

ALGORITMA QUICKPROP PADA PENGENALAN SUARA

Abdul Aziz¹, Ednawati Rainarli²

Program Studi Teknik Informatika

^{1,2} Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia

Jl. Dipati Ukur 114 Bandung

E-mail : aziz.rpl1@gmail.com¹, irene_edna@yahoo.com²

ABSTRAK

Speaker recognition merupakan suatu proses pengenalan suara berdasarkan seorang pembicara. Angga setiawan, dkk telah menggunakan *backpropagation* untuk melakukan *speaker recognition*. Hasil penelitian diperoleh nilai akurasi sebesar 83,99%. Selain itu Windra Swastika juga telah membandingkan penggunaan *quickprop* dengan *backpropagation* untuk masalah *image recognition*. Hasil penelitian menunjukkan bahwa metode *quickprop* bekerja lebih baik dari metode *backpropagation*. Oleh karena itu, algoritma klasifikasi yang akan digunakan pada penelitian ini adalah metode *quickprop*. Pada penelitian ini, data suara yang digunakan berasal dari 5 orang pembicara laki-laki yang mengucapkan 5 kata dalam bahasa Indonesia. Kata yang digunakan berasal dari penelitian menurut *Leipzig Corpora Collection* yang disimpan dalam bentuk file .wav. Sebelum dilakukan pengenalan, data suara terlebih dahulu dilakukan proses ekstraksi ciri dengan metode MFCC. Hasil dari ekstraksi ciri tersebut yang akan disimpan dan kemudian akan digunakan sebagai masukan untuk proses klasifikasi *Quickprop*. Berdasarkan pengujian *K-Fold Cross Validation* terhadap parameter yang digunakan didapatkan rata-rata akurasi yaitu sebesar 92% dan kata yang dipilih dapat digunakan dalam proses pengenalan.

Kata Kunci : *Speaker Recognition*, *Quickprop*, Ekstraksi Ciri, MFCC.

1. PENDAHULUAN

Speaker Recognition adalah suatu proses identifikasi dan verifikasi untuk mengenali seorang pembicara yang telah diketahui identitasnya berdasarkan data suara yang telah diinputkan. Untuk dapat mengenali data suara seseorang berdasarkan sebuah kata yang diucapkan, data suara akan melalui proses ekstraksi ciri, pencocokan pola, sehingga informasi yang terkandung di dalam data suara dapat digunakan [1]. Jaringan Syaraf Tiruan (JST) merupakan metode yang dapat digunakan untuk proses pembelajaran sebagai pencocokan pola pada kasus pengenalan suara. Terdapat banyak metode JST, salah satunya adalah *Learning Vector*

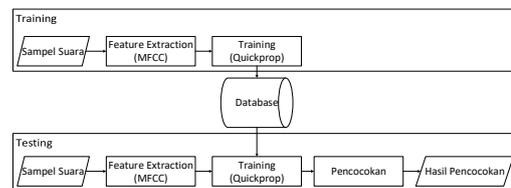
Quantization (LVQ), *Backpropagation* dan *Quickprop*.

Pada penelitian sebelumnya yang dilakukan oleh Angga Setiawan, dkk. menerapkan ekstraksi ciri dengan metode *Mel Frequency Cepstrum Coefficients (MFCC)* dan pencocokan pola melalui JST dengan metode pembelajaran LVQ hasilnya diperoleh rata-rata persentase keberhasilan pengenalan data suara sebesar 83,99% [2]. Pada penelitian lain yang dilakukan oleh Yulia Nur Utami, dkk. menerapkan MFCC sebagai ekstraksi ciri dan menggunakan *JST Backpropagation* sebagai pencocokan pola, mendapatkan hasil 80,23% akurasi keberhasilan [3]. Selain itu Windra Swastika, dalam penelitian kasus lain yang membandingkan metode *Quickprop* dengan metode *Backpropagation* sebagai metode pembelajaran pada kasus *image recognition*, metode *Quickprop* dapat bekerja lebih baik dari metode *Backpropagation* [4]. Dari beberapa penelitian tersebut *Quickprop* terlihat mampu memberikan hasil yang baik ketika digunakan untuk pengenalan pola, namun performa dari *Quickprop* untuk kasus *speaker recognition* kata dalam Bahasa Indonesia belum diketahui.

