation* dan grafik MSE Epoch.
11. Memilih tombol simpan	12. Menyimpan data hasil pengujian *Backpropagation.*
13. Memilih tombol ulangi	14. Menampilkan halaman pengujian yang baru.
Eksepsi	
7. Melakukan validasi data: Jika data belum lengkap, menampilkan pesan "Lengkapi data terlebih dahulu", kemudian kembali ke tahap 3.	

Tabel 3.16 Skenario Use Case Pengolahan Jenis Narkoba

Nama Use Case	Pengolahan Jenis Narkoba
Deskripsi	Pengguna dapat mengelola data jenis narkoba.
Aktor	Petugas
Kondisi Awal	Sumber datajenis narkoba tersedia
Kondisi Akhir	Sistem menyimpan perubahan data jenis narkoba
Extend	Tambah Jenis Narkoba, Ubah Jenis Narkoba, Hapus Jenis Narkoba
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Pengguna berada pada tampilan beranda sistem pendeteksi penyalahgunaan narkoba	
2. Pengguna memilih menu narkoba	3. Menampilkan halaman menu narkoba.

Tabel 3.17 Skenario Use Case Tambah Jenis Narkoba

Nama Use Case	Tambah Data Jenis Narkoba
Deskripsi	Pengguna dapat menambah data jenis narkoba yang baru.
Aktor	Petugas
Kondisi Awal	Data jenis narkoba yang baru belum ada
Kondisi Akhir	Sistem menyimpan data jenis narkoba yang baru
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih tombol tambah data jenis narkoba baru	2. Menampilkan halaman tambah data jenis narkoba.
3. Mengisi data jenis narkoba yang baru (data tidak harus berurutan)	
4. Memilih tombol simpan	5. Melakukan validasi data.
	6. Menyimpan data baru ke database.
	7. Menampilkan pesan “Data berhasil disimpan”
Eksepsi	
6. Jika data tidak valid: a. Jika data belum lengkap, menampilkan pesan “Lengkapi data terlebih dahulu”, kemudian kembali ke tahap 3.	

Tabel 3.18 Skenario Use Case Ubah Jenis Narkoba

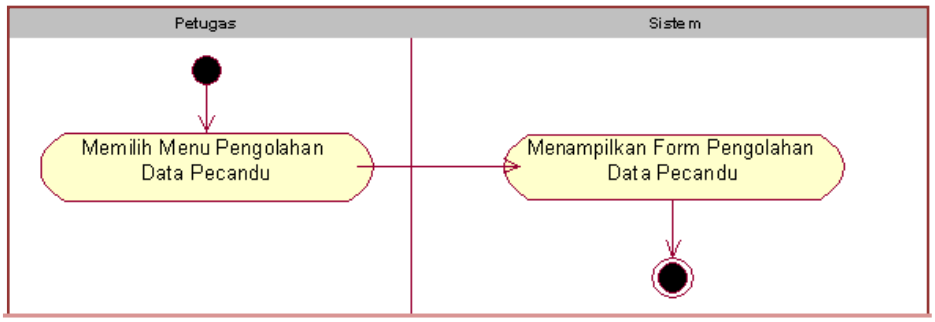
Nama Use Case	Ubah Jenis Narkoba
Deskripsi	Pengguna dapat mengubah data jenis narkoba
Aktor	Petugas
Kondisi Awal	Data jenis narkoba tersedia dan data tidak sesuai
Kondisi Akhir	Sistem menyimpan perubahan data jenis narkoba
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih data jenis narkoba yang ingin diubah	
2. Memilih tombol ubah jenis narkoba	3. Menampilkan halaman ubah data jenis narkoba
4. Mengganti data dengan data jenis narkoba yang baru (data tidak harus berurutan)	
5. Memilih tombol ubah	6. Melakukan validasi data.
	7. Menyimpan perubahan data baru ke database.
	8. Menampilkan pesan "Data berhasil diubah"
Eksepsi	
7. Jika data tidak valid: a. Jika data belum lengkap, menampilkan pesan "Lengkapi data terlebih dahulu", kemudian kembali ke tahap 3.	

Tabel 3.19 Skenario Use Case Hapus Jenis Narkoba

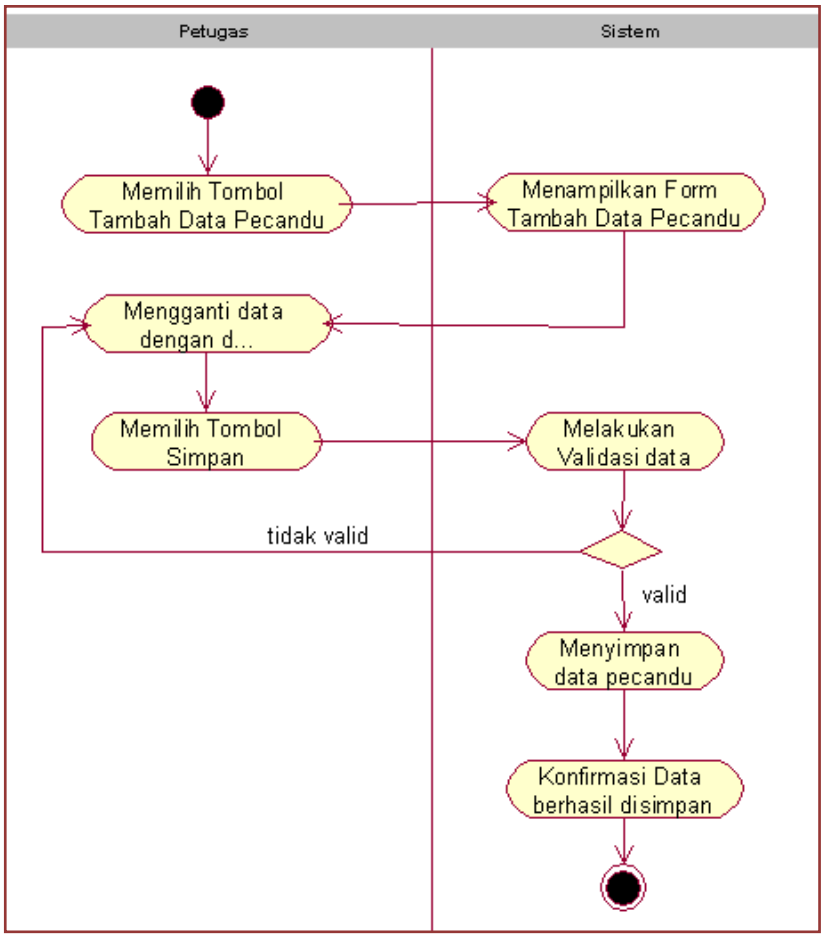
Nama Use Case	Hapus Jenis Narkoba
Deskripsi	Pengguna dapat menghapus data jenis narkoba
Aktor	Petugas
Kondisi Awal	Data jenis narkoba tersedia
Kondisi Akhir	Sistem menghapus data jenis narkoba.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih data jenis narkoba yang akan dihapus	2. Menampilkan konfirmasi "Apakah yakin akan menghapus data?"
3. Memilih tombol YA	4. Data jenis narkoba dihapus.
	5. Menampilkan pesan "Data Berhasil dihapus".
Alternatif	
3. Jika pengguna memilih tombol TIDAK, data jenis narkoba tidak dihapus dan kembali ke halaman narkoba.	

3.1.5.2 Activity Diagram

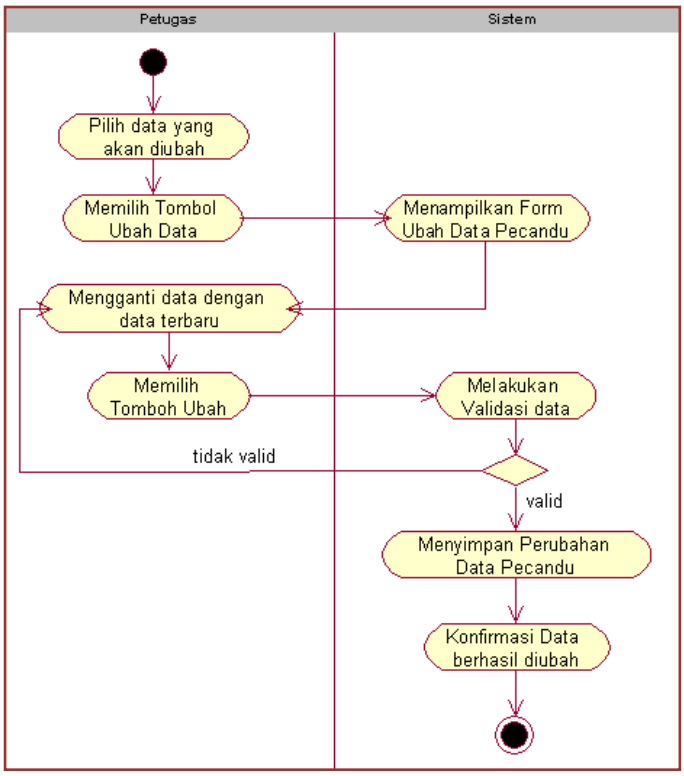
Diagram aktivitas (*Activity Diagram*) memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Berikut gambar diagram aktivitas dari sistem yang akan dibangun.



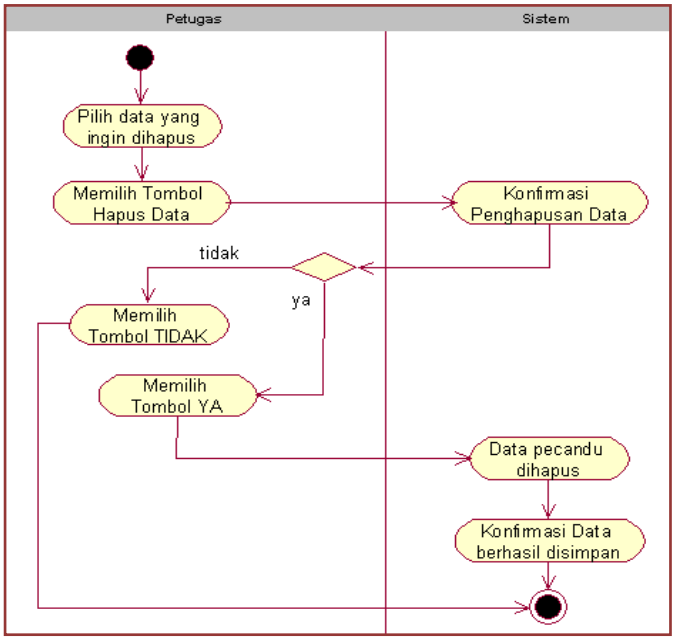
Gambar 3.5 Activity Diagram : Pengolahan Data Pecandu



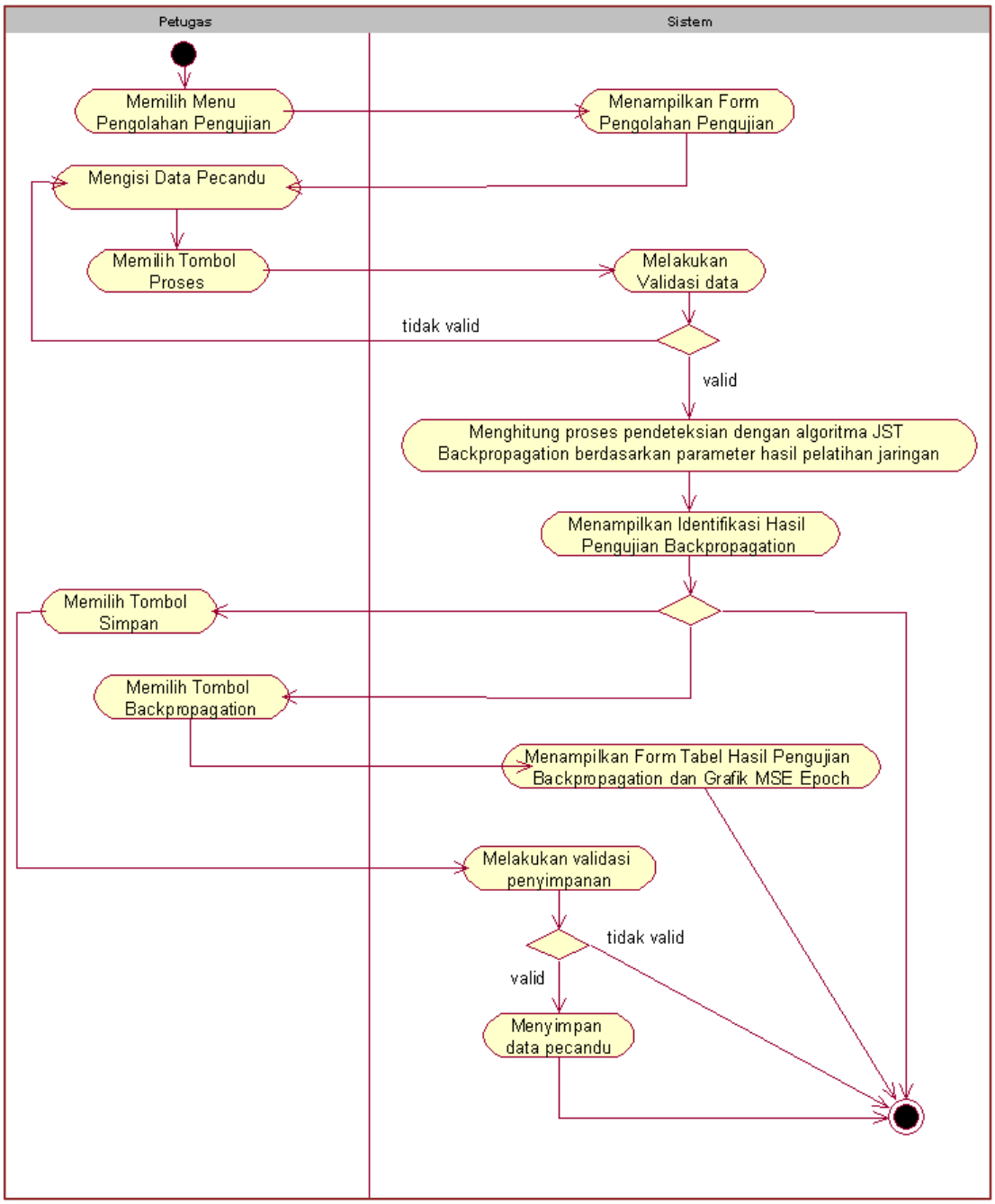
Gambar 3.6 Activity Diagram : Tambah Data Pecandu



