

**IMPLEMENTASI
METODE JARINGAN SARAF TIRUAN *BACKPROPAGATION*
PADA PENGENALAN SUARA MANUSIA**

SKRIPSI

**OLEH:
MOHAMMAD BAGUS DIMAS PRAYUGO
NIM. 19610001**



**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

**IMPLEMENTASI
METODE JARINGAN SARAF TIRUAN *BACKPROPAGATION*
PADA PENGENALAN SUARA MANUSIA**

SKRIPSI

**Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat.)**

**Oleh
MOHAMMAD BAGUS DIMAS PRAYUGO
NIM. 19610001**

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

**IMPLEMENTASI
METODE JARINGAN SARAF TIRUAN *BACKPROPAGATION*
PADA PENGENALAN SUARA MANUSIA**


SKRIPSI

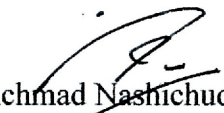
**Oleh
Mohammad Bagus Dimas Prayugo
NIM. 19610001**

Telah Diperiksa dan Disetujui Untuk Diuji
Malang, 21 Juni 2023

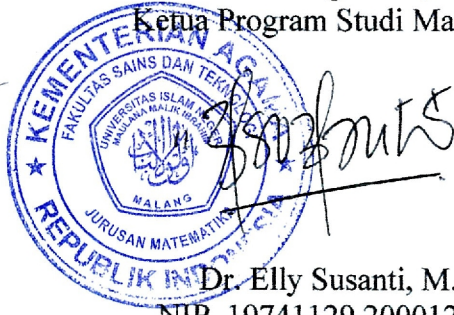
Dosen Pembimbing I,

Dosen Pembimbing II,


Hisyam Fahmi, M.Kom.
NIP. 19890727 201903 1 018


Achmad Nashichuddin, M.A.
NIP. 19730705 200003 1 002

Mengetahui,
Ketua Program Studi Matematika



Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005




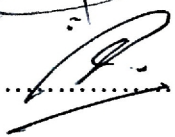
**IMPLEMENTASI
METODE JARINGAN SARAF TIRUAN *BACKPROPAGATION*
PADA PENGENALAN SUARA MANUSIA**

SKRIPSI

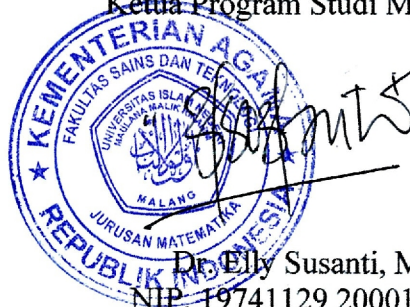
Oleh
Mohammad Bagus Dimas Prayugo
NIM. 19610001

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)

Tanggal 23 Juni 2023

Ketua Penguji	: Juhari, M.Si.	
Anggota Penguji 1	: Mohammad Nafie Juhari, M.Si.	
Anggota Penguji 2	: Hisyam Fahmi, M.Kom.	
Anggota Penguji 3	: Achmad Nashichuddin, M.A.	

Mengetahui,
Ketua Program Studi Matematika



Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005

HALAMAN PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Mohammad Bagus Dimas Prayugo
NIM : 19610001
Program Studi : Matematika
Fakultas : Sains dan Teknologi
Judul Skripsi : Implementasi Metode Jaringan Saraf Tiruan

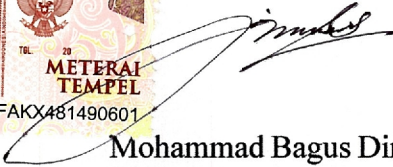
Backpropagation Pada Pengenalan Suara Manusia

Menyatakan dengan sebenar-benarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan dan pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 23 Juni 2023

Yang membuat pernyataan,




Mohammad Bagus Dimas Prayugo
NIM. 19610001

HALAMAN MOTO

“Sepi Ing Pamrih, Rame Ing Gawe, Banter Tan Mblancangi, Dhuwur Tan
Ngungkuli”

HALAMAN PERSEMBAHAN

Bismillahirrahmanirrahim, dengan mengucapkan syukur kepada Allah SWT. skripsi ini penulis persembahkan kepada ibunda tercinta Nuryati, ayahanda tercinta Ridwan, Kakak tersayang Siti Nur Rofiqoh dan Edi Susilo yang senantiasa mendoakan, memberikan motivasi, nasihat dan dukungan kepada penulis baik materiil maupun moril.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah, segala puji dan syukur senantiasa penulis panjatkan kepada Allah *subhanahu wa ta'ala* atas segala rahmat, taufik, dan hidayah yang telah diberikan sehingga penulis dapat menyelesaikan proposal skripsi dengan judul “Implementasi Metode Jaringan Saraf Tiruan *Backpropagation* Pada Pengenalan Suara Manusia. Shalawat dan salam senantiasa tercurahkan kepada Nabi Muhammad *shallallahu 'alaihi wa sallam* yang telah membimbing manusia dalam jalan kebenaran, yakni agama Islam dan dinantikan syafaat beliau kelak di akhirat.

Dalam proses penyelesaian proposal skripsi ini, penulis banyak mendapat bimbingan, dukungan, dan arahan serta sumbangsi dari berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. M. Zainuddin, MA. selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si. selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Elly Susanti, M.Sc. selaku Ketua Program Studi Matematika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Hisyam Fahmi, M.Kom. selaku Dosen Pembimbing I yang telah memberikan berbagai pengetahuan, pengalaman, arahan, nasihat, serta motivasi kepada penulis.
5. Achmad Nashichuddin, M.A. selaku Dosen Pembimbing II yang telah memberikan ilmu, nasihat, bimbingan, pengalaman, serta motivasi kepada penulis.
6. Juhari, M.Si. selaku Ketua Penguji dalam ujian skripsi, yang telah memberikan arahan, kritik, saran dan pengetahuan yang bermanfaat kepada penulis.
7. Mohammad Nafie Jauhari, M.Si, selaku Dosen Anggota Penguji I yang telah memberikan berbagai arahan, kritik dan saran serta ilmu yang bermanfaat kepada penulis.

8. Seluruh dosen Program Studi Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
9. Kedua orang tua serta seluruh keluarga yang senantiasa mendoakan, memberikan semangat, dukungan, nasihat, serta kasih sayang sehingga penulis dapat menyelesaikan tugas akhir.
10. Rekan penulis yang berjuang bersama dalam mengerjakan skripsi yang senantiasa memberikan dukungan dan semangat kepada penulis.
11. Seluruh teman-teman yang telah bersedia meluangkan waktunya sebagai partisipan dalam pengambilan data penelitian.
12. Teman-teman penulis khususnya KKM DR ~Vacation yang senantiasa memberikan semangat, dukungan dan bantuan baik secara langsung maupun tidak langsung.
13. Seluruh teman-teman Program Studi Matematika angkatan 2019 yang senantiasa memberikan bantuan, dukungan dan semangat kepada penulis dalam berbagai kondisi.

Semoga Allah *subhanahu wa ta'ala* selalu memberikan balasan atas segala bantuan dan kebaikan yang telah diberikan kepada penulis. Penulis berharap agar laporan ini dapat bermanfaat bagi penulis serta pembaca untuk menambah wawasan keilmuan yang selalu berkembang.

Wassalamu 'alaikum Warahmatullahi Wabarakatuh

Malang, 23 Juni 2023

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
HALAMAN PERNYATAAN KEASLIAN TULISAN	v
HALAMAN MOTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xiv
ABSTRAK	xv
ABSTRACT	xvi
مستخلص البحث	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Tujuan Penelitian	5
1.4 Manfaat Penelitian	5
1.5 Batasan Masalah	6
1.6 Definisi Istilah	7
BAB II KAJIAN TEORI	8
2.1 Teori Pendukung	8
2.1.1 Jaringan Saraf Tiruan	8
2.1.1.1 Arsitektur Jaringan Saraf Tiruan	9
2.1.1.2 Konsep Jaringan Saraf Tiruan	12
2.1.1.3 Fungsi Aktivasi Jaringan Saraf Tiruan	12
2.1.2 Jaringan Saraf Tiruan <i>Backpropagation</i>	13
2.1.3 <i>Linear Predictive Coding</i> (LPC)	20
2.1.4 <i>Fast Fourier Transform</i> (FFT)	23
2.1.5 <i>Mean Squared Error</i> (MSE)	25
2.1.6 <i>K-Fold Cross Validation</i>	25
2.1.7 Suara	26
2.1.8 Pengenalan Suara	29
2.1.9 Jaringan Saraf	31
2.2 Keragaman Suara Manusia dalam Perspektif Islam	32
2.3 Kajian Penelitian dengan Teori Pendukung	34
BAB III METODE PENELITIAN	36
3.1 Jenis Penelitian	36
3.2 Data dan Sumber Data	36
3.3 Perancangan Sistem	37
3.4 Teknik Analisis Data	37
3.5 Diagram Alir Penelitian	40
BAB IV HASIL DAN PEMBAHASAN	41

4.1	Pengumpulan Data	41
4.2	<i>Preprocessing</i> Data	41
4.2.1	<i>Filter</i> Suara	41
4.2.2	Ekstraksi Fitur Suara	42
4.2.3	Transformasi Sinyal Suara	51
4.2.4	Normalisasi Data	52
4.3	Proses <i>Training</i> Data	53
4.4	Proses <i>Testing</i> Data	63
4.5	Evaluasi	66
4.6	Implementasi Metode <i>JST Backpropagation</i> Dalam Pandangan Islam	70
BAB V PENUTUP		73
5.1	Kesimpulan.....	73
5.2	Saran.....	73
DAFTAR PUSTAKA		75
LAMPIRAN.....		78
RIWAYAT HIDUP		93

DAFTAR TABEL

Tabel 2.1	Interpretasi Nilai Akurasi.....	26
Tabel 2.2	Perbedaan Sinyal Analog dan Sinyal Diskrit.....	27
Tabel 4.1	<i>Output Array Filter</i> Suara.....	42
Tabel 4.2	<i>Output Array Preemphasis</i>	43
Tabel 4.3	<i>Output Array Frame Blocking</i>	44
Tabel 4.4	<i>Output Array Frame</i> Ke-2	45
Tabel 4.5	<i>Output Array Windowing Frame</i> Ke-2	46
Tabel 4.6	<i>Output Array Windowing</i>	47
Tabel 4.7	<i>Output Array</i> Autokorelasi	48
Tabel 4.8	<i>Output Array</i> Autokorelasi Frame Ke-2	49
Tabel 4.9	<i>Output</i> Koefisien LPC	50
Tabel 4.10	<i>Output Array</i> FFT	52
Tabel 4.11	<i>Output Array</i> Normalisasi.....	52
Tabel 4.12	Bobot Awal <i>Input Layer</i> ke <i>Hidden Layer</i>	53
Tabel 4.13	Bobot Awal <i>Hidden Layer</i> ke <i>Output Layer</i>	54
Tabel 4.14	Bias Awal <i>Input Layer</i> ke <i>Hidden Layer</i>	54
Tabel 4.15	Sinyal Masukan <i>Hidden Layer</i>	54
Tabel 4.16	Sinyal Masukan <i>Hidden Layer</i> dengan Fungsi Aktivasi	55
Tabel 4.17	Perbaikan Bobot w_{jk}	56
Tabel 4.18	Faktor Koreksi δ_{in_j}	57
Tabel 4.19	Faktor Koreksi δ_j	58
Tabel 4.20	Perbaikan Bobot v_{ij}	58
Tabel 4.21	Faktor Koreksi Δv_{0j}	59
Tabel 4.22	Bobot Baru v_{ij}	59
Tabel 4.23	Bias Baru v_{0j}	60
Tabel 4.24	Bobot Baru w_{jk}	61
Tabel 4.25	Percobaan Pelatihan Jaringan Saraf Tiruan	61
Tabel 4.26	Sinyal Masukan <i>Hidden Layer</i>	63
Tabel 4.27	Sinyal Masukan <i>Hidden Layer</i> dengan Fungsi Aktivasi	64
Tabel 4.28	Hasil Pengujian Model Arsitektur Jaringan Saraf Tiruan.....	65
Tabel 4.29	Pembagian <i>Training</i> dan <i>Testing</i> dengan <i>Cross Validation</i>	66
Tabel 4.30	Hasil Proses <i>Cross Validation</i>	70

DAFTAR GAMBAR

Gambar 2.1	Jaringan Saraf Tiruan <i>Single Layer</i>	10
Gambar 2.2	Jaringan Saraf Tiruan <i>Multi Layer</i>	11
Gambar 2.3	Jaringan Saraf Tiruan Lapisan Kompetitif	11
Gambar 2.4	Model Jaringan Saraf Tiruan <i>Backpropagation</i>	14
Gambar 2.5	Bentuk Sinyal Analog	27
Gambar 2.6	Bentuk Sinyal Digital	28
Gambar 2.7	Anatomi Pembentukan Suara Pada Manusia	28
Gambar 2.8	Blok Diagram Proses Pembelajaran Pola	30
Gambar 2.9	Blok Diagram Proses Pengenalan Pola	30
Gambar 2.10	Struktur Jaringan Saraf Manusia	32
Gambar 3.1	Diagram Alir Penelitian	40
Gambar 4.1	Sinyal Suara Asli dengan Sinyal Suara <i>Filter</i>	41
Gambar 4.2	Sinyal Suara <i>Filter</i> Dengan Sinyal Suara <i>Preemphasis</i>	43
Gambar 4.3	<i>Frame Blocking</i> Sinyal Suara	44
Gambar 4.4	Sinyal Suara pada <i>Frame</i> Ke-2	45
Gambar 4.5	<i>Windowing</i> Sinyal Suara <i>Frame</i> Ke-2	46
Gambar 4.6	<i>Windowing</i> Sinyal Suara	46
Gambar 4.7	Autokorelasi Sinyal Suara	48
Gambar 4.8	Autokorelasi Sinyal Suara <i>Frame</i> Ke-2	49
Gambar 4.9	Koefisien LPC Sinyal Suara	50
Gambar 4.10	FFT Sinyal Suara	51
Gambar 4.11	Model Arsitektur JST 64-15-1-1	62
Gambar 4.12	Hasil <i>Training</i> JST Pada MATLAB	62
Gambar 4.13	Hasil <i>Testing</i> JST Pada MATLAB	65
Gambar 4.14	<i>Training</i> pada <i>Fold</i> 1	67
Gambar 4.15	<i>Testing</i> pada <i>Fold</i> 1	67
Gambar 4.16	<i>Training</i> pada <i>Fold</i> 2	68
Gambar 4.17	<i>Testing</i> pada <i>Fold</i> 2	68
Gambar 4.18	<i>Training</i> pada <i>Fold</i> 3	68
Gambar 4.19	<i>Testing</i> pada <i>Fold</i> 3	68
Gambar 4.20	<i>Training</i> pada <i>Fold</i> 4	69
Gambar 4.21	<i>Testing</i> pada <i>Fold</i> 4	69
Gambar 4.22	<i>Training</i> pada <i>Fold</i> 5	69
Gambar 4.23	<i>Testing</i> pada <i>Fold</i> 5	70

DAFTAR LAMPIRAN

Lampiran 1	<i>Array</i> Data Suara Asli	78
Lampiran 2	<i>Array Filter</i> Suara.....	78
Lampiran 3	<i>Array Preemphasis</i>	78
Lampiran 4	<i>Array Frame Blocking</i>	78
Lampiran 5	<i>Array Frame Blocking</i> pada <i>Frame</i> ke-2.....	78
Lampiran 6	<i>Array Windowing</i> pada <i>Frame</i> ke-2	78
Lampiran 7	<i>Array Windowing</i>	78
Lampiran 8	<i>Array</i> Autokorelasi	78
Lampiran 9	<i>Array</i> Autokorelasi pada <i>Frame</i> ke-2.....	78
Lampiran 10	<i>Array</i> Analisa LPC	79
Lampiran 11	<i>Array</i> FFT Suara.....	79
Lampiran 12	<i>Array</i> Normalisasi Satu Suara	79
Lampiran 13	<i>Array</i> Normalisasi Data Suara.....	79
Lampiran 14	Bobot Awal v_{ij}	79
Lampiran 15	Bobot Awal w_{jk}	79
Lampiran 16	Bias Awal v_{0j}	79
Lampiran 17	Sinyal <i>Input</i> $z_{in,j}$ pada <i>Training</i>	79
Lampiran 18	Sinyal <i>Input</i> Z_j pada <i>Training</i>	79
Lampiran 19	Perbaikan Bobot W_{jk}	79
Lampiran 20	Faktor Koreksi $\delta_{in,j}$	80
Lampiran 21	Faktor Koreksi δ_j	80
Lampiran 22	Perbaikan Bobot v_{ij}	80
Lampiran 23	Faktor Koreksi Δv_{0j}	80
Lampiran 24	Bobot Baru v_{ij}	80
Lampiran 25	Bias Baru v_{0j}	80
Lampiran 26	Bobot Baru w_{jk}	80
Lampiran 27	Sinyal <i>Input</i> $z_{in,j}$ pada <i>Testing</i>	80
Lampiran 28	Sinyal <i>Input</i> Z_j pada <i>Testing</i>	80
Lampiran 29	Hasil Pelatihan Model Terbaik Data Latih	81
Lampiran 30	Hasil Pengujian Model Arsitektur Jaringan Saraf Tiruan	85
Lampiran 31	<i>Script Code</i> MATLAB	86

ABSTRAK

Prayugo, Mohammad Bagus Dimas. 2023. **Implementasi Metode Jaringan Saraf Tiruan *Backpropagation* Pada Pengenalan Suara Manusia**. Skripsi. Program Studi Matematika, Fakultas Sains dan teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Hisyam Fahmi, M.Kom. (II) Achmad Nashichuddin, M.A.

Kata Kunci: *Backpropagation*, *Fast Fourier Transform*, Jaringan Saraf Tiruan, *Linear Predictive Coding*, Pengenalan Suara

Pengenalan suara merupakan suatu proses identifikasi suara dengan menggunakan parameter tertentu yang diambil oleh penangkap suara. Perkembangan teknologi memunculkan suatu peristiwa yang membutuhkan model perhitungan dalam pengenalan suara yang berguna dalam ilmu pengetahuan. Model perhitungan didapatkan dengan bantuan sistem komputer. Salah satu dari sistem komputer adalah Jaringan Saraf Tiruan (JST). Sistem jaringan saraf tiruan meniru cara kerja biologis jaringan saraf. Salah satu metode jaringan saraf tiruan yang paling umum digunakan ialah *Backpropagation*. Penelitian ini dilakukan dengan menerapkan metode jaringan saraf tiruan *Backpropagation* untuk pengenalan suara manusia. Penelitian ini bertujuan untuk mengetahui model arsitektur dan tingkat akurasi yang didapatkan. Pada penelitian ini mengambil pengenalan suara dikarenakan suara memiliki pola yang cukup lengkap dan unik sehingga proses identifikasi sinyal suara dibantu dengan perhitungan yang dapat mengekstraksi ciri atau fitur suara. Suara dilakukan *filter* untuk menghilangkan *noise*. Digunakan *Linear Predictive Coding* (LPC) untuk ekstraksi fitur suara. Selanjutnya ekstraksi fitur suara dalam bentuk domain waktu ditransformasi menjadi bentuk domain frekuensi menggunakan *Fast Fourier Transform* (FFT). Data suara dibagi 80% sebagai data *training* dan 20% sebagai data *testing*. Pelatihan menggunakan data *training* untuk mendapatkan model arsitektur jaringan saraf tiruan yang sesuai dari beberapa model parameter yang berbeda. Pelatihan dilakukan dengan menghitung nilai optimal bobot dan bias jaringan agar dapat mengenali pola suara dengan baik. Didapatkan model terbaik arsitektur jaringan saraf tiruan 64-15-1-1. Model arsitektur jaringan yang didapat diuji dengan data *testing* untuk menguji arsitektur dalam mengenali pola suara. Perhitungan evaluasi menggunakan *K-Fold Cross Validation* untuk mengetahui tingkat akurasi model arsitektur jaringan. Didapatkan nilai akurasi sebesar 0,95 terhadap data *training* dan 0,088886 terhadap data *testing*. Model arsitektur jaringan saraf tiruan sangat baik dalam mengenali suara terhadap data *training* dan kurang baik dalam data *testing*. Diharapkan metode ini dapat membantu pada proses penelitian terkait pengenalan.

