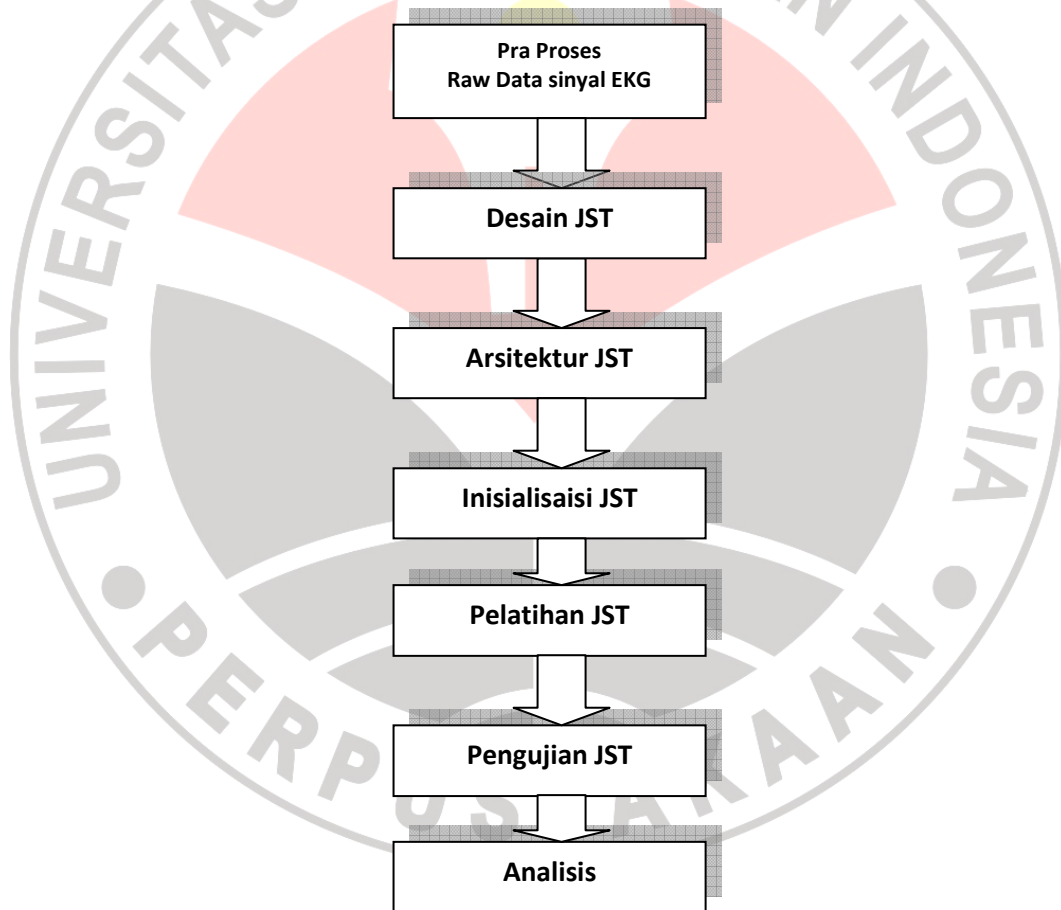


BAB III

METODE PENELITIAN

Pada penelitian ini, akan dilakukan beberapa langkah untuk mencapai penelitian tersebut. Langkah – langkah tersebut dapat digambarkan melalui diagram alir sebagai berikut.



Gambar 3.1. Diagram Alir penelitian

Penelitian ini akan dimulai dengan pengumpulan *raw* data sinyal pada EKG. Data tersebut merupakan data statistik dari hasil pemrosesan sinyal yang

berdasarkan pada parameter sinyal pada EKG dan jenis keadaan jantung. Data sinyal pada EKG tersebut kemudian akan dipisahkan kedalam dua bagian, yaitu data untuk pelatihan dan data untuk pengujian.

Tahap selanjutnya adalah penentuan desain jaringan. Pada tahap ini akan ditentukan defisiensi masalah yaitu penentuan pola masukan dan keluaran untuk keperluan pelatihan dan pengujian pada Jaringan Saraf Tiruan (JST). Tahap ini kemudian diikuti dengan penentuan algoritma pelatihan.

Tahap berikutnya adalah penentuan arsitektur jaringan berdasarkan desain jaringan yang telah didefinisikan. Setelah itu, dilakukan inisialisasi terhadap jaringan yang akan dilatih dan diuji.

Selanjutnya adalah tahap pelatihan terhadap data yang telah diklasifikasikan dan ditentukan arsitekturnya. Tujuan dari pelatihan ini adalah menentukan nilai SSE (*Sum square error*), nilai epoch, jumlah neuron layer tersembunyi dan konstanta belajar. Metode pelatihan JST yang digunakan dalam penelitian ini adalah *Backpropagation*. Algoritma pelatihan ini dirancang untuk memperkecil MSE (*mean square error*) antara input sebenarnya output yang diinginkan.

