



BAB III

PERANCANGAN SISTEM

3.1 Deskripsi Sistem

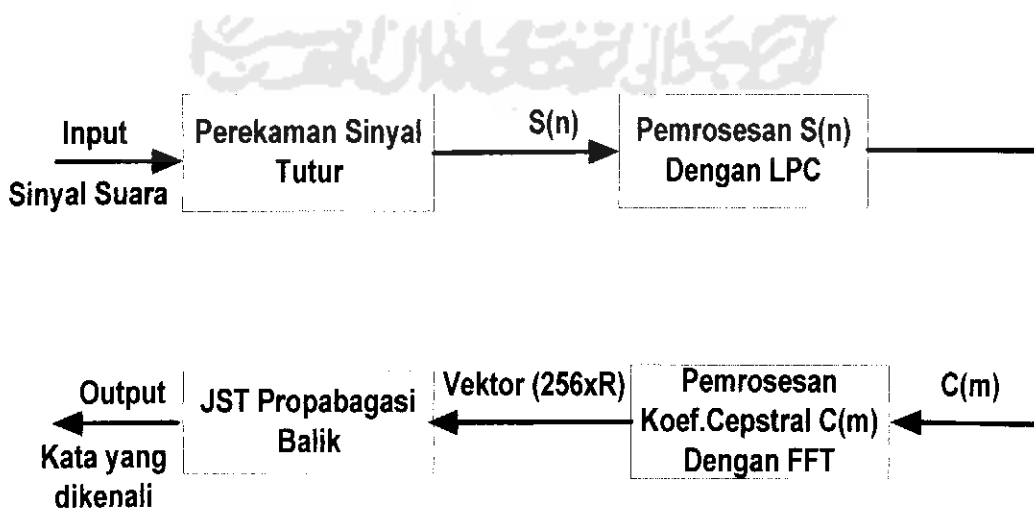
Jumlah data tutur yang digunakan dalam penelitian ini sebanyak 100 kata dari 5 penutur yang berbeda. Setiap penutur memberikan data suara 2 *file* dimana setiap *file* memuat 10 kata yaitu : 'mama', 'papa', 'didi', 'budi', 'dodi', 'buka', 'tutup', 'mulai', 'maju', 'belok'. Perangkat keras yang digunakan adalah komputer dengan spesifikasi berikut : *Prossesor Intel Pentium 4, 2.26 GHz, RAM 128 MHz, soundcard onboard*, serta dilengkapi *microphone*. Sedangkan perangkat lunak yang digunakan adalah MATLAB versi 6.5. *Toolbox* MATLAB yang digunakan: *Signal Processing Toolbox, Neural Network Toolbox*. *Signal Processing Toolbox* digunakan untuk menangani pengolahan suara (sinyal tutur), sedangkan *Neural Network Toolbox* digunakan untuk menangani jaringan saraf tiruan propagasi balik.

Pada sistem pengenalan suara ini dibuat 4 buah GUI (*Graphical User Interface*) yang masing-masing digunakan untuk perekaman, pemrosesan sinyal tutur dengan LPC dan FFT, pelatihan, pengujian serta pengenalan langsung sinyal tutur dan GUI program utama untuk sistem pengenalan suara manusia. Cara menampilkannya adalah dengan menjalankan (*run*) pada *debug* yang ada pada masing-masing M-file GUInya atau dapat dilakukan melalui program utama dengan menekan tombol-tombol yang ada sesuai dengan fungsinya. GUI untuk

program utama ditunjukkan pada gambar III.1 dan blok diagram sistem pengenalan suara manusia ditunjukkan pada gambar III.2.