ABSTRACT

Prayugo, Mohammad Bagus Dimas. 2023. **The Implementation of Backpropagation Neural Network Method in Human Speech Recognition**. Undergraduate Thesis. Mathematics Department, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) Hisyam Fahmi, M.Kom. (II) Achmad Nashichuddin, M.A.

Keywords: Backpropagation, Fast Fourier Transform, Neural Network, Linear Predictive Coding, Speech Recognition

Speech recognition is a process of voice identification using specific parameters taken by the sound catcher. The technology development gave rise to a problem that requires a calculation model in speech recognition to be useful in science. The calculation model is obtained by using a computer system. One of the computer systems is the Artificial Neural Network (ANN). Artificial neural network systems are imitate the biological workings of neural networks. One of the most generally used artificial neural network methods is Backpropagation. This research was performed by applying the Backpropagation artificial neural network method for human voice recognition. In this research, voice recognition is used because the voice has a sufficient complete and unique pattern so that the process of identifying voice signals is assisted by calculations which can extract the characteristics or features of the voice. The voice is filtered to remove noise. Linear Predictive Coding (LPC) is used for voice feature extraction. Then, the extracted voice features in the time domain are transformed into the frequency domain by fast Fourier transform (FFT). Voice data is divided 80% as training data and 20% as testing data. Training uses training data to get the suitable artificial neural network architecture model from a few different parameter models. Training performed by calculating the optimal value of the weight and bias of the network to recognize sound patterns well. The best model of artificial neural network architecture 64-15-1-1 is obtained. The network architecture model obtained was tested with testing data to test the architecture in recognizing voice patterns. The evaluation calculation is using K-Fold Cross Validation to determine the accuracy of the network architecture model. An accuracy value of 0.95 was obtained for the training data and 0.088886 for the testing data. The artificial neural network architecture model is best at recognizing voices on training data and poorly on testing data. Hopefully, this method can help in the research process related to recognition.

مستخلص البحث

فرايوكا، محمد باكوس ديماس. ٢٠٢٣. تنفيذ طريقة الشبكة العصبية الاصطناعية باستخدام الانتشار العكسي في التعرف على صوت الإنسان. البحث الجمعي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشر ف: (١) هثام فهمي، الماجستير، (٢) احمد ناصح الدين، الماجستير

الكلمات المفتاحية: الانتشار العكسي، التحويل السريع لفورييه، الشبكة العصبية، التشفير التنبؤي الخطي، التعرف على الصوت

التعرف على الصوت هو عملية تعريف الصوت باستخدام المعيار المعينة يتخذها ماسك الصوت. أدى تطور التكنولوجيا إلى ظهور حدث الذي يتطلب نمودجا حسابيا في التعرف على الكلام يكون مفيدا في العلوم. يتم الحصول على نمودج الحساب بمساعدة نظام الكمبيوتر. أحد أنظمة الكمبيوتر هو الشبكة العصبية الاصطناعية (*JST*). يحاكي نظام الشبكة العصبية الاصطناعية الأعمال البيولوجية للشبكات العصبية. واحدة من أكثر طرق الشبكات العصبية الاصطناعية شيوعا هي الانتشار العكسي. تم إجراء هذا البحث من خلال تطبيق طريقة الشبكة العصبية الاصطناعية الانتشار العكسي للتعرف على الكلام البشري. في هذه الدراسة، يتم أخذ التعرف على الصوت لأن الصوت له نمط كامل وفريد إلى حد ما بحيث يتم مساعدة عملية تحديد الإشارات الصوتية من خلال الحسابات التي يمكن أن تستخرج خصائص أو ميزات الصوت. تمت تصفية الصوت لإزالة الضوضاء. يستخدم الترميز التنبؤي الخطي (*LPC*) لاستخراج ميزات الصوت. علاوة على ذلك، يتم تحويل استخراج ميزات الصوت في شكل مجالات زمنية إلى شكل مجالات تردد باستخدام تحويل فورييه السريع (*FFT*). يتم تقسيم البيانات الصوتية بنسبة ٨٠٪ كبيانات تدريب و ٢٠٪ كبيانات اختبار. يستخدم التدريب بيانات التدريب للحصول على نمودج بنية الشبكة العصبية الاصطناعية المقابلة من عدة نماذج معلمات مختلفة. يتم التدريب عن طريق حساب القيمة المثلى لوزن الشبكة والتحيز من أجل التعرف على أنماط الصوت بشكل جيد. تم الحصول على أفضل نمودج لبنية الشبكة العصبية الاصطناعية ٦٤-١٥-١-١. يتم اختبار نمودج بنية الشبكة الذي تم الحصول عليه من خلال اختبار البيانات لاختبار البنية في التعرف على أنماط الصوت. يستخدم حساب التقييم كضعاف التحقق المتقاطع لتحديد مستوى دقة نمودج بنية الشبكة. تم الحصول على قيمة دقة ٠,٩٥ لبيانات التدريب و ٠,٨٨٨٨٦ لبيانات الاختبار. نمودج بنية الشبكة العصبية الاصطناعية جيد جدا في التعرف على الصوت مقابل بيانات التدريب وأقل جودة في اختبار البيانات. من المأمول أن تساعد هذه الطريقة في عملية البحث المتعلقة بالتعرف.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan dalam bidang teknologi pada zaman sekarang disebut sebagai inti perkembangan dari hampir seluruh aspek kehidupan. Pesatnya perkembangan teknologi tidak terlepas dari adanya pengaruh dari komputer. Sistem pada komputer dapat membuat semua berjalan menjadi lebih cepat dan efisien, sehingga dalam proses pengolahan informasi komputer menjadi alat yang sangat penting. Salah satu dari sistem komputer ini adalah Jaringan Saraf Tiruan (JST). *Artificial Neural Network* (ANN) dalam istilah bahasa Indonesia disebut sebagai Jaringan Saraf Tiruan (JST) merupakan salah satu metode pengolahan data yang terdiri dari kelompok unit pemrosesan kecil yang dimodelkan sesuai jaringan saraf pada manusia (Solikhun dkk., 2017). Sistem jaringan saraf tiruan meniru dari cara kerja biologis jaringan saraf. Neuron buatan dirancang untuk meniru karakteristik neuron biologis (Yolanda dkk., 2018). Diperlukan data dari masalah pada jaringan saraf tiruan yang akan diselesaikan dan kemudian informasi yang didapat dari data diseleksi melalui pelatihan.

Teknik yang terdapat pada jaringan saraf tiruan banyak digunakan dalam berbagai macam bidang terutama pada sistem pengenalan pola citra, suara, dan sebagainya. Manusia dapat mengenali identitas seseorang, baik dari gerakan, wajah maupun dari suara dengan mudah. Misalkan ketika sedang berbicara melalui telepon, kita akan dengan mudah mengenali orang tersebut. Kemudahan tersebut timbul karena pada otak manusia terdapat suatu jaringan yang terdiri dari komponen

yang saling terhubung atau disebut sebagai neuron-neuron dalam memproses informasi yang diterimanya, akan tetapi jika permasalahan tersebut diselesaikan komputer, maka akan terdapat berbagai kesulitan yang ditimbulkan (Zaitun dkk., 2015).

Salah satu karunia yang diberikan oleh Allah SWT. kepada makhluk hidup ciptaan-Nya yakni adanya suara. Salah satu makhluk hidup ciptaan Allah SWT. ialah manusia. Allah SWT. menciptakan manusia dalam bentuk yang paling sempurna. Sehingga memiliki kelebihan seperti akal, pikiran, pendengaran serta suara. kesempurnaan manusia terdapat disebutkan oleh Allah SWT. dalam Al-Qur'an surah At-Tiin [95] ayat 4 (Kemenag, 2019) sebagai berikut:

Artinya: "Sesungguhnya Kami telah menciptakan manusia dalam bentuk yang sebaik-baiknya" (Q.S. At-Tiin: 4).

Salah satu bukti dari kesempurnaan manusia adalah dengan beragamnya suara manusia. Allah SWT. menciptakan manusia dengan berbagai macam suara, yang di mana tiap manusia diberikan beragam suara sehingga mempunyai ciri khas karakter masing-masing pada manusia (Fatman & Islamiyati, 2020).

Melalui suara, kita dapat berkomunikasi, memahami dan mengerti dari apa yang sedang diperbincangkan serta menyampaikan suatu informasi kepada orang lain. Suara merupakan salah satu objek yang digunakan untuk mengidentifikasi kepribadian seseorang. Dengan mendengarkan suara, manusia dapat mengenali kondisi seseorang tersebut, sebagai contoh identitas pembicara, jenis kelamin, gaya bahasa, aksen, emosi serta kondisi kesehatan. Seiring dengan berjalannya waktu dan kemajuan teknologi memunculkan suatu peristiwa yang membutuhkan model perhitungan untuk melakukan pengenalan terhadap suara

yang dapat berguna dalam ilmu pengetahuan, serta pembuatan aplikasi yang praktis seperti sistem keamanan menggunakan suara.

Salah satu pengaplikasian pada bidang teknologi yakni pengenalan pola. Hal ini selanjutnya berkaitan dengan bidang *computer vision* atau visi komputer, di mana mencoba untuk meniru kemampuan mata serta sistem otak manusia untuk membentuk dan melakukan interpretasi terhadap citra (Loppies, 2018). Dikarenakan pola pada jenis suara yang cukup lengkap dan unik, sehingga proses identifikasi sinyal suara dibantu dengan perhitungan yang dapat mengekstraksi ciri atau fitur sinyal suara. Dalam hal ini digunakan metode jaringan saraf tiruan untuk membantu dalam mengidentifikasi gambar pola, salah satunya gambar sinyal suara yang diubah menjadi bentuk grafik.

Berbagai macam sistem pengenalan suara saat ini dikembangkan oleh tiap negara di dunia dengan bermacam-macam bahasa. Hal ini membuat sistem pengenalan suara dapat diterapkan pada berbagai macam bidang, seperti keamanan berbasis suara dan pembelajaran berbasis suara. Pada sistem keamanan berbasis suara dinilai lebih aman dan efektif. Hal ini dikarenakan suara memiliki ciri khas masing-masing sehingga berbeda dengan suara orang yang lain sehingga lebih sulit untuk diretas. Begitu pula dengan sistem pembelajaran berbasis suara, sehingga memudahkan aktivitas seseorang dengan memberikan perintah melalui suara seperti *Google Assistant*.

Metode jaringan saraf tiruan yang paling umum digunakan salah satunya ialah metode *backpropagation*. Metode *backpropagation* merupakan metode yang dilakukan dengan berdasarkan pendekatan nilai hasil *output* terhadap nilai pembanding (Zaitun dkk., 2015). Berbagai aplikasi yang bisa digunakan dengan

jaringan saraf tiruan *backpropagation*, diantaranya ialah dalam hal pengenalan suara. Penggunaan metode ini merujuk pada penelitian yang telah dilakukan oleh Avisena Abdillah Alwi, Putra Pandu Adikara, dan Indriati (Alwi dkk., 2020). Dalam penelitian tersebut data yang digunakan bersumber dari *Mozilla Common Voice* dan *Free ST American English Corpus* dan atribut yang digunakan hanya jenis kelamin dan umur. Dari hasil pengujian jenis kelamin dan rentang umur dengan data dari *Mozilla Common Voice* mendapatkan akurasi sebesar 0,18357. Sedangkan ketika pengujian untuk klasifikasi jenis kelamin saja didapat akurasi data dari *Mozilla Common Voice* sebesar 0,62504 dan data dari *Free ST American English Corpus* sebesar 0,9349. Penelitian lainnya dilakukan oleh Odi Nurdiawan (Nurdiawan, 2018). Dalam penelitian tersebut data yang digunakan yakni sandi morse suara berupa tiupan peluit. Data sampel yang digunakan berupa sandi morse dalam angka 0-9. Terdapat secara keseluruhan sebanyak 350 sampel suara sebagai data *training* dan sebanyak 150 sampel suara sebagai data *testing*. Dalam penelitian tersebut, didapatkan hasil pelatihan model sebesar 85,16% dan hasil pengujian sebesar 67,99%.

Oleh karena itu, pada penelitian ini menggunakan metode jaringan saraf tiruan *backpropagation* di mana memiliki nilai akurasi yang cukup tinggi. Pada penelitian ini akan digunakan *software* MATLAB dalam proses penelitian dan menggunakan proses tambahan yakni *Fast Fourier Transform* yang belum pernah dipakai pada penelitian-penelitian sebelumnya. Penggunaan *Fast Fourier Transform* disini bertujuan untuk transformasi dari sinyal dalam domain waktu menjadi sinyal dalam domain frekuensi sehingga mempermudah dalam proses identifikasi. Diharapkan dengan dilakukannya penelitian ini dapat diperoleh nilai

akurasi yang lebih baik daripada penelitian-penelitian sebelumnya. Selain itu, dapat berkontribusi untuk pengembangan aplikasi praktis berbasis suara kedepannya.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas maka diambil rumusan masalah sebagai berikut:

1. Bagaimana model arsitektur metode jaringan saraf tiruan *backpropagation* dalam mengenali suara manusia?
2. Bagaimana akurasi dari metode jaringan saraf tiruan *backpropagation* dalam mengenali suara manusia?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas maka tujuan penelitian ini adalah sebagai berikut:

1. Mengetahui model arsitektur metode jaringan saraf tiruan *backpropagation* dalam mengenali suara manusia.
2. Mengetahui akurasi dari metode jaringan saraf tiruan *backpropagation* dalam mengenali suara manusia.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini berdasarkan uraian-uraian di atas dipaparkan sebagai berikut:

1. Manfaat Teoritis

- a. Mampu merancang model arsitektur jaringan saraf tiruan *backpropagation* dalam pengenalan suara manusia.
 - b. Mampu mengetahui proses dan kinerja jaringan saraf tiruan *backpropagation* dalam pengenalan suara manusia.
 - c. Dapat mengetahui ciri khas atau fitur setiap suara manusia.
2. Manfaat Praktis
- a. Bagi Program Studi, metode ini dapat digunakan sebagai wawasan ilmu pengetahuan dan bahan pembelajaran bagi mahasiswa terkait jaringan saraf tiruan *backpropagation*.
 - b. Bagi pembaca, metode ini dapat memberikan informasi dan referensi terkait penerapan jaringan saraf tiruan *backpropagation* serta penerapan dalam aplikasi pengenalan suara.

1.5 Batasan Masalah

Batasan masalah yang digunakan pada penelitian ini adalah sebagai berikut:

1. Perekaman suara dilakukan dengan memakai *Smartphone* dan file rekaman disimpan dalam format WAV.
2. Durasi rata-rata rekaman 2 detik.
3. Perancangan program menggunakan *Software* MATLAB.
4. Data yang digunakan yakni suara dari 15 partisipan dengan masing-masing partisipan sebanyak 15 suara.
5. Data suara terdiri dari 11 suara pria dan 4 suara wanita.

1.6 Definisi Istilah

Terdapat beberapa istilah yang digunakan pada penelitian ini sebagai berikut:

- Data Training* : Data yang digunakan untuk melatih algoritma program untuk mendapatkan model yang sesuai.
- Data Testing* : Data yang digunakan untuk menguji model dan mengetahui tingkat keakuratan.
- Learning Rate* : Salah satu parameter *training* yang digunakan untuk menghitung nilai koreksi bobot pada proses pelatihan.
- Preprocessing* : Proses pengolahan data untuk mengubah data mentah ke dalam bentuk yang lebih mudah dipahami sistem.

BAB II

KAJIAN TEORI

2.1 Teori Pendukung

2.1.1 Jaringan Saraf Tiruan

Jaringan saraf tiruan pertama kali didesain oleh Warren McCulloch dan Walter Pitts pada tahun 1943. Konsep dasar jaringan saraf tiruan yakni dengan mengaplikasikan model matematik yang diterapkan seperti fungsi kinerja neuron pada otak (Harum, 2018). Pengembangan metode seperti saraf biologi ini merupakan suatu kemajuan pada ilmu komputer (Kusumadewi, 2003).

Jaringan saraf tiruan (JST) adalah metode pemisahan dan pengelompokan data yang mempunyai sistem kerja seperti jaringan saraf pada manusia. Jaringan saraf tiruan dibuat untuk menyelesaikan permasalahan, seperti identifikasi atau klasifikasi pola melalui proses pelatihan. Jaringan saraf tiruan atau *artificial neural network* merupakan bentuk representasi buatan otak manusia dalam menggambarkan proses pembelajaran otak manusia (Adinugraha, 2016). Pengertian buatan tersebut dipakai karena pada jaringan saraf diaplikasikan dengan bantuan program komputer sehingga dapat menyelesaikan proses pembelajaran.

Model jaringan saraf tiruan meniru bentuk jaringan saraf pada tubuh manusia. Rata-rata orang menggunakan jaringan saraf tiruan disebabkan struktur sistemnya dapat berubah-ubah ketika informasi yang didapat berubah, baik dari segi internal maupun dari segi eksternal yang diolah pada sistem tersebut. Pembuatan sistem jaringan saraf tiruan dimaksudkan untuk memungkinkan komputer dapat mengidentifikasi bentuk, pola, atau struktur tertentu. Hal ini dikarenakan pada

sistem komputer tidak memiliki kecerdasan, walaupun sebenarnya suatu sistem komputer dapat melakukan perhitungan, seperti dalam waktu singkat mengidentifikasi wajah manusia daripada manusia itu sendiri (Adinugraha, 2016).

Pada dasarnya, prinsip kerja pada jaringan saraf tiruan dalam proses perhitungan dapat menghitung semua fungsi yang dapat dihitung. Jaringan saraf tiruan dapat melakukan proses yang dilakukan oleh komputer digital pada umumnya. Implementasi jaringan saraf tiruan biasanya dipakai dalam melakukan klasifikasi serta masalah yang dapat mentolerir ketidakakuratan (Effendi, 2013). Seperti banyaknya data pelatihan yang dimiliki, akan tetapi mempunyai peraturan yang tidak bisa diterapkan dengan mudah.