Setelah dilakukan tahap pelatihan adalah tahap pengujian terhadap data pengujian. Data pengujian diambil dari tinjauan data EKG dalam bentuk yang telah ditentukan. Tujuan dari pengujian ini adalah mengetahui apakah JST mampu mengklasifikasikan pola sinyal pada EKG dan menghasilkan acuan untuk keperluan diagnosis penyakit jantung.

Perangkat lunak yang digunakan dalam perancangan sistem pada penelitian ini adalah dengan menggunakan *MATLAB V7.0.4*, karena memiliki bahasa tingkat tinggi dan dapat digunakan untuk komputasi teknik, penghitungan, visualisasi dan pemrograman. Selain itu juga memiliki *neural network (NN) toolbox*, sehingga memudahkan dalam perancangan program JST maupun pensimulasian dari sistem yang telah dilatih. Beberapa kegunaan lain dari *MATLAB* di antaranya adalah untuk pengembangan algoritma, pemodelan, simulasi dan pembuatan antarmuka GUI (*Graphical User Interface*).

3.1 Raw Data Sinyal Elektrik Jantung pada EKG

Data sinyal EKG akan dibedakan kedalam 2 kelompok data, yaitu data untuk pelatihan dan data untuk pengujian. Data ini diperoleh dari situs <http://www.physionet.org>. Situs ini merupakan situs resmi yang menyediakan data-data EKG dari seluruh penjuru dunia yang dapat diakses secara gratis dan diperuntukan untuk penelitian-penelitian seputar dunia medis, terutama penelitian seputar EKG.

Data yang digunakan dalam penelitian ini adalah sebanyak 170 sampel yang kemudian dibagi menjadi 3 yaitu 70% atau 119 sampel, 50 % atau 85 sampel dan 30 % sampel untuk pelatihan JST dan data untuk pengujian merupakan sisa dari proses pelatihan. Data selengkapnya dapat dilihat pada lampiran I.

3.2 Desain Jaringan

Pada perancangan desain jaringan dilakukan penentuan pola masukan dan pola keluaran. Pola keluaran ditentukan berdasarkan parameter sinyal pada EKG sedangkan pola keluaran berdasarkan jenis kondisi jantung.

Yang tergolong sebagai data masukan adalah merupakan parameter gelombang EKG yang terdiri dari 11 parameter, yang dijelaskan secara berurutan sebagai berikut.

1. P_{amp} adalah amplitudo dari gelombang P, yang merupakan proses depolarisasi pada otot-otot arial jantung. P_{amp} ini memiliki satuan mV . Gelombang P yang normal biasanya sebesar 0.25.
2. P_{width} merupakan lebar dari gelombang P dalam satuan sekon pada ketinggian yang sedang.
3. PR_{int} merupakan interval antara gelombang P dengan gelombang R dalam satuan sekon. Interval PR diukur dari awal gelombang P sampai awal kompleks QRS, yang biasanya pada kondisi normal memiliki durasi 0.12-0.20 detik. Pada interval ini berhubungan dengan 3 sampai 5 kotak kecil pada kertas perekam. Jika durasi interval PR < 0.12 detik atau $>$ dari 0.20 detik maka dikatakan jantung dalam keadaan abnormal.
4. Q_{amp} adalah amplitudo gelombang Q dalam satuan mV .
5. R_{amp} adalah amplitudo gelombang R dalam satuan mV .

6. S_{amp} adalah amplitudo gelombang S dalam satuan mV .
7. R_{width} merupakan lebar dari gelombang R dalam satuan sekon.
8. RR_{int} merupakan interval antara puncak gelombang R dalam satuan sekon. Interval PR diukur dari awal gelombang P sampai awal kompleks QRS, yang biasanya memiliki durasi 120-200 ms. Pada interval ini berhubungan dengan 3 sampai 5 kotak kecil pada kertas perekam. Interval RR juga akan menunjukkan besarnya *Heart beat* (detak jantung) yang dapat diperoleh dengan menggunakan persamaan.

$$HR = \frac{60}{RR_{int}}$$

9. T_{amp} adalah amplitudo gelombang T dalam satuan mV . Gelombang T menggambarkan repolarisasi atau kembalinya ventrikel. Pada sebagian besar sadapan, gelombang T menunjukkan positif. Namun, gelombang T negatif normal di sadapan aVR.
10. T_{width} merupakan lebar merupakan lebar dari gelombang T dalam satuan sekon.
11. RT_{int} merupakan interval antara gelombang P dengan gelombang R dalam satuan sekon.

Parameter yang diperlukan sebagai pola keluaran adalah dengan mengambil variabel dari kondisi jantung, yaitu meliputi kondisi normal, *Atrial Tachycardia*, *Premature Ventricular Complex (PVC)*, *Atrial Flutter*, *Atrial*

Fibrillation, Junction Tachycardia, Ventricular Tachycardia dan *Ventricular Fibrillation*. Secara lengkap dijelaskan sebagai berikut.