Berdasarkan uraian di atas, pada penelitian ini akan menerapkan metode pembelajaran *Quickprop* untuk kasus pengenalan suara. Pada tahap ekstraksi fitur menggunakan MFCC sebagai ekstraksi ciri suara yang berguna bagi proses pengenalan suara.

2. LANDASAN TEORI

Sistem *speaker recognition* yang akan dibangun pada penelitian ini terdiri dari beberapa proses, yaitu Ekstraksi Ciri (*Feature Extraction*), Normalisasi, kemudian bagian *Quickprop* (pelatihan maupun pengujian). Secara umum gambaran sistem yang akan dibangun dapat dilihat pada gambar 1.



Gambar 1. Gambaran Umum Sistem

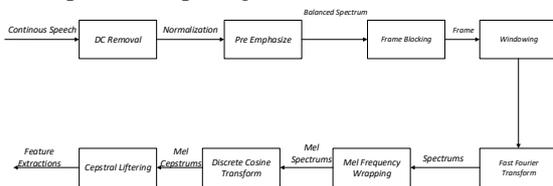
Dataset (data latih dan data uji) yang digunakan sebagai data masukkan pada proses pelatihan (*training*) dan proses pengujian (*testing*) merupakan file suara hasil perekaman 5 orang pembicara dengan menyebutkan kalimat dengan 5 kata bahasa indonesia yang paling sering digunakan[5] yaitu (adalah, bisa, milik, rupa, sebut) yang kemudian akan dipotong menjadi satu kata yang ingin digunakan sehingga berdurasi 1 detik menggunakan format .wav dengan sinyal suara mono.

Pada proses pelatihan (training) akan dilakukan proses konversi dari sinyal analog menjadi sinyal digital terhadap sinyal suara terpilih yang kemudian akan masuk kedalam proses ekstraksi ciri. Setelah didapatkan sinyal digital, maka metode ekstraksi ciri yang digunakan adalah metode *Mel Frequency Cepstrum Coefficients* (MFCC) sehingga bisa digunakan sebagai bahan inputan pelatihan menggunakan *Quickprop*.

Pada proses pengujian (testing) akan dilakukan proses yang sama seperti pada proses pelatihan, namun setelah dilakukan proses ekstraksi ciri dan normalisasi kemudian dilakukan proses pengenalan menggunakan *Quickprop* yang bobotnya telah dilatih pada proses pelatihan. Hasil dari proses pengujian ini sistem akan menampilkan nama pembicara dari data uji terpilih.

2.1 MFCC

Mel Frequency Cepstrum Coefficients (MFCC) merupakan salah satu metode yang banyak digunakan dalam bidang *speech technology*, baik *speaker recognition* maupun *speech recognition* [3]. Pada metode *Mel Frequency Cepstrum Coefficients* ini terdiri dari 8 tahapan proses yang terdiri dari *DC Removal*, *Pre-Emphasize Filtering*, *Frame Blocking*, *Windowing*, *Fast Fourier Transform* (FFT), *Mel-Frequency Wrapping*, *Discrete Cosine Transform* (DCT) dan *Cepstral Lifting*. Tahapan pada proses MFCC dapat dilihat pada gambar 2.



Gambar 2. Blok Diagram MFCC

2.1.1 DC-Removal

Proses *DC removal* dilakukan untuk menghitung rata-rata dari data sampel suara, dan mengurangi nilai setiap sampel suara dengan nilai rata-rata tersebut. Tujuannya adalah mendapat normalisasi dari data suara input [7].

$$dr_i = s_i - \bar{x}, 0 \leq i \leq N - 1 \quad (1)$$

Keterangan :

dr_i : titik sinyal hasil proses *DC Removal*.

s_i : titik sinyal ke- i .