Selain itu, jaringan saraf tiruan memiliki kemampuan untuk mengambil pola yang telah dipelajari dengan sempurna. Dari proses tersebut didapatkan pula tanggapan yang serupa dan diterima dengan baik terhadap pola yang telah dipelajari sebelumnya. Oleh sebab itu, jaringan saraf tiruan dapat bermanfaat ketika suatu informasi baru dimasukkan, maka jaringan saraf tiruan akan menerima dan menanggapi informasi tersebut dengan baik. Dan dari proses tersebut didapatkan pula hasil yang paling mendekati dari informasi semula.

2.1.1.1 Arsitektur Jaringan Saraf Tiruan

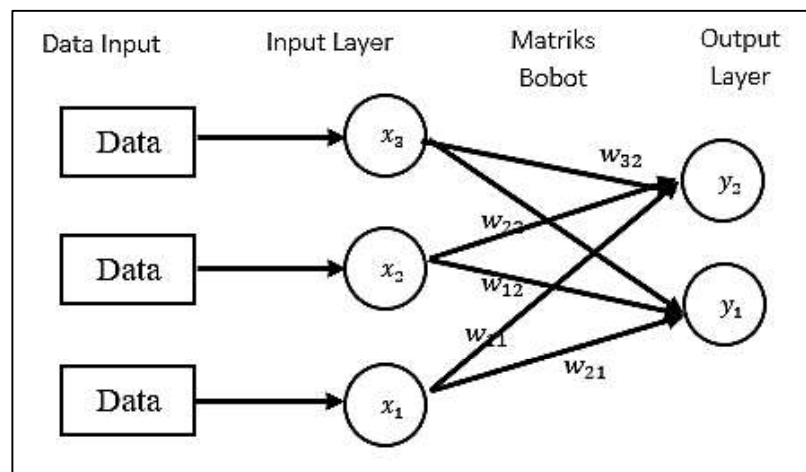
Arsitektur suatu jaringan berperan penting untuk menentukan keberhasilan tujuan yang diperoleh karena dengan arsitektur yang sama tidak dapat menyelesaikan semua permasalahan. Sebagaimana telah disebutkan pada definisi diatas bahwa otak manusia terdiri dari sekitar 10^{10} unit neuron yang bekerja secara paralel dan bertukar informasi melalui konektornya. Dengan cara yang serupa, pada

jaringan saraf tiruan tersusun atas unit komputasi sederhana yang disebut sebagai “neuron buatan”, di mana setiap unit terhubung ke unit lain melalui konektor bobot (*weight*) dan kemudian unit-unit tersebut menghitung jumlah bobot yang telah ter-*input* dan dicari *output*-nya menggunakan fungsi aktivasi.

Jaringan saraf tiruan mempunyai beberapa arsitektur yang diterapkan pada berbagai macam aplikasi. Perancangan dilakukan dengan memakai sebuah aturan yang bersifat *universal* di mana seluruh model jaringan mempunyai konsep dasar yang sama (Humaini, 2015). Terdapat 3 lapisan pada arsitektur jaringan saraf tiruan, yakni sebagai berikut:

1. Jaringan dengan lapisan tunggal (*single layer*)

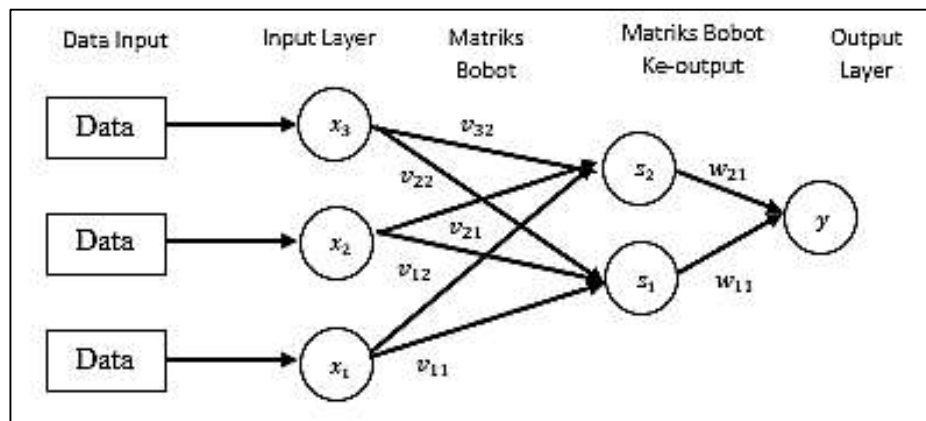
Jaringan *single layer* tersusun atas satu *input layer* dan satu *output layer*. Terdapat satu lapisan saja pada jaringan *single layer* dengan bobot-bobot saling terhubung. Tiap neuron atau unit pada *input layer* terhubung dengan *neural* pada *output layer*. Jaringan *single layer* hanya menerima *input*, kemudian mengolahnya menjadi *output* secara langsung tanpa melalui *hidden layer*.



Gambar 2.1 Jaringan Saraf Tiruan *Single Layer*

2. Jaringan dengan lapisan banyak (*multi layer*)

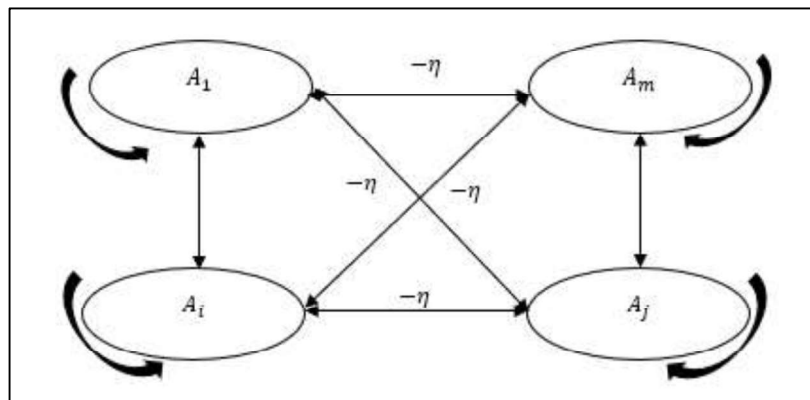
Jaringan *multi layer* tersusun lebih dari satu lapisan yang terdapat di antara *input layer* dan *output layer* (mempunyai minimal satu *hidden layer*). Kelebihan jaringan *multi layer* yakni dapat menyelesaikan permasalahan yang kompleks daripada jaringan *single layer*. Akan tetapi, pada proses pelatihan membutuhkan waktu yang cukup lama.



Gambar 2.2 Jaringan Saraf Tiruan *Multi Layer*

3. Jaringan dengan lapisan kompetitif

Umumnya, hubungan antar neuron pada lapisan ini tidak diperlihatkan pada diagram arsitektur. Pada jaringan lapisan kompetitif, sekumpulan neuron akan bersaing untuk menjadi neuron aktif yang digunakan pada pengolahan.



Gambar 2.3 Jaringan Saraf Tiruan Lapisan Kompetitif

2.1.1.2 Konsep Jaringan Saraf Tiruan

Konsep dasar pada jaringan saraf tiruan ditinjau berdasarkan skema interkoneksi dan kerangka kerja. Kerangka kerja pada jaringan saraf tiruan ditinjau berdasarkan jumlah *layer* atau lapisan dan banyaknya *node* pada tiap *layer*.

Terdapat 3 bagian lapisan pada jaringan saraf tiruan, yakni sebagai berikut (Agustin & Prahasto, 2012):

1. *Input Layer* (Lapisan Masukan)

Node-node pada *input layer* disebut sebagai unit masukan. Tiap unit masukan menerima pola masukan dari luar. *Input* yang dimasukkan adalah penggambaran dari suatu masalah.

2. *Hidden Layer* (Lapisan Tersembunyi)

Node-node pada *hidden layer* disebut sebagai unit tersembunyi. Pada *hidden layer* terdapat *output* yang secara langsung tidak dapat diamati.

3. *Output Layer* (Lapisan Keluaran)

Node-node pada *output layer* disebut sebagai unit *output*. Keluaran dari *layer* ini adalah solusi dari jaringan saraf tiruan terhadap suatu masalah.

2.1.1.3 Fungsi Aktivasi Jaringan Saraf Tiruan

Fungsi aktivasi pada jaringan saraf tiruan digunakan untuk menentukan *output* suatu neuron. Argumen fungsi aktivasi yakni *net input* yang terdiri dari gabungan linier *input* dan bobotnya (Siang, 2005). Berikut beberapa fungsi aktivasi jaringan saraf tiruan yang sering digunakan:

1. Fungsi *threshold* atau ambang batas

$$f(u) = \begin{cases} 1 & \text{jika } u \geq a \\ 0 & \text{jika } u < a \end{cases} \quad (2.1)$$

Fungsi ambang batas pada beberapa kasus dibuat tidak bernilai 0 atau 1, akan tetapi bernilai -1 atau 1. Sehingga

$$f(u) = \begin{cases} 1 & \text{jika } u \geq a \\ -1 & \text{jika } u < a \end{cases} \quad (2.2)$$

2. Fungsi *sigmoid*

$$f(u) = \frac{1}{1 + e^{-u}} \quad (2.3)$$

Dikarenakan letak nilai fungsi antara 0 dan 1 dan juga dapat diturunkan, maka fungsi *sigmoid* sering dipakai sebagai fungsi aktivasi.

$$f'(u) = f(u)(1 - f(u)) \quad (2.4)$$

3. Fungsi identitas

Penggunaan fungsi identitas terjadi apabila *output* yang dihasilkan jaringan saraf tiruan adalah sebarang bilangan riil.

$$f(u) = u \quad (2.5)$$

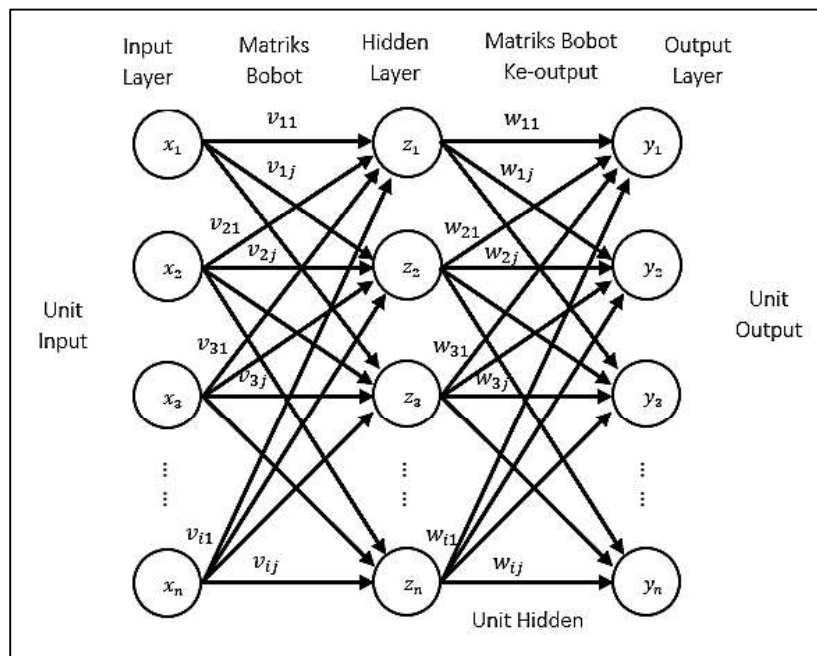
Dalam jaringan saraf tiruan terdapat beberapa metode dalam penggunaannya, yakni LVQ (*Linear Vector Quantization*), *perceptron*, *backpropagation*, dan *multilayer perceptron* (Arini, 2018). Adapun untuk penelitian yang akan dilakukan yakni dengan memakai metode perambatan galat mundur atau *backpropagation*,

2.1.2 Jaringan Saraf Tiruan *Backpropagation*

Backpropagation atau perambatan galat mundur merupakan metode jaringan saraf tiruan yang umum dipakai untuk meminimalkan dengan menyesuaikan bobot atau nilai berdasarkan selisih dari *output* yang diinginkan. *Backpropagation* merupakan salah satu metode sistematis dalam melatih banyak

lapisan jaringan saraf tiruan. Metode *backpropagation* mempunyai dasar matematis yang solid, objektif, serta algoritmanya berbentuk nilai koefisien dan persamaan dalam rumus dengan melalui model yang dikembangkan untuk meminimalkan jumlah kuadrat galat *error* (Loppies, 2018).

Model jaringan *backpropagation* tersusun atas *multi layer* atau banyak *layer* yang terdiri dari *input layer*, setidaknya satu *hidden layer* serta *output layer*. Pada tiap *layer* terdapat sejumlah neuron atau unit pemrosesan data atau yang menjalankan fungsinya masing-masing. Kemungkinan beberapa implementasi jaringan membutuhkan beberapa lapisan tersembunyi, tetapi dengan memakai satu lapisan tersembunyi saja sudah cukup untuk digunakan pada berbagai macam penerapan. Adapun sebagai contoh yakni terdapat pada gambar 2.4 berikut.



Gambar 2.4 Model Jaringan Saraf Tiruan *Backpropagation*

Metode *backpropagation* melakukan pelatihan terhadap jaringan. Hal tersebut dilakukan untuk memperoleh keseimbangan tiap jaringan dalam mengenali

pola yang digunakan. Kemampuan jaringan dalam merespon pola *input* yang benar mirip dengan pola yang dipakai ketika proses pelatihan. Pada penelitian ini menggunakan metode jaringan saraf tiruan *backpropagation*. Pada metode ini tersusun atas *input layer*, *hidden layer*, *hidden neuron*, fungsi aktivasi, dan jumlah iterasi atau *epoch*.

Berikut beberapa Langkah-langkah yang dilakukan pada metode *backpropagation*:

1. Inisialisasi bobot dan bias.

Bobot dan bias dilakukan Inisialisasi awal secara acak dengan nilai acak mendekati 0.

2. Proses *feedforward*

Setiap *input* ($x_i, i = 1, 2, \dots, n$) menerima sinyal *input* x_i kemudian sinyal diteruskan pada *hidden layer* ke seluruh unit. Data *training* digunakan sebagai data *input*. Setiap *hidden* $Z_j, j = 1, 2, \dots, p$ akan menambahkan bobot dan bias sinyal *input* dengan persamaan (2.6) berikut.

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.6)$$

Dimana:

z_{in_j} : Sinyal keluaran *hidden layer*

x_i : Neuron *input*

v_{0j} : bias *input layer*

v_{ij} : bobot *input layer*

Dalam menghitung sinyal keluaran *hidden unit* menggunakan fungsi aktivasi *sigmoid*, kemudian mengirim sinyal *output* ke semua unit pada *output* dengan persamaan (2.7).

$$Z_j = f(z_{in_j}) \quad (2.7)$$

Tiap *output* ($y_k, k = 1, 2, \dots, m$) menambahkan bobot dan bias sinyal *input* dengan persamaan (2.8).

$$y_{in_k} = w_{0k} + \sum_{j=1}^n w_{jk} Z_j \quad (2.8)$$

Dimana:

y_{in_k} : Sinyal keluaran *output layer*

w_{0k} : Bobot bias ke *output layer*

Z_j : Hasil fungsi aktivasi *hidden layer*

w_{jk} : Bobot *output layer*

Untuk menghitung sinyal keluaran dari *hidden unit* dengan fungsi aktivasi *purelin* atau identitas, kemudian mengirim sinyal *output* ke seluruh unit dalam unit *output* dengan persamaan (2.9).

$$Y_k = f(y_{in_k}) \quad (2.9)$$

3. Proses *backpropagation*

Setiap *output* ($Y_k, k = 1, 2, \dots, m$) menerima *target pattern* (*desired output*) yang sesuai dengan *input training pattern* guna menghitung kesalahan atau *error* tiap target dengan *output* yang didapatkan oleh jaringan dengan persamaan (2.10).

$$\delta_k = (t_k - Y_k)Y_k(1 - Y_k) \quad (2.10)$$

Keterangan:

δ_k : Faktor koreksi *error output layer*

t_k : Target

Y_k : Hasil *output training*

Menghitung perubahan bobot W_{jk} dengan persamaan (2.11).

$$\Delta W_{jk} = \alpha \delta_k Z_j \quad (2.11)$$

Keterangan:

ΔW_{jk} : Perubahan bobot *output layer*

α : *Learning rate*

δ_k : Faktor koreksi *error output layer*

Z_j : Hasil fungsi aktivasi *hidden layer*

Menghitung perubahan bias W_{0k} dengan persamaan (2.12).

$$\Delta W_{0k} = \alpha \delta_k \quad (2.12)$$

Keterangan:

ΔW_{0k} : Perubahan bias *output layer*

$\alpha \delta_k$: *Learning rate*

δ_k : Faktor koreksi *error output layer*

Menghitung faktor koreksi *input hidden layer* dengan persamaan (2.13).

$$\delta_{in_j} = \delta_k w_{jk} \quad (2.13)$$

Keterangan:

δ_{in_j} : Faktor koreksi *input hidden layer*

δ_k : Faktor koreksi *error output layer*

w_{jk} : Bobot *hidden layer* ke *output layer*

Menghitung faktor koreksi *error hidden unit* dengan persamaan (2.14).

$$\delta_j = \delta_{in_j} Z_j (1 - Z_j) \quad (2.14)$$

Keterangan:

δ_j : Faktor koreksi *error hidden layer*

δ_{in_j} : Faktor koreksi *input hidden layer*

Z_j : Hasil fungsi aktivasi *hidden layer*

Menghitung koreksi bobot *hidden layer* dengan persamaan (2.15).

$$\Delta v_{ij} = \alpha \delta_j x_j \quad (2.15)$$

Keterangan:

Δv_{ij} : perubahan bobot *hidden layer*

α : *Learning rate*

δ_j : Faktor koreksi *error unit*

x_j : Nilai *input*

Menghitung koreksi bias *hidden layer* dengan persamaan (2.16).

$$\Delta v_{0j} = \alpha \delta_j \quad (2.16)$$

Keterangan:

Δv_{0j} : Perubahan bias *layer*

α : *Learning rate*

δ_j : Faktor koreksi *error hidden unit*

4. Menghitung bobot dan bias baru

Perhitungan terhadap bobot baru *input layer* ke *hidden layer* dengan persamaan (2.17).

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.17)$$

Keterangan:

$v_{ij}(\text{baru})$: Bobot baru *input layer* ke *hidden layer*

$v_{ij}(\text{lama})$: Bobot lama *input layer* ke *hidden layer*

Δv_{ij} : Perubahan bobot *hidden layer*

Perhitungan terhadap bias baru *input layer* ke *hidden layer* dengan persamaan (2.18).

$$v_{0j}(\text{baru}) = v_{0j}(\text{lama}) + \Delta v_{0j} \quad (2.18)$$

Keterangan:

$v_{0j}(\text{baru})$: Bias baru *input layer* ke *hidden layer*

$v_{0j}(\text{lama})$: Bias lama *input layer* ke *hidden layer*

Δv_{0j} : Perubahan bias *layer*

Perhitungan terhadap bobot baru *hidden layer* ke *output layer* dengan persamaan (2.19).