1. Keadaan Normal

Ritme keadaan normal merupakan ritme jantung dalam kondisi sehat. Pada kondisi ini dapat dengan mudah dilakukan diagnosis dengan 3 defleksi P-QRS-T. Ritme ini memiliki frekuensi antara 60 dan 100 BPM (*Beat Per Minutes*). Grafik pada EKG pada kondisi ini ditunjukkan pada gambar 3.2.



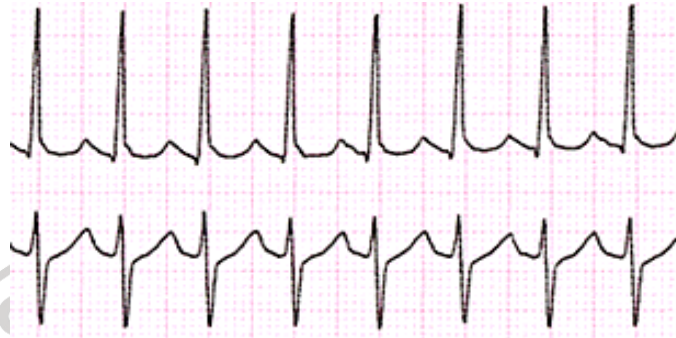
Gambar 3.2. Sinyal EKG Keadaan normal

Dan memiliki karakteristik sebagai berikut: Gelombang P = tegak lurus, interval P-R = 0.12-0.20 detik, Interval QRS = 0.04-0.10 detik. Nilai keluaran yang diberikan adalah 1 0 0 0 0 0 0.

2. Atrial Tachycardia

Karakteristik dari *Atrial Flutter* adalah memiliki frekuensi detak sebesar 160-240 BPM. Pada kondisi ini kompleks QRS terlihat normal, tetapi kadang-kadang melebar. Penyebab dari jenis gangguan jantung ini adalah dikerenakan

oleh kelebihan mengkonsumsi rokok dan kafein dan kekurangan kalsium. Nilai keluaran yang diberikan adalah 0 1 0 0 0 0 0.



Gambar 3.3. Sinyal EKG *Atrial Tachycardia*

3. **Atrial Flutter**

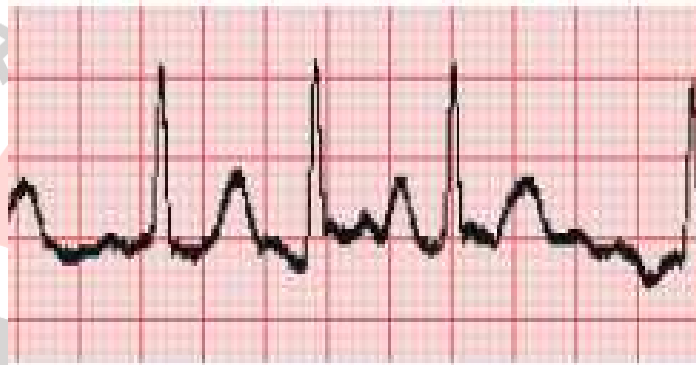
Karakteristik dari *Atrial Flutter* adalah memiliki frekuensi detak sebesar 250-300 BPM, Gelombang P = tidak teridentifikasi, Interval QRS = 0.04-0.10 detik. Penyebab dari timbulnya ritme ini adalah dapat disebabkan oleh keracunan obat, *Hypoxia* atau kekurangan Oksigen (O_2) dan penyebab yang paling sering terjadi adalah penyempitan jantung. Nilai output yang diberikan adalah 0 0 1 0 0 0 0. Grafik pada EKG pada kondisi ini ditunjukkan pada gambar 3.4 berikut.



Gambar 3.4 . Sinyal EKG *Atrial Flutter*

4. *Atrial Fibrillation*

Grafik pada EKG pada kondisi ini ditunjukkan pada gambar 3.5. Karakteristik *Atrial Fibrillation* adalah memiliki frekuensi detak jantung sebesar 400-700 BPM, Gelombang P = tidak teridentifikasi dan tidak berurutan secara teratur. penyebab dari ritme ini adalah disebabkan oleh *Hypoxia*, penyebaran penyakit jantung, dll. Nilai keluaran yang diberikan adalah 0 0 0 1 0 0 0 0.



Gambar 3.5. Sinyal EKG *Atrial Fibrillation*

5. *Junctional Tachycardia*

Karakteristik dari *Junctional Tachycardia* adalah memiliki frekuensi detak sebesar 101-200 BPM, Gelombang P = terbalik sebelum dan sesudah QRS, interval P-R < 0.12 detik ketika gelombang P terbalik dengan QRS, Interval QRS = 0.04-0.10 detik. Penyebab dari ritme ini adalah Kelebihan dalam mengkonsumsi kafein, merokok dan alkohol dan *hypoklemia* (kekurangan kalsium). Nilai keluaran yang diberikan adalah 0 0 0 0 1 0 0 0. Grafik pada EKG pada kondisi ini ditunjukkan pada gambar 3.6 berikut.