\bar{x} : nilai rata-rata titik sinyal.

i : 0, 1, 2, ..., N-1

N : panjang sinyal (banyak titik sinyal).

2.1.2 Pre-Emphasize Filtering

Setelah melalui proses *DC-Removal*, tahap selanjutnya adalah *Pre-Emphasize Filtering* merupakan salah satu jenis *filter* yang sering digunakan sebelum sebuah sinyal diproses lebih lanjut. *Filter* ini mempertahankan frekuensi-frekuensi tinggi pada sebuah spektrum yang umumnya tereliminasi pada saat proses produksi suara [3].

$$pf_i = dr_i - (dr_{i-1} \times \alpha) \quad (2)$$

Keterangan :

pf_i : sinyal hasil *pre-emphasize filter* ke- i .

dr_i : sinyal sebelum *pre-emphasize filter* ke- i .

α : *pre-emphasize filter*.

i : 0, 1, 2, ..., N-1

N : panjang sinyal (banyaknya titik sinyal).

Kemudian persamaan untuk mendapatkan hasil *pre-emphasize filtering* dapat direpresentasikan pada persamaan 3.

$$p_i = dr_i + pf_i \quad (3)$$

Keterangan :

p_i : titik sinyal baru ke- i .

dr_i : sampel sebelum *pre-emphasize* ke- i (hasil *DC Removal*)

pf_i : *pre-emphasize filter* pada sampel ke- i .

i : 0, 1, 2, ..., N-1

2.1.3 Frame Blocking

Pada langkah ini, titik-titik sinyal dibagi menjadi *frame-frame*. Setiap *frame* memiliki ukuran atau panjang yang sama, dan juga setiap *frame* melakukan *overlapping* terhadap *frame* lainnya. Jarak *overlap* antar *frame* yang bisa digunakan sekitar $\frac{1}{2}$ sampai $\frac{1}{3}$ panjang *frame*. Pada penelitian ini *overlap* yang digunakan adalah $\frac{1}{2}$ dari panjang *frame*. Durasi *frame* yang digunakan pada penelitian ini adalah 0.025 detik, sedangkan nilai *sample rate* yang digunakan adalah 16000 [3]. Adapun proses menghitung jumlah *frame blocking* dapat dilihat pada persamaan 4.

$$\text{jumlah frame} = ((NS - SP)/M) + 1 \quad (4)$$

Keterangan :

NS : jumlah sampel

SP : *Sample Point* dalam tiap *frame* (*Sample Rate* \times panjang *frame* dalam detik (s))

M : $SP/2$ (*Overlap*)

2.1.4 Windowing

Proses *windowing* ini dilakukan pada setiap *frame* hasil dari proses *frame blocking*. Hal ini

dilakukan untuk mengurangi terjadinya kebocoran spektral atau *aliasing* [3]. *Aliasing* adalah sinyal baru dimana memiliki frekuensi yang berbeda dengan sinyal aslinya. Efek ini dapat terjadi karena rendahnya jumlah *sampling rate*, ataupun karena proses *frame blocking* dimana menyebabkan sinyal menjadi *discontinue*. Untuk mengurangi kemungkinan terjadinya kebocoran spektral, maka hasil dari proses *framing* harus melewati proses *window* [7]. Adapun proses untuk menghitung dapat dilihat pada persamaan 5.

$$w_i = p_i \times fw_i \quad (5)$$

Keterangan :

w_i : nilai titik sinyal hasil *windowing*.

p_i : hasil *pre-emphasize* ke-1 dari *frame* ke- i .

fw_i : fungsi *window* ke-1 untuk *frame* ke- i .

NS : banyaknya *sample point* pada tiap *frame*.

1 : *sample point* ke-1 pada *frame* (dimana nilai $n = 0, 1, \dots, NS-1$).