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.19)$$

Keterangan:

$w_{jk}(\text{baru})$: Bobot baru *hidden layer* ke *output layer*

$w_{jk}(\text{lama})$: Bobot lama *hidden layer* ke *output layer*

Δw_{jk} : Perubahan bobot *output layer*

Perhitungan terhadap bias baru *hidden layer* ke *output layer* dengan persamaan (2.20).

$$w_{0k}(\text{baru}) = w_{0k}(\text{lama}) + \Delta w_{0k} \quad (2.20)$$

Keterangan:

$w_{0k}(\text{baru})$: Bias baru *hidden layer* ke *output layer*

$w_{0k}(\text{lama})$: Bias lama *hidden layer* ke *output layer*

Δw_{0k} : Perubahan bias *output layer*

2.1.3 *Linear Predictive Coding (LPC)*

Linear Predictive Coding (LPC) merupakan teknik analisis sinyal percakapan yang memberikan kualitas yang baik dan ekstraksi fitur yang efisien dalam perhitungan (Dinuriyati, 2019). LPC merupakan salah satu metode parametrik yang dapat digunakan untuk mengidentifikasi sinyal. pertama kali LPC dipakai untuk membuat alat sintesis sinyal suara pada tahun 1978. Analisis dalam LPC dilakukan dengan memperkirakan *formant*, melakukan *inverse filtering* atau pemisahan *formant* dari sinyal, kemudian mencari estimasi residu atau intensitas serta sisa frekuensi dari sinyal suara. Dikarenakan sinyal suara seiring waktu bervariasi, maka estimasi tersebut dilakukan pada setiap *frame* atau potongan kecil sinyal (Faradiba, 2017).

Dalam LPC terdapat beberapa langkah yang digunakan untuk pengenalan suara sebagai berikut:

1. *Preemphasis*

Pada langkah ini, data sinyal diambil untuk meminimalkan daerah perubahan sinyal. Bentuk *digital* dari cuplikan suara disaring untuk meratakan spektral sinyal dengan menggunakan *Finite Impulse Response (FIR) filter* orde satu. Adapun tujuan dari proses *filter* ini yakni untuk memperoleh bentuk spektral frekuensi sinyal suara yang lebih halus. Adapun untuk proses *preemphasis* menggunakan persamaan (2.21) berikut.

$$\tilde{s}(n) = s(n) - \tilde{\alpha} \cdot s(n - 1) \quad (2.21)$$

Keterangan:

$\tilde{s}(n)$: Fungsi *preemphasis*

$s(n)$: Data sinyal ke- n

$s(n - 1)$: Data sinyal ke- $n - 1$

\tilde{a} : Koefisien konstan pada *frame* analisis suara.

2. *Frame Blocking*

Sinyal suara yang telah melalui proses *pre-emphasis*, kemudian suara dibagi menjadi beberapa *frame* dimana tiap *frame* berisi N cuplikan suara dan beberapa *frame* yang berdekatan dipisah oleh M cuplikan. Adapun untuk menentukan banyaknya *frame blocking* pada tiap suara menggunakan persamaan (2.22) berikut.

$$frame = \frac{Fs - N}{M} + 1 \quad (2.22)$$

Keterangan:

Fs : *Sample rate* masukan suara

M : Pergeseran *frame*

N : Panjang *frame*

Dalam menghitung nilai tiap *frame*, ditentukan indeks awal dan akhir terlebih dahulu dengan persamaan (2.23) dan (2.24) berikut.

$$indeks\ awal = (i - 1)M + 1 \quad (2.23)$$

$$indeks\ akhir = (i - 1)M + N \quad (2.24)$$

3. *Windowing*

Pada langkah *windowing*, dilakukan fungsi *weighting* pada tiap *frame* yang sudah dibuat *frame blocking*. Jenis *windowing* yang sering digunakan yakni *Hamming Window*, dimana memiliki bentuk umum pada persamaan (2.25) berikut.

$$w(n)_i = L_i \left(0,54 - 0,46 \cdot \cos \left(\frac{2\pi n}{N - 1} \right) \right) \quad (2.25)$$

$$0 \leq n \leq N - 1$$

Keterangan:

$w(n)$: Fungsi *windowing* ke- n

N : Panjang *frame*

L_i : Sinyal masukan ke- i

4. Analisa Autokorelasi

Tiap *frame* yang telah dimasukkan pada *windowing* dilakukan autokorelasi dengan nilai autokorelasi tertinggi dimana merupakan orde dari analisis LPC. Adapun untuk analisa autokorelasi menggunakan persamaan (2.26) berikut.

$$r_t(m) = \sum_{n=1}^{N-1-m} x_t(n) \cdot x_t(n + m) \quad (2.26)$$

Keterangan:

$r_t(m)$: Koefisien autokorelasi dengan lag- m pada *frame* ke- t

m : Nilai lag, dengan $m = 0,1,2, \dots, p$

$x_t(n)$: Sinyal masukan ke- n pada *frame* ke- t

N : Panjang sinyal

5. Analisa LPC

Hasil autokorelasi selanjutnya dilakukan analisa LPC untuk diubah menjadi parameter LPC. Metode umum dalam menyelesaikan nilai autokorelasi menjadi koefisien LPC yakni menggunakan metode *Levinson-Durbin*. Adapun untuk algoritma *Levinson-Durbin* pada persamaan (2.27).

$$E_0 = r[0] \quad (2.27)$$

$$k_i = \frac{1}{E_{i-1}} \left(r[i] - \sum_{j=1}^{i-1} (\alpha_{j,i-1} r[|i-j|]) \right) \quad (2.28)$$

$$\alpha_{j,i} = \alpha_{j,i-1} - k_i \alpha_{i-j,i-1} \quad (2.29)$$

Untuk $j = 1$ hingga $j = i - 1$

$$\alpha_{i,i} = k_i \quad (2.30)$$

$$E_i = (1 - k_i^2) E_{i-1} \quad (2.31)$$

Berdasarkan persamaan (2.27) hingga (2.31) direkursi untuk $i = 1, 2, \dots, p$ sehingga didapatkan koefisien LPC pada persamaan (2.32).

$$\alpha_m = \alpha_m^p; \quad 1 \leq m \leq p \quad (2.32)$$

Keterangan:

E : *Error* prediksi

α : Koefisien LPC

k : Faktor refleksi

2.1.4 *Fast Fourier Transform* (FFT)

Fast Fourier Transform atau FFT merupakan teknik komputasi matematis di mana dipakai untuk mentransformasikan sinyal dalam domain waktu menjadi sinyal dalam domain frekuensi. FFT melakukan pembagian sinyal menjadi frekuensi yang berbeda-beda dengan fungsi eksponensial yang kompleks (Kusuma, 2021). FFT merupakan algoritma perhitungan pada transformasi *fourier* diskrit dengan efisien dan cepat. Hal ini dikarenakan perbedaan terkait sampel suara dengan sampel suara lainnya lebih jelas.

Data yang didapat oleh FFT merupakan data *input* untuk jaringan saraf tiruan. *Fourier transform* dapat didefinisikan pada persamaan (2.33).

$$S(f) = \int_{-\infty}^{\infty} s(t)e^{j2\pi ft} dt \quad (2.33)$$

Keterangan:

$S(f)$: Sinyal pada frekuensi

$s(t)$: Sinyal pada waktu

$s(t)e^{j2\pi ft} dt$: Konstanta nilai sinyal

f : Frekuensi

t : Waktu

Berdasarkan persamaan (2.33) terlihat bahwa FFT dapat dipakai dalam menghitung nilai frekuensi, fase gelombang sinyal dan amplitudo. Sedangkan untuk perhitungan spektrum frekuensi suatu sinyal dalam komputer, diperlukan algoritma *Discrete Fourier Transform* (DFT) untuk mengganti sinyal dalam domain waktu menjadi sinyal dalam domain frekuensi. Adapun rumus untuk DFT pada persamaan

$$F(u) = \sum_{x=0}^{x=N-1} f(x) \exp \left[-\frac{2i\pi ux}{N} \right] \quad (2.34)$$

Keterangan:

$F(u)$: Koefisien frekuensi pada frekuensi diskrit u

N : Panjang sinyal

$f(x)$: Sampel sinyal pada indeks waktu diskrit x

i : Satuan imajiner ($\sqrt{-1}$)

exp : Fungsi eksponensial

2.1.5 Mean Squared Error (MSE)

Mean Squared Error (MSE) merupakan salah satu metode perhitungan yang digunakan dalam evaluasi dari hasil prediksi. Setiap *error* atau sisa dikuadratkan (Ramadhanty dkk., 2017). Metode ini mengatur kesalahan atau *error* dari prediksi yang besar dikarenakan *error* tersebut dikuadratkan. Adapun untuk perhitungan MSE pada persamaan (2.35).

$$MSE = \frac{1}{N} \sum_{k=1}^N (t_k - y_k)^2 \quad (2.35)$$

Keterangan:

MSE : Perhitungan kesalahan *mean square error*

t_k : Target ke- k

y_k : Hasil *output training* ke- k

N : Banyaknya data

2.1.6 K-Fold Cross Validation

Cross Validation (CV) atau estimasi rotasi merupakan teknik yang digunakan untuk melakukan validasi keakuratan sebuah model yang dibuat berdasarkan data tertentu (Supartini dkk., 2017). Pada CV memberikan solusi pada suatu masalah dengan membagi data menjadi *fold* serta mengecek tiap *fold* digunakan sebagai set *testing* pada beberapa titik CV. Data yang digunakan pada CV terdiri dari data *training* dan data *testing*.

Proses CV membagi data yang diberikan menjadi sebanyak K bagian atau *fold* secara acak. Pada tiap *fold* digunakan sebagai set *testing* pada beberapa titik CV. Sebanyak K percobaan dilakukan dengan tiap percobaan menggunakan partisi

ke- K sebagai data *testing* dan sisa partisi lainnya digunakan sebagai data *training*. Banyaknya percobaan yang dilakukan sesuai dengan banyaknya partisi yang dilakukan.

Perhitungan akurasi digunakan untuk mengetahui dan menggambarkan terkait hasil pengujian apakah sesuai dengan nilai yang sebenarnya. Nilai akurasi dapat dinyatakan berdasarkan rentang nilai 0 sampai 100 dalam persen atau rentang nilai 0 sampai 1. Semakin tinggi nilai akurasi pada pengujian yang diperoleh, maka sistem tersebut berjalan dengan sangat baik. Adapun interpretasi nilai akurasi suatu sistem pada Tabel 2.1.

Tabel 2.1 Interpretasi Nilai Akurasi

Akurasi	Keterangan
< 0,25	Kurang Baik
0,25 – 0,5	Cukup Baik
0,5 – 0,75	Baik
> 0,75	Sangat Baik

2.1.7 Suara

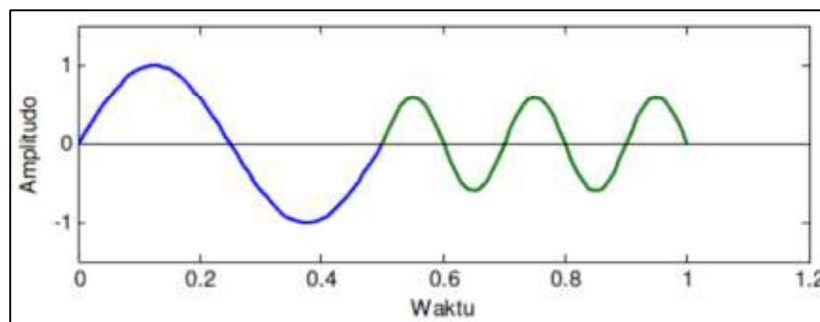
Suara atau bunyi dalam ilmu fisika merupakan gelombang longitudinal yang merambat melalui medium atau zat perantara tertentu (Kustaman, 2017). Adapun medium atau perantara yang dapat dilalui suara dapat berupa zat padat, cair, dan gas. Dalam Kamus Besar Bahasa Indonesia (KBBI), bunyi ialah sesuatu yang terdengar dan ditangkap oleh telinga. Suara dihasilkan dari sumber suara berupa suatu getaran yang merambat bergerak melalui zat perantara atau medium yang kemudian dapat diterima oleh indra pendengaran. Pada hakikatnya zat perantara suara dapat bermacam-macam bentuk dan sifatnya seperti zat padat, cair, maupun gas. Hal tersebut juga berpengaruh dari seberapa jauh sifat benda tersebut untuk dapat menghantarkan suara melalui udara.

Sinyal merupakan peristiwa yang timbul dari lingkungan tertentu dan dapat dinyatakan secara kuantitatif (Dinuriyati, 2019). Peristiwa di sini memiliki makna bahwa sinyal membawa frekuensi, dan adapun untuk kuantitatif di sini memiliki makna bahwa sinyal dapat dihitung dengan persamaan matematika meskipun tidak sempurna akan tetapi serupa dan mendekati. Begitu pula dengan suara, suara dapat diukur dan dihitung. Gelombang suara yang dapat didengar manusia mempunyai frekuensi berkisar 20 Hz sampai 20.000 Hz. Berdasarkan bentuk data, sinyal suara dibagi menjadi sinyal analog dan sinyal digital (Kusuma, 2021). Berikut perbedaan antara sinyal analog dan sinyal digital terdapat pada Tabel 2.2.

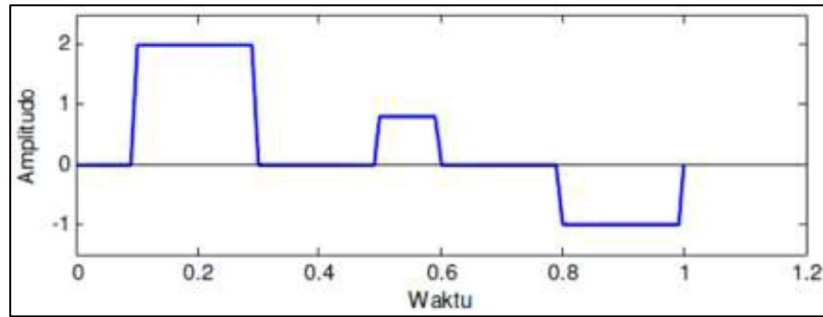
Suara memiliki peran penting bagi semua makhluk hidup, salah satunya manusia. Suara sebagai alat utama dalam berkomunikasi, menyatakan kepribadian, emosional, dan kondisi pembicara. Secara fisiologis, terdapat beberapa proses pada produksi suara manusia, yakni proses pernapasan, proses pembunyian, proses resonansi, dan proses artikulasi (Seikel dkk., 2009).

Tabel 2.2 Perbedaan Sinyal Analog dan Sinyal Diskrit

No	Sinyal Analog	Sinyal Digital
1	Bersifat Kontinyu	Bersifat Diskrit
2	Variasi nilai amplitudo tidak terbatas	Variasi nilai amplitudo terbatas
3	Peluang <i>error</i> besar	Peluang <i>error</i> kecil
4	Koreksi <i>error</i> sulit	Koreksi <i>error</i> mudah

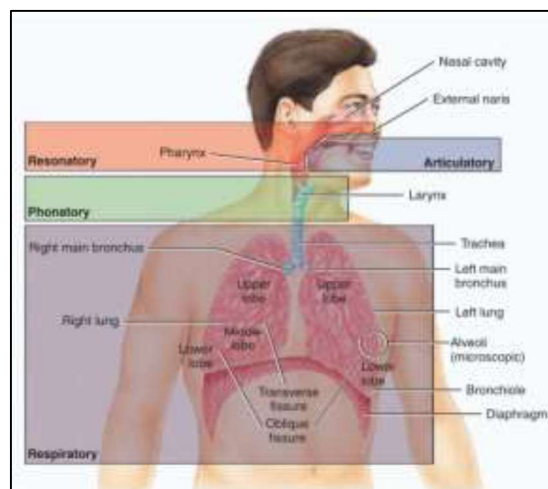


Gambar 2.5 Bentuk Sinyal Analog



Gambar 2.6 Bentuk Sinyal Digital

Proses pernapasan merupakan proses utama yang dilakukan paru-paru sebagai sumber energi. Proses pembunyian merupakan proses di mana energi berupa udara dari paru-paru melalui pita suara yang terdapat pada laring sehingga menghasilkan bunyi dengan atau tanpa bunyi. Proses resonansi terjadi karena cepatnya proses buka-tutup pita suara sehingga mengakibatkan aliran udara terhalang dan pita suara bergetar sehingga menghasilkan suara. Proses artikulasi merupakan proses di mana membentuk suara dari pita suara menjadi suara vokal dan konsonan sehingga dapat ditangkap dan dimengerti.



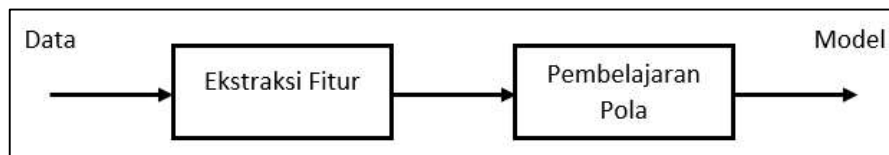
Gambar 2.7 Anatomi Pembentukan Suara Pada Manusia

2.1.8 Pengenalan Suara

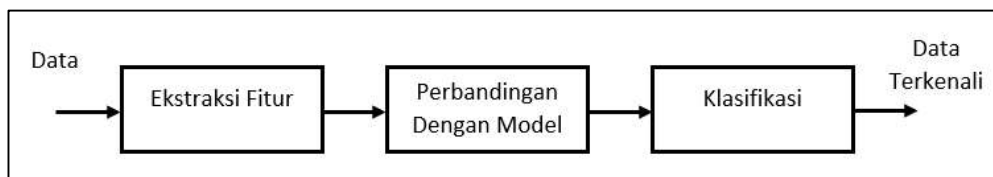
Pengenalan suara merupakan suatu proses identifikasi suara dengan berdasarkan parameter tertentu yang diambil oleh perangkat *input* suara. Salah satu model perkembangan dari *human-machine interaction* adalah pengenalan suara, sehingga mesin dapat dioperasikan hanya dengan suara manusia (Pradipta dkk., 2019). Pengenalan suara dalam hal ini merupakan suatu sistem yang digunakan untuk mengenali suara manusia berdasarkan kata yang diucapkan dan diubah menjadi data yang dapat dipahami komputer. Pengenalan suara dibagi menjadi tiga bentuk pendekatan, yakni pendekatan kecerdasan buatan (*artificial intelligent approach*), pendekatan akustik forensik (*accoustic phonetik approach*), dan pendekatan pengenalan pola (*pattern recognition approach*) (Dinuriyati, 2019).

Pattern recognition approach atau pendekatan pengenalan pola merupakan pendekatan yang digunakan untuk menggambarkan dan mengklasifikasikan pengukuran kuantitatif suatu ciri atau fitur dari suatu objek (Yodha & Kurniawan, 2014). Tujuan pendekatan pengenalan pola yakni untuk menentukan kategori atau kelompok pola berdasarkan fitur atau ciri pada pola tersebut sehingga terdapat perbedaan antara objek satu dengan objek yang lain. Terdapat dua langkah dalam pendekatan pengenalan pola, yakni pembelajaran pola dan pengenalan pola lewat perbandingan pola (Hapsari, 2014). Pada proses pembelajaran pola, sistem akan dilakukan pembelajaran terhadap pola dari suatu data supaya sistem yang digunakan dapat mengenali data tersebut ketika dimasukkan data baru dengan pola yang sama. Pada tahap pengenalan pola melalui perbandingan pola, data baru dimasukkan kemudian sistem akan mengidentifikasi data tersebut berdasarkan

perbandingan pola yang didapatkan pada tahap pembelajaran kemudian diklasifikasikan dengan pola terbaik yang sesuai.