Gambar 3.6. Sinyal EKG *Junctional Tachycardia*

6. Premature Ventricular Complex (PVC)

Pada keadaan ini irama jantung tidak beraturan dan akan terjadi detak jantung yang tidak diharapkan. Kompleks QRS akan terlihat sangat lebar, hal ini disebabkan karena selama kondisi ini depolarisasi pada ventrikel tidak secara bersamaan. Penyebab dari gangguan ini adalah karena stress, ketidakseimbangan elektrolit dan lemak yang berlebihan. Nilai keluaran yang diberikan adalah 0 0 0 0 0 1 0 0.

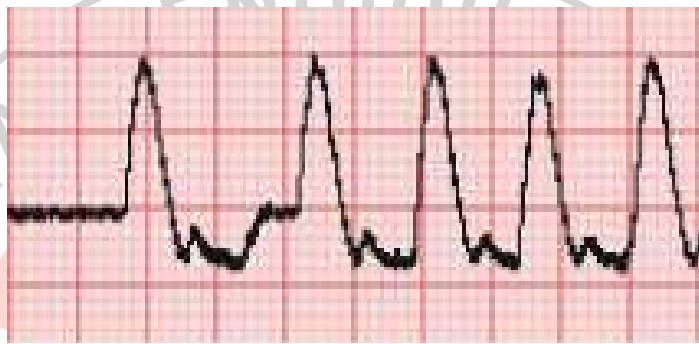


Gambar 3.7. Sinyal EKG *Premature Ventricular Complex (PVC)*

7. Ventricular Tachycardia

Karakteristik dari *Ventricular Tachycardia* adalah memiliki frekuensi detak sebesar 100-250 BPM. Terdapat tiga deret atau lebih lebar kompleks QRS.

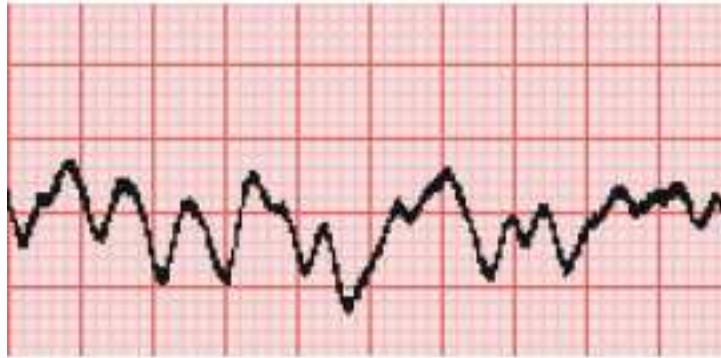
Ditemukan *ventricular ischemia* (gejala jantung koroner). Penyebab dari ritme ini adalah kondisi stress, pengaruh nikotin, alkohol dan cafein. Disebabkan juga kekurangan O₂ dan keseimbangan elektrolit. Nilai keluaran yang diberikan adalah 0 0 0 0 0 0 1 0. Grafik pada EKG pada kondisi ini ditunjukkan pada gambar 3.8 berikut.



Gambar 3.8. Sinyal EKG *Ventricular Tachycardia*

8. *Ventricular fibrillation*

Karakteristik dari *Ventricular Tachycardia* adalah memiliki frekuensi detak sebesar 100-250 BPM. Kondisi ritme dalam kondisi parah dan kacau. Ini adalah gejala dari kematian. Penyebab dari ritme ini adalah ketidakseimbangan elektrolit dan kejutan elektrik jantung. Nilai keluaran yang diberikan adalah 0 0 0 0 0 0 0 1. Grafik pada EKG pada kondisi ini ditunjukkan pada gambar 3.9 berikut.



Gambar 3.9. Sinyal EKG *Ventricular Fibrillation*

Untuk melakukan klasifikasi JST maka pola keluaran tersebut harus diubah kedalam bentuk biner. Dalam penelitian ini, pola keluaran dibagi menjadi 2 jenis pola, yaitu pola dengan 8 target keluaran dan Pola-pola ini kemudian akan dilatih sehingga jaringan dapat mengenali pola-pola tersebut. jenis pola tersebut dapat dilihat pada tabel 3.2 sebagai berikut.