Fungsi *window* yang paling sering digunakan dalam aplikasi *speaker recognition* adalah *Hamming Window*. Berikut persamaan dari *Hamming Window* dapat dilihat pada persamaan 6.

$$fw_i = 0.54 - 0.46 \cos \frac{2\pi l}{SP-1} \quad (6)$$

Keterangan :

fw_i : fungsi *hamming window* pada *frame* ke- i .

SP : *Sample Point* (panjang *frame*).

l : 0, 1, ..., SP-1

2.1.5 FFT

Setelah melewati tahapan *windowing*, maka tahap selanjutnya adalah *Fast Fourier Transform* (FFT). Setiap *frame* dikonversi dari domain waktu ke domain frekuensi untuk mendapatkan spektrum frekuensinya. Hal ini dilakukan untuk mempermudah komputasi dan analisa [2]. Persamaan FFT dapat dilihat sebagai berikut.

$$f_k = \sum_{i=0}^{SP-1} (w_i \cos \frac{2\pi i k}{SP}) - j \sum_{i=0}^{SP-1} (w_i \sin \frac{2\pi i k}{SP}), 0 \leq k \leq SP-1 \quad (7)$$

Keterangan :

f_k : spektrum FFT ke- k .

N : jumlah sampel yang akan diproses ($N \in \mathbb{N}$).

l : 0, 1, 2, ..., SP-1 (jumlah sampel dalam *frame*).

k : 0, 1, 2, ..., SP-1 (variabel frekuensi diskrit, merupakan hasil FFT).

w_i : nilai titik sinyal ke- n (hasil *windowing* ke- n pada *frame*).

j : bilangan imajiner.

Dimana untuk mendapatkan hasil dari FFT dengan menggunakan persamaan 8.

$$|fft_k| = [R^2 + I^2]^{\frac{1}{2}} \quad (8)$$

Keterangan :

R : bilangan real (hasil perhitungan $w_i \cos \frac{2\pi i k}{N}$).

I : bilangan imajiner (hasil perhitungan $j(w_i \sin \frac{2\pi i k}{N})$).

Hasil dari FFT kemudian akan disebut sebagai magnitude FFT yang akan dipetakan kedalam skala mel.

2.1.6 Mel-Frequency Wrapping

Magnitude yang telah didapatkan dari proses FFT kemudian akan masuk ke tahap Mel-Frequency Wrapping. *Mel Frequency Wrapping* umumnya dilakukan dengan menggunakan *filterbank*. *Filterbank* adalah salah satu bentuk dari *filter* yang dilakukan dengan tujuan untuk mengetahui ukuran energi dari *frequency band* tertentu dalam sinyal suara [5]. Persamaan yang digunakan untuk mendapatkan hasil *Mel-Frequency Wrapping* dapat dilihat sebagai berikut.

$$MF_m = \sum_{k=0}^{SP-1} fft_k \times H_{mk} \quad (9)$$

Keterangan :

MF_m : hasil *Mel Frequency Wrapping* ke- m .

fft_k : hasil FFT ke- k .

SP : banyaknya nilai hasil FFT.

m : 0, 1, 2, ..., jumlah *Mel Filterbank*

Untuk mencari koefisien mel *filterbank* dapat dibuat dengan cara membuat triangular *filterbank*, menggunakan persamaan 10 berikut[2].

$$mel(f) = 2595 \times \log_{10} \left(1 + \frac{f}{700} \right) \quad (10)$$

Keterangan :

$mel(f)$: frekuensi *Mel Scale*.

f : frekuensi yang digunakan.

Dan persamaan invers mel scale dapat dilihat pada persamaan 11.

$$mel^{-1}(f) = 700 \times \left(10^{\frac{f}{2595}} - 1 \right) \quad (11)$$

Keterangan :

$mel^{-1}(f)$: frekuensi *Mel Scale* invers.

f : frekuensi yang digunakan.