Gambar 2.8 Blok Diagram Proses Pembelajaran Pola



Gambar 2.9 Blok Diagram Proses Pengenalan Pola

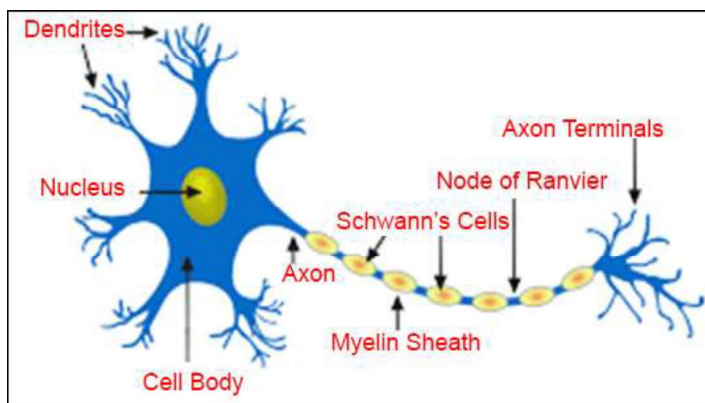
Terdapat dua macam pengenalan suara, yakni pengenalan ucapan (*Speech Recognition*) dan pengenalan pembicara (*Speaker Recognition*) (Pradipta dkk., 2019). Pengenalan ucapan merupakan pengenalan suara berdasarkan kata yang diucapkan. Sedangkan pengenalan pembicara merupakan pengenalan suara berdasarkan suara orang yang berbicara. Terdapat tiga tahap dalam pengenalan suara, yakni ekstraksi fitur, pemodelan, dan pengenalan (Hapsari, 2014). Ekstraksi fitur berfungsi untuk mendapatkan fitur atau ciri suara. Beberapa metode ekstraksi fitur yang dapat digunakan pada suara yakni *Linear Predictive Coding (LPC)*, *Mel-Frequency Cepstrum RASTA Filtering*, dan *Probabilistic Linear Discriminate Analysis*. Pemodelan dan pengenalan suara dilakukan dengan menggunakan metode yang dapat digunakan untuk mengidentifikasi suara, salah satunya dengan metode jaringan saraf tiruan *backpropagation*.

2.1.9 Jaringan Saraf

Tubuh manusia mempunyai susunan yang kompleks. Susunan yang kompleks tersebut memungkinkan manusia untuk melakukan aktivitas seperti bergerak, berbicara, mendengar, melihat, berpikir, dan sebagainya. Salah satu komponen penyusun dalam tubuh manusia adalah jaringan. Jaringan merupakan sekelompok sel yang memiliki struktur dan fungsi yang sama (Chalik, 2016). Jaringan saraf merupakan jaringan saraf yang menjadi inti dari seluruh aktivitas sel beserta jaringan lainnya. Jaringan saraf tersebar di seluruh tubuh manusia dan saling terhubung. Jaringan saraf berperan untuk mengatur dan mengkoordinasikan semua aktivitas yang dilakukan oleh tubuh. Jaringan saraf mendeteksi rangsangan perubahan suatu kondisi baik dari luar maupun dalam tubuh kemudian ditanggapi oleh sinyal elektrik yang menghasilkan impuls atau aksi yang ditransfer ke organ tubuh.

Jaringan saraf tersusun atas neuron atau sel saraf. Terdapat kurang lebih 10^{10} sel saraf pada otak manusia yang saling berhubungan (Loppies, 2018). Dalam otak manusia terdiri dari neuron-neuron di mana neuron bekerja sesuai dengan sinyal yang didapatkan dari neuron lain dan disalurkan kembali ke seluruh tubuh untuk menanggapi sinyal tersebut. Satu neuron tersusun atas *dendrit*, akson dan badan sel (Setiawan, 2021). *Dendrit* merupakan percabangan badan sel yang membantu menghantarkan serta menerima rangsangan dari badan sel. Badan sel adalah bagian terbesar neuron dan memiliki banyak komponen penting yang bertujuan menerima rangsangan dari *dendrit* dan mengirimkannya ke akson. Akson atau *neurit* adalah serabut neuron yang serupa dengan *dendrit*, akan tetapi hanya terdapat satu buah dan memiliki ukuran yang panjang dan lebih besar. Akson

berfungsi untuk menyalurkan impuls menuju ke efektor dari badan sel, Adapun sebagai contohnya seperti sel kelenjar atau sel otot.



Gambar 2.10 Struktur Jaringan Saraf Manusia

2.2 Keragaman Suara Manusia dalam Perspektif Islam

Allah SWT. menciptakan manusia dengan begitu kompleks sehingga menjadi makhluk yang paling sempurna dari semua ciptaan-Nya yang lain. Manusia diberikan nafsu, akal, pikiran, pendengaran serta suara. Kesempurnaan manusia disebutkan dalam Al-Qur'an surah At-Tiin [95] ayat 4 (Kemenag, 2019) sebagai berikut:

Artinya: "Sesungguhnya Kami telah menciptakan manusia dalam bentuk yang sebaik-baiknya." (Q.S. At-Tiin: 4).

Wahbah Zuhaili dalam kitab tafsirnya, yakni *At-Tafsir Al-Munir Fil Aqidah Wa Syari'ah Wal Manhaj* menjelaskan bahwa makna lafadz *Ahsani Taqwim*" atau "bentuk yang sebaik-baiknya" adalah sebaik-baiknya bentuk, anggota tubuh yang sempurna dengan susunan yang seimbang serta tertata rapi. Diberikan ilmu, pemikiran, kepemimpinan, kalam (komunikasi), dan kebijaksanaan. Hal tersebut semakin mempertegas bahwa manusia sebagai khalifah (pemimpin) di bumi. Segala keistimewaan dan kemuliaan ada dalam diri manusia (Pratiwi dkk., 2021).

Kemuliaan akan tetap ada apabila manusia mampu menjaga hubungan baik dengan Allah SWT. serta sesama manusia. Apabila hubungan tersebut tidak dijaga dengan baik, maka kemuliaan yang ada dalam diri manusia berubah menjadi kehinaan. Keistimewaan dalam diri manusia akan berubah menjadi kerendahan.

Suara merupakan salah satu anugerah yang diberikan oleh Allah SWT. kepada manusia. Anugerah tersebut berupa beragamnya jenis suara yang diberikan kepada jutaan manusia di seluruh bumi. Beragamnya jenis suara tersebut dipengaruhi oleh perbedaan bentuk, ukuran tebal dan tipisnya pita suara. Perbedaan itulah yang membuat suara manusia berbeda dengan manusia yang lain. Dengan adanya keragaman suara tersebut mengisyaratkan bagi sesama manusia untuk saling mengenal dan memahami karakter setiap individu. Mempelajari tentang suara merupakan salah satu memuji Allah SWT. sebagai Sang Pencipta. Orang yang senantiasa berpikir dan mempelajari tentang segala sesuatu ciptaan Allah SWT dijelaskan dalam Al-Quran surah Ali-Imran [2] ayat 190-191 (Kemenag, 2019).

Artinya: “Sesungguhnya dalam penciptaan langit dan bumi, dan silih bergantinya malam dan siang terdapat tanda-tanda bagi orang-orang yang berakal. (Yaitu) orang-orang yang mengingat Allah sambil berdiri atau duduk atau dalam keadaan berbaring dan mereka memikirkan tentang penciptaan langit dan bumi (seraya berkata): “Ya Rabb, Kami tiadalah Engkau menciptakan ini dengan sia-sia. Mahasuci Engkau, maka peliharalah kami dari siksa Neraka.” (Q.S. Ali-Imran: 190-191)

M. Quraissy Syihab dalam kitab tafsirnya, yakni *Al-Misbah* menjelaskan bahwa dalam ayat tersebut, Allah menguraikan sedikit dari penciptaan-Nya kemudian memerintahkan manusia agar memikirkan ciptaan tersebut. Berdasarkan tujuan utama diturunkannya surah Ali-Imran ialah untuk membuktikan terkait tauhid, kekuasaan dan keesaan Allah SWT. Hakikat tersebut kemudian dipertegas pada ayat berikutnya. Memicu manusia untuk berpikir merupakan salah satu dari

bukti adanya kebenaran hal tersebut. Hal ini dikarenakan dalam penciptaan, yakni adanya benda-benda luar angkasa seperti matahari, bumi, dan bulan serta banyaknya bintang-bintang dalam sistem tata surya sangat teliti sehingga menjadikannya perbedaan siang dan malam serta perbedaan waktu terdapat tanda-tanda kekuasaan Allah SWT bagi orang-orang yang memiliki akal yang murni atau ulul albab (Sofia, 2021). Mempelajari tentang segala sesuatu tentang ciptaan-Nya baik dari struktur, bentuk dan fungsi merupakan salah satu bentuk dalam diri ulul albab, dimana menggunakan seluruh akal yang murni dalam mempelajari sebagai bukti memuji dan mengingat akan kuasa Allah SWT. atas segalanya.

2.3 Kajian Penelitian dengan Teori Pendukung

Penelitian ini disusun berdasarkan beberapa teori pendukung terkait jaringan saraf tiruan *backpropagation*. Adapun penelitian terkait yakni pengenalan jenis kelamin dan rentang umur berdasarkan suara oleh Avisena Abdillah Alwi, Putra Pandu Adikara, dan Indriati (Alwi dkk., 2020). Pada penelitian tersebut digunakan data suara yang bersumber dari *Mozilla Common Voice* dan *Free ST American English Corpus*. Ekstraksi fitur yang digunakan yakni *Linear Predictive Coding* (LPC). Identifikasi menggunakan metode *backpropagation* dan didapatkan hasil akurasi kurang baik pada data *Mozilla Common Voice* dan hasil akurasi cukup baik pada data *Free ST American English Corpus*. Hal tersebut dikarenakan terdapat berbagai macam variasi kata pada kedua *dataset* tersebut. Adapun untuk pertimbangan penelitian selanjutnya terkait pengenalan sandi morse oleh Odi Nurdiawan (Nurdiawan, 2018). Pada penelitian tersebut digunakan sandi morse suara berupa tiupan peluit. Data sampel yang digunakan berupa sandi morse dalam

angka 0-9. Dengan ekstraksi fitur *Linear Predictive Coding* (LPC) dan identifikasi suara menggunakan jaringan saraf tiruan *backpropagation*, didapatkan hasil pelatihan sebesar 85,16% dan pengujian sebesar 67,99%.

Pada penelitian ini, peneliti menggunakan *dataset* berupa rekaman suara dengan pengucapan satu kata untuk menyamakan suara. *Preprocessing* dilakukan terhadap data suara yang digunakan untuk mendapatkan data suara yang optimal dengan pemotongan suara hening dan menghilangkan *noise* yang terdapat dalam data suara. Hasil *preprocessing* kemudian dilakukan ekstraksi fitur suara menggunakan *Linear Predictive Coding* (LPC). Hasil ekstraksi fitur suara dalam bentuk sinyal domain waktu sehingga diperlukan transformasi menjadi bentuk sinyal dalam domain frekuensi dengan *Fast Fourier Transform* (FFT). Data FFT digunakan sebagai data *input* pada jaringan saraf tiruan *backpropagation*. Data suara dibagi menjadi dua, yakni data *training* dan data *testing*. Kemudian dilakukan perancangan model untuk arsitektur jaringan saraf tiruan *backpropagation* yang digunakan dalam identifikasi suara. Apabila model yang didapatkan sesuai, selanjutnya evaluasi dilakukan untuk mengetahui tingkat akurasi keberhasilan dari metode tersebut.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian yang digunakan pada penelitian ini merupakan penelitian kuantitatif evaluasi. Pada penelitian ini, pengolahan data dipilih cara yang lebih banyak kualitas suatu data dalam bentuk angka-angka selanjutnya dilakukan pengujian berdasarkan perhitungan rumus-rumus berdasarkan pola hubungan antar variabel yang akan dibuktikan. Hasil kesimpulan dinyatakan dalam bentuk pernyataan yang didukung oleh angka dengan beranggapan bahwa angka-angka dapat mempermudah menyatakan, membandingkan, dan memperjelas akurasi. Peneliti melakukan penelitian berdasarkan tahapan-tahapan yang terstruktur dan sistematis. Data yang digunakan pada penelitian ini yakni berupa data sinyal suara yang diubah menjadi data angka. Selanjutnya data diproses dengan metode jaringan saraf tiruan *backpropagation*.

3.2 Data dan Sumber Data

Data yang digunakan pada penelitian ini yakni data primer, di mana data suara dikumpulkan secara langsung dari partisipan. Data suara didapatkan dengan mengambil 15 orang partisipan dengan masing-masing partisipan sebanyak 15 sampel suara. Adapun untuk kata yang diucapkan yakni kata “HAI” untuk tiap orang dengan durasi rata-rata 2 detik. Penggunaan kata tersebut bertujuan untuk menyamakan suara sehingga memudahkan dalam proses identifikasi. Suara yang

direkam merupakan suara dengan format WAV. Data suara dibagi menjadi dua, yakni data *training* sebesar 80% dan data *testing* sebesar 20%.

3.3 Perancangan Sistem

Sistem yang akan dirancang yakni sistem yang bisa digunakan untuk mengidentifikasi berbagai macam serta banyaknya jenis model suara yang tepat dengan permasalahan yang telah dirumuskan. Sistem ini akan mengidentifikasi suara dari subjek yang sebelumnya sudah dimasukkan ke dalam satu *folder*, jadi suara yang belum dimasukkan ke dalam *folder* tidak dapat dikenali. Sistem ini akan dirancang untuk mengenali suara seseorang setelah melewati beberapa proses.

3.4 Teknik Analisis Data

Adapun untuk proses pada sistem dapat diuraikan sebagai berikut:

1. Proses *Preprocessing*

Pada proses *Preprocessing* terdapat beberapa tahapan sebagai berikut

- 1) Data suara akan dilakukan *preprocessing* terlebih dahulu untuk mendapatkan suara yang optimal. Suara dilakukan proses *filter* untuk mengurangi *noise* yang terdapat pada suara.
- 2) Hasil *preprocessing* suara kemudian dilakukan ekstraksi fitur dengan *Linear Predictive Coding*. Terdapat beberapa proses pada *Linear Predictive Coding*, yakni *preemphasis* menggunakan persamaan (2.21), *frame blocking* menggunakan persamaan (2.22) hingga (2.24), *windowing* menggunakan persamaan (2.25), analisa autokorelasi

menggunakan persamaan (2.26) dan analisa LPC menggunakan persamaan (2.27) hingga (2.32).

- 3) Setelah didapatkan hasil dari *Linear Predictive Coding*, kemudian akan dilakukan proses *Fast Fourier Transform* untuk mengubah sinyal analog menjadi sinyal digital menggunakan persamaan (2.34).
- 4) Dikarenakan hasil *Fast Fourier Transform* cukup besar, maka data dilakukan normalisasi terlebih dahulu.
- 5) Data *Fast Fourier Transform* selanjutnya akan digunakan pada proses *training* dan *testing* dalam jaringan saraf tiruan *backpropagation*.
- 6) Pada jaringan saraf tiruan *backpropagation*, data akan dibagi menjadi dua, yakni data *training* dan data *testing*.

2. Proses *Training Data*

Pada proses *training data* terdapat beberapa tahapan sebagai berikut:

- 1) Suara yang digunakan sebagai data *training* sebanyak 80% menjadi 180 data suara.
- 2) Fungsi aktivasi yang digunakan yakni fungsi *sigmoid* pada persamaan (2.3).
- 3) Parameter yang akan digunakan pada jaringan saraf tiruan yakni terdiri dari *error* sebesar 0,01, *learning rate* (α) dengan nilai 0,01, maksimum *epoch* sebesar 3000, perhitungan *error* menggunakan MSE (*Mean Squared Error*).
- 4) Data dilakukan proses pelatihan sehingga menghasilkan nilai *error* terkecil.

3. Proses *Testing Data*

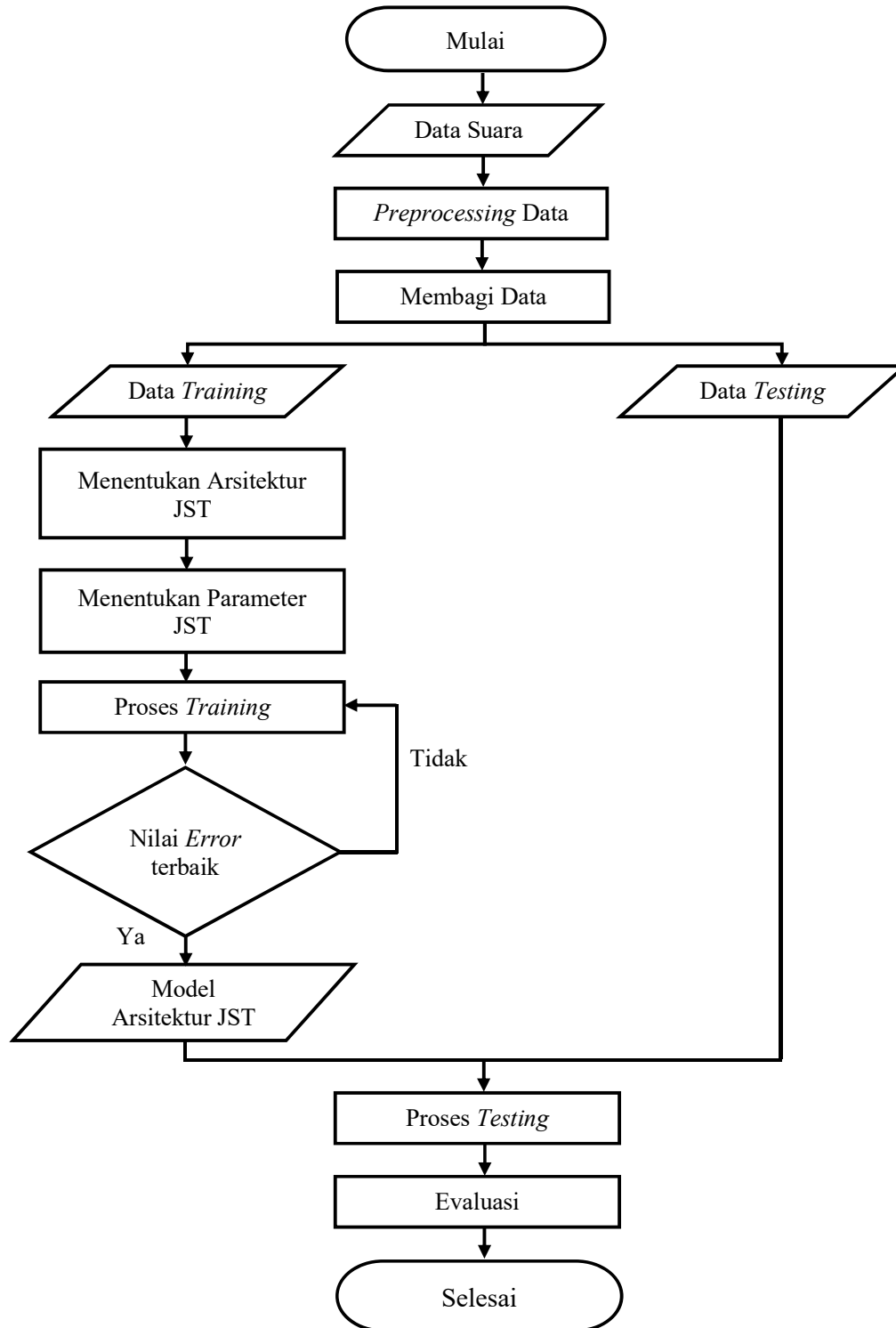
Pada proses *testing data* terdapat beberapa tahapan sebagai berikut:

- 1) Suara yang digunakan sebagai data *testing* sebanyak 20% menjadi 45 data suara.
- 2) Model arsitektur jaringan saraf tiruan akan dilakukan pengujian dengan menggunakan data uji yang belum pernah dilakukan pelatihan untuk mengetahui apakah model tersebut dapat mengenali data dengan baik.