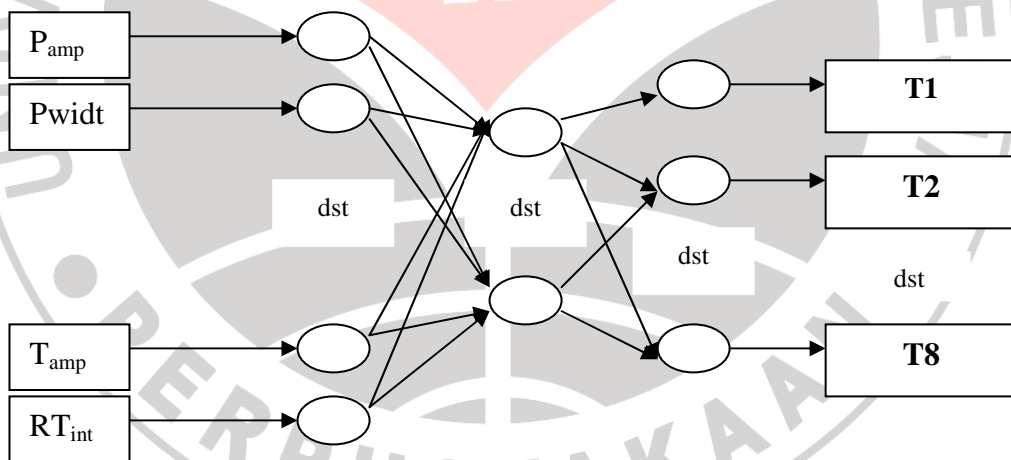
Tabel 3.2. Pola 8 Vektor keluaran

No	Klasifikasi data	File	Data keluaran biner
1	Kondisi normal	KN.m	1 0 0 0 0 0 0 0
2	<i>Atrial Tachycardia</i>	AT.m	0 1 0 0 0 0 0 0
3	<i>Atrial Flutter</i>	AFI.m	0 0 1 0 0 0 0 0
4	<i>Atrial Fibrillation</i>	AF.m	0 0 0 1 0 0 0 0
5	<i>Junction Tachycardia</i>	JT.m	0 0 0 0 1 0 0 0
6	<i>Premature Ventricular Complex (PVC)</i>	JT.m	0 0 0 0 0 1 0 0
7	<i>Ventricular Tachycardia</i>	VT.m	0 0 0 0 0 0 1 0
8	<i>Ventricular fibrillation</i>	VF.m	0 0 0 0 0 0 0 1

3.3 Arsitektur Jaringan

Pada penelitian ini akan digunakan arsitektur jaringan yang dapat dilihat pada tabel 3.4. Arsitektur JST terdiri dari 3 parameter jaringan dengan 11 keluaran, 8 keluaran yang didefinisikan sebagai JST_1, JST_2 dan JST_3.

Pada arsitektur ini akan digunakan fungsi transfer *logsig*. Hal ini dikarenakan *range* keluaran adalah 0 dan 1 yang sangat baik untuk pelatihan dengan vektor biner. Pola masukan didasarkan pada parameter sinyal pada EKG dan 8 neuron dalam layer keluaran adalah untuk klasifikasi 8 kelas kondisi jantung. Adapun arsitektur JST dalam penelitian kali ini dapat dilihat pada gambar 3.10 sebagai berikut.



Gambar 3.10. Arsitektur JST

Vektor target terdiri dari 8 elemen bit dan jaringan akan merespon nilai bit 1 atau 0 pada perbedaan posisi kelas data yang diperlihatkan pada jaringan. Setiap nilai keluaran ditentukan oleh fungsi transfer. Sebagai contoh, keadaan jantung normal direpresentasikan oleh 1 pada elemen pertama dan 0 pada elemen lainnya.

Jika kondisi jantung normal, maka vektor keluaran akan menunjukkan [1 0 0 0 0 0 0 0].

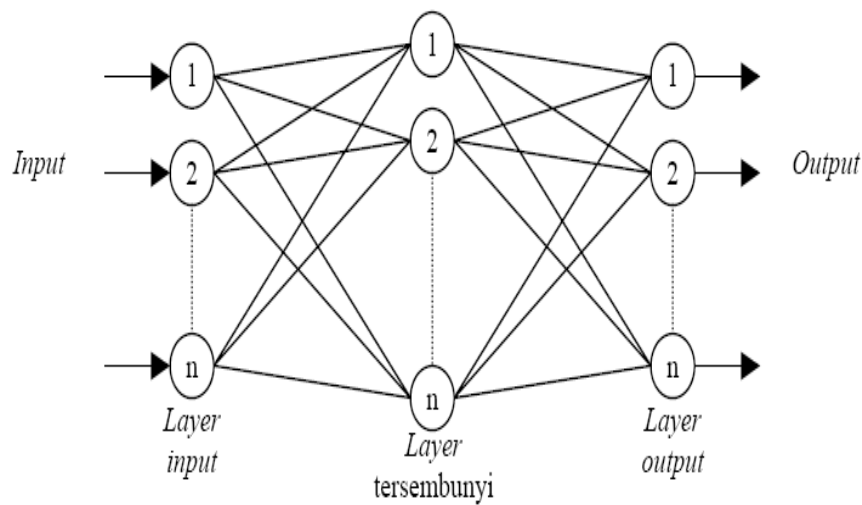
Dalam penelitian ini akan dikembangkan 3 arsitektur berdasarkan pada pembagian data pelatihan jaringan. Data pelatihan untuk penelitian ini diambil sebanyak 170 sampel data, yang kemudian dibagi kedalam 3 jenis, yaitu arsitektur JST_1 dengan 70 % dari jumlah data maka terdapat 119 neuron *input*, arsitektur JST_2 dari 50 % maka terdapat 85 neuron *input* dari jumlah data dan 30 % dari jumlah data. Untuk selengkapnya ketiga data arsitektur jaringan tersebut dapat dilihat pada tabel 3.4 berikut.