Kemudian untuk *filterbank boundary* yang digunakan dapat dilihat pada persamaan 12 berikut

$$fb_m = \left(\frac{SP}{FS} \right) \times mel^{-1} \left(mel(Nlow) + m \frac{mel(Nhigh) - mel(Nlow)}{Nfbank} \right) \quad (12)$$

Keterangan :

fb_m : hasil *boundary point filter* ke- m .

SP : *Sample Point* (banyaknya sampel point).

FS : frekuensi sampling.

m : jumlah *triangular filterbank*.

Kemudian setelah didapatkan nilai *filterbank boundary*, dilakukan aturan pada persamaan 13 berikut [8].

$$H_{mk} = \begin{cases} 0, & \text{untuk } k < fb_m \\ \frac{k - fb_{m-1}}{fb_m - fb_{m-1}}, & \text{untuk } fb_{m-1} \leq k < fb_m \\ \frac{fb_{m+1} - k}{fb_{m+1} - fb_m}, & \text{untuk } fb_m \leq k < fb_{m+1} \\ 0, & \text{untuk } k > fb_{m+1} \end{cases} \quad (13)$$

Keterangan :

fb_m : hasil *boundary point filter*.

H_{mk} : koefisien dari *mel filterbank*.

2.1.7 DCT

Discrete Cosine Transform merupakan langkah terakhir dari proses utama MFCC *feature extraction*. Konsep dasar dari DCT adalah mendekorelasikan *mel cepstrums* sehingga menghasilkan representasi yang baik dari *property* spektral lokal. Pada dasarnya konsep dari DCT sama dengan *inverse fourier transform*. Namun hasil dari DCT mendekati PCA (*Principal Component Analysis*). PCA adalah metode statistika klasik yang digunakan secara luas dalam analisa data dan kompresi. Hal inilah yang menyebabkan seringkali DCT menggantikan *inverse fourier transform* dalam proses MFCC *feature extraction* [8]. Berikut ini persamaan 14 untuk menghitung DCT.

$$C_{kofef} = \sqrt{\frac{2}{Nfbank}} \sum_{m=0}^{Nfbank-1} (\log MF_m) \cos \left[kofef \left(\frac{2m-1}{2} \right) \frac{\pi}{Nkofef} \right] \quad (14)$$

Keterangan :

MF_m : hasil Mel Frequency Wrapping ke-m.

M : 0, 1, 2, ..., SP-1 (indeks hasil filterbank ke-m).

$Nfbank$: jumlah *Mel Filterbank* (40).

$Nkofef$: jumlah koefisien yang diharapkan (pada penelitian ini 13).

Koefisien ke nol dari DCT pada umumnya akan dihilangkan, walaupun sebenarnya mengindikasikan energi dari *frame* sinyal tersebut. Hal ini dilakukan karena, berdasarkan penelitian-penelitian yang pernah dilakukan, koefisien ke nol ini tidak *reliable* terhadap *speaker recognition* [6].

2.1.8 Cepstral Liftering

Hasil dari fungsi DCT adalah *cepstrum* yang sebenarnya sudah merupakan hasil akhir dari proses MFCC. Tetapi untuk meminimalisasi sensitifitas dari koefisien MFCC yang telah didapat, maka *cepstrum* hasil dari DCT akan diolah lagi pada blok *cepstral liftering*. *Cepstral liftering* menghaluskan spektrum hasil dari *main processor* sehingga dapat digunakan lebih baik untuk *pattern matching* [7]. Adapun persamaan 15 untuk menghitung cepstral liftering adalah sebagai berikut.

$$Cepstral_n = \left\{ C \times 1 + \frac{L}{2} \sin \left(\frac{n\pi}{L} \right) \right\} \quad (15)$$

Keterangan :

L : jumlah *cepstral coefficients*.

n : indeks dari *cepstral coefficients*.

C : hasil dct.