4. Proses Evaluasi

Proses evaluasi dilakukan untuk menganalisis dan mengetahui tingkat akurasi pelatihan dan pengujian model arsitektur jaringan saraf tiruan. Dalam menentukan tingkat akurasi menggunakan *K-Fold Cross Validation* (CV). Berdasarkan tingkat akurasi tersebut akan diketahui apakah model jaringan saraf tiruan tersebut sudah baik atau belum.

3.5 Diagram Alir Penelitian



Gambar 3.1 Diagram Alir Penelitian

BAB IV

HASIL DAN PEMBAHASAN

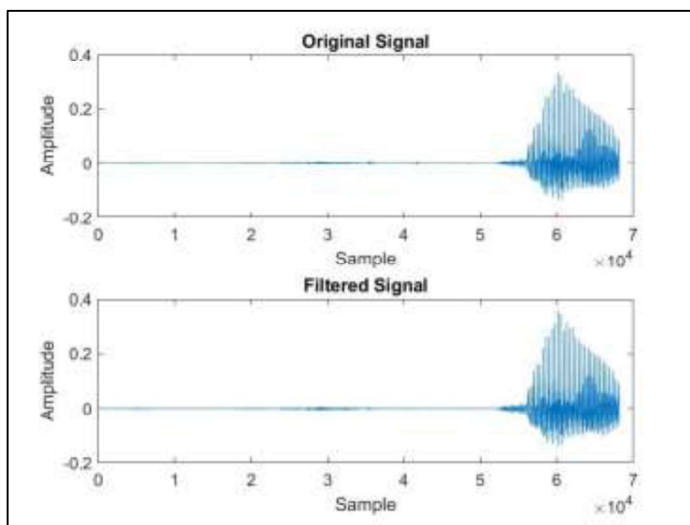
4.1 Pengumpulan Data

Data penelitian didapatkan sebanyak 225 rekaman suara yang diambil dari 15 partisipan dengan tiap partisipan sebanyak 15 rekaman suara. Pengambilan data dilakukan dengan melakukan rekaman suara langsung kepada partisipan melalui *smartphone*. Data yang telah didapatkan dimasukkan ke dalam satu folder untuk mempermudah dalam pemrosesan data.

4.2 *Preprocessing Data*

4.2.1 *Filter Suara*

Data suara dilakukan proses *filter* untuk mengurangi *noise* atau gangguan pada saat proses rekaman suara. Adapun *filter* suara yang digunakan yakni *Band Pass Filter* (BPF). Adapun untuk sebagai contoh hasil *filter* yang didapatkan menggunakan program MATLAB pada gambar Gambar 4.1.



Gambar 4.1 Sinyal Suara Asli dengan Sinyal Suara *Filter*

Hasil dari *filter* suara disimpan dalam bentuk rekaman suara baru. Setiap suara mempunyai panjang *array* yang berbeda-beda. Sebagai contoh salah satu panjang array suara didapatkan sebanyak 68160 pada Tabel 4.1 (lebih lengkapnya dapat dilihat pada Lampiran 1 dan Lampiran 2).

Tabel 4.1 *Output Array Filter Suara*

No	<i>Filter Suara</i>
1	0
2	0
3	0
⋮	⋮
68159	-0,0656314160965912
68160	-0,0658394910155097

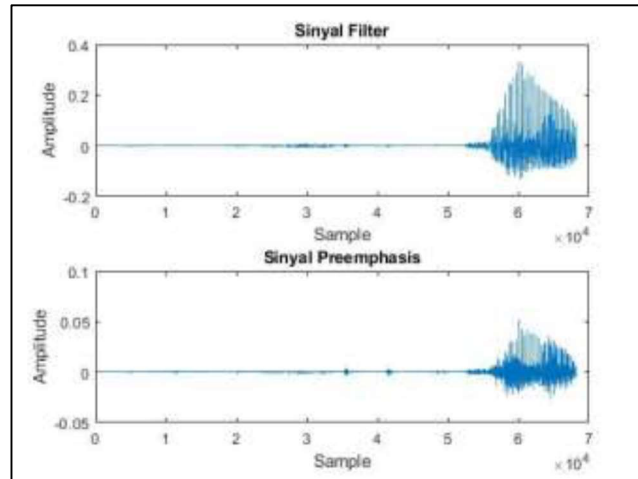
Selanjutnya ekstraksi fitur suara dilakukan pada hasil *filter* menggunakan *Linear Predictive Coding* (LPC) untuk mendapatkan ciri atau fitur suara.

4.2.2 Ekstraksi Fitur Suara

Proses ekstraksi fitur suara menggunakan *Linear Predictive Coding* (LPC). Terdapat beberapa proses yang dilakukan dalam *Linear Predictive Coding* (LPC). Proses pertama yakni *preemphasis* adapun contoh hasil dari *preemphasis* berdasarkan persamaan (2.21) dengan $\tilde{a} = 0,95$, $n = 1$ sehingga nilai $s(1) = 0$

$$\tilde{s}(1) = 0 - 0,95 \cdot 0 = 0$$

didapatkan hasil *preemphasis* untuk $\tilde{s}(1) = 0$. Dilakukan perhitungan seterusnya sesuai dengan panjang suara sehingga didapatkan hasil *preemphasis* sebagai contoh dalam satu suara pada Gambar 4.2 dan Tabel 4.2 (lebih lengkapnya dapat dilihat pada Lampiran 3).



Gambar 4.2 Sinyal Suara *Filter* Dengan Sinyal Suara *Preemphasis*

Tabel 4.2 *Output Array Preemphasis*

$\tilde{s}(1)$	$\tilde{s}(2)$	$\tilde{s}(3)$...	$\tilde{s}(68159)$	$\tilde{s}(68160)$
0	0	0	...	-0,00306	-0,00319

Hasil *preemphasis* kemudian dilakukan *frame blocking*. *Frame blocking* bertujuan untuk membagi sinyal suara menjadi beberapa *frame*. Adapun sebagai contoh hasil dari *frame blocking* yang didapatkan dengan persamaan (2.22). Dengan $N = 256$, $M = 256$, F_s merupakan banyak data *preemphasis*.

$$frame = \frac{68160 - 256}{256} + 1$$

$$frame = 266,25$$

$$frame \approx 266$$

Dalam contoh suara tersebut akan dibagi menjadi 266 *frame*. Untuk menghitung nilai masukan tiap *frame* digunakan persamaan (2.23) dan (2.24).

Sebagai contoh menentukan *frame* ke-1

$$indeks\ awal = (1 - 1)256 + 1 = 1$$

$$indeks\ akhir = (1 - 1)256 + 256 = 256$$

Sehingga untuk *frame* 1 diisi dengan nilai pada *preemphasis* dimulai dari kolom 1 sampai 256. Untuk *frame* ke-2 didapatkan indeks awal dan akhir sebagai berikut.

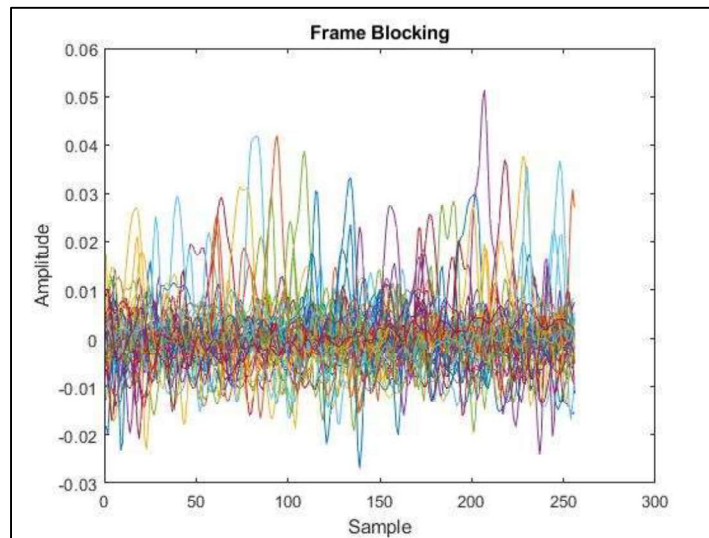
$$\text{indeks awal} = (2 - 1)256 + 1 = 257$$

$$\text{indeks akhir} = (2 - 1)256 + 256 = 512$$

Untuk *frame* 2 diisi dengan nilai pada *preemphasis* dimulai dari kolom 257 sampai 512. Seterusnya sampai *frame* 266 sehingga didapatkan matriks *frame blocking* pada Tabel 4.3 (lebih lengkapnya dapat dilihat pada Lampiran 4). Jika ditampilkan dalam bentuk grafik untuk setiap *frame* ditunjukkan dengan warna yang berbeda-beda pada Gambar 4.3.

Tabel 4.3 *Output Array Frame Blocking*

	1	2	3	...	255	256
Frame 1	0	0	0	...	-0,0000015	-0,0000015
Frame 2	-0,000032	-0,0000031	-0,0000031	...	-0,0000015	0,0000289
Frame 3	0	0	0	...	-0,0000015	0,0000288
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Frame 265	0,001738	0,003018	0,003658	...	0,0019836	0,0056244
Frame 266	0,009015	0,010242	0,008426	...	0,0022263	0,0019424

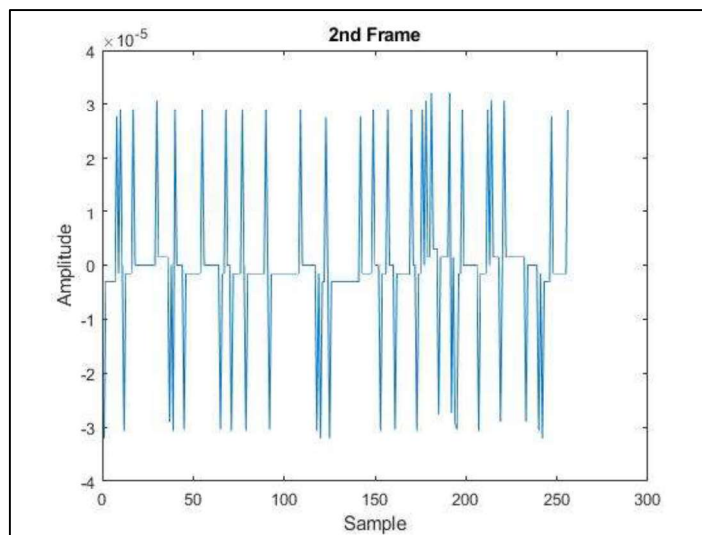


Gambar 4.3 *Frame Blocking Sinyal Suara*

Pada satu suara terdapat 266 *frame* dan panjang tiap *frame* 256 panjang gelombang sinyal yang bertumpuk menjadi satu. Sebagai contoh gelombang *frame* ke-2 pada Tabel 4.4 (lebih lengkapnya dapat dilihat pada Lampiran 5) dan jika ditampilkan dalam bentuk grafik pada Gambar 4.4.

Tabel 4.4 *Output Array Frame Ke-2*

	1	2	3	...	255	256
<i>Frame 2</i>	-0,000032	-0,0000031	-0,0000031	...	-0,0000015	0,0000289



Gambar 4.4 Sinyal Suara pada *Frame* Ke-2

Hasil suara *frame blocking* selanjutnya dilakukan *windowing* dengan persamaan (2.25). Dengan $N = 256$ dan $L_1 = -0,000032$. Untuk menentukan nilai *windowing* pada *frame* ke-2 sebagai berikut.

$$w(2)_1 = -0,000032 \left(0,54 - 0,46 \cdot \cos \left(\frac{2\pi 1}{256 - 1} \right) \right)$$

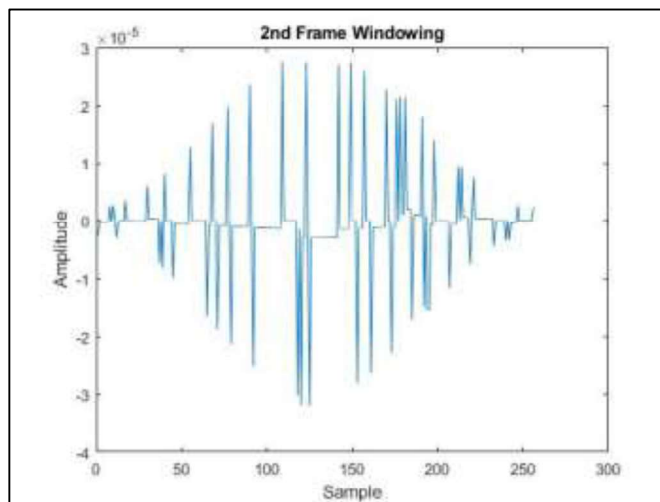
$$w(2)_1 = -0,000026$$

Dilakukan perhitungan seterusnya sesuai dengan panjang suara dan banyaknya *frame* sehingga didapatkan hasil *windowing* pada *frame* ke-2 pada Tabel 4.5 (lebih

lengkapya dapat dilihat pada Lampiran 6) dan jika ditampilkan dalam bentuk grafik pada Gambar 4.5.

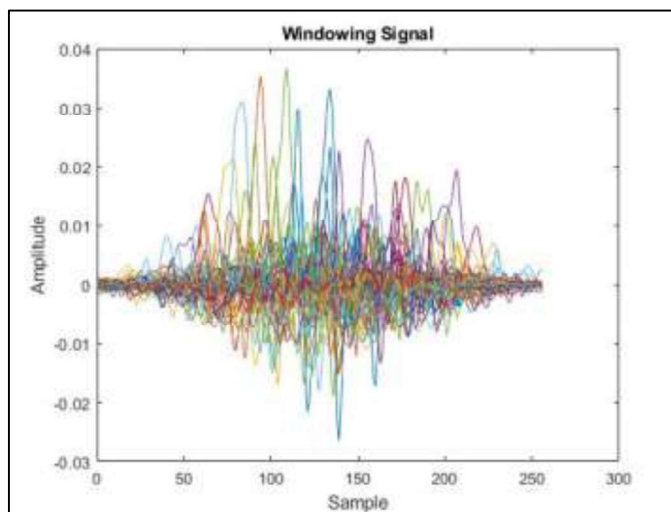
Tabel 4.5 *Output Array Windowing Frame Ke-2*

	1	2	3	...	255	256
Frame 2	-0,000026	-0,0000024	-0,0000025	...	-0,00000012	0,00000231



Gambar 4.5 *Windowing Sinyal Suara Frame Ke-2*

Berikut didapatkan hasil perhitungan *windowing* keseluruhan pada Tabel 4.6 (lebih lengkapnya dapat dilihat pada Lampiran 7) dan jika ditampilkan dalam bentuk grafik pada Gambar 4.6.



Gambar 4.6 *Windowing Sinyal Suara*

Tabel 4.6 Output Array Windowing

	1	2	3	...	255	256
Frame 1	0	0	0	...	-0,00000012	-0,00000012
Frame 2	-0,0000026	-0,0000024	-0,0000025	...	-0,00000012	0,00000231
Frame 3	0	0	0	...	-0,00000012	0,00000232
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Frame 266	0,000721	0,000821	0,000679	...	0,0001784	0,0001554

Hasil suara *windowing* selanjutnya dilakukan analisa autokorelasi dengan persamaan (2.26). Dengan $N = 266$, $m = 0,1,2, \dots, p$, $x_t(n)$ merupakan sinyal masukan ke- t pada *frame* ke- n . Untuk menentukan nilai autokorelasi sinyal masukan ke-2 sebagai berikut.

Untuk $m = 0$

$$\begin{aligned}
 r_2(0) &= \sum_{n=1}^{266-0} x_1(n) \cdot x_1(n+0) \\
 &= (-0,0000026 \cdot (-0,0000026)) + (-0,0000024 \cdot (-0,0000024)) \\
 &\quad + (-0,0000025 \cdot (-0,0000025)) + \dots \\
 &\quad + (0,00000231 \cdot 0,00000231) \\
 &= -0,000001625
 \end{aligned}$$

Untuk $m = 1$

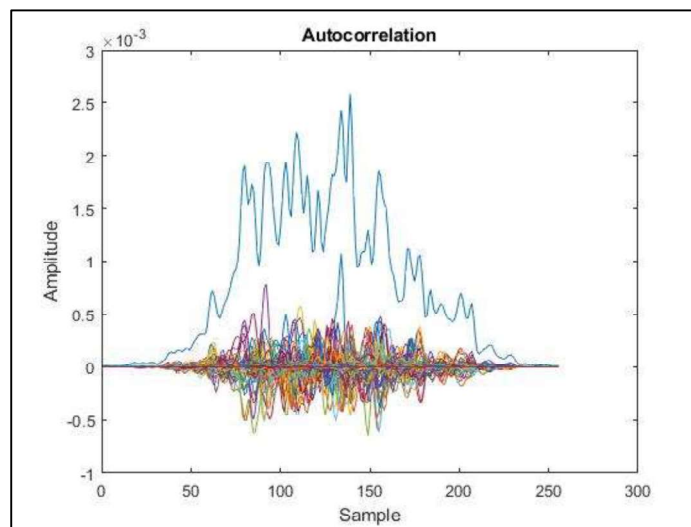
$$r_2(1) = \sum_{n=1}^{266-1} x_1(n) \cdot x_1(n+1) = \sum_{n=1}^{265} x_1(n) \cdot x_1(n+1)$$

$$\begin{aligned}
&= (-0,0000026 \cdot (-0,0000024)) + (-0,0000024 \cdot (-0,0000025)) \\
&\quad + (-0,0000025 \cdot (-0,0000025)) + \dots \\
&\quad + (-0,00000012 \cdot 0,00000231) \\
&= -0,000001277
\end{aligned}$$

Perhitungan dilakukan seterusnya hingga $m = 266$ dan sebanyak sinyal masukan sehingga didapatkan hasil autokorelasi pada Tabel 4.7 (lebih lengkapnya dapat dilihat pada Lampiran 8) dan jika ditampilkan dalam bentuk grafik pada Gambar 4.7.