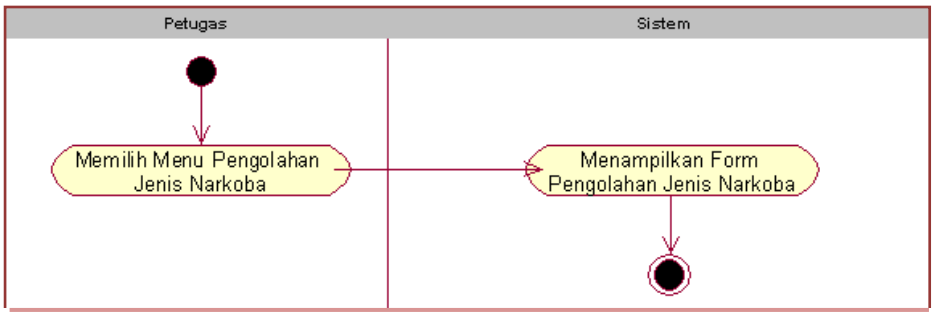
Gambar 3.7 Activity Diagram : Ubah Data Pecandu



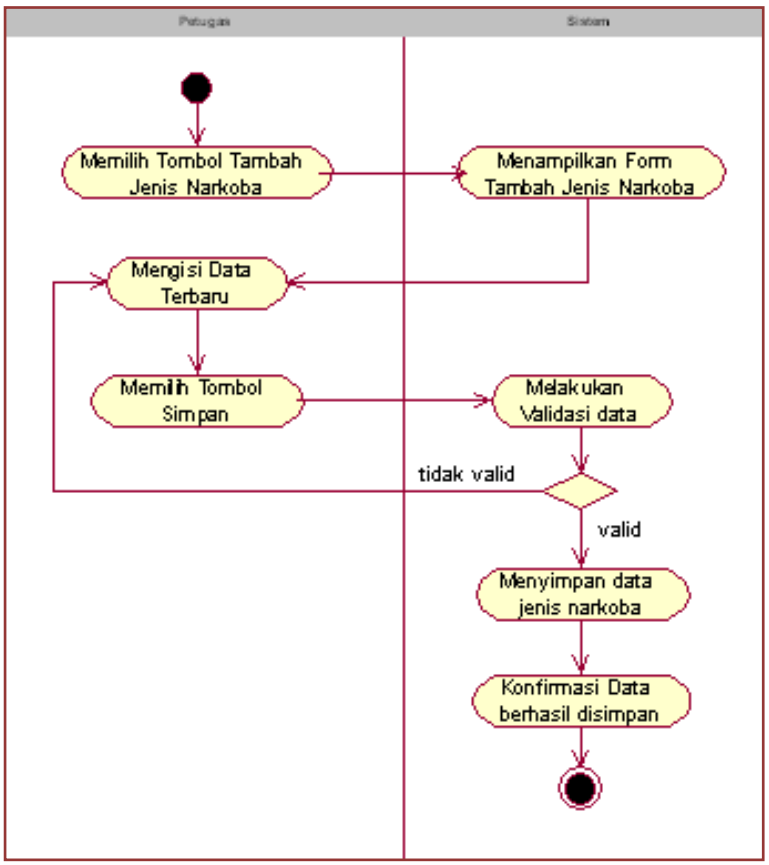
Gambar 3.8 Activity Diagram : Hapus Data Pecandu



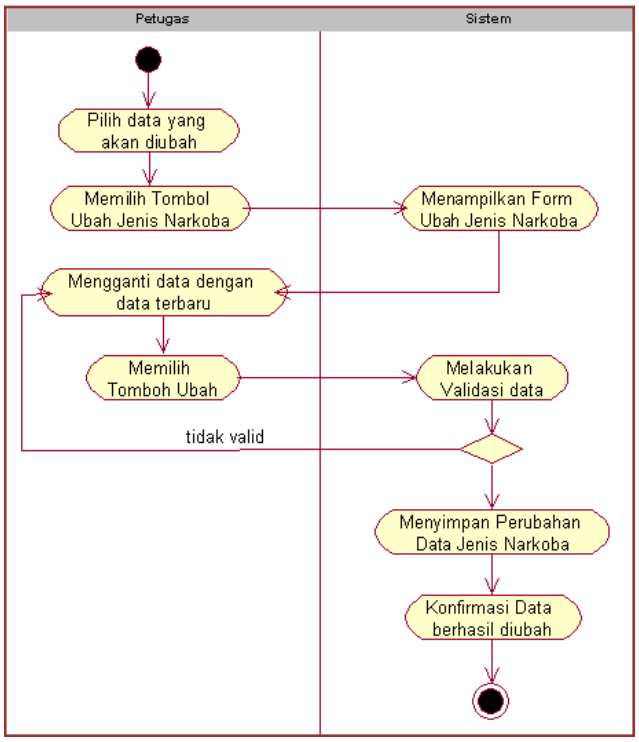
Gambar 3.9 Activity Diagram : Pengolahan Pengujian



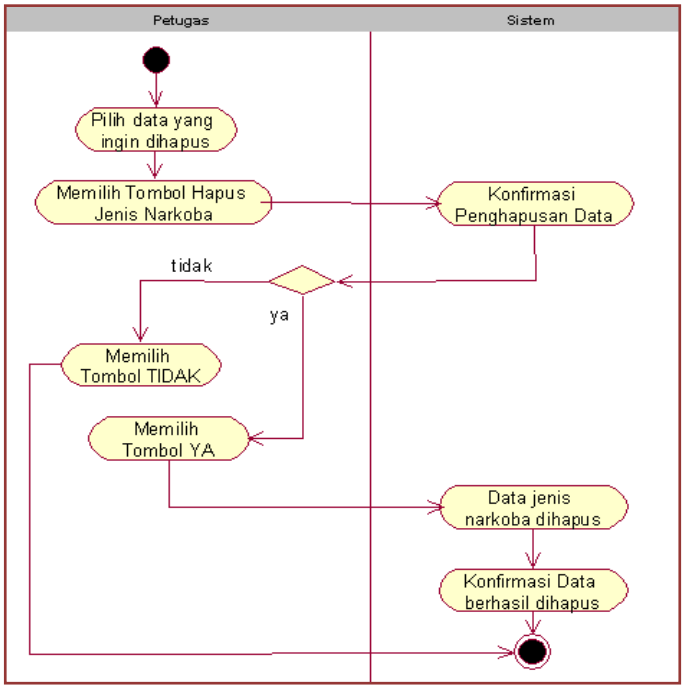
Gambar 3.10 Activity Diagram : Pengolahan Jenis Narkoba



Gambar 3.11 Activity Diagram : Tambah Jenis Narkoba



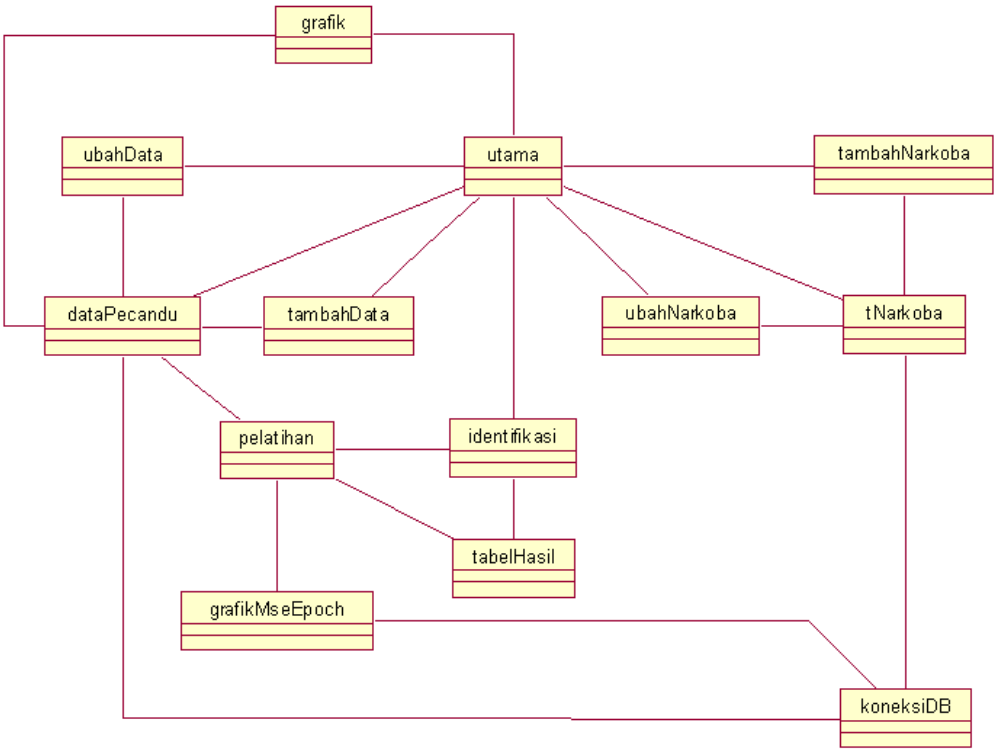
Gambar 3.12 Activity Diagram : Ubah Jenis Narkoba



Gambar 3.13 Activity Diagram : Hapus Jenis Narkoba

3.1.5.3 Class Diagram

Class diagram membantu memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling sering ditemui dalam pemodelan sistem berbasis *object-oriented.* Berikut ini *class diagram* dari aplikasi yang akan dibangun.



Gambar 3.14 Class Diagram

Tabel 3.20 Attribute dan Method Class Diagram

Nama *Class*	***Attribute***	***Method***

utama	dataPecandu dp = new dataPecandu();
	tNarkoba tnb = new tNarkoba();
	pelatihan lth = new pelatihan();
	pengujian uji = new pengujian();
	grafik gr = new grafik();
	NumberFormat NFCI = NumberFormat.getIntegerInstance();
	String pola_nama = "[a-z,A-Z]*";
	String pola_angka = "[0-9]*";
	String pola_simbol = "[*,-,@,!,$,<]*";
	SimpleDateFormat sdf;
		public utama();
		public void disablePengujian(Boolean x);
		public void clearNarkoba();
		public void resetPengujian();
		public void autoNomorPasien() ;
		public String tgl();
		public void isiTpasien();
		public void isiTNarkoba(String ket) ;
		private void makeSize();
		public static void main(String args[]);

koneksiDB	private String dbuser = "root";
	private String dbpasswd = "";
	private Statement stmt = null;
	private Connection con = null;
	private ResultSet rs = null;
		public koneksiDB();
		public ResultSet getData(String SQLString);
		public void query(String SQLString);
		public PreparedStatement getPreparedStatement(String query) ;

dataPecandu	private String idPasien;
	private String nama;
	private int usia;
	private String x1,x2,x3,x4,x5,x6,x7, x8,x9,x10,x11,x12x13,x14,x15,x16, x17,x18,x19,x20,x21,x22,x23;
	private String identifikasi;
		public String getIdPasien();
		public void setIdPasien(String IdPasien);
		public String getNama();
		public void setNama(String nama);
		public int getUsia();
		public void setUsia(int u);
		public String getX1();
		public void setX1(String x);
		public String getX2();
		public void setX2(String x) ;
		public String getX3();
		public void setX3(String x);
		public String getX4();
		public void setX4(String x) ;
		public String getX5();
		public void setX5(String x);
		public String getX6();
		public void setX6(String x) ;
		public String getX7();
		public void setX7(String x);
		public String getX8();
		public void setX8(String x) ;
		public String getX9();
		public void setX9(String x);
		public String getX10();
		public void setX10(String x) ;
		public String getX11();
		public void setX11(String x);
		public String getX12();
		public void setX12(String x) ;
		public String getX13();
		public void setX13(String x);
		public String getX14();
		public void setX14(String x) ;
		public String getX15();
		public void setX15(String x);
		public String getX16();
		public void setX16(String x) ;
		public String getX17();
		public void setX17(String x);
		public String getX18();
		public void setX18(String x) ;
		public String getX19();
		public void setX19(String x);
		public String getX20();

		public void setX20(String x) ;
		public String getX21();
		public void setX21(String x);
		public String getX22();
		public void setX22(String x) ;
		public void setX23(String x);
		public String getIdentifikasi();
		public void setIdentifikasi(String identifikasi);
		public void insertPasien();
		public ResultSet getPasien(String ID);
		public ResultSet searchPasien(String nama);
		public void deletePasien(String ID);
		public void updatePasien();
		public int getMaxPasien();

tNarkoba	private String turunan;
	private String keterangan;
	private String pengertian;
	private String gejalaEfekSamping;
	private String gambar;
		public String getTurunan();
		public void setTurunan(String trn);
		public String getKeterangan();
		public void setKeterangan(String ket);
		public String getPengertian();
		public void setPengertian(String arti);
		public String getGejalaEfekSamping();
		public void setGejalaEfekSamping(String efek);
		public String getGambar();
		public void setGambar(String gbr);
		public void insertNarkoba();
		public ResultSet getNarkoba(String trn);
		public ResultSet searchNarkoba(String trn);
		public void deleteNarkoba(String trn);
		public void updateNarkoba();
		public int getMaxNarkoba();

identifikasi	private String idPasien;
	private String nama;
	private int usia;
	private int x1,x2,x3,x4,x5,x6,x7,x8,x9, x10,x11,x12,x13,x14,x15,x16,x17,x18, x19,x20,x21,x22,x23;
	private String identifikasi, hasil, kode;
	private double[][] A1;
	private double[][] A2;
	private double[][] B1 = new double[1][5];
	private double[][] B2 = new double[1][3];
	public pelatihan latih = new pelatihan();
		public String getIdPasien();
		public void setIdPasien(String IdPasien);
		public String getNama();
		public void setNama(String nama);
		public int getUsia();
		public void setUsia(int u);
		public String getX1();
		public void setX1(String x);
		public String getX2();
		public void setX2(String x) ;
		public String getX3();
		public void setX3(String x);
		public String getX4();
		public void setX4(String x) ;
		public String getX5();
		public void setX5(String x);
		public String getX6();

		public void setX6(String x) ;
		public String getX7();
		public void setX7(String x);
		public String getX8();
		public void setX8(String x) ;
		public String getX9();
		public void setX9(String x);
		public String getX10();
		public void setX10(String x) ;
		public String getX11();
		public void setX11(String x);
		public String getX12();
		public void setX12(String x) ;
		public String getX13();
		public void setX13(String x);
		public String getX14();
		public void setX14(String x) ;
		public String getX15();
		public void setX15(String x);
		public String getX16();
		public void setX16(String x) ;
		public String getX17();
		public void setX17(String x);
		public String getX18();
		public void setX18(String x) ;
		public String getX19();
		public void setX19(String x);
		public String getX20();
		public void setX20(String x) ;
		public String getX21();
		public void setX21(String x);
		public String getX22();
		public void setX22(String x) ;
		public void setX23(String x);
		public String getIdentifikasi();
		public String getHasil()
		public void setHasil(String hasil);
		public String getKode();
		public void setKode(String Kode);
		public double[][] getA1() ;
		public void setA1(double[] aa1);
		public double[][] getA2();
		public void setA2(double[] aa2);
		public double[][] getB1();
		public void setB1(double[][] bb1);
		public double[][] getB2();
		public void setB2(double[][] bb2);
		public void prosesUji(int x1, int x2,int x3,int x4,int x5,int x6,int x7,int x8,int x9,int x10,int x11,int x12,int x13,int x14,int x15,int x16,int x17,int x18,int

		x19,int x20,int x21,int x22,int x23);