2.2 Normalisasi

Hasil dari proses ekstraksi ciri yang telah dilakukan harus melalui proses normalisasi sehingga

dapat diproses oleh *Quickprop* karena proses pelatihan maupun pengujian akan lebih mudah dilakukan dengan nilai diskrit dibandingkan dengan nilai kontinu. Normalisasi adalah penskalaan terhadap nilai-nilai yang masuk ke dalam suatu range tertentu. Hal ini dilakukan agar nilai *input* dan target *output* sesuai dengan range dari fungsi aktivasi yang digunakan dalam jaringan. Bila fungsi aktivasi yang digunakan adalah sigmoid biner, maka persamaan 16 normalisasi yang dapat digunakan yaitu [9].

$$normal_i = \frac{0.8(Cepstral_n - a)}{b - a} + 0.1 \quad (16)$$

Keterangan :

$normal_i$: hasil normalisasi ke-i.

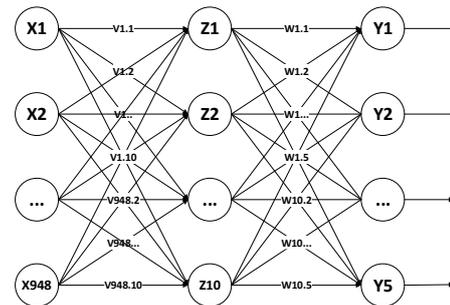
$Cepstral_n$: hasil Cepstral Liftering ke-n.

a : nilai minimum pada *frame*.

b : nilai maksimum pada *frame*.

2.3 Quickprop

Quickprop atau *Quickpropagation* adalah salah satu metode pembelajaran dengan JST (Jaringan Syaraf Tiruan). *Quickprop* berasal dari *Backpropagation* tetapi menggunakan cara yang berbeda dalam meng-update bobot sinapsis. *Quickprop* banyak digunakan dalam banyak kasus, salah satu diantaranya adalah *pattern recognition*. Diantara metode pembelajaran dengan JST, *Quickprop* termasuk metode pembelajaran yang cukup cepat dalam pembelajaran dan menghasilkan akurasi yang cukup tinggi dalam pengenalan [4]. Berikut adalah arsitektur dari jaringan quickprop pada gambar 3.



Gambar 3. Arsitektur Quickprop

2.3.1 Pelatihan Quickprop

Secara prosedural tahapan-tahapan pada algoritma pelatihan jaringan *Quickprop* dapat diuraikan sebagai berikut :

Langkah-1: Inisialisasi semua bobot pada *layer* tersembunyi (*hidden*) dan *layer* keluaran (*output*), lalu tetapkan fungsi aktivasi yang digunakan untuk setiap *layer*.

Langkah-2 : Hitung keluaran yang didapatkan dari neuron dalam *hidden layer*.

$$v_j(p) = \sum_{i=1}^r x_i(p) \times w_{ij}(p) \quad (17)$$

Keterangan :

v : *output* pada *layer* saat ini.

j : *neuron* pada *layer* keluaran.

x : *input* pada *layer* saat ini.

r : jumlah masukan pada *neuron* pada *layer* saat ini.

w : bobot

Langkah-3: Kemudian aktivasi keluaran pada langkah ke 2 dengan menggunakan fungsi aktivasi sigmoid

$$y_j(p) = \frac{1}{1+e^{-v_j(p)}} \quad (18)$$

Langkah-4 : Lakukan perhitungan yang sama seperti pada langkah ke 2 dan 3 untuk mendapatkan nilai keluaran pada *output layer*.

Langkah-5 : Hitung perbedaan antara output dari langkah ke 3 dengan target yang diinginkan menggunakan *cost function* seperti pada persamaan 19.

$$E = \frac{1}{2} \sum_j^N (d_j - o_j)^2 \quad (19)$$

Keterangan :

d_j : output yang diharapkan

o_j : output aktual.

Langkah-6 : Perubahan bobot atau *update* bobot menggunakan persamaan 20 seperti berikut ini.