Tabel 3.4. Data JST

No.	Jaringan	Data Latih	Neuron Input	Neuron Hidden	Konstanta Belajar	Neuron Output
1.	JST_1	119	11	25 - 400	0.005-0.010	8
2.	JST_2	85	11	10 - 500	0.003-0.010	8
3.	JST_3	51	11	50 - 825	0.001-0.010	8

3.4 Inisialisasi Jaringan

Inisialisai jaringan adalah pengenalan gambaran hubungan antara lapisan yang digunakan dalam proses pembelajaran jaringan. Setiap unit sel pada satu lapisan dihubungkan secara penuh terhadap sel-sel unit pada lapisan didepannya sehingga akan ditemukan bobot dan bias dari hubungan antara lapisan tersebut.



Gambar 3.11. Arsitek Jaringan Saraf tiruan (*Pola dan Schowengerdt, 1997*)

Gambar 3.10 menunjukkan bahwa jaringan terdiri atas 3 lapisan, yaitu lapisan masukan, lapisan tersembunyi dan lapisan keluaran. Lapisan masukan terdiri dari 8 unit sel saraf yang merupakan variabel-variabel dari amplitudo dan interval dari gelombang sinyal pada EKG. Lapisan tersembunyi dengan jumlah unit sel saraf yang berubah-ubah sesuai dengan kecepatan konvergensi dan akan terdapat 8 sel unit saraf keluaran.

Model jaringan pada penelitian ini adalah dengan menggunakan metode *backpropagation* untuk menganalisis dan mendiagnosis kondisi jantung berdasarkan sinyal pada EKG. JST-BP memiliki beberapa layer yang mana layer terhubung secara keseluruhan terhadap layer yang lainnya. Saat jaringan akan memberikan masukan, maka saat itu juga akan diaktifkan nilai *forward* yang berasal dari input *layer* pada unit-unit yang terproses. Kemudian masing-masing layer internal diberikan kepada *layer* keluaran yang terproses oleh unit-unit

keluaran. Unit-unit dari keluaran akan memberikan respons terhadap jaringan. Bilamana jaringan terdapat koreksi-koreksi pada parameter didalamnya, mekanisme perbaikan akan dimulai dari output unit dan *Error Back Propagation* kemudian akan kembali ke masing-masing unit internal untuk dipakai pada input layer.

Algoritma Pembelajaran Jaringan saraf tiruan (JST) dengan menggunakan metode *backpropagation* adalah sebagai berikut :

Langkah 0: Inisialisasi bobot

Langkah 1: Jika kondisi henti tidak terpenuhi, lakukan langkah 2 – 9

Langkah 2: Untuk setiap pola input, lakukan step 3 – 8

Feedforward (perambatan Maju):

Pembelajaran pada layer 1

Langkah 3: Setiap neuron masukan (X_i $i= 1...n$) menerima sinyal masukan x_i dan menyebarkannya ke semua neuron pada lapis tersembunyi.

Langkah 4: Setiap neuron pada lapis tersembunyi-1 (Z_j , $j=1...p$) menjumlahkan bobot-bobot sinyal masukan,

$$z_in_j = \sum_{i=1}^n x_i v_{ij} \quad (3.1)$$

dan menerapkannya pada fungsi aktivasi untuk menghitung sinyal keluarannya,

$$z_j = f(z_in_j) \quad (3.2)$$

dan mengirim sinyal ini ke semua neuron pada lapis keluaran

Setiap neuron pada lapis tersembunyi-2 ($Z'_j, j=1 \dots m$) menjumlahkan bobot-bobot sinyal masukan,

$$z'_{in_j} = \sum_{i=1}^n x_i v_{ij} \quad (3.3)$$

dan menerapkannya pada fungsi aktivasi untuk menghitung sinyal keluarannya,

$$z'_j = f(z_{in_j}) \quad (3.4)$$

dan mengirim sinyal ini ke semua neuron pada lapis keluaran

Langkah 5: Setiap neuron keluaran ($Y_k, k=1 \dots o$) menjumlahkan bobot sinyal yang masuk,

$$y_{in_k} = \sum_{j=1}^p z'_j w_{jk} \quad (3.5)$$

dan mengaplikasikan fungsi aktivasinya untuk menghitung sinyal yang akan dikeluarkannya

$$y_k = f(y_{in_k}) \quad (3.6)$$

Backpropagasi error

Langkah 6: Setiap neuron keluaran ($Y_k, k=1 \dots o$) menerima sebuah pola target yang berhubungan dengan pola masukan pelatihan dan menghitung kesalahan informasi

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (3.7)$$

menghitung koreksi bobot (yang digunakan untuk memperbaiki w_{jk} nanti),

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (3.8)$$

dan mengirim δ_k ke neuron lapis tersembunyi.