Gambar III.1 GUI Program Utama Sistem Pengenalan Suara Manusia



Gambar III.2 Blok Diagram Sistem Pengenalan Suara Manusia

3.2 Perekaman Sinyal Tutur

Untuk membangun sistem pengenalan sinyal tutur diperlukan tahapan-tahapan yang berbeda. Tahapan yang pertama kali dilakukan adalah perekaman. Sebelum melakukan perekaman, terlebih dahulu dibuat GUI untuk perekaman. Bentuk tampilan GUI untuk perekaman adalah sebagai berikut :



Gambar III.3 GUI Untuk Perekaman Suara

Perekaman merupakan pengubahan sinyal tutur analog ke sinyal tutur digital dengan bantuan *soundcard* pada PC (*Personal Computer*) melalui piranti *microphone*. Parameter yang dapat dirubah dari perekaman ini adalah frekuensi cuplik (f_s) dan durasi perekaman (t). Nilai (f_s, t) ini berpengaruh terhadap jumlah

hasil cuplik. Pada sistem ini, frekuensi cuplik (f_s) yang digunakan adalah 8 kHz dengan durasi perekaman (t) untuk setiap kata adalah 0,5 detik. Frekuensi pencuplikan ditetapkan berdasarkan asumsi bahwa sinyal percakapan (*speech*) berada pada daerah frekuensi 300 – 3400 Hz sehingga memenuhi kriteria Nyquist yang menyatakan :

$$f_s \geq 2f_h, f_h = f_{in} \text{ tertinggi.} \quad (3.1)$$

Selain itu, penetapan frekuensi cuplik juga akan mengurangi banyaknya hasil cuplikan, dimana jika dibandingkan dengan frekuensi diatas 8 kHz misalnya 10 kHz, tentunya akan menghasilkan hasil cuplikan yang lebih banyak dan akan menambah jumlah perhitungan yang ada. Durasi perekaman untuk satu kata yaitu 0,5 detik, ditetapkan berdasarkan perkiraan ketika melakukan perekaman untuk satu kata. Seperti halnya frekuensi cuplik, durasi perekaman juga ditetapkan selain untuk efisiensi perhitungan juga untuk mengurangi banyaknya *noise* yang masuk ketika perekaman. Misalnya, ketika melakukan perekaman, setelah kata diucapkan ternyata waktu perekamannya masih tersisa, maka sisa waktu tersebut akan terisi *noise*.

Perekaman dilakukan dengan menekan tombol “OK” untuk tiap kata yang ada. Perintah yang digunakan untuk merekam adalah *wavrecord(Sn,fs,Type data)*. Setelah direkam kemudian disimpan dalam *folder* dengan menekan tombol “SELESAI”. Tombol ini akan menyimpan hasil rekaman suara dan menutup GUI perekaman. Satu file berisi 10 tutur kata. Dalam perekaman ini, amplitudo sinyal dibuat agar berkisar -1 sampai +1 volt. Untuk membunyikan sinyal tutur yang telah direkam, dapat dilakukan dengan menekan tombol “BUNYI”. Perintah yang

digunakan untuk membunyikan sinyal tutur adalah “*wavplay(Sn,fs)*”. Perintah ini memerlukan parameter sinyal yang akan dibunyikan dan juga frekuensi cuplik untuk membunyikan. Data perekaman dapat dipanggil, dengan mengeksekusi perintah “*load(nama_file.mat)*” pada *command window*.

3.3 Pemrosesan Sinyal Tutur Dengan LPC

Pemrosesan sinyal tutur dengan metode LPC (*Linear Predictive coding*) lebih dikenal dengan ekstraksi ciri. Ekstraksi ciri dilakukan untuk mengkonversi sinyal menjadi ciri-ciri (*features*) yang mengandung informasi semantik. Ekstraksi ciri ini dilakukan terhadap sinyal tutur yang telah direkam sebelumnya. Proses ekstraksi ciri tersebut adalah sebagai berikut :

1. **Preemphasis**: yaitu hasil cuplikan kata dalam bentuk digital difilter dengan *FIR filter* orde satu.
2. **Frame Blocking**: yaitu pembagian sinyal kata dalam bentuk kerangka-kerangka (*frame*). Antar kerangka diberi jarak sebesar M cuplikan. Bila jumlah cuplikan sinyal setiap tutur kata sebanyak S_n cuplikan, maka akan dihasilkan jumlah kerangka S_n/M , dengan M merupakan jarak antar kerangka. Berdasarkan table II.1, proses *frame blocking* dilakukan tiap 30 ms sehingga jumlah cuplikan kata setiap *frame* yaitu $30 \text{ ms} \times 8000 \text{ Hz} = 240$ cuplikan. Sedangkan jarak antar *frame* 10 ms sehingga jarak antar *frame* yaitu $10 \text{ ms} \times 8000 \text{ Hz} = 80$ cuplikan.
3. **Windowing**: yaitu proses pembobotan (*weighting*) pada setiap *frame* yang telah dibentuk. Bentuk *window* yang digunakan adalah *Hamming window*.