pelatihan	private int JHneuron ;
	private int JOneuron;
	private double lRate;
	private int epoch;
	private double mse;
	private double maxMSE;
	private double w1[][];
	private double w2[][];
	private double b1[][];
	private double b2[][];
	public dataPecandu dp = new dataPecandu();
		public int getJHneuron();
		public void setJHneuron();
		public int getJOneuron();
		public void setJOneuron();
		public double getLearningRate();
		public void setLearningRate();
		public int getEpoch();
		public void setEpoch()
		public double getMSE();
		public void setMSE();
		public double[][] getW1();
		public void setW1(double[][] W1);
		public double[][] getW2();
		public void setW2(double[][] W2);
		public double[][] getB1() ;
		public void setB1(double[][] B1);
		public double[][] getB2() ;
		public void setB2(double[][] B2);
		public double[][] acakW1();
		public double[][] acakW2();
		public double[][] acakB1();
		public double[][] acakB2();
		public int[][] getPola();
		public int[][] getTarget();
		public void insertMseEpoch(double mse, int epoh);
		public void latihJST();

tambahData	String pola_nama = "[a-z, A-Z]*";
	String pola_angka = "[0-9]*";
	dataPecandu dp = new dataPecandu();
	utama u = new utama();
		public tambahData(java.awt.Frame parent, boolean modal) ;
		public void autoNumberPasien();

ubahData	String pola_nama = "[a-z, A-Z]*";
	String pola_angka = "[0-9]*";
	NumberFormat NFCI = NumberFormat.getIntegerInstance();
	dataPecandu dp = new dataPecandu();
	utama u = new utama();
		public ubahData(java.awt.Frame parent, boolean modal) ;
		public void getNumberPasien(String id) ;
		private void bTutupActionPerformed (java.awt.event.ActionEvent evt) ;
		private void bHapusActionPerformed (java.awt.event.ActionEvent evt) ;
		private void bUbahActionPerformed (java.awt.event.ActionEvent evt);

tambahNarkoba	String pola_huruf = "[a-z, A-Z]*";
	tNarkoba tn = new tNarkoba();
	utama u = new utama();
		public tambahNarkoba(java.awt.Frame parent, boolean modal) ;
		private void bBrowseActionPerformed (java.awt.event.ActionEvent evt) ;
		private void bTutupActionPerformed (java.awt.event.ActionEvent evt) ;
		private void bHapusActionPerformed (java.awt.event.ActionEvent evt) ;
		private void bSimpanActionPerformed;

ubahNarkoba	String pola_huruf = "[a-z, A-Z]*";
	tNarkoba tn = new tNarkoba();
	utama u = new utama();
	public ubahNarkoba(java.awt.Frame parent, boolean modal) ;
		public void getDataNarkoba(String trn) ;
		private void bUbahActionPerformed (java.awt.event.ActionEvent evt);
		private void bBrowseActionPerformed (java.awt.event.ActionEvent evt) ;
		private void bTutupActionPerformed (java.awt.event.ActionEvent evt) ;
		private void bHapusActionPerformed (java.awt.event.ActionEvent evt) ;

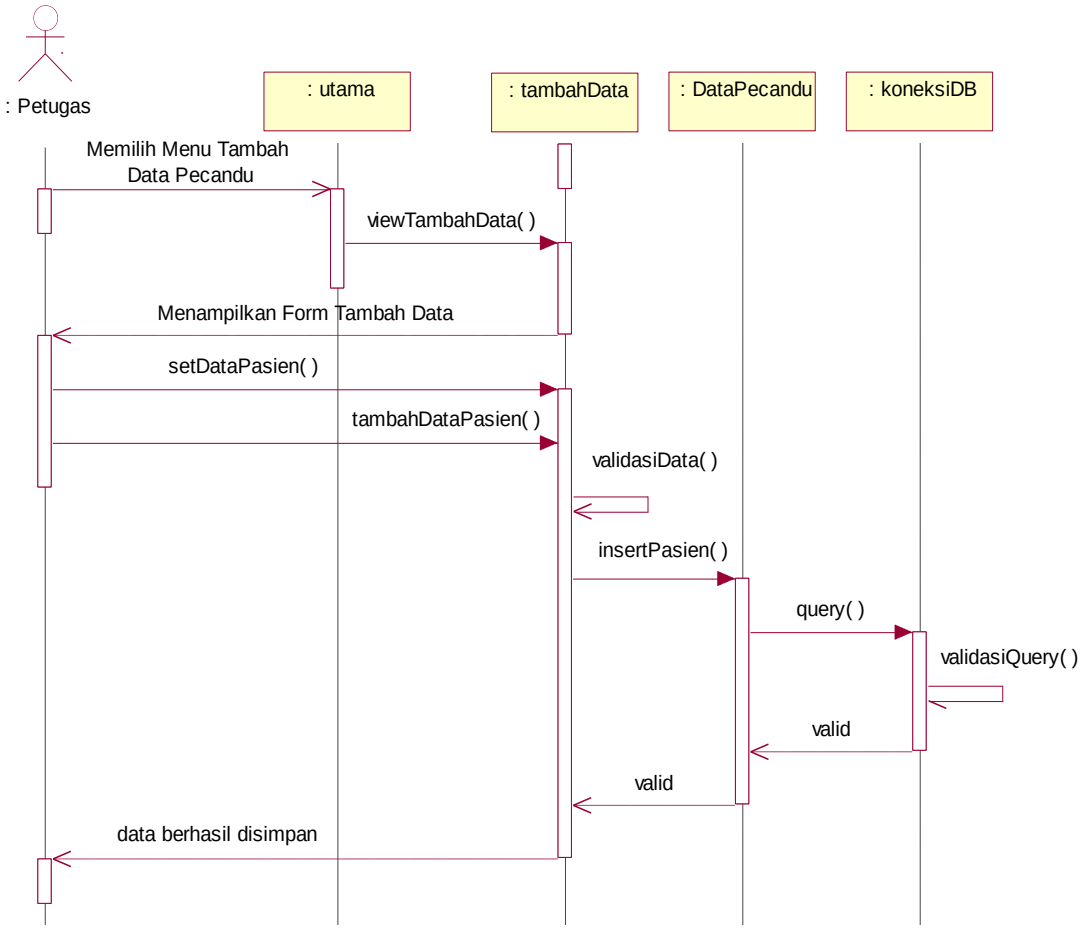
grafik	dataPecandu dp = new dataPecandu();
	public int x, nar, psi, zat;
		public int getNarkotika()
		public int getPsikotropika();
		public int getZatAdiktif();
		public int JumlahPasien(String ID);
		public static void main(String[] ar, int n, int p, int z);

tabelHasil	pengujian uji = new pengujian();
	pelatihan latih = new pelatihan();
	DecimalFormat df = new DecimalFormat();
		public tabelHasil(java.awt.Frame parent, boolean modal);
		public void isiTabelHasilA1(double[][] a1);
		public void isiTabelHasilA2(double[][] a2);
		public void isiTabelHasilB1(double[][] b1);
		public void isiTabelHasilB2(double[][] b2);

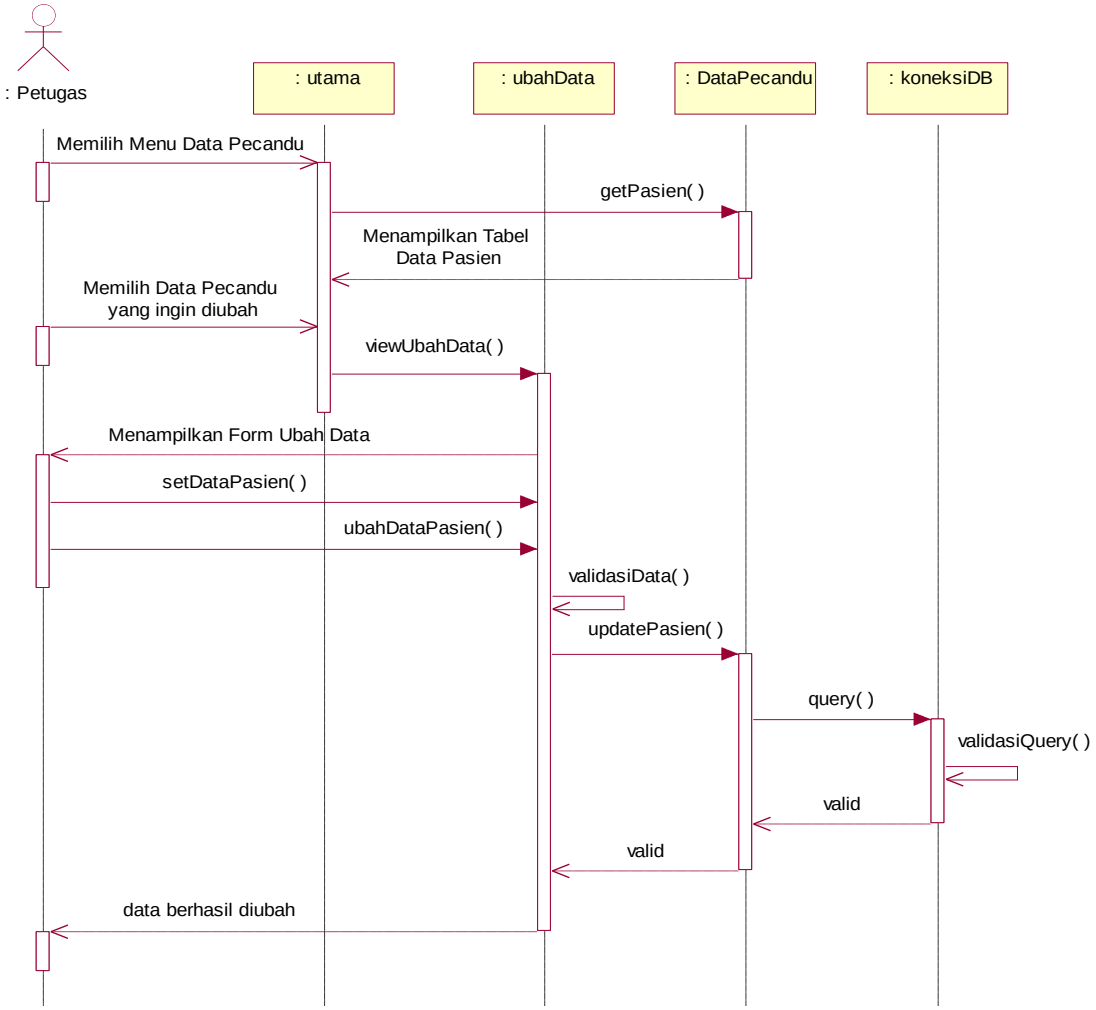
grafikEpochMSE	private String query; private JDBCCategoryDataset data; private JFreeChart jChart;	public static void main(String[] ar) ;

3.1.5.4 Sequence Diagram

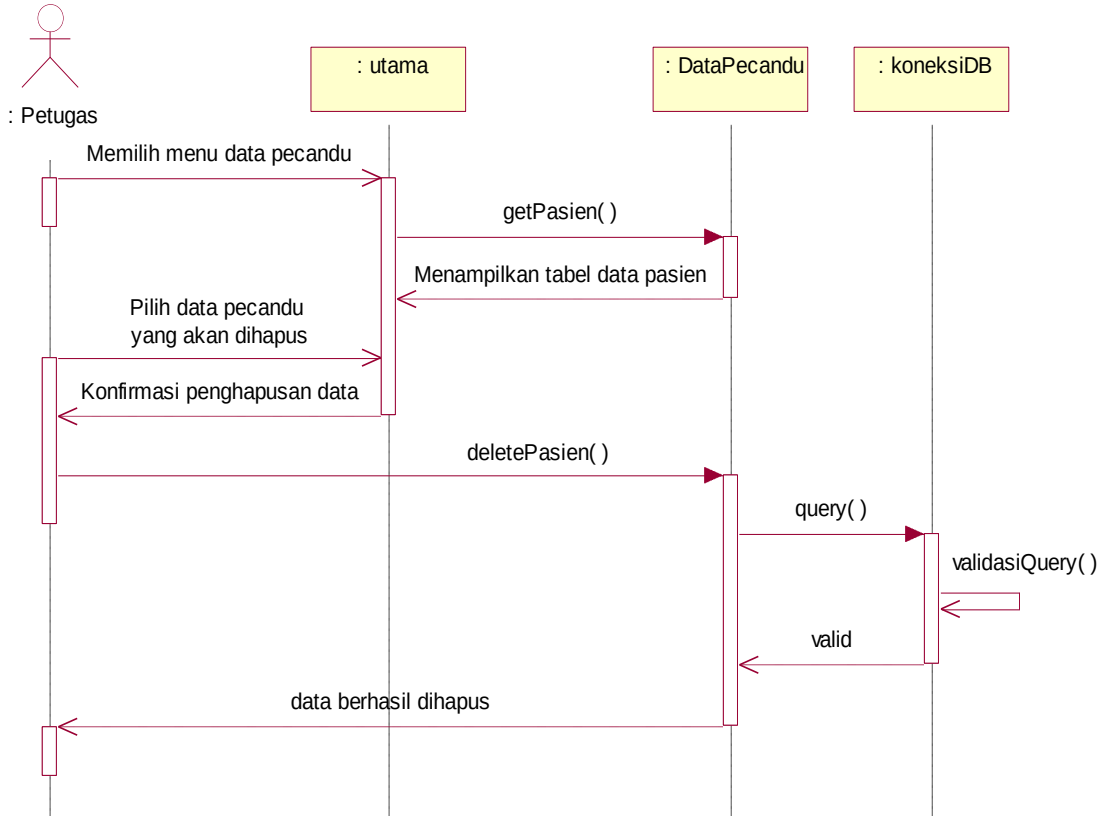
Sequence diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Sequence diagram yang ada pada aplikasi ini digambarkan sebagai berikut.