Langkah 7: Setiap neuron tersembunyi-2 ($Z'_j, j=1 \dots m$) menjumlahkan bobot setiap neuron yang telah dikali dengan kesalahan informasinya,

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (3.9)$$

mengalikan dengan turunan fungsi aktivasinya untuk menghitung kesalahan informasinya,

$$\delta_j = \delta_{in_j} f'(z'_{in_j}) \quad (3.10)$$

menghitung koreksi bobotnya (yang digunakan untuk memperbaiki v_{ij} nanti)

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (3.11)$$

Perbaiki bobot untuk proses selanjutnya

Langkah 8: Setiap neuron keluaran ($Y_k, k=1 \dots o$) memperbaiki bobotnya ($j=0 \dots m$)

$$w_{jk} (new) = w_{jk} (old) + \Delta w_{jk} \quad (3.12)$$

Setiap neuron tersembunyi-1 ($Z_j, j=1 \dots p$) memperbaiki bobotnya

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \quad (2.13)$$

Setiap neuron tersembunyi-2 ($Z'_j, j=1 \dots m$) memperbaiki bobotnya

$$v'_{ij}(new) = v'_{ij}(old) + \Delta v'_{ij} \quad (3.14)$$

Langkah 9: Lakukan pengujian kondisi henti.

Nilai *SSE* yang diizinkan dalam proses pelatihan dalam penelitian ini adalah dengan nilai galat sebesar 0.1. Apabila nilai galat dari *SSE* lebih besar 0.1 maka jaringan dianggap belum mengenali pola. Sementara nilai konstanta belajar yang diberikan adalah 0.1. Nilai ini menunjukkan kecepatan belajar dari suatu jaringan.

Untuk mempercepat waktu dalam pelatihan, prosedur perubahan bobot dapat dimodifikasi dengan menggunakan konstanta belajar. Dalam perubahan bobot menggunakan konstanta belajar nilai bobot $t + 1$, hasilnya juga ditentukan oleh nilai bobot ke (t) dan $t (t-1)$, yaitu selisihnya dikalikan dengan suatu konstanta belajar yang bernilai antara 0 sampai 1

3.5 Metode Pelatihan Jaringan

Pelatihan dilakukan dengan menggunakan perangkat lunak (*software*) *MATLAB* versi 7.0.4 yang telah menyediakan fungsi-fungsi pelatihan dan pengujian pada JST dengan algoritma *backpropagation*.

Proses pelatihan dilakukan dengan beberapa tahapan, antara lain dengan identifikasi bobot dan *bias* dengan menggunakan fungsi *initff*, yang memiliki persamaan sebagai berikut.

$$[W1,b1,W2,b2) = \text{initff} (p,S1,F1,S2,F2)$$

Setelah dilakukan inisialisasi maka jaringan *backpropagation* siap dilatih dengan menggunakan fungsi :

$$[W1,b1,W2,b2,epoch,tr) = \text{trainbp} (W1,b1, F1,W2,b2,F2,p,t,tp)$$

Dengan:

W1,W2 = bobot awal jaringan

b1,b2 = *bias*

epoch = jumlah iterasi

tr = jumlah galat

F1,F2 = fungsi aktivasi

P = pola masukan

S = jumlah neuron

t = target

fungsi pelatihan diatas merupakan fungsi pelatihan *trainbp*, yang merupakan fungsi pelatihan *backpropagation* yang memiliki fungsi ini memiliki nilai Tp dengan persamaan:

$$Tp = [\text{disp_freq max_epoch err_goal lr}]$$

Dengan:

Disp_freq = selang tampilan

Max_epoch = jumlah maksimal iterasi

Err_goal = nilai galai yang diperbolehkan

Lr = konstanta belajar

Proses pelatihan ini dilakukan secara berulang sampai di hasilkan suatu jaringan yang dapat memberikan suatu tanggapan yang benar terhadap nilai masukan. Tanggapan yang benar ini ditunjukkan oleh nilai SSE yang biasanya mempunyai nilai galat dibawah 0.1. Apabila nilai RMS/SSE menunjukkan dibawah 0.1 maka jaringan sudah boleh dikatakan terlatih. Sedangkan jumlah iterasi maksimum sebesar 50000 dengan selang tampilan per itersi sebesar 50. Apabila pada pelatihan nilai epoch lebih dari 50000 maka jaringan dianggap tidak konvergen, artinya JST tidak mampu melakukan pembelajaran pada sistem. Selanjutnya, pada pelatihan akan dicari ini akan dicari parameter optimal dengan memvariasikan nilai layer tersembunyi dan konstanta belajar.