Tabel 4.7 Output Array Autokorelasi

	1	2	3	...	255	256
Frame 1	1,035e-05	9,4385e-06	9,0077e-06	...	1,2732e-05	1,1781e-05
Frame 2	-1,625e-06	-1,277e-06	1,6758e-08	...	-4,477e-08	-8,437e-07
Frame 3	-6,133e-07	3,4884e-07	1,387e-06	...	-2,764e-06	-1,558e-06
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Frame 265	-1,848e-09	-2,007e-10	-1,668e-10	...	-4,125e-11	3,0549e-10
Frame 266	0	0	0	...	-2,181e-11	-1,896e-11

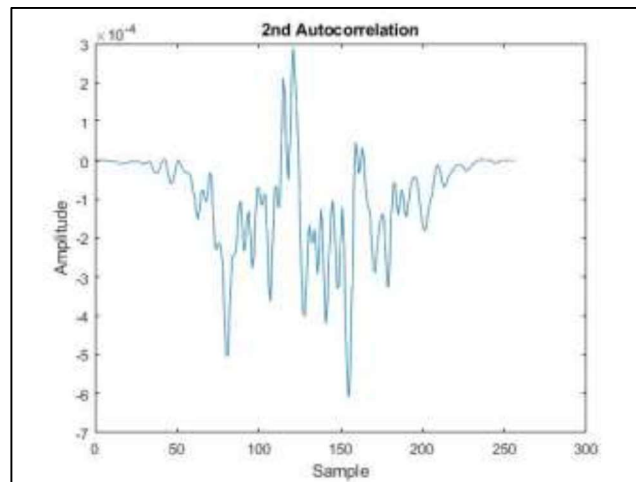


Gambar 4.7 Autokorelasi Sinyal Suara

Adapun untuk contoh autokorelasi ke-2 pada Tabel 4.8 (lebih lengkapnya dapat dilihat pada Lampiran 9) dan jika ditampilkan dalam bentuk grafik pada Gambar 4.8.

Tabel 4.8 *Output Array Autokorelasi Frame Ke-2*

	1	2	3	...	255	256
Frame 2	-1,6253e-06	-1,2776e-06	1,6758e-08	...	-4,4779e-08	-8,4371e-07



Gambar 4.8 Autokorelasi Sinyal Suara *Frame Ke-2*

Hasil koefisien autokorelasi selanjutnya dilakukan analisa LPC dengan persamaan (2.27) sampai (2.32).

$$E_0 = -0,000001625$$

Untuk $i = 1$,

$$k_1 = \frac{-0,000001277 - \sum_{j=1}^{i-1} \alpha_{j,i-1} r[|i-1|]}{-0,000001625}$$

$$k_1 = \frac{-0,000001277}{-0,000001625}$$

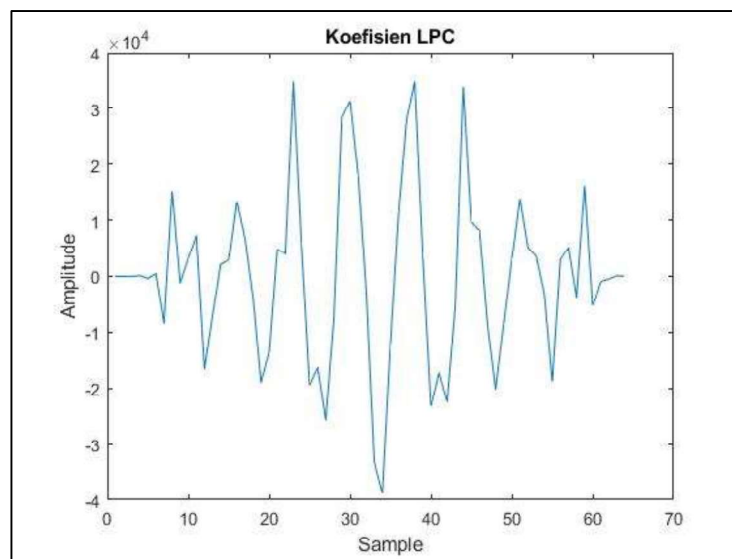
$$k_1 = 0,78584$$

$$\alpha_{1,1} = 0,78584$$

$$\begin{aligned}
 E_1 &= (1 - (0,78584)^2) \cdot (-0,000001625) \\
 &= -0,0000006215
 \end{aligned}$$

Iterasi dilakukan dengan nilai $i = 1$ hingga $p = 64$, sehingga didapatkan $\alpha_{1,64}$ hingga $\alpha_{64,64}$ yang merupakan hasil koefisien LPC pada Tabel 4.9 (lebih lengkapnya dapat dilihat pada

Lampiran 10) dan jika ditampilkan dalam bentuk grafik pada Gambar 4.9.



Gambar 4.9 Koefisien LPC Sinyal Suara

Tabel 4.9 Output Koefisien LPC

α_1	α_2	α_3	...	α_{63}	α_{64}
0	0,9112	-47,3374	...	123,9453	2,6373

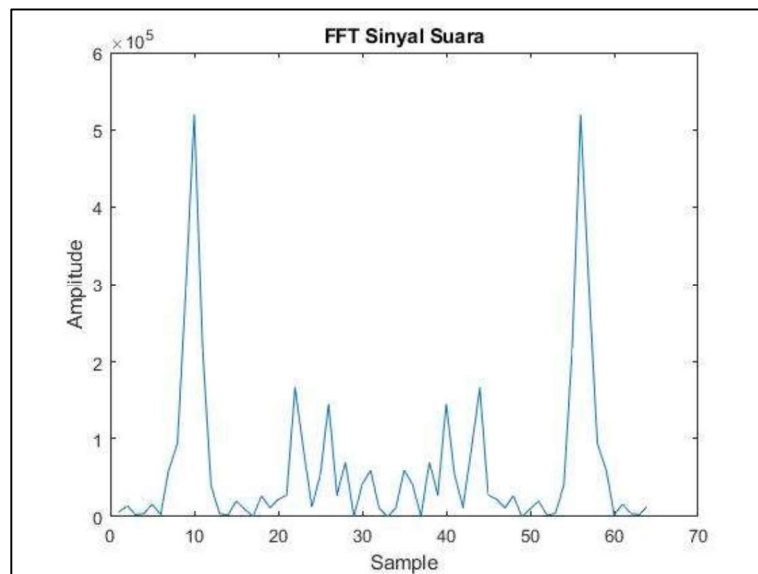
Setelah hasil ekstraksi fitur suara didapatkan, selanjutnya dilakukan transformasi sinyal suara untuk mengubah dari sinyal analog menjadi sinyal diskrit dengan *Fast Fourier Transform* atau FFT.

4.2.3 Transformasi Sinyal Suara

Transformasi suara dilakukan menggunakan *Fast Fourier Transform* (FFT). Transformasi suara mengubah sinyal suara dalam domain waktu menjadi domain frekuensi menggunakan persamaan (2.34). Dengan $N = 64$, sebagai contoh perhitungan untuk mencari $F(1)$ berikut.

$$\begin{aligned}
 F(1) &= \sum_{x=0}^{x=64-1} f(x) \exp\left[-\frac{2j\pi ux}{64}\right] \\
 &= \sum_{x=0}^{63} f(x) \exp\left[-\frac{2j\pi ux}{64}\right] \\
 &= 0 + 0,82604 + (-38,9022) + \dots + 0,28255 + 0,00545 \\
 &= 29,2724
 \end{aligned}$$

Dilakukan perhitungan seterusnya hingga $u = 64$, sehingga didapatkan hasil FFT pada Tabel 4.10 (lebih lengkapnya dapat dilihat pada Lampiran 11) dan jika ditampilkan dalam bentuk grafik pada Gambar 4.10.



Gambar 4.10 FFT Sinyal Suara

Tabel 4.10 *Output Array FFT*

F(1)	F(2)	F(3)	...	F(63)	F(64)
29,2724	70,0798	291,163	...	291,163	70,0798

Hasil transformasi sinyal suara akan dijadikan sebagai data *input* dalam proses *training* dan *testing* pada jaringan saraf tiruan *backpropagation*.

4.2.4 Normalisasi Data

Normalisasi data dilakukan untuk membuat beberapa variabel memiliki rentang nilai yang sama. Pada proses FFT menghasilkan nilai masukan data yang cukup besar sehingga dilakukan normalisasi dengan rentang -1 sampai 1. Adapun untuk perhitungan normalisasi sebagai berikut.

$$x'_n = \frac{x_n - x_{min}}{x_{max} - x_{min}} \cdot 2 - 1$$

Dimana x_n merupakan operasi pada cuplikan suara ke- n dan menentukan nilai terkecil dan terbesar dari x . Berikut contoh untuk mencari hasil normalisasi ke-1.

$$\begin{aligned} x'_1 &= \frac{x_1 - x_{min}}{x_{max} - x_{min}} \cdot 2 - 1 \\ &= \frac{5979,343 - 2,7971}{519176,55 - 2,7971} \cdot 2 - 1 = -0,9770 \end{aligned}$$

Dilakukan perhitungan seterusnya sesuai banyak masukan data sehingga didapatkan hasil normalisasi pada Tabel 4.11 (lebih lengkapnya dapat dilihat pada Lampiran 12).

Tabel 4.11 *Output Array Normalisasi*

x'₁	x'₂	x'₃	...	x'₆₃	x'₆₄
-0,97698	-0,948201	-0,98963	...	-0,98963	-0,948201

Berdasarkan rangkaian *preprocessing* data, didapatkan data yang akan digunakan pada jaringan saraf tiruan *backpropagation* pada Lampiran 1. Data suara

dibagi menjadi 2, yakni data *training* sebanyak 80% atau 180 data suara dan data *testing* sebanyak 20% atau 45 data suara.

4.3 Proses *Training* Data

Proses *training* data dilakukan untuk melatih dan mendapatkan model jaringan saraf tiruan yang digunakan untuk mengidentifikasi sinyal suara manusia. Berdasarkan proses tersebut menghasilkan *output* suara yang dapat dikenali dengan baik sesuai dengan target ataupun yang tidak dikenali. Penelitian ini menggunakan parameter yang digunakan berupa nilai *error* = 0,01, *learning rate* (α) = 0,01, fungsi aktivasi yang digunakan gabungan dari *sigmoid* dan *purelin*, pelatihan *hidden layer* yang digunakan sebanyak 2, 3, 4, dan 5, jumlah neuron yang digunakan sebanyak 10 dan 15, *epoch* maksimal yang digunakan 3000.

Inisialisasi bobot dan bias awal acak. Bobot dan bias terdiri dari bobot dan bias dari *input layer* ke *hidden layer* dan dari *hidden layer* ke *output layer*. Adapun bobot dan bias awal pada Tabel 4.12, Tabel 4.13, Tabel 4.14 dan bias awal $w_{0k} = -0,0444$ (lebih lengkapnya dapat dilihat pada

Lampiran 14, Lampiran 15, dan Lampiran 16). Bobot dan bias tersebut digunakan pada proses *feedforward*. Perhitungan dilakukan untuk mencari nilai sinyal masukan *hidden layer* dengan persamaan (2.6). Sebagai contoh dilakukan perhitungan pada satu sampel suara.

Tabel 4.12 Bobot Awal *Input Layer* ke *Hidden Layer*

v_{ij}	1	2	3	...	63	64
1	-0,2141	0,0700	-0,0675	...	-0,0487	0,0418
2	0,1870	-0,0424	-0,0613	...	0,0676	0,0803
3	0,1137	-0,1757	0,0460	...	-0,1128	0,0328
⋮	⋮	⋮	⋮	⋮	⋮	⋮
14	0,0465	0,0611	-0,0229	...	-0,0345	-0,0248
15	-0,2066	0,0637	-0,0502	...	-0,0629	-0,0021

Tabel 4.13 Bobot Awal *Hidden Layer* ke *Output Layer*

	w_{jk}
1	-0,0304
2	-0,0393
3	-0,0127
⋮	⋮
14	0,0231
15	-0,0046

Tabel 4.14 Bias Awal *Input Layer* ke *Hidden Layer*

	v_{0j}
1	-0,0768
2	0,1336
3	0,0164
⋮	⋮
14	0,0335
15	0,1391

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

$$z_{in_1} = -0,0768 + \sum_{i=1}^{64} x_i v_{i1}$$

$$\begin{aligned} &= -0,0768 + (-1 \cdot (-0,2141)) + (-0,9883 \cdot 0,0700) \\ &\quad + (-0,9671 \cdot (-0,0675)) + \dots + (-0,9671 \cdot (-0,0487)) \\ &\quad + (-0,9883 \cdot 0,0418) = 0,4907 \end{aligned}$$

Perhitungan dilakukan hingga sebanyak j bobot, dimana j merupakan banyak neuron yang ditentukan. Didapatkan hasil perhitungan z_{in_j} pada Tabel 4.15 (lebih lengkapnya dapat dilihat pada Lampiran 17).

Tabel 4.15 Sinyal Masukan *Hidden Layer*

z_{in_1}	z_{in_2}	z_{in_3}	...	$z_{in_{14}}$	$z_{in_{15}}$
0,4907	0,3857	-0,1523	...	0,4831	-0,3031

Sinyal masukan *hidden layer* dihitung dengan fungsi aktivasi pada persamaan (2.7).

Fungsi aktivasi yang digunakan yakni *sigmoid* pada persamaan (2.3).

$$Z_j = f(z_{in_j})$$

$$Z_j = \frac{1}{1 + e^{-z_{in_j}}}$$

$$Z_1 = \frac{1}{1 + e^{-z_{in_1}}}$$

$$= \frac{1}{1 + e^{-(0,4907)}} = 0,6203$$

Perhitungan dilakukan hingga $j = 15$, didapatkan hasil sinyal masukan *hidden layer* pada Tabel 4.16 (lebih lengkapnya dapat dilihat pada Lampiran 18).

Tabel 4.16 Sinyal Masukan *Hidden Layer* dengan Fungsi Aktivasi

Z_1	Z_2	Z_3	...	Z_{14}	Z_{15}
0,6203	0,5953	0,4620	...	0,6185	0,4248

Selanjutnya dilakukan perhitungan untuk mencari keluaran *output layer* dengan persamaan (2.8).

$$y_{in_k} = w_{0k} + \sum_{j=1}^n w_{jk} Z_j$$

$$= -0,0444 + \sum_{j=1}^{15} w_{jk} Z_j$$

$$= -0,0444 + (-0,0304 \cdot 0,6203) + (-0,0393 \cdot 0,5953) + (-0,0127 \cdot 0,4620)$$

$$+ \dots + (0,0231 \cdot 0,6185) + (-0,0046 \cdot 0,4248) = -0,1867$$

Didapatkan keluaran *output layer* $y_{in_k} = -0,1867$. Sinyal keluaran *output layer* dihitung dengan fungsi aktivasi pada persamaan (2.9). Fungsi aktivasi yang digunakan yakni *purelin* pada persamaan (2.5).

$$Y_k = f(y_{in_k})$$

$$Y_k = u = -0,1867$$

Didapatkan keluaran *output layer* dengan fungsi aktivasi $Y_k = -0,1867$.

Selanjutnya proses *backpropagation*, dimana menghitung *error* antara target dengan *output* yang dihasilkan. Mencari faktor koreksi *error output layer* dengan persamaan (2.10), dimana t_k ialah target tiap sinyal suara, untuk $t_k = 1$.

$$\delta_k = (t_k - Y_k)Y_k(1 - Y_k)$$

$$\delta_1 = (1 - (-0,1867))(-0,1867)(1 - (-0,1867))$$

$$\delta_1 = -0,2629$$

Menghitung perubahan bobot W_{jk} dengan persamaan (2.11).

$$\Delta W_{jk} = \alpha \delta_k Z_j$$

$$\Delta W_{11} = 0,01 \cdot (-0,2629) \cdot 0,6203 = -0,00163$$

$$\Delta W_{12} = 0,01 \cdot (-0,2629) \cdot 0,5953 = -0,00157$$

⋮

$$\Delta W_{115} = 0,01 \cdot (-0,2629) \cdot 0,4248 = -0,00112$$

Perhitungan dilakukan sebanyak Z_j , dengan $j = 15$. Didapatkan hasil perbaikan bobot w_{jk} pada Tabel 4.17 (lebih lengkapnya dapat dilihat pada Lampiran 19).

Tabel 4.17 Perbaikan Bobot w_{jk}

	ΔW_{jk}
1	-0,00163
2	-0,00157
3	-0,00121
⋮	⋮
14	-0,00163
15	-0,00112

Menghitung perubahan bias w_{0k} dengan persamaan (2.12).

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

$$= 0,01 \cdot (-0,2629) = -0,00263$$

Didapatkan hasil perbaikan bias $w_{0k} = -0,00263$. Menghitung faktor koreksi *input hidden layer* dengan persamaan (2.13).

$$\begin{aligned}\delta_{in_j} &= \delta_k w_{jk} \\ \delta_{in_1} &= -0,2629 \cdot (-0,00163) = 0,0080 \\ \delta_{in_2} &= -0,2629 \cdot (-0,00157) = 0,0103 \\ &\vdots \\ \delta_{in_{15}} &= -0,2629 \cdot (-0,00112) = 0,0012\end{aligned}$$

Perhitungan dilakukan sebanyak w_{jk} , didapatkan hasil δ_{in_j} pada Tabel 4.18 (lebih lengkapnya dapat dilihat pada Lampiran 20).

Tabel 4.18 Faktor Koreksi δ_{in_j}

	δ_{in_j}
1	0,0080
2	0,0103
3	0,0033
\vdots	\vdots
14	-0,0061
15	0,0012

Menghitung faktor koreksi *error hidden layer* dengan persamaan (2.14).

$$\begin{aligned}\delta_j &= \delta_{in_j} z_j (1 - z_j) \\ \delta_1 &= 0,0080 \cdot 0,6203 (1 - 0,6203) = 0,00188 \\ \delta_2 &= 0,0103 \cdot 0,5953 (1 - 0,5953) = 0,00249 \\ &\vdots \\ \delta_{15} &= 0,0012 \cdot 0,4248 (1 - 0,4248) = 0,00030\end{aligned}$$

Didapatkan hasil perhitungan δ_j keseluruhan pada Tabel 4.19 (lebih lengkapnya dapat dilihat pada Lampiran 21).

Tabel 4.19 Faktor Koreksi δ_j

	δ_j
1	0,00188
2	0,00249
\vdots	\vdots
15	0,00030

Menghitung perbaikan bobot *hidden layer* dengan persamaan (2.15), dimana x_j merupakan target tiap sinyal suara, untuk ini $x_j = 1$.

$$\Delta v_{ij} = \alpha \delta_j x_j$$

$$\Delta v_{11} = 0,01 \cdot 0,00188 \cdot 1 = 0,0000188$$

$$\Delta v_{21} = 0,01 \cdot 0,00249 \cdot 1 = 0,0000249$$

$$\vdots$$

$$\Delta v_{151} = 0,01 \cdot 0,00030 \cdot 1 = 0,0000030$$

$$\Delta v_{12} = 0,01 \cdot 0,00188 \cdot 1 = 0,0000188$$

$$\vdots$$

$$\Delta v_{1564} = 0,01 \cdot 0,00030 \cdot 1 = 0,0000030$$

Perhitungan dilakukan hingga sebanyak matriks bobot *input layer*, yakni matriks 15×64 didapatkan pada Tabel 4.20 (lebih lengkapnya dapat dilihat pada Lampiran 22).

Tabel 4.20 Perbaikan Bobot v_{ij}

Δv_{ij}	1	2	3	...	63	64
1	0,0000188	0,0000188	0,0000188	...	0,0000188	0,0000188
2	0,0000249	0,0000249	0,0000249	...	0,0000249	0,0000249
3	0,0000083	0,0000083	0,0000083	...	0,0000083	0,0000083
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
14	-0,0000143	-0,0000143	-0,0000143	...	-0,0000143	-0,0000143
15	0,0000030	0,0000030	0,0000030	...	0,0000030	0,0000030

Menghitung koreksi bias *hidden layer* dengan persamaan (2.16).