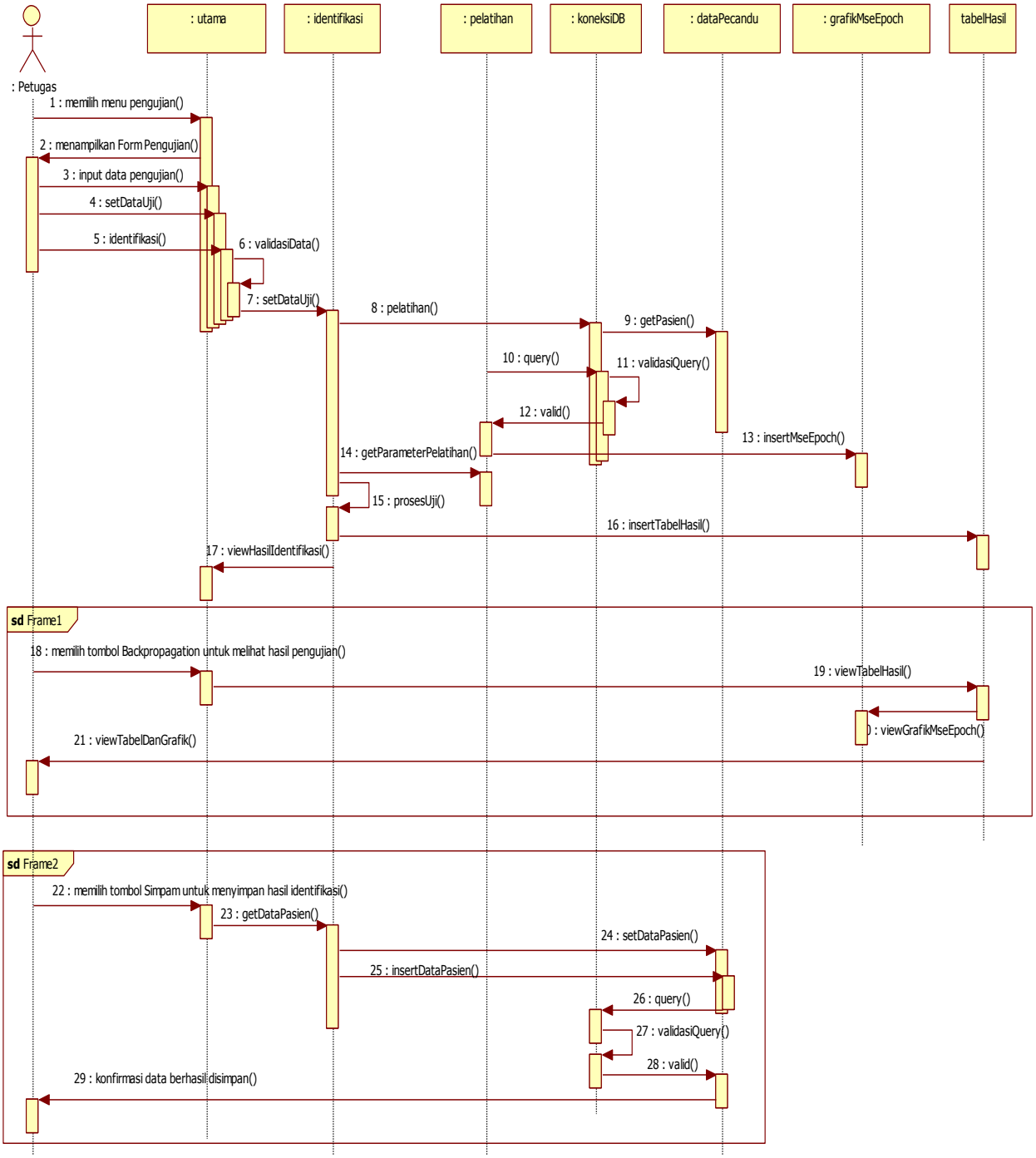
Gambar 3.15 Sequence Diagram : Tambah Data Pecandu



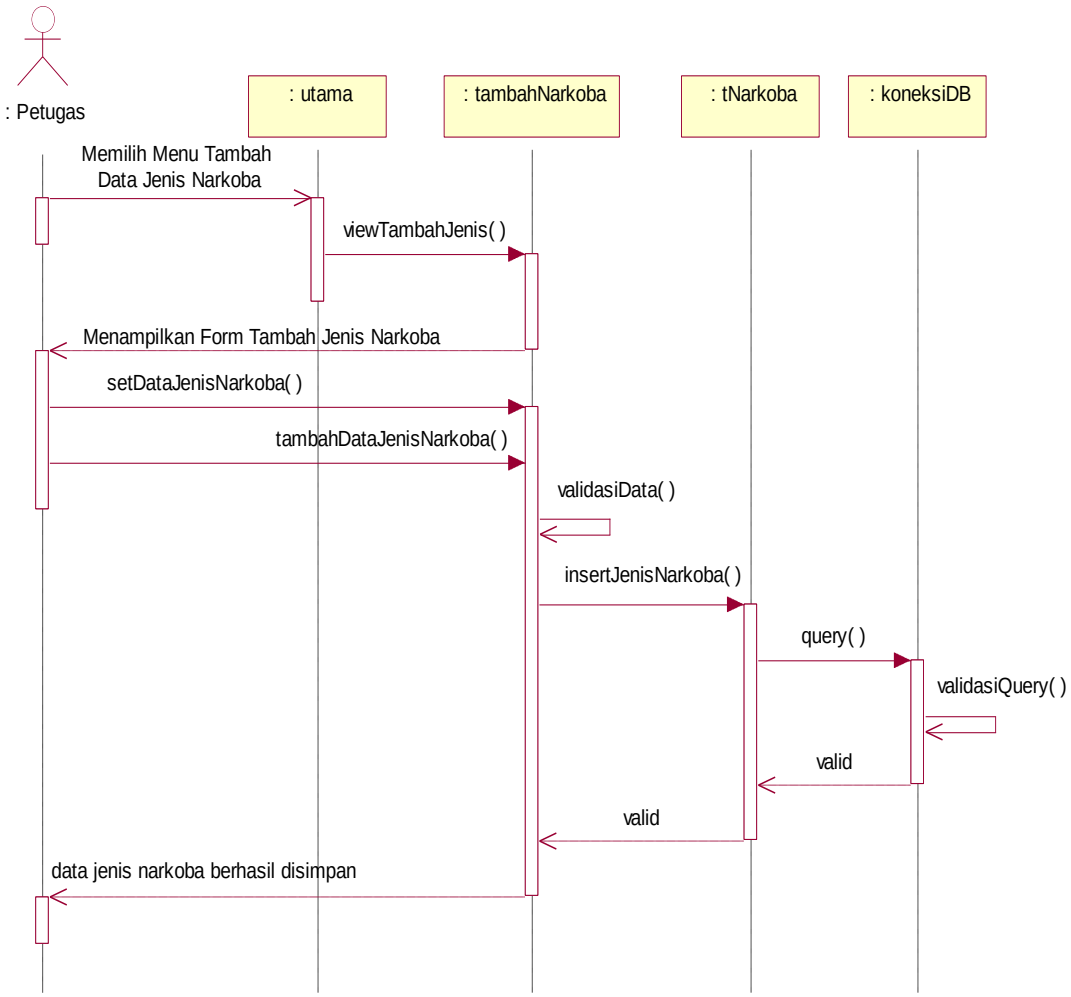
Gambar 3.16 Sequence Diagram : Ubah Data Pecandu



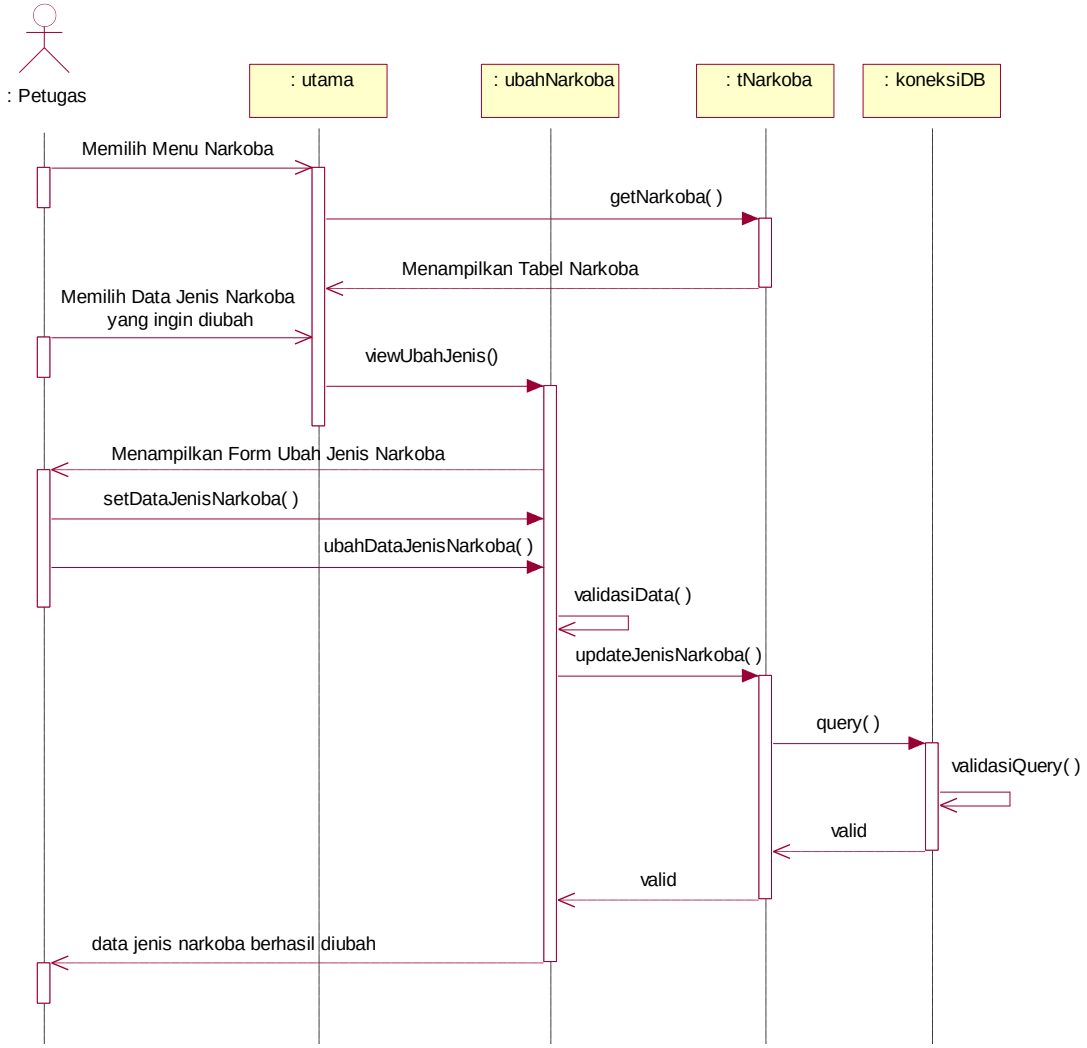
Gambar 3.17 Sequence Diagram : Hapus Data Pecandu



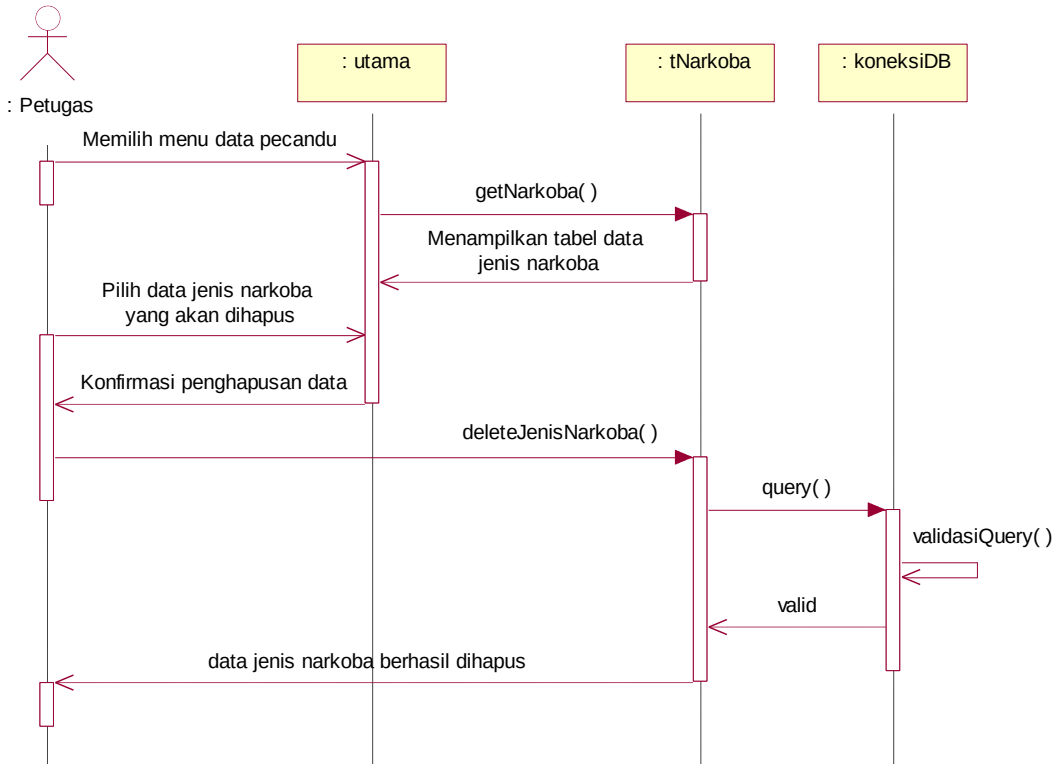
Gambar 3.18 Sequence Diagram : Pengolahan Pengujian



Gambar 3.19 Sequence Diagram : Tambah Data Jenis Narkoba



Gambar 3.20 Sequence Diagram : Ubah Data Jenis Narkoba



Gambar 3.21 Sequence Diagram : Hapus Data Jenis Narkoba

3.2 Perancangan Sistem

Perancangan merupakan bagian dari metodologi pembangunan suatu perangkat lunak yang harus dilakukan setelah melalui tahapan analisis. Pada bagian ini akan dijelaskan perancangan sistem yang dimaksudkan yang meliputi perancangan basis data, perancangan struktur menu, perancangan antarmuka, perancangan pesan, jaringan semantik. Semua perancangan yang telah disebutkan diatas harus saling berhubungan agar menciptakan suatu sistem yang baik.

3.2.1 Perancangan Basis Data

Tahapan perancangan basis data dalam sistem pendeteksi penyalahgunaan narkoba ini terdiri dari perancangan struktur tabel serta perancangan pengkodean.

3.2.1.1 Struktur Tabel

Struktur tabel merupakan isi data yang berada dalam satu *record* pada satu tabel. Tabel-tabel yang digunakan dalam *database* aplikasi sistem pendeteksi penyalahgunaan narkoba adalah tabel untuk mengolah data pecandu narkoba, tabel data tentang jenis narkoba serta tabel nilai MSE dan Epoch.