4. **Analisa Autokorelasi:** yaitu masing-masing *frame* yang telah *diwindowing* diautokorelasikan dengan nilai autokorelasi tertinggi adalah orde dari analisa LPC.
5. **Analisa LPC :** yaitu proses ekstraksi ciri dimana pada tahap ini nilai auto korelasi pada setiap *frame* diubah menjadi koefisien LPC. Untuk menghasilkan vektor ciri yang akan diproses pada tahap selanjutnya.
6. **Mengkonversi Parameter LPC Menjadi Koefisien Cepstral.** Koefisien *cepstral* merupakan koefisien transformasi Fourier. Jika orde LPC adalah p , maka diperoleh p elemen, yaitu p elemen koefisien *cepstral*.

Pemrosesan suara dengan LPC pada sistem pengenalan suara manusia adalah sebagai berikut :

Frekuensi cuplik (f_s)	= 8000 Hz untuk setiap kata
Jumlah penutur	= 5 orang, masing-masing mengucapkan 2×10 kata
Jumlah total kata (R)	= 100 kata
Jenis kata	= 10 kata
Jumlah cuplikan setiap kata	= $(0,5/1) \times 8000 = 4000$ cuplikan
Jumlah cuplikan per <i>frame</i> (N)	= 240 cuplikan
Jarak antar <i>frame</i> (M)	= 80 cuplikan
Orde LPC (p)	= 10
Jumlah <i>frame</i> setiap kata	= Jumlah cuplikan/ $M = 4000/80 = 50$ <i>frame</i>
Hasil proses LPC	= 50×10 untuk setiap kata

Hasil proses LPC berupa matriks diubah kedalam suatu vektor dengan cara menyusun kolom-kolomnya menjadi sebuah vektor baris. Sehingga hasil proses LPC menjadi 500×1 vektor ciri untuk setiap kata. Hasil proses LPC ini merupakan koefisien *cepstral* yang merupakan input bagi proses FFT yaitu 500 data.

3.4 Pemrosesan Sinyal Tutar Dengan FFT

Proses *Fast Fourier Transform* dilakukan setelah didapat koefisien *cepstral* yaitu 500 data. FFT yang digunakan adalah 512 *point* dan karena hasil FFT simetris maka keluaran FFT hanya diambil sebanyak 256 data. Proses FFT dilakukan dengan menggunakan perintah “*fft*” pada data tutur yang telah diekstraksi. Proses dilakukan untuk mendapatkan sinyal pada kawasan frekuensi dimana jumlah titik data N merupakan bilangan yang dapat difaktorkan sehingga seluruh jumlah titik dapat dipecah ke dalam kelompok-kelompok yang lebih kecil. Proses FFT ini bertujuan agar perbedaan antar pola kata yang satu dengan yang lain terlihat lebih jelas serta untuk lebih meningkatkan unjuk kerja sistem sehingga ekstrasi parameter sinyal memberikan hasil yang lebih baik. GUI untuk proses LPC dan FFT ditunjukkan pada gambar III.4.



Gambar III.4 GUI Untuk Pemrosesan Sinyal Tutar Dengan LPC dan FFT

3.5 Membangun Jaringan Saraf Tiruan Propagasi Balik

Arsitektur jaringan saraf tiruan yang sering digunakan oleh algoritma propagasi balik adalah jaringan *feedforward* dengan banyak lapisan. Pada sistem pengenalan suara manusia, jaringan saraf tiruan dibangun dengan jaringan *feedforward* yaitu dengan menggunakan instruksi "newff". Struktur jaringan saraf tiruan propagasi balik dengan *newff* adalah sebagai berikut :

net = newff (PR, [S1 S2 ...SN1], {TF1 TF2...TFN1}, BTF)

PR : matriks berukuran Rx2 yang berisi nilai minimum dan maksimum, dengan

R adalah jumlah variabel input.

S_i : jumlah neuron pada lapisan ke- i , dengan $i = 1, 2, \dots, N1$.

TF_i : fungsi aktivasi pada lapisan ke- i , dengan $i = 1, 2, \dots, N1$; (default : *tansig*).