Source code untuk pelatihan pada MATLAB terlihat pada tabel berikut

```

ntwarn off

p=[input]

p=p'

t=[output]

t=t'

tp=[50 10000 errol_goal lr]

[w1,b1,w2,b2]=initff(p,x,'logsig't,'logsig')

[w1,b1,w2,b2]=trainbp(w1,b1,w2,b2',p,t,tp)

```

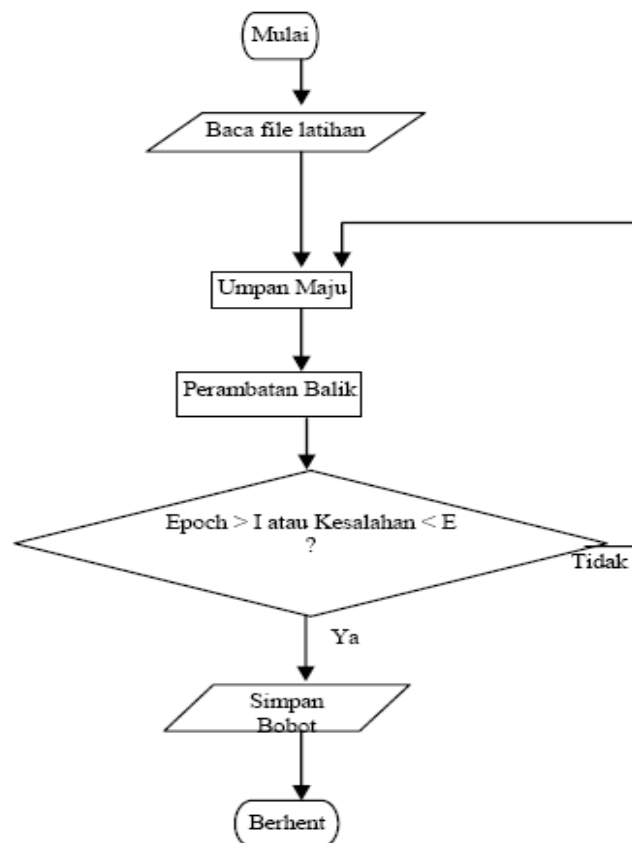
Pada proses pelatihan adalah dimulai dengan mempelajari kedelapan pola dengan umpan balik dan umpan maju kesalahannya. Kemudian setelah itu, dilakukan perhitungan nilai *SSE* dari jaringan. kedelapan pola tersebut akan dipelajari kembali apabila nilai *SSE* nya lebih besar dari nilai galat yang telah ditetapkan.

Untuk menghasilkan data pelatihan akan divariasikan beberapa nilai terhadap layer tersembunyi dan konstanta belajar. Pada palatihan pertama akan dilakukan variasi terhadap layar tersembunyi dengan nilai sebesar 5-800. Setelah dihasilkan nilai optimal, tahap selanjutnya adalah mevariasikan nilai konstanta belajar yang berkisar antara 0.001-0.02. Sehingga akan diperoleh nilai *epoch* dan *SSE* yang optimal untuk pelatihan jaringan

Proses pelatihan JST yang dirancang dapat dihentikan oleh salah satu dari dua kategori yaitu *SSE* lebih kecil dari kesalahan yang ditetapkan atau bila *epoch*

pelatihan melebihi nilai tertentu yang telah ditetapkan sebelumnya seperti yang digambarkan dengan diagram alir pada Gambar 3.12.

Pelatihan JST dimulai dengan mempelajari pola pertama dengan umpan maju dan umpan balik kesalahannya, dilanjutkan dengan pola kedua, kemudian pola ketiga sampai pola seterusnya. Setelah kesemua pola tersebut dipelajari oleh JST, kemudian dihitung nilai SSEnya. Sampai proses ini, dikatakan bahwa JST telah melalui 1 *epoch*. Jika SSEnya lebih besar dari nilai yang ditetapkan atau jumlah epoch yang telah dilakukan kurang dari nilai yang telah ditetapkan, maka JST akan mempelajari kembali kesemua pola tersebut.



Gambar 3.12. Diagram alir metode proses pelatihan JST (Nazrul Effendi,2005)

3.4 Metode Pengujian Jaringan

Pengujian Jaringan dilakukan untuk menguji apakah JST yang telah dilatih dapat digunakan untuk mengklasifikasi pola sinyal pada EKG dan untuk mendapatkan parameter JST yang tepat untuk hal tersebut.

Pengujian dilakukan melalui 2 tahap, yaitu pengujian terhadap data yang telah dilatih dan pengujian terhadap data baru. Setelah bobot akhir dihasilkan maka jaringan tersebut dapat diuji dengan menggunakan fungsi sebagai berikut.

$$A = \text{Simuff}(P, w_1, b_1, \text{'logsig'}, w_2, b_2, \text{'logsig'})$$

Untuk mengetahui nilai error dalam akan digunakan fungsi

$$\text{train_error} = \text{mse}(A-t)$$

Model yang dikembangkan adalah klasifikasi pola yang bertujuan untuk mengenali dengan baik. Untuk menguji kinerja model yang dikembangkan, dilakukan uji tingkat akurasi yang terdiri atas :

1. Tingkat akurasi interenal yaitu tingkat keberhasilan JST dilihat dari nilai error terhadap jumlah seluruh data yang digunakan pada saat dilakukan pengujian.
2. Tingkat akurasi eksternal yaitu tingkat akurasi tiap kelas yang merupakan perbandingan antara pengenalan yang benar terhadap jumlah kelas yang terjadi.

Berdasarkan hasil uji tingkat akurasi, dapat didefinisikan kinerja model yang dikembangkan.