1. Tabel mseEpoch

Tabel mseEpoch digunakan meyimpan nilai MSE dan Epoch yang merupakan hasil pelatihan jaringan dengan algoritma JST backpropagation. Struktur tabel mseEpoch adalah sebagai berikut :

Tabel 3.21 Tabel mseEpoch

Nama *Field*	Tipe	Panjang	Kunci	Keterangan
Epoch	int	11	*Primary Key*	*Not Null*
MSE	double	-		*Not Null*

2. Tabel tKategori

Tabel tKategori digunakan meyimpan data jenis narkoba yang terdiri dari Narkotika, Psikotropika dan Zat Adiktif Lain. Struktur tabel tKategori adalah sebagai berikut :

Tabel 3.22 Tabel tKategori

Nama *Field*	Tipe	Panjang	Kunci	Keterangan
Jenis	Varchar	20	*Primary Key*	*Not Null*
Kode	Varchar	3		*Not Null*

3. Tabel tNarkoba

Tabel tNarkoba digunakan meyimpan data turunan narkoba yang berisi penjelasan masing-masing turunan serta gejala efek samping dari pemakaian obat-obatan tersebut. Struktur tabel tNarkoba adalah sebagai berikut :

Tabel 3.23 Tabel tNarkoba

Nama *Field*	Tipe	Panjang	Kunci	Keterangan
Turunan	Varchar	30	*Primary Key*	*Not Null*
Pengertian	Text	-		*Not Null*
GejalaEfekSamping	Text	-		*Not Null*
Gambar	Text	-		*Not Null*
Keterangan	Varchar	20		*Not Null*

4. Tabel dataPecandu

Tabel dataPecandu digunakan meyimpan data pecandu narkoba yang merupakan data Masukan untuk menjalankan sistem pendeteksi penyalahgunaan narkoba dengan algoritma JST backpropagation. Struktur tabel dataPecandu adalah sebagai berikut :

Tabel 3.24 Tabel dataPecandu

Nama *Field*	Tipe	Panjang	Kunci	Keterangan
idPasien	Varchar	4	*Primary Key*	*Not Null*
Nama	Varchar	50		*Not Null*
Usia	Integer	-		*Not Null*
X1	Set(‘0’,’1’)	-		*Not Null*
X2	Set(‘0’,’1’)	-		*Not Null*
X3	Set(‘0’,’1’)	-		*Not Null*
X4	Set(‘0’,’1’)	-		*Not Null*
X5	Set(‘0’,’1’)	-		*Not Null*
X6	Set(‘0’,’1’)	-		*Not Null*
X7	Set(‘0’,’1’)	-		*Not Null*
X8	Set(‘0’,’1’)	-		*Not Null*
X9	Set(‘0’,’1’)	-		*Not Null*
X10	Set(‘0’,’1’)	-		*Not Null*
X11	Set(‘0’,’1’)	-		*Not Null*
X12	Set(‘0’,’1’)	-		*Not Null*
X13	Set(‘0’,’1’)	-		*Not Null*
X14	Set(‘0’,’1’)	-		*Not Null*
X15	Set(‘0’,’1’)	-		*Not Null*
X16	Set(‘0’,’1’)	-		*Not Null*
X17	Set(‘0’,’1’)	-		*Not Null*
X18	Set(‘0’,’1’)	-		*Not Null*
X19	Set(‘0’,’1’)	-		*Not Null*
X20	Set(‘0’,’1’)	-		*Not Null*
X21	Set(‘0’,’1’)	-		*Not Null*
X22	Set(‘0’,’1’)	-		*Not Null*
X23	Set(‘0’,’1’)	-		*Not Null*
identifikasi	Set(‘Narkotika’, ’Psikotropika’, ’Zat Adiktif Lain’)	-		*Not Null*

3.2.1.2 Perancangan Pengkodean

Kode merupakan penyajian dalam mengklasifikasikan data sehingga mudah dalam proses masukan ke dalam sistem. Penggunaan kode biasanya digunakan untuk mengidentifikasikan data, simbol kode biasanya digunakan pada hampir semua proses yang ada kaitannya dengan data. Pada aplikasi ini terdapat ketentuan kode untuk tabel dataPecandu, yaitu :

1. Kode Pasien

   Panjang kode : 4 digit

   Tipe kode : String

   Struktur kode : AXXX

   Perincian :

   A = Menunjukkan jenis kode pasien yaitu P.

   XXX = Menunjukkan no urutan pasien.

   Contoh:

   P002 = kode pasien dengan urutan penyimpanan no 2.

3.2.2 Perancangan Method

Bagian ini menjelaskan *method-method* yang ada pada *class diagram.* diawali dari nama metode, deskpripsi dan algoritma dengan pendekatan *pseudocode.*

a. *Method Class* Data Pecandu

Tabel 3.25 Method Class Data Pecandu

Nama Metode	insertPasien()
Deskripsi	Entri data pecandu baru

Algoritma	`insertPasien(String idPasien, String nama,int usia, int x1, int x2, int x3, int x4, int x5, int x6, int x7, int x8, int x9, int x10, int x11, int x12, int x13, int x14, int x15, int x16, int x17, int x18, int x19, int x20, int x21, int x22, int x23, String identifikasi){`
	`koneksiDB db=new koneksiDB();`
	`String insert="insert into dataPecandu values ('"+idPasien+"','"+nama+"',"+usia+",'"+x1+"','"+x2+"','"+x3+"','"+x4+"','"+x5+"','"+x6+"','"+x7+"','"+x8+"','"+x9+"','"+x10+"','"+x11+"','"+x12+"','"+x13+"','"+x14+"','"+x15+"','"+x16+"','"+x17+"','"+x18+"','"+x19+"','"+x20+"','"+x21+"','"+x22+"','"+x23+"','"+identifikasi+"')";`

	db.query(insert); }
Nama Metode	updatePasien()
Deskripsi	Memperbaharui data pecandu

Algoritma	updatePasien(String idPasien, String nama,int usia, int x1, int x2, int x3, int x4, int x5, int x6, int x7, int x8, int x9, int x10, int x11, int x12, int x13, int x14, int x15, int x16, int x17, int x18, int x19, int x20, int x21, int x22, int x23, String identifikasi){
	koneksiDB db=new koneksiDB();
	String update=" update datapecandu set identifikasi='"+identifikasi+"',nama='"+nama+"',usia="+usia+",x1='"+x1+"',x2='"+x2+"',x3='"+x3+"',x4='"+x4+"',x5='"+x5+"',x6='"+x6+"',x7='"+x7+"',x8='"+x8+"',x9='"+x9+"',x10='"+x10+"',x11='"+x11+"',x12='"+x12+"',x13='"+x13+"',x14='"+x14+"',x15='"+x15+"',x16='"+x16+"',x17='"+x17+"',x18='"+x18+"',x19='"+x19+"',x20='"+x20+"',x21='"+x21+"',x22='"+x22+"',x23='"+x23+"' where idPasien='"+idPasien+"'";
	db.query(insert);
	}

Nama Metode	deletePasien()
Deskripsi	Menghapus data pecandu berdasarkan id

Algoritma	deletePasien(String id){
	koneksiDB db=new koneksiDB();
	String delete="delete from dataPecandu where idPasien='"+id+"'";
	db.query(delete);
	}

Nama Metode	searchPasien()
Deskripsi	Mencari data pecandu berdasarkan nama pasien

Algoritma	searchPasien(String nama){
	koneksiDB db=new koneksiDB();
	result=db.getData("select * from datapecandu where namaPasien='"+nama+"'");
	}

Nama Metode	getPasien()
Deskripsi	Mengambil data pecandu berdasarkan id

Algoritma	getPasien(String ID){
	koneksiDB db=new koneksiDB();
	result=db.getData("select * from datapecandu where idPasien='"+ID+"'");
	}

b. *Method Class* Pelatihan

Tabel 3.26 Method Class Pelatihan

Nama Metode	acakW1()
Deskripsi	Mengacak nilai variabel W1 (nilai Vij, bobot awal input ke *hidden)*

Algoritma	double[][] acakW1(){
	double[][] W1= new double[23][getJHneuron()];
	double[] row = new double[getJHneuron()];
	DecimalFormat df = new DecimalFormat();

	`Random rand = new Random();`
	`df.setMaximumFractionDigits(4);`
	
	`//bangkitkan bobot awal input ke hidden (Vij)`
	`for (int gg=0;gg<23;gg++){`
	`for (int ii=0;ii<getJHneuron();ii++){`
	`row[ii]=`
	`Double.parseDouble(df.format(rand.nextDouble()));`
	`W1[gg][ii]=row[ii];`
	`}`
	`}`
	`return W1;`
	`}`

Nama Metode	acakW2()
Deskripsi	Mengacak nilai variabel W2 (nilai Wjk, bobot awal *hidden* ke output)

Algoritma	`double[][] acakW2(){`
	`double[][] W2 = new double[getJHneuron()][getJOneuron()];`
	`double[] row = new double[getJOneuron()];`
	`DecimalFormat df = new DecimalFormat();`
	`Random rand = new Random();`
	`df.setMaximumFractionDigits(4);`
	
	`//bangkitkan bobot awal hidden ke output (Wjk)`
	`for (int hh=0;hh<getJHneuron();hh++){`
	`for (int jj=0;jj<getJOneuron();jj++){`
	`row[jj]=`
	`Double.parseDouble(df.format(rand.nextDouble()));`
	`W2[hh][jj]=row[jj];`
	`}`
	`}`
	`return W2;`
	`}`

Nama Metode	acakB1()
Deskripsi	Mengacak nilai variabel B1 (nilai Vo, bobot awal bias ke *hidden)*

Algoritma	`double[][] acakB1(){`
	`double[][] B1=new double[1][getJHneuron()];`
	`double[] row = new double[getJHneuron()];`
	`DecimalFormat df = new DecimalFormat();`
	`Random rand = new Random();`
	`df.setMaximumFractionDigits(4);`
	
	`//bangkitkan bobot awal bias ke hidden (V0ij)`
	`for (int g=0;g<1;g++){`
	`for (int i=0;i<getJHneuron();i++){`
	`row[i]=`
	`Double.parseDouble(df.format(rand.nextDouble()));`
	`B1[g][i]=row[i];`
	`}`
	`}`
	`return B1;`
	`}`

Nama Metode	acakB2()
Deskripsi	Mengacak nilai variabel B2 (nilai Wo, bobot awal bias ke output)

Algoritma	`double[][] acakB2(){`
	`double[][] B2=new double[1][getJOneuron()];`
	`double[] row = new double[getJOneuron()];`

	DecimalFormat df = new DecimalFormat();
	Random rand = new Random();
	df.setMaximumFractionDigits(4);
	
	//bangkitkan bobot awal bias ke output (W0jk)
	for (int h=0;h<1;h++){
	for (int j=0;j<getJOneuron();j++){
	row[j]=
	Double.parseDouble(df.format(rand.nextDouble()));
	B2[h][j]=row[j];
	}
	}
	return B2;
	}

Nama Metode	getPola()
Deskripsi	Mengambil pola data pecandu (variabel x1 hingga x23) dari database untuk menjadi data latih

Algoritma	int[][] getPola(){
	koneksiDB db = new koneksiDB();
	int i = dp.getMaxPasien();
	int j =0;
	int[][] pola = new int[i][23];
	ResultSet rs = null;
	try {
	rs = db.getData("select * from datapecandu group by idPasien");
	
	while (rs.next() && j<i) {
	pola[j][0]=rs.getInt("X1");
	pola[j][1]=rs.getInt("x2");
	pola[j][2]=rs.getInt("x3");
	pola[j][3]=rs.getInt("x4");
	pola[j][4]=rs.getInt("x5");
	pola[j][5]=rs.getInt("x6");
	pola[j][6]=rs.getInt("x7");
	pola[j][7]=rs.getInt("x8");
	pola[j][8]=rs.getInt("x9");
	pola[j][9]=rs.getInt("x10");
	pola[j][10]=rs.getInt("x11");
	pola[j][11]=rs.getInt("x12");
	pola[j][12]=rs.getInt("x13");
	pola[j][13]=rs.getInt("x14");
	pola[j][14]=rs.getInt("x15");
	pola[j][15]=rs.getInt("x16");
	pola[j][16]=rs.getInt("x17");
	pola[j][17]=rs.getInt("x18");
	pola[j][18]=rs.getInt("x19");
	pola[j][19]=rs.getInt("x20");
	pola[j][20]=rs.getInt("x21");
	pola[j][21]=rs.getInt("x22");
	pola[j][22]=rs.getInt("x23");
	
	j++;
	}
	rs.close();
	} catch (Exception e) {
	javax.swing.JOptionPane.showMessageDialog(null, "error

	`: " + e.getMessage());`
	`   }`
	`   return pola;`
	`}`

Nama Metode	getTarget()
Deskripsi	Mengambil pola target data pecandu dari database untuk menjadi data latih

Algoritma	`int[][] getTarget(){`
	`   koneksiDB db = new koneksiDB();`
	`   String identifikasi;`
	`   int i = dp.getMaxPasien();`
	`   int j =0;`
	`   int[][] target = new int[i][getJOneuron()];`
	`   ResultSet rs = null;`
	
	`   try {`
	`       rs = db.getData("select * from datapecandu group by idPasien");`
	
	`       while (rs.next() && j<i) {`
	`          identifikasi = rs.getString("identifikasi");`
	`          if (identifikasi.equals("Narkotika")) {`
	`              target[j][0]=0;`
	`              target[j][1]=0;`
	`              target[j][2]=1;`
	`         }else if(identifikasi.equals("Psikotropika")){`
	`              target[j][0]=0;`
	`              target[j][1]=1;`
	`              target[j][2]=0;`
	`         }else{`
	`              target[j][0]=1;`
	`              target[j][1]=0;`
	`              target[j][2]=0;`
	`         }`
	`         j++;`
	`       }`
	`       rs.close();`
	`    } catch (Exception e) {`
	`        javax.swing.JOptionPane.showMessageDialog(null, "error : " + e.getMessage());`
	`    }`
	`    return target;`
	`}`

Nama Metode	latihJST()
Deskripsi	Proses pelatihan JST yang menghasilkan nilai parameter jaringan yaitu nilai Vij, Wjk, Vo, Wo)

Algoritma	`public void latihJST(){`
	`  double LR, maksMSE;`
	`  int epoh, JH, JO, jumPola, dimPola;`
	`  double[] row = new double[getJHneuron()];`
	`  double[] row2 = new double[getJOneuron()];`
	`  double[] error = new double[getJOneuron()];`
	
	`  DecimalFormat df = new DecimalFormat();`
	`  Random rand = new Random();`
	`  df.setMaximumFractionDigits(4);`
	