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij} + \frac{\frac{\partial E}{\partial w_{ij}}(p)}{\frac{\partial E}{\partial w_{ij}}(p-1) - \frac{\partial E}{\partial w_{ij}}(p)} \Delta w_{ij}(p-1) \quad (20)$$

Keterangan :

$w_{ij}(p+1)$: Perubahan bobot / bobot baru.

$\Delta w_{ij}(p-1)$: delta bobot pada *epoch* sebelumnya.

$\frac{\partial E}{\partial w_{ij}}(p)$: Derivatif *error*.

$\frac{\partial E}{\partial w_{ij}}(p-1)$: Derivatif *error* pada *epoch* sebelumnya.

Langkah-7 : proses pelatihan akan berhenti apabila nilai *Error* sudah lebih kecil dengan nilai yang sudah ditetapkan.

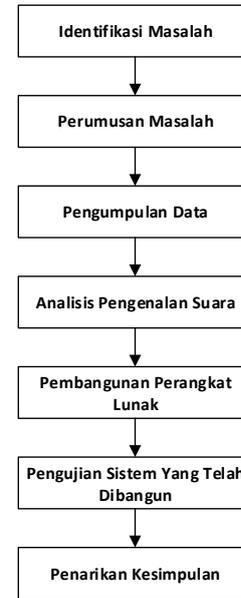
2.3.1 Pengujian Quickprop

Secara prosedural tahapan-tahapan pada algoritma pengujian jaringan *Quickprop* hampir sama dengan tahapan pada pelatihan, namun langkah inisialisasi diganti dengan bobot hasil proses pelatihan sebelumnya dan langkah yang digunakan hanyalah langkah 2 dan 3.

3. METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini adalah penelitian kuantitatif. Disebut metode kuantitatif karena data penelitian berupa

angka-angka dan analisis menggunakan statistik. Teknik sampling yang digunakan yaitu *simple random sampling*, dikatakan *simple* (sederhana) karena pengambilan anggota sampel dari populasi dilakukan secara acak [10]. Alur penelitian yang akan dilakukan pada penelitian ini dapat dilihat pada gambar 4 dibawah ini.



Gambar 4. Alur Penelitian

4. HASIL DAN PEMBAHASAN

Pengujian dilakukan menggunakan *K-Folds Cross Validation*, dengan menggunakan nilai $K=5$ yang menunjukkan jumlah dataset, dimana dataset dibagi menjadi 5 bagian yaitu K_1, K_2, K_3, K_4, K_5 . Jumlah dataset yang digunakan sebanyak 500 data. Tiap dataset (K) berjumlah 100 data.

Pada pengujian ini nilai parameter optimal yang digunakan yaitu *hidden layer*=10, *learning rate*=0.3, *error*=0.01 dan *epoch*=500. Berikut merupakan beberapa percobaan dari parameter berbeda yang dilakukan pada proses pelatihan dan pengujian dengan dataset yang sama sebanyak 500 dapat dilihat pada tabel 1.

Tabel 1. Kombinasi Parameter

No	Learning Rate	Akurasi (%)
1	0.2	76.8
2	0.3	98.4
3	0.4	82.8
4	0.5	78.6

4.1 Hasil pengujian

Setelah dilakukan pengujian dengan menggunakan *5-fold cross validation* pada algoritma

quickprop yang pengujiannya dilakukan sebanyak 5 kali, maka didapatkan rekap hasil pengujian yang dapat dilihat pada tabel 2.