BTF : fungsi pelatihan jaringan (default : *trainlm*).

Beberapa parameter yang ditentukan sebelum melatih jaringan antara lain yaitu jumlah iterasi (*epoch*), target error (*goal*), *momentum*, *learning rate* dan target keluaran.

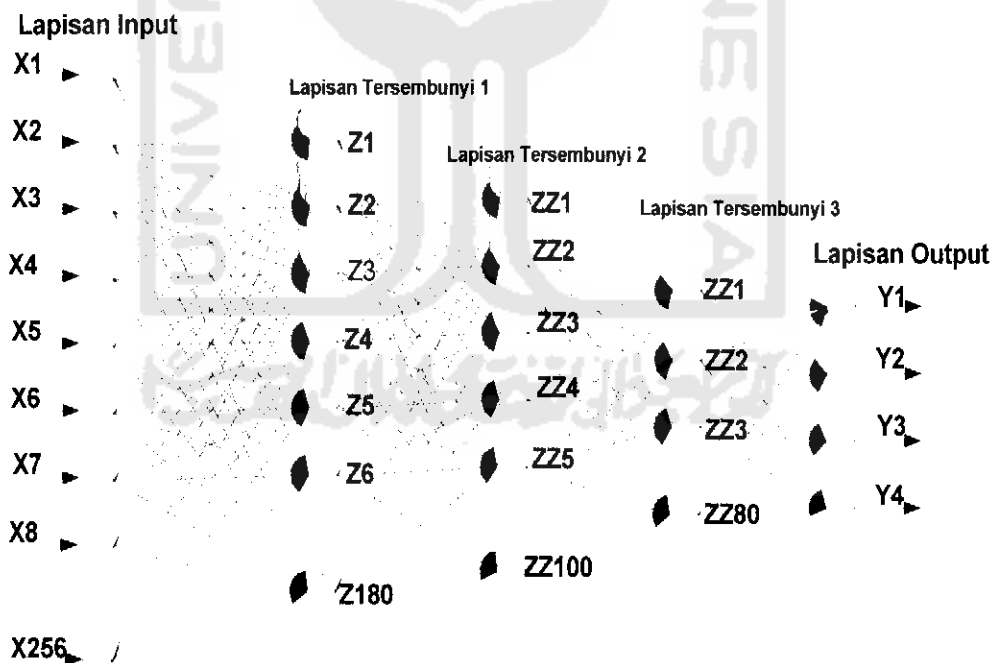
Pada perancangan sistem ini, input jaringan saraf tiruan berasal dari keluaran FFT dengan FFT yang digunakan 512 *point*. Hasil proses FFT yang diambil hanya 256 data, karena sifat FFT yang simetris. Keluaran FFT ini menjadi input bagi jaringan saraf tiruan yaitu 256 data input untuk setiap katanya sedangkan outputnya berjumlah 4 buah. Target pelatihan yang digunakan sebanyak 10 buah yang masing-masing merupakan bilangan biner dengan pola sebagai berikut :

Tabel III.1 Target Keluaran Jaringan Saraf Tiruan Propagasi Balik Pada sistem Pengenalan Suara Manusia

Kata	Pola Biner
Mama	0000
Papa	0001
Didi	0010
Budi	0011
Dodi	0100
Buka	0101
Tutup	0110

Mulai	0111
Maju	1000
Belok	1001

Struktur jaringan saraf yang digunakan pada sistem ini adalah jaringan saraf tiruan propagasi balik dengan tiga lapisan tersembunyi dengan susunan 180, 100, 80 untuk setiap lapisan tersembunyi. Gambar III.5 menunjukkan struktur jaringan saraf propagasi balik yang digunakan pada sistem pengenalan suara manusia.