	`  jumPola = dp.getMaxPasien();`

```
dimPola = 23;
JO = getJOneuron();
JH = getJHneuron();
LR = getLearningRate();
epoh = getEpoch();
maksMSE = getMSE();

int[][] CP = new int[jumPola][dimPola];
int[][] CT = new int[jumPola][JO];
int[][] P = getPola();
int[][] T = getTarget();
double[][] W1 = new double[dimPola][JH];
double[][] W2 = new double[JH][JO];
double[][] B1 = new double[1][JH];
double[][] B2 = new double[1][JO];

W1 = acakW1();
W2 = acakW2();
B1 = acakB1();
B2 = acakB2();
double MSEepoch = maksMSE + 1;
double[] MSE = new double[epoh+1];
int ee = 1;

while ((ee <= epoh) && (MSEepoch > maksMSE)){
   MSEepoch = 0;
   for (int pp = 0; pp < jumPola; pp++){
       for (int i = 0; i < dimPola; i++){
           CP[pp][i] = P[pp][i];
       }

       for (int i = 0; i < JO; i++){
           CT[pp][i] = T[pp][i];
       }

       for (int ii = 0; ii < JH; ii++){
           double v=0.0;
           double a;

           for (int p = 0; p < dimPola; p++){
               v = v+(CP[pp][p]*W1[p][ii]);
           }
           v = v+B1[0][ii];
           a = Math.pow(2.7183, -v);
           a = 1/(1+a);
           a = Double.parseDouble(df.format(a));
           row[ii]=a;
       }
       double[][] Aa1 = {row};

       for (int jj = 0; jj < JO; jj++){
           double v = 0.0;
           double a;

           for (int p = 0; p < JH; p++){
               v = v+(Aa1[0][p]*W2[p][jj]);
           }
           v = v+B2[0][jj];
```

```
                a = Math.pow(2.7183, -v);
                a = 1/(1+a);
                a = Double.parseDouble(df.format(a));
                row2[jj]= a;
            }
            double[][] Aa2 = {row2};

            for (int i = 0; i < JO; i++){
                error[i] = CT[pp][i] - Aa2[0][i];
            }

            for (int kk=0; kk < JO; kk++){
               MSEepoch=MSEepoch+(error[kk]*error[kk]);
            }
//================================================

            double[] D2=new double[JO];
            for (int kk=0; kk<JO; kk++){
              D2[kk]=Aa2[0][kk]*(1-Aa2[0][kk])*error[kk];
            }

            double[][] dW2= new double[JH][JO];
            double[] delta2 = new double[JO];
            double[][] rowDW2 = new double[JH][JO];
            for (int jj = 0; jj < JH; jj++){
                for (int kk = 0; kk < JO; kk++){
                    delta2[kk] = LR*D2[kk]*Aa1[0][jj];
                    rowDW2[jj][kk]=delta2[kk];
                }
            }
            dW2=rowDW2;

            double[] rowDB2 = new double[JO];
            for (int kk = 0; kk < JO; kk++){
                rowDB2[kk] = LR*D2[kk];
            }
            double[][] dB2 ={rowDB2};

            double dD2[]=new double[JH];
            for (int i = 0; i < JH; i++){
                double a=0.0;
                for (int j = 0; j < JO; j++){
                    double a1=W2[i][j]*D2[j];
                    a=a+a1;
                }
                dD2[i]=a;
            }

            double[] D1=new double[JH];
            for (int jj = 0; jj < JH; jj++){
              double a=0.0;

              a=(1/(1+(Math.pow(2.7183, -(Aa1[0][jj]))))) *(1-
(1/(1+(Math.pow(2.7183, -Aa1[0][jj])))))*dD2[jj];

              D1[jj]=Double.parseDouble(df.format(a));
            }
```

```
            double[] delta1=new double[JH];
            double[][] rowDW1=new double[dimPola][JH];
            for (int ii = 0; ii < dimPola; ii++){
                for (int jj = 0; jj < JH; jj++){
                    delta1[jj] = LR*D1[jj]*CP[pp][ii];
                    rowDW1[ii][jj]=delta1[jj];
                }
            }
            double[][] dW1=rowDW1;

            double[] rowDB1 = new double[JH];
            for (int kk = 0; kk < JH; kk++){
                rowDB1[kk] = LR*D1[kk];
            }
            double[][] dB1 ={rowDB1};

            for (int ii = 0; ii < dimPola; ii++){
                for (int jj = 0; jj < JH; jj++){
                    W1[ii][jj]=W1[ii][jj]+dW1[ii][jj];
                }
            }
            this.setW1(W1);

            for (int ii = 0; ii < JH; ii++){
                for (int jj = 0; jj < JO; jj++){
                    W2[ii][jj]=W2[ii][jj]+dW2[ii][jj];
                }
            }
             this.setW2(W2);

            for (int ii = 0; ii < 1; ii++){
                for (int jj = 0; jj < JH; jj++){
                    B1[ii][jj]=B1[ii][jj]+dB1[ii][jj];
                }
            }
             this.setB1(B1);

            for (int ii = 0; ii < 1; ii++) {
                for (int jj = 0; jj < JO; jj++) {
                    B2[ii][jj]=B2[ii][jj]+dB2[ii][jj];
                }
            }
             this.setB2(B2);

        }
        double nilaiMSE=MSEepoch/jumPola;
        MSE[ee] = nilaiMSE;
        ee  = ee + 1;
    }
}
```

c. *Method Class* Pengujian

Tabel 3.27 Method Class Pengujian

Nama Metode	setDataUji()
Deskripsi	Mengatur nilai data pengujian (variabel x1 hingga x23)

Algoritma	`setDataUji(int x1, int x2,int x3,int x4,int x5,int x6,int x7,int x8,int x9,int x10,int x11,int x12,int x13,int x14,int x15,int x16,int x17,int x18,int x19,int x20,int x21,int x22,int x23){`
	`int[] dataUji ={x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,`
	`x11,x12,x13,x14,x15,x16,x17,x18,x19,`
	`x20,x21,x22,x23};`
	`testSet = dataUji;`
	`}`

Nama Metode	prosesUji()
Deskripsi	Melakukan proses pengujian data pecandu dan menampilkan hasil identifikasi

Algoritma	`public void prosesUji(int x1, int x2,int x3,int x4,int x5,int x6,int x7,int x8,int x9,int x10,int x11,int x12,int x13,int x14,int x15,int x16,int x17,int x18,int x19,int x20,int x21,int x22,int x23){`
	`int[] testSet = setDataUji(x1,x2,x3,x4,x5,x6,x7,`
	`x8,x9,x10,x11,x12,x13,x14,x15,x16,x17,x18,x19,x20,`
	`x21, x22,x23);`
	`int jumPola = 23;`
	`int[] CP = new int[23];`
	`int kelas[] = new int[3];`
	`String id = null, kd = null;`
	`double[] row = new double[5];`
	`double[] row2 = new double[3];`
	
	`pelatihan.latihJST();`
	`double W1[][] = null; W1=pelatihan.getW1();`
	`double W2[][] = null; W2= pelatihan.getW2();`
	`double B1[][] = null; B1= pelatihan.getB1();`
	`double B2[][] = null; B2= pelatihan.getB2();`
	`double mse= pelatihan.getMSE();`
	`int epoh= pelatihan.getEpoch();`
	`int JHneuron= pelatihan.getJHneuron();`
	`int JOneuron= pelatihan.getJOneuron();`
	
	`for (int ii = 0; ii < JHneuron; ii++) {`
	`double v=0.0;`
	`double a;`
	`for (int p = 0; p < jumPola; p++) {`
	`v = v+(testSet[p]*W1[p][ii]);`
	`}`
	`v = v+B1[0][ii];`
	`a = Math.pow(2.7183, -v);`
	`a = 1/(1+a);`
	`row[ii]=a;`
	`}`
	`double[][] Aa1 = {row};`
	`setA1(row);`
	
	`for (int jj = 0; jj < JOneuron; jj++) {`
	`double v = 0.0;`
	`double a;`

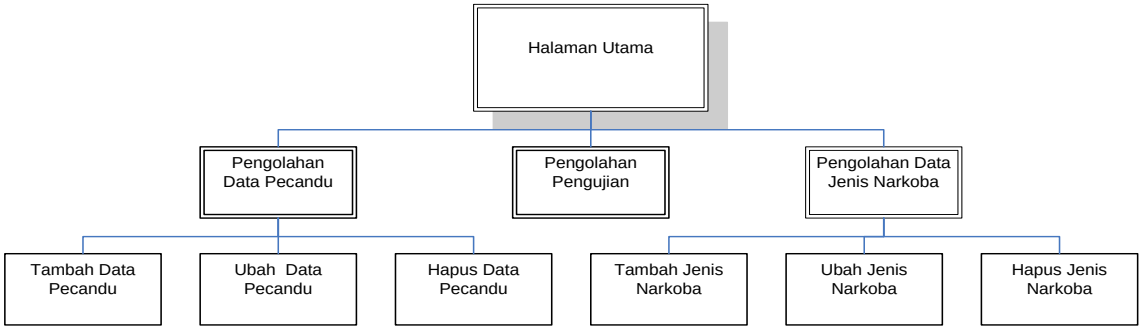
```
        for (int p = 0; p < JHneuron; p++) {
           v = v+(Aa1[0][p]*W2[p][jj]);
        }
        v = v+B2[0][jj];
        a = Math.pow(2.7183, -v);
        a = 1/(1+a);
        row2[jj]= a;
     }
     double[][] Aa2 = {row2};
     setA2(row2);

     String kls="";
     for (int j = 0; j < JOneuron; j++) {
        if (Aa2[0][j] < 0.5) {
            kelas[j] = 0;
        } else {
            kelas[j] = 1;
        }
        kls = kls+""+kelas[j];
     }

     if (kls.equals("001")){
        id = "Narkotika"; kd = "001";
     } else if (kls.equals("010")){
        id = "Psikotropika"; kd = "010";
     } else if(kls.equals("100")){
        id = "Zat Adiktif Lain"; kd = "100";
     } else {
        id = "Tidak Teridentifikasi"; kd = kls;
     }
     setHasil(""+id);
     setKode(kd);
}
```

3.2.3 Perancangan Struktur Menu

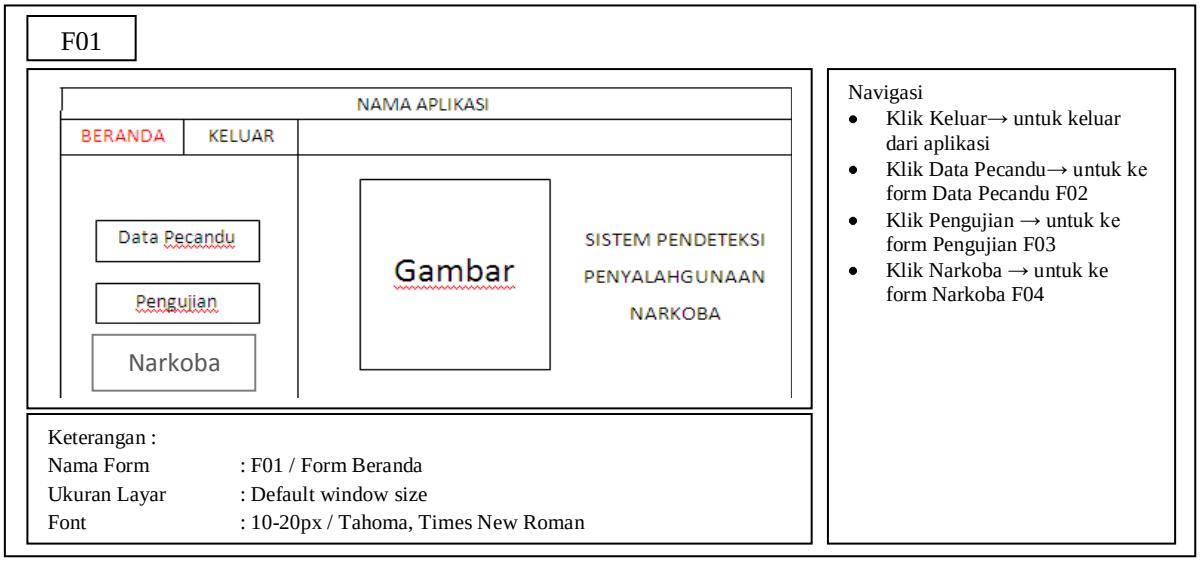
Arsitektur struktur menu merupakan gambaran perancangan menu yang akan dimiliki oleh sistem. Adapun perancangan struktur menu yang terdapat dalam Sistem Pendeteksi Penyalahgunaan Narkoba dengan Algoritma JST Backpropagation adalah sebagai berikut.