Tabel 2. Hasil Pengujian K-Fold Cross Validation

Pengujian	Benar	Salah	Akurasi (%)
1	92	8	92
2	95	5	95
3	93	7	43
4	92	8	92
5	88	12	88
Rata-rata akurasi (%)			92

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian, analisis dan perancangan, implementasi serta sampai pada tahap pengujian, dapat diambil kesimpulan bahwa parameter kata-kata dalam bahasa Indonesia yang dipilih telah mampu digunakan untuk proses pengenalan pembicara pada *speaker recognition* dengan menggunakan metode *Quickprop* sebagai algoritma klasifikasi dengan memanfaatkan hasil ekstraksi ciri MFCC. Pada proses pengujian dengan parameter optimal yang digunakan yaitu *hidden layer=10*, *learning rate=0.3*, *error=0.01*, maksimum *epoch=500* telah didapatkan nilai akurasi tertinggi sebesar 95% sedangkan nilai akurasi terendah sebesar 88% serta jumlah rata-rata nilai akurasi sebesar 92%. Nilai akurasi yang didapatkan akan berbeda karena dipengaruhi oleh parameter pelatihan, data latih serta data uji yang digunakan.

5.2 Saran

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan, penelitian ini masih dapat dikembangkan lagi sesuai dengan kebutuhan pengguna. Adapun saran untuk penelitian kedepannya yaitu :

1. Menambahkan *filter* yang dapat memperkecil *noise* atau gangguan serta memperkecil data suara tanpa mengurangi informasi penting yang akan dilakukan proses ekstraksi ciri sehingga dapat meningkatkan akurasi dan mempercepat proses ekstraksi.
2. Menambahkan variasi terhadap jumlah parameter yang akan digunakan seperti jumlah *hidden layer*, *learning rate*, *error* yang akan berpengaruh kepada hasil pengujian.
3. Hal lain yang perlu ditambahkan yaitu sistem dapat mengenali nama pembicara serta kata yang diucapkannya.

DAFTAR PUSTAKA

- [1] J. P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437-1462, 1997.
- [2] A. Setiawan, A. Hidayatno, R. R. Isnanto, "Aplikasi Pengenalan Ucapan dengan Ekstraksi Mel-Frequency Cepstrum Coefficients (MFCC) Melalui Jaringan Syaraf Tiruan (JST) Learning Vector Quantization (LVQ) untuk Mengoperasikan Kursor Komputer," *TRANSMISI*, vol. 13, no. 3, pp. 82-86, 2012.
- [3] Y. N. Utami, R. Rumani, N. Anbaranti, "Perancangan Speaker Recognition pada Sistem Kendali Lampu Berbasis Mikrokontroler," *e-Proceeding of Engineering*, vol. 2, no. 2, pp. 3332-3346, 2015.
- [4] W. Swastika, "Quickprop Method to Speed up Learning Process of Artificial Neural Network in Money's Nominal Value Recognition Case," *AIP Publishing*, vol. 1825, no. 1, pp. 1-6, 2017.
- [5] L. University, "Corpora and Language Statistics," *Deutscher Wortschatz*, 1998 - 2018. [Online]. Available: http://cls.corpora.uni-leipzig.de/en/ind_mixed_2013. [Accessed 3 Maret 2018].
- [6] A. D. Andriana, "Perangkat Lunak Untuk Membuka Aplikasi Pada Komputer Dengan Perintah Suara Menggunakan Metode Mel Frequency Cepstrum Coefficients," *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, vol. 2 No. 1, 2013.
- [7] D. Putra, A. Resmawan, "Verifikasi Biometrika Suara Menggunakan Metode MFCC dan DTW," *Lontar Komputer*, vol. 2 No. 1, pp. 8-21, 2011.
- [8] F. Y. Leu, G. L. Lin, "An MFCC-based speaker identification system," *International Conference on Advanced Information AINA*, pp. 1055-1062, 2017.
- [9] Z. Ramadhan, S. N. Endah, "Perintah Suara Berbahasa Indonesia untuk Membuka dan Menutup Aplikasi dalam Sistem Operasi Windows Menggunakan Metode Mel Frequency Cepstrum Coefficient dan Metode Backpropagation," *Seminar Nasional Ilmu Komputer*, pp. 33-41, 2016.
- [10] P. D. Sugiyono, *Metode Penelitian Pendidikan (Pendekatan Kuantitatif, Kualitatif, dan R&D)*, Bandung: Alfabeta, 2013.