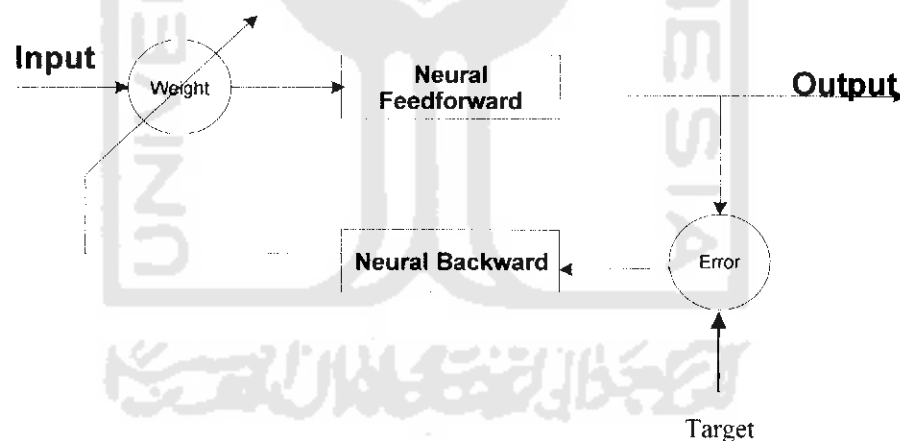


Gambar III.5 Struktur Jaringan Saraf Tiruan Propagasi Balik Untuk Pengenalan

Suara Manusia Dengan Tiga Lapisan tersembunyi

3.6 Pelatihan Jaringan Saraf Tiruan Propagasi Balik

Jaringan saraf tiruan propagasi balik merupakan jaringan saraf tiruan yang menggunakan algoritma pembelajaran terawasi (*supervised learning*) untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersebut. Algoritma propagasi balik menggunakan *error* output untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap perambatan maju (*feedforward*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi. Blok diagram pembelajaran jaringan saraf tiruan propagasi balik ditunjukkan pada gambar III.6 :



Gambar III.6 Blok Diagram Pembelajaran Jaringan Saraf Tiruan Propagasi Balik

Pada perancangan sistem ini, input jaringan saraf tiruan berasal dari keluaran FFT dengan FFT yang digunakan 512 *point*. Hasil proses FFT yang diambil hanya 256 data, karena sifat FFT yang simetris. Keluaran FFT ini menjadi

input bagi jaringan saraf tiruan yaitu 256 data input untuk setiap katanya sedangkan outputnya berjumlah 4 buah. Target pelatihan yang digunakan sebanyak 10 buah yang masing-masing merupakan bilangan biner.

Pelatihan jaringan dilakukan dengan mengambil input dari pembicara sebanyak 5 orang dimana masing-masing pembicara mengucapkan 10 buah pola kata yaitu: 'mama', 'papa', 'didi', 'budi', 'dodi', 'buka', 'tutup', 'mulai', 'maju', 'belok'. Setiap pembicara membuat dua *file* perekaman sehingga jumlah keseluruhan data yaitu 100 data. Pola kata dari masing-masing pembicara tersebut disimpan dan diproses terlebih dahulu dan kemudian dilatihkan secara bersamaan ke dalam jaringan saraf tiruan propagasi balik. Pola kata dimasukkan secara urut mulai pembicara 1 dengan masing-masing pola katanya selanjutnya pembicara 2 sampai seterusnya pembicara 5. Setelah semua data dimasukkan maka proses pelatihan dilakukan sampai *error* yang dihasilkan mencapai nilai yang telah ditentukan yaitu 0.0001. Untuk pelatihan jaringan saraf tiruan propagasi balik dilakukan dengan menggunakan perintah "*train(net,input_jst,target_jst)*".

3.7 Pengujian Jaringan Saraf Tiruan Propagasi Balik

Pada tahap awal, uji pengenalan dilakukan terhadap sinyal suara yang sama persis dengan data pelatihan. Kemudian dilakukan pengujian terhadap sinyal suara secara langsung dari *microphone* oleh orang yang sama dengan yang dilatihkan (pembicara 1 sampai dengan pembicara 5). Pengujian jaringan dilakukan dengan menggunakan perintah "*sim(data.net,input_jst)*". Sedangkan untuk pengenalan langsung, dilakukan dengan menekan tombol "PENGENALAN

KATA” pada GUI pelatihan dan pengujian serta pengenalan yaitu dengan mengucapkan kata yang akan dikenali. GUI untuk pelatihan dan pengujian serta pengenalan jaringan saraf tiruan propagasi balik ditunjukkan pada gambar III.7.



Gambar III.7 GUI Untuk Pelatihan, Pengujian JST PB Serta Pengenalan Kata