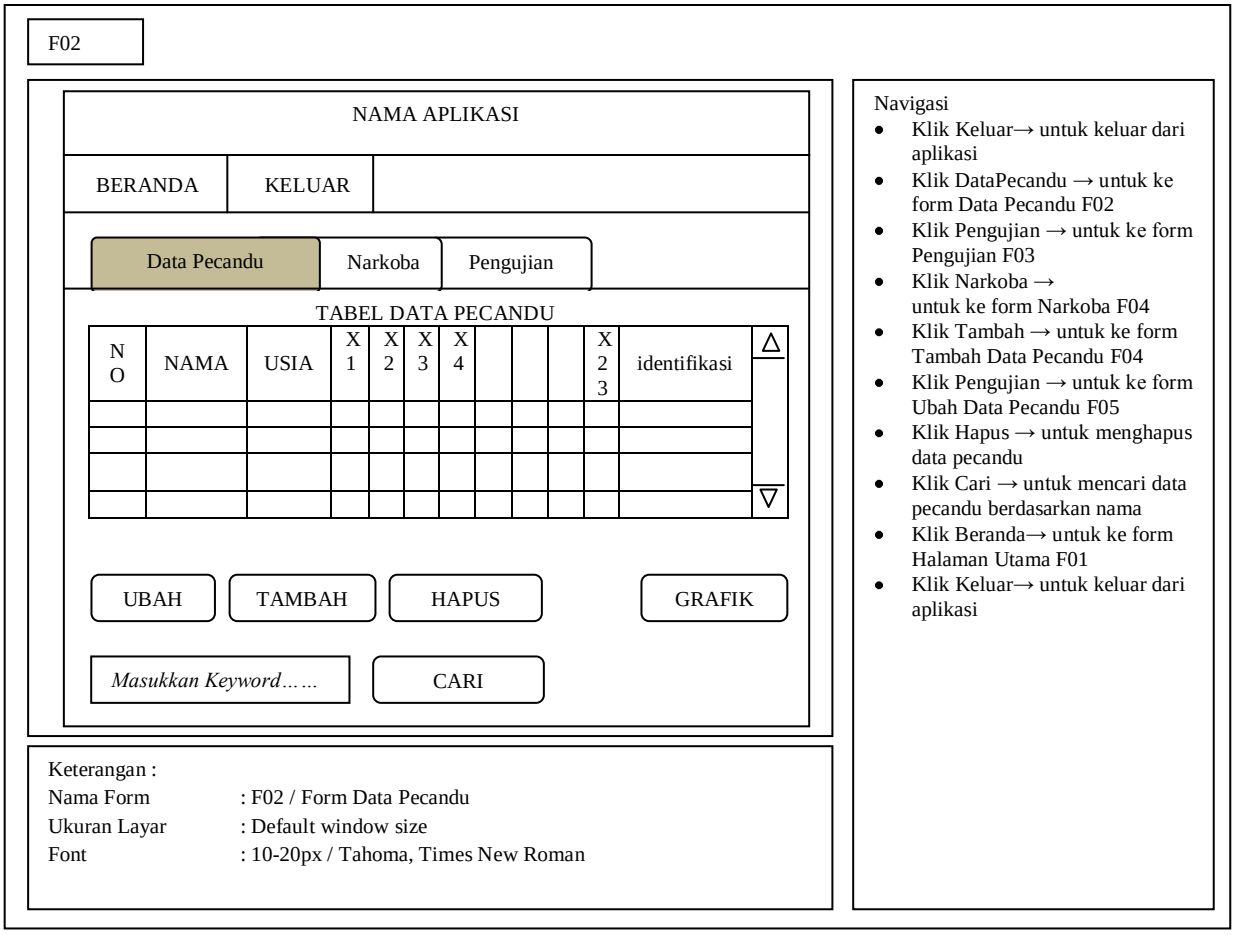
Gambar 3.22 Struktur Menu

3.2.4 Perancangan Antarmuka

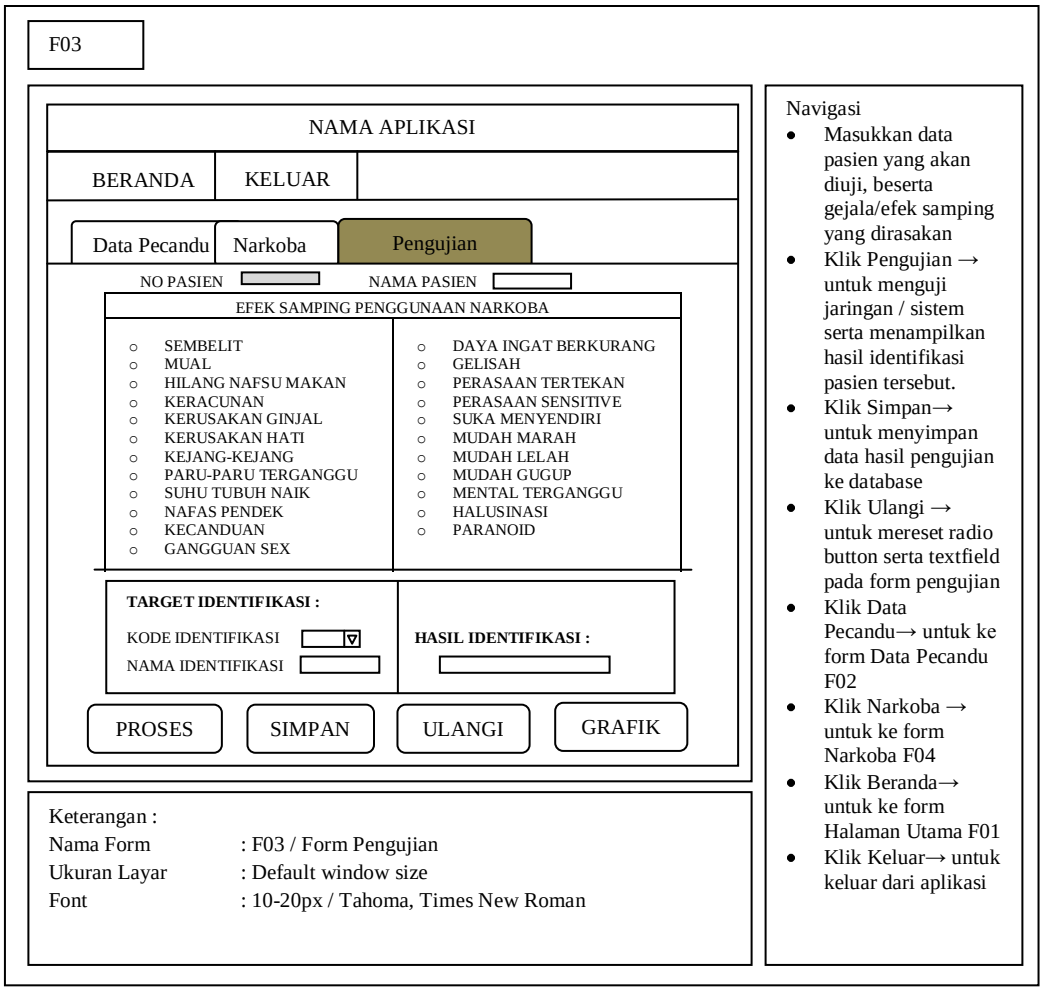
Pada tahap ini dirancang desain tampilan dari sistem pendeteksi penyalahgunaan narkoba yang akan dibuat yang berperan sebagai media komunikasi yang digunakan sebagai sarana berdialog antara program dan *user*. Berikut adalah sketsa dari aplikasi yang akan dibangun.



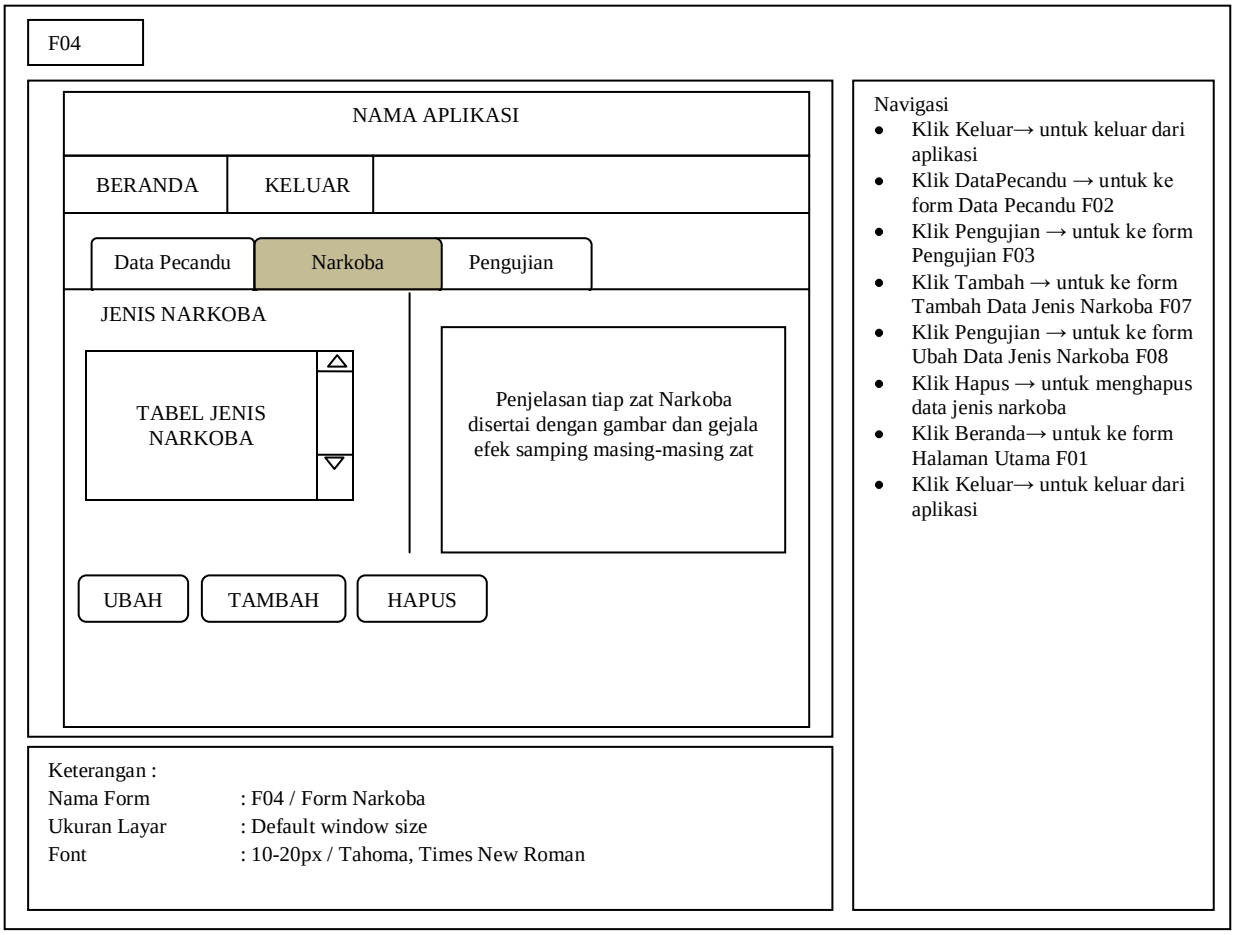
Gambar 3.23 Halaman utama / Beranda Sistem Pendeteksi Penyalahgunaan Narkoba



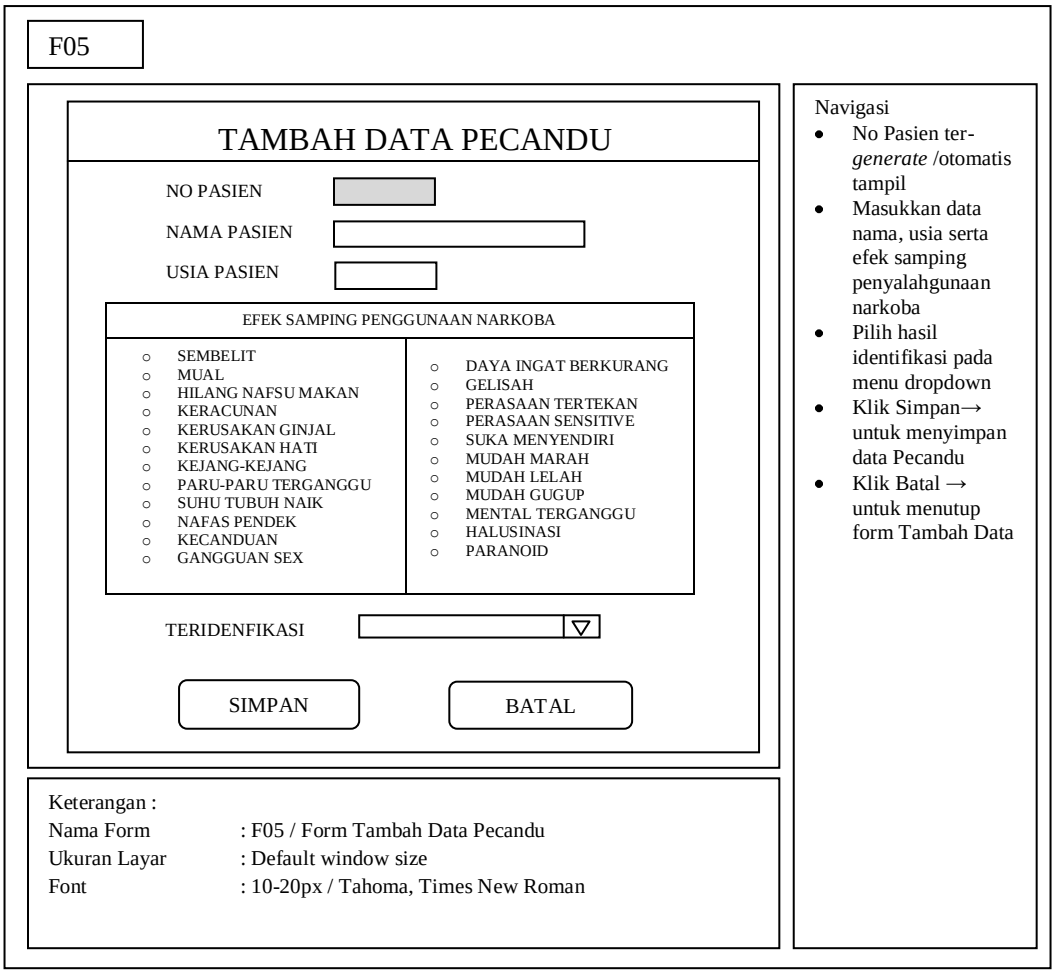
Gambar 3.24 Halaman Data Pecandu pada Sistem Pendeteksi Penyalahgunaan Narkoba



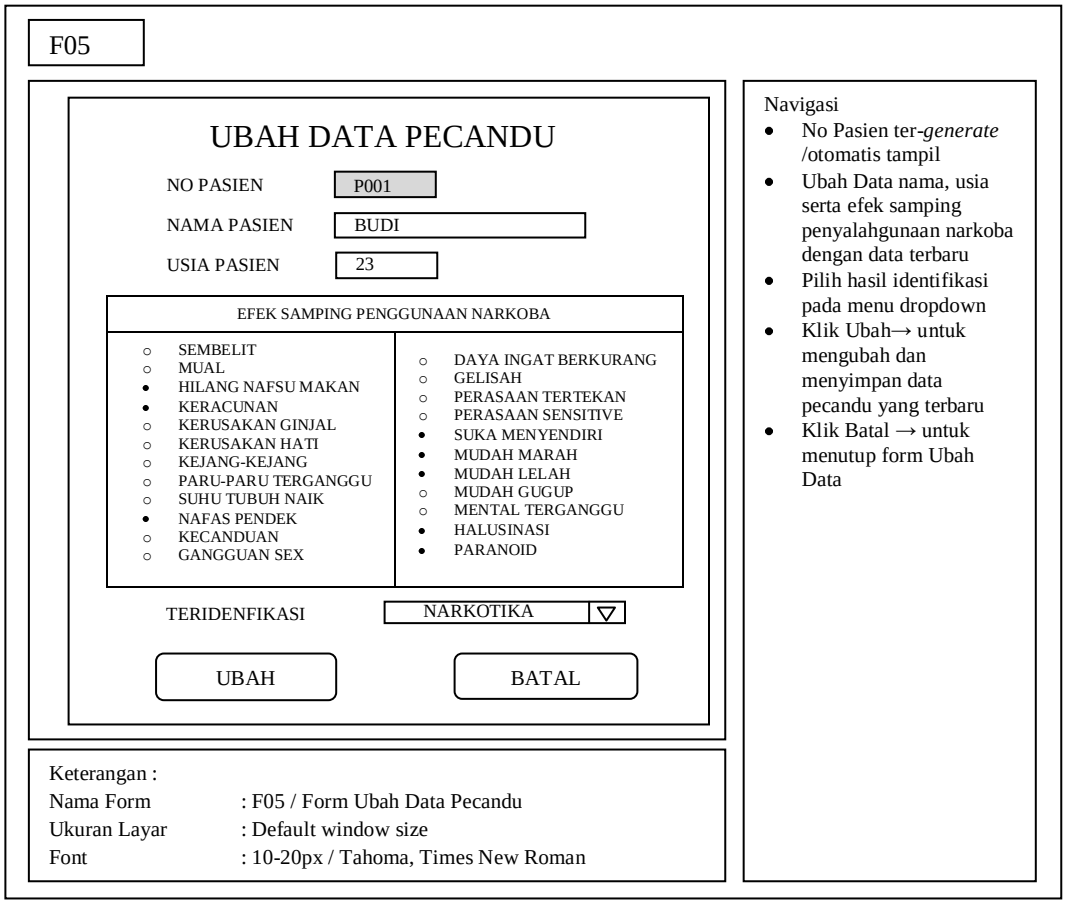
Gambar 3.25 Halaman Pengujian pada Sistem Pendeteksi Penyalahgunaan Narkoba



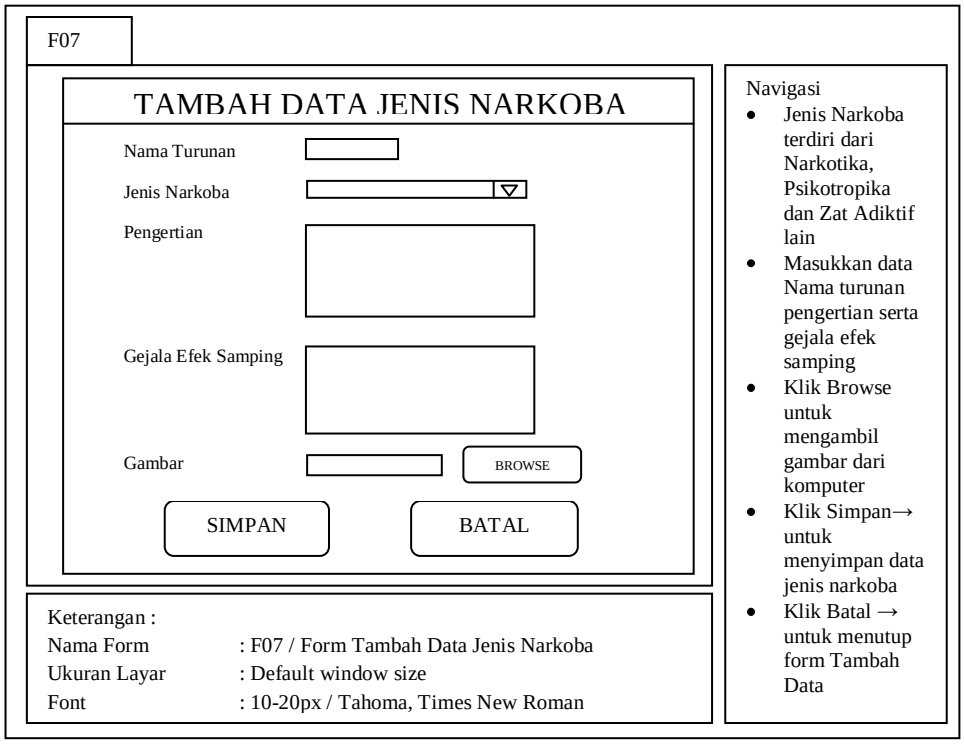
Gambar 3.26 Halaman Narkoba pada Sistem Pendeteksi Penyalahgunaan Narkoba



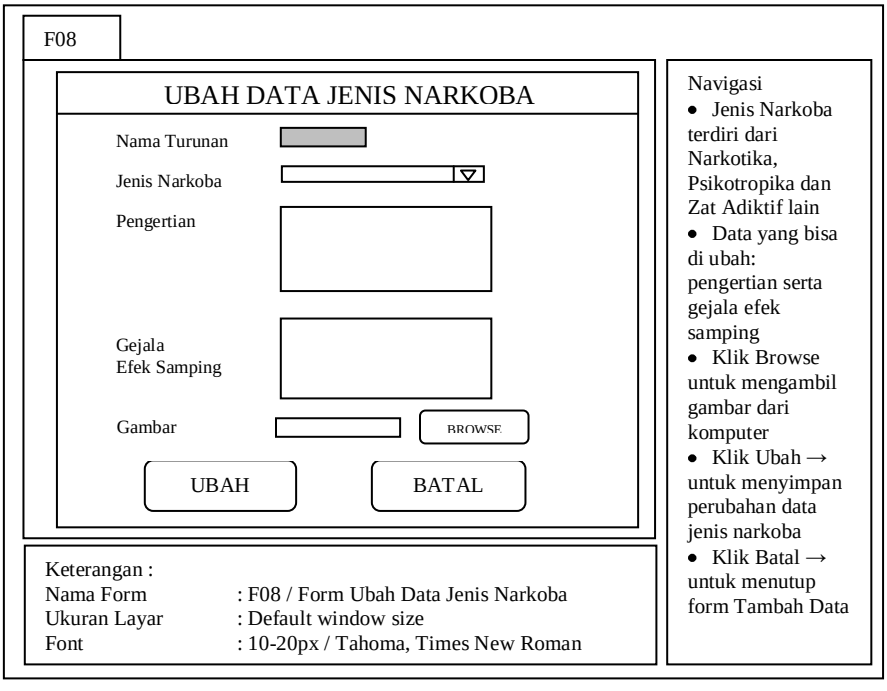
Gambar 3.27 Halaman Tambah Data Pecandu pada Sistem Pendeteksi Penyalahgunaan Narkoba



Gambar 3.28 Halaman Ubah Data Pecandu pada Sistem Pendeteksi Penyalahgunaan Narkoba



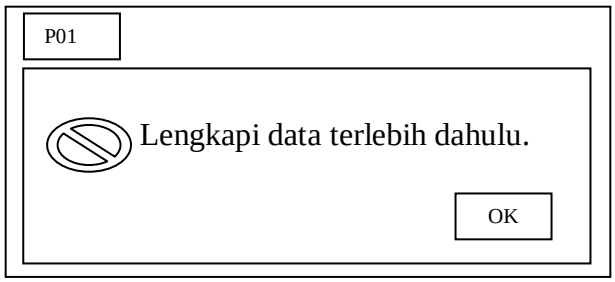
Gambar 3.29 Halaman Tambah Data Jenis Narkoba pada Sistem Pendeteksi Penyalahgunaan Narkoba



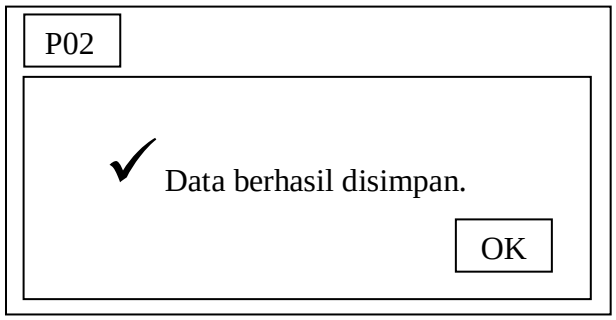
Gambar 3.30 Halaman Ubah Data Jenis Narkoba pada Sistem Pendeteksi Penyalahgunaan Narkoba

3.2.5 Perancangan Pesan

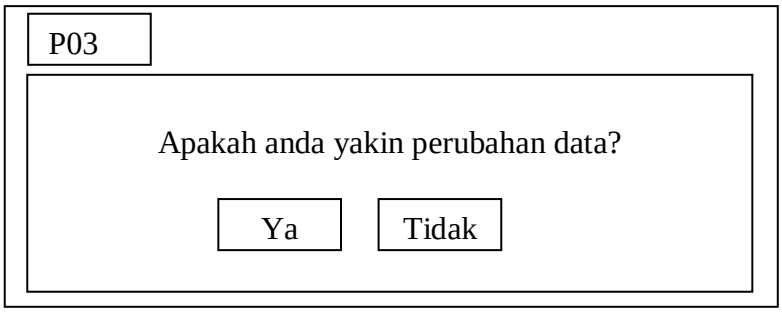
Perancangan pesan adalah rancangan antarmuka dari pesan yang akan ditampilkan apabila terjadi kesalahan dan pemberitahuan dalam proses yang dilakukan. Berikut adalah perancangan pesan yang ada pada Sistem Pendeteksi Penyalahgunaan Narkoba.



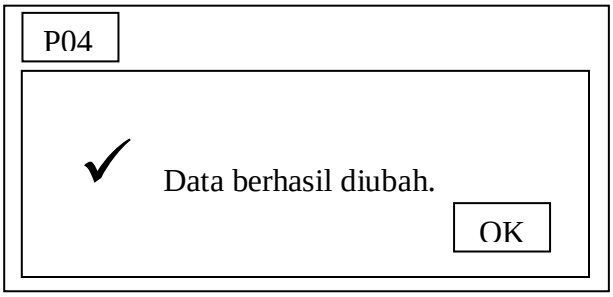
Gambar 3.31 Pesan Kesalahan Pengisian Data Belum Lengkap



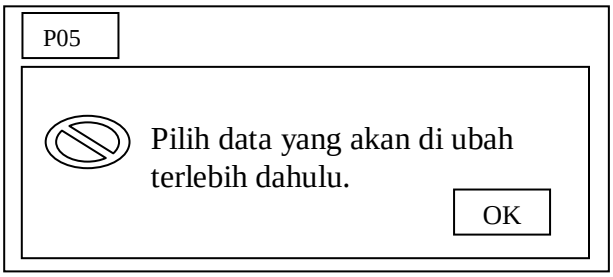
Gambar 3.32 Pesan Konfirmasi Data Berhasil Disimpan



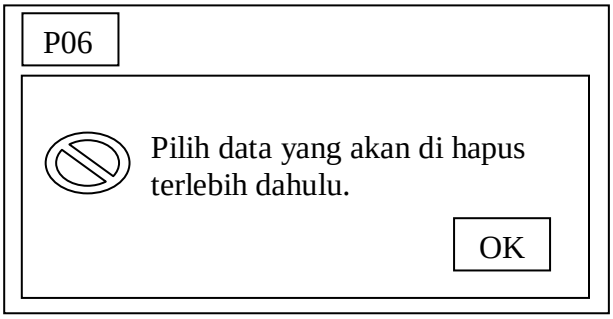
Gambar 3.33 Pesan Konfirmasi Perubahan Data



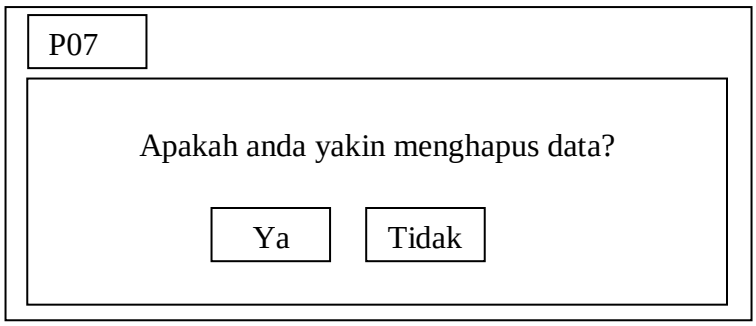
Gambar 3.34 Pesan Konfirmasi Data Berhasil Diubah



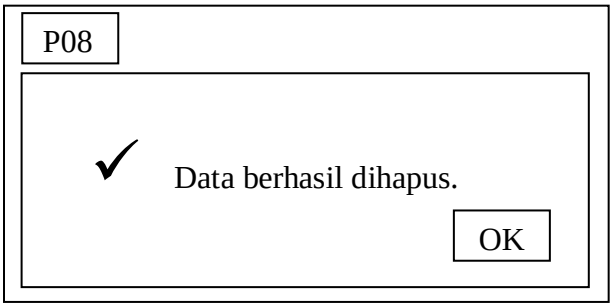
Gambar 3.35 Pesan Kesalahan Tombol Ubah Data



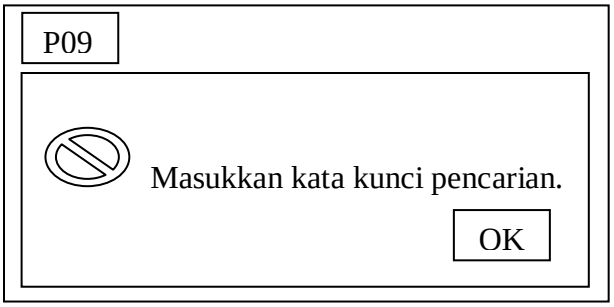
Gambar 3.36 Pesan Kesalahan Tombol Hapus Data



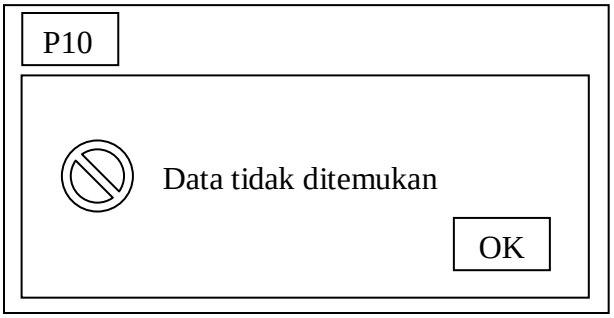
Gambar 3.37 Pesan Konfirmasi Hapus Data



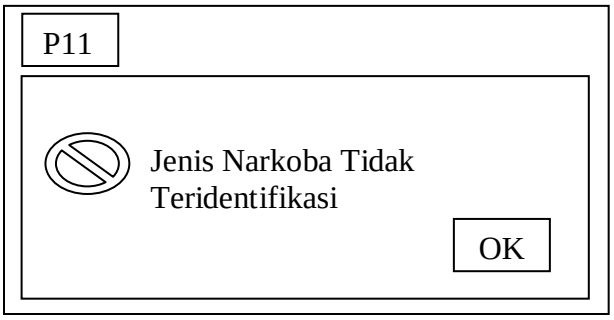
Gambar 3.38 Pesan Konfirmasi Data Berhasil Dihapus



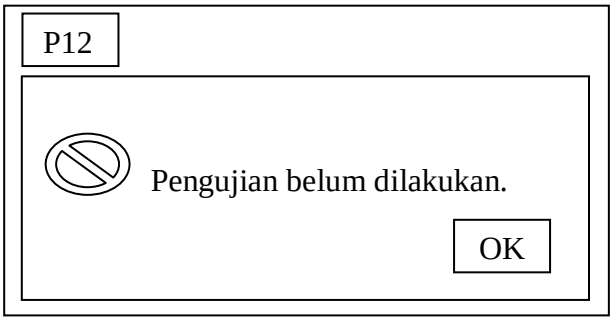
Gambar 3.39 Pesan Kesalahan Tombol Cari Data



Gambar 3.40 Pesan Kesalahan Pencarian Data Tidak Ditemukan



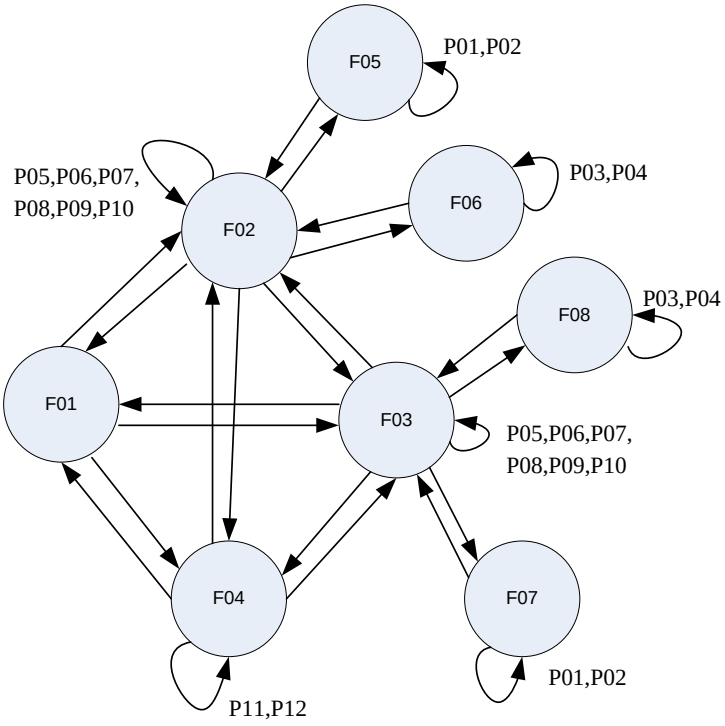
Gambar 3.41 Pesan Kesalahan Pengujian untuk Jenis Narkoba Tidak Teridentifikasi



Gambar 3.42 Pesan Kesalahan Tombol Simpan Pengujian

3.2.6 Jaringan Semantik

Jaringan semantik menggambarkan keterhubungan navigasi menu dari satu halaman ke halaman lainnya. Jaringan semantik untuk sistem pendeteksi penyalahgunaan narkoba adalah sebagai berikut.



Gambar 3.43 Jaringan Semantik Sistem Pendeteksi Penyalahgunaan Narkoba