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

$$\Delta v_{01} = 0,01 \cdot 0,00188 = 0,0000188$$

$$\Delta v_{02} = 0,01 \cdot 0,00249 = 0,0000249$$

⋮

$$\Delta v_{015} = 0,01 \cdot 0,00030 = 0,0000030$$

Perhitungan dilakukan hingga sebanyak bias *hidden layer* awal, didapatkan hasil Δv_{0j} pada Tabel 4.21 (lebih lengkapnya dapat dilihat pada Lampiran 23).

Tabel 4.21 Faktor Koreksi Δv_{0j}

	Δv_{0j}
1	0,0000188
2	0,0000249
3	0,0000083
⋮	⋮
14	-0,0000143
15	0,0000030

Selanjutnya dilakukan perhitungan untuk mendapatkan bobot dan bias baru.

Menghitung bobot baru *input layer* ke *hidden layer* dengan persamaan (2.17).

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

$$v_{11}(\text{baru}) = -0,2141 + 0,0000188 = -0,2140626$$

$$v_{21}(\text{baru}) = 0,1870 + 0,0000249 = 0,1869871$$

⋮

$$v_{1564}(\text{baru}) = -0,0021 + 0,0000030 = -0,0021408$$

Perhitungan dilakukan hingga sebanyak matriks bobot *input layer*, yakni matriks 15×64 didapatkan pada Tabel 4.22 (lebih lengkapnya dapat dilihat pada Lampiran 24).

Tabel 4.22 Bobot Baru v_{ij}

$v_{ij}(\text{baru})$	1	2	3	...	63	64
1	-0,2140626	0,0700295	-0,0675267	...	-0,0486906	0,0417948
2	0,1869871	-0,0424092	-0,0612401	...	0,0676294	0,0803609
3	0,1136642	-0,1757202	0,0460459	...	-0,1127713	0,0327656

⋮	⋮	⋮	⋮	⋮	⋮	⋮
14	0,0464559	0,0610674	-0,0229418	...	-0,0345588	-0,0248018
15	-0,2065877	0,0637518	-0,0501750	...	-0,0629123	-0,0021408

Menghitung bias baru *input layer* ke *hidden layer* dengan persamaan (2.18).

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

$$v_{01}(\text{baru}) = -0,0768 + 0,0000188 = -0,0768140$$

$$v_{02}(\text{baru}) = 0,1336 + 0,0000249 = 0,1336690$$

⋮

$$v_{015}(\text{baru}) = 0,1391 + 0,0000030 = 0,1391529$$

Perhitungan dilakukan hingga sebanyak bias v_{0j} , didapatkan hasil bias baru v_{0j} pada Tabel 4.23 (lebih lengkapnya dapat dilihat pada Lampiran 25).

Tabel 4.23 Bias Baru v_{0j}

	$v_{0j}(\text{baru})$
1	-0,0768140
2	0,1336690
3	0,0164214
⋮	⋮
14	0,0334946
15	0,1391529

Menghitung bobot baru *hidden layer* ke *output layer* dengan persamaan (2.19).

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

$$w_{11}(\text{baru}) = -0,0304 + (-0,00163) = -0,0320$$

$$w_{12}(\text{baru}) = -0,0393 + (-0,00157) = -0,0408$$

⋮

$$w_{115}(\text{baru}) = -0,0046 + (-0,00112) = -0,0058$$

Perhitungan dilakukan hingga sebanyak bobot w_{jk} , didapatkan hasil bobot baru w_{jk} pada Tabel 4.24 (lebih lengkapnya dapat dilihat pada Lampiran 26).

Tabel 4.24 Bobot Baru w_{jk}

	$w_{jk}(\text{baru})$
1	-0,0320
2	-0,0408
3	-0,0139
⋮	⋮
15	-0,0058

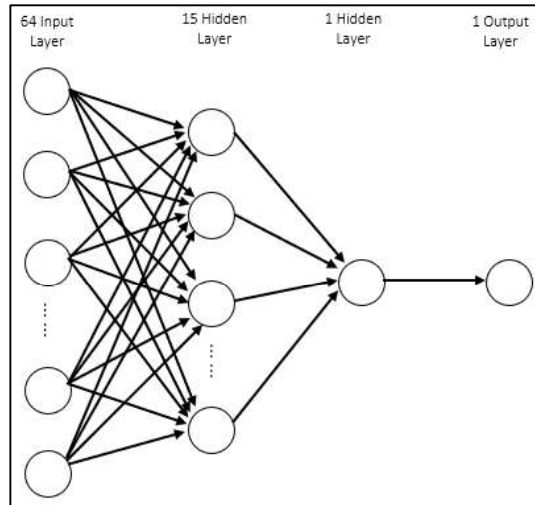
Menghitung bias baru *hidden layer* ke *output layer* dengan persamaan (2.20).

$$\begin{aligned}
 w_{0k}(\text{baru}) &= w_{0k}(\text{lama}) + \Delta w_{0k} \\
 &= -0,0444 + (-0,00263) = -0,0471
 \end{aligned}$$

Pelatihan dilakukan hingga semua *epoch* terpenuhi. Perhitungan dilakukan mulai dari 1 *epoch* dimana semua data terhitung, kemudian sampai *epoch* maksimum yang telah ditentukan. Pelatihan untuk mencari model arsitektur jaringan dilakukan dengan parameter *hidden layer* dan jumlah neuron yang berbeda-beda. Berikut hasil pelatihan jaringan saraf tiruan dengan beberapa parameter yang digunakan pada Tabel 4.25.

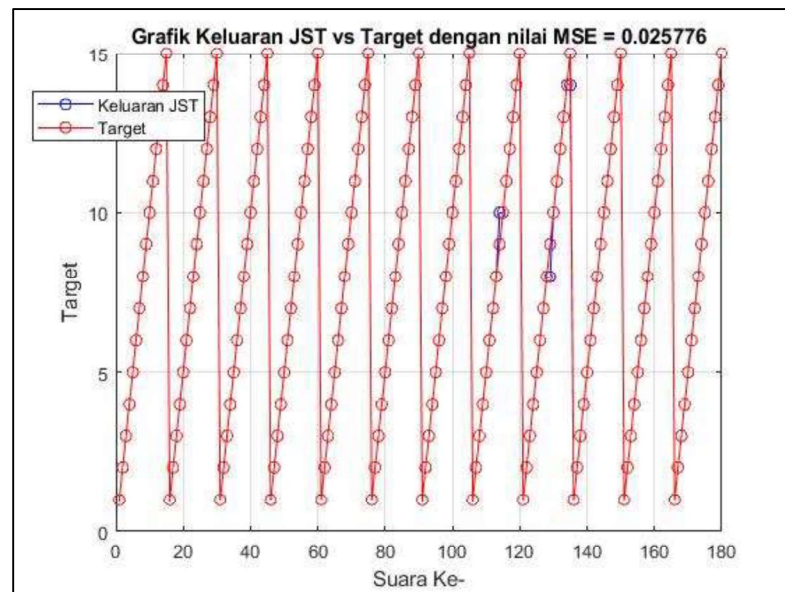
Tabel 4.25 Percobaan Pelatihan Jaringan Saraf Tiruan

<i>Hidden Layer</i>	Neuron	<i>epoch</i>	Waktu (detik)	MSE
2	10;1	3000	13	0,71392
	15;1	3000	15	0,025776
3	10;10;1	3000	13	0,05553
	15;15;1	3000	20	0,10211
4	10;10;10;1	3000	54	0,27732
	15;15;15;1	3000	23	0,24138
5	10;10;10;10;1	3000	34	1,4221
	15;15;15;15;1	3000	15	0,14197



Gambar 4.11 Model Arsitektur JST 64-15-1-1

Berdasarkan Tabel 4.25 didapatkan model pelatihan yang optimal dengan 2 *hidden layer* dan 15;1 neuron. Waktu pelatihan yang diperoleh yakni 15 detik dengan 3000 *epoch*. Nilai MSE yang diperoleh untuk mendekati *error target* 0,01 adalah 0,025776. Hasil pelatihan terhadap data latih terdapat pada Lampiran 29.



Gambar 4.12 Hasil *Training* JST Pada MATLAB

4.4 Proses *Testing Data*

Proses *testing* data dilakukan untuk menguji model jaringan saraf tiruan apakah dapat mengenali data atau tidak. Data uji yang digunakan yakni data suara yang belum pernah dilakukan pelatihan sebelumnya. Berikut contoh perhitungan untuk proses *testing* data dengan inisialisasi parameter bobot dan bias yang digunakan pada proses *training* data.

Perhitungan pengujian hanya pada proses *feedforward*, yakni terlebih dahulu mencari nilai sinyal masukan *hidden layer* dengan persamaan (2.6).

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

$$z_{in_1} = -0,0768 + \sum_{i=1}^{64} x_i v_{ij}$$

$$= -0,0768 + (-0,9996 \cdot (-0,2141)) + (-0,99931 \cdot 0,0700)$$

$$+ (-0,998 \cdot (-0,0675)) + \dots + (-0,998 \cdot (-0,0487))$$

$$+ (-0,99931 \cdot 0,0418) = 0,6663$$

Perhitungan dilakukan hingga sebanyak j bobot, dimana j merupakan banyak neuron yang ditentukan. Didapatkan hasil perhitungan z_{in_j} pada Tabel 4.26 (lebih lengkapnya dapat dilihat pada Lampiran 27).

Tabel 4.26 Sinyal Masukan *Hidden Layer*

z_{in_1}	z_{in_2}	z_{in_3}	...	$z_{in_{14}}$	$z_{in_{15}}$
0,6663	0,8389	0,4253	...	0,3967	-0,0397

Sinyal masukan *hidden layer* dihitung dengan fungsi aktivasi pada persamaan (2.7).

Fungsi aktivasi yang digunakan yakni *sigmoid* pada persamaan (2.3).

$$Z_j = f(z_{in_j})$$

$$Z_j = \frac{1}{1 + e^{-z_{in_j}}}$$

$$Z_1 = \frac{1}{1 + e^{-z_{in_1}}}$$

$$= \frac{1}{1 + e^{-(0,6663)}} = 0,660671$$

Perhitungan dilakukan hingga $j = 15$, didapatkan hasil sinyal masukan *hidden layer* pada Tabel 4.27 (lebih lengkapnya dapat dilihat pada Lampiran 28).

Tabel 4.27 Sinyal Masukan *Hidden Layer* dengan Fungsi Aktivasi

z_1	z_2	z_3	...	z_{14}	z_{15}
0,660671	0,698224	0,604745	...	0,59789	0,509914

Selanjutnya dilakukan perhitungan untuk mencari keluaran *output layer* dengan persamaan (2.8).

$$y_{in_k} = w_{0k} + \sum_{j=1}^n w_{jk} Z_j$$

$$= -0,0444 + \sum_{j=1}^{15} w_{jk} Z_j$$

$$= -0,0444 + (-0,0304 \cdot 0,660671) + (-0,0393 \cdot 0,698224)$$

$$+ (-0,0127 \cdot 0,604745) + \dots + (0,0231 \cdot 0,59789)$$

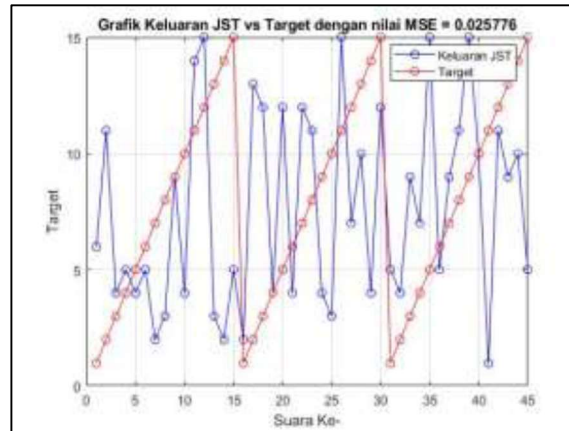
$$+ (-0,0046 \cdot 0,509914) = -0,1909$$

Didapatkan keluaran *output layer* $y_{in_k} = -0,1909$. Sinyal keluaran *output layer* dihitung dengan fungsi aktivasi pada persamaan (2.9). Fungsi aktivasi yang digunakan yakni *purelin* pada persamaan (2.5).

$$Y_k = f(y_{in_k})$$

$$Y_k = u = -0,1909$$

Didapatkan keluaran *output layer* dengan fungsi aktivasi $Y_k = -0,1867$.



Gambar 4.13 Hasil *Testing* JST Pada MATLAB

Pengujian model secara keseluruhan menggunakan program MATLAB.

Didapatkan hasil pengujian terhadap model pada Tabel 4.28.

Tabel 4.28 Hasil Pengujian Model Arsitektur Jaringan Saraf Tiruan

Nama	Target Uji	Hasil Uji	Error	Klasifikasi	Kesimpulan
Ali	1	5,6402	-4,6402	6	Salah
Andri	2	10,7842	-8,7842	11	Salah
Cahya	3	3,8312	-0,8312	4	Salah
Dimas	4	5,0061	-1,0061	5	Salah
Doni	5	4,4029	0,5971	4	Salah
Fitrah	6	4,5494	1,4506	5	Salah
Ike	7	1,7348	5,2652	2	Salah
Iqbal	8	3,1693	4,8307	3	Salah
Lia	9	8,6279	0,3721	9	Benar
Mustofa	10	4,1827	5,8173	4	Salah
Nana	11	13,6905	-2,6905	14	Salah
Sabit	12	15,3993	-3,3993	15	Salah
Sania	13	2,6657	10,3343	3	Salah
Wahyu	14	1,8571	12,1429	2	Salah
Yuslih	15	4,8243	10,1757	5	Salah
Ali	1	1,6703	-0,6703	2	Salah
Andri	2	12,5831	-10,5831	13	Salah
Cahya	3	12,2201	-9,2201	12	Salah
Dimas	4	3,9521	0,0479	4	Benar
Doni	5	12,3192	-7,3192	12	Salah
Fitrah	6	3,8715	2,1285	4	Salah
Ike	7	11,8766	-4,8766	12	Salah
Iqbal	8	11,0569	-3,0569	11	Salah
Lia	9	4,0996	4,9004	4	Salah
Mustofa	10	3,4450	6,5550	3	Salah
Nana	11	15,0999	-4,0999	15	Salah
Sabit	12	7,4986	4,5014	7	Salah
Sania	13	9,8384	3,1616	10	Salah
Wahyu	14	4,4508	9,5492	4	Salah
Yuslih	15	11,8495	3,1505	12	Salah

Ali	1	4,7248	-3,7248	5	Salah
Andri	2	3,8301	-1,8301	4	Salah
Cahya	3	8,6617	-5,6617	9	Salah
Dimas	4	7,1884	-3,1884	7	Salah
Doni	5	15,2413	-10,2413	15	Salah
Fitrah	6	4,8782	1,1218	5	Salah
Ike	7	9,3911	-2,3911	9	Salah
Iqbal	8	10,6307	-2,6307	11	Salah
Lia	9	14,5292	-5,5292	15	Salah
Mustofa	10	10,0769	-0,0769	10	Benar
Nana	11	-0,7537	11,7537	1	Salah
Sabit	12	10,9117	1,0883	11	Salah
Sania	13	8,5857	4,4143	9	Salah
Wahyu	14	9,5585	4,4415	10	Salah
Yuslih	15	5,0993	9,9007	5	Salah

4.5 Evaluasi

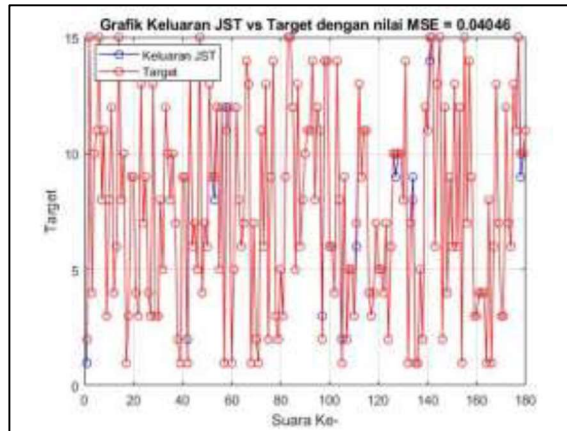
Pada subbab ini berisi evaluasi terkait hasil pada proses pelatihan dan pengujian model jaringan saraf tiruan. Proses evaluasi dilakukan menggunakan metode *K-Fold Cross Validation* (CV). Model diuji dengan membagi data menjadi 5 *fold*. 225 Data keseluruhan didapatkan 45 *subset* data (*S*) hasil pengacakan dengan nilai $K = 5$. Data dibagi menjadi data *training* dan *testing* dengan persentase pembagian 80%-20% untuk tiap *fold*.

Tabel 4.29 Pembagian *Training* dan *Testing* dengan *Cross Validation*

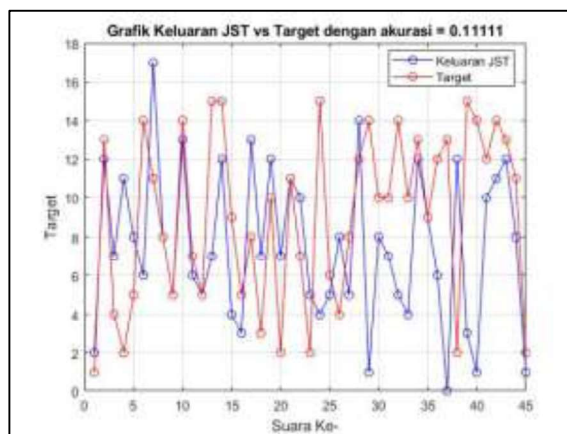
<i>Fold</i>	<i>Training</i>	<i>Testing</i>
F1	S1, S2, S3, S4	S5
F2	S1, S2, S3, S5	S4
F3	S1, S2, S4, S5	S3
F4	S1, S3, S4, S5	S2
F5	S2, S3, S4, S5	S1

Proses selanjutnya yakni setiap data dilakukan pengujian satu persatu *fold*, sehingga didapatkan nilai rata-rata akurasi pengenalan suara manusia. Berdasarkan Tabel 4.29, berikut proses *cross validation* yang didapatkan untuk tiap *fold*.

Percobaan *training* pertama pada Gambar 4.14 didapatkan nilai $MSE = 0,04046$ dan *testing* pertama pada Gambar 4.15 didapatkan akurasi 0,11111.



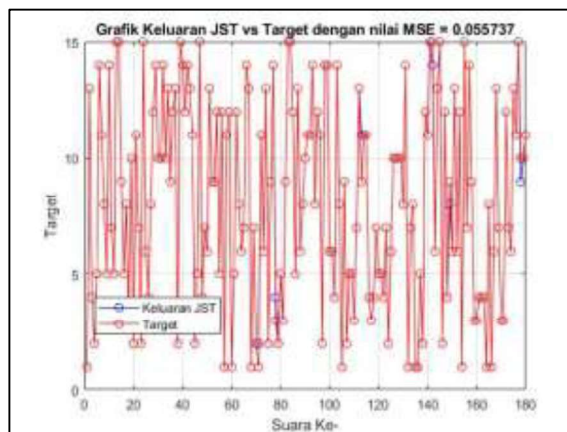
Gambar 4.14 *Training* pada *Fold 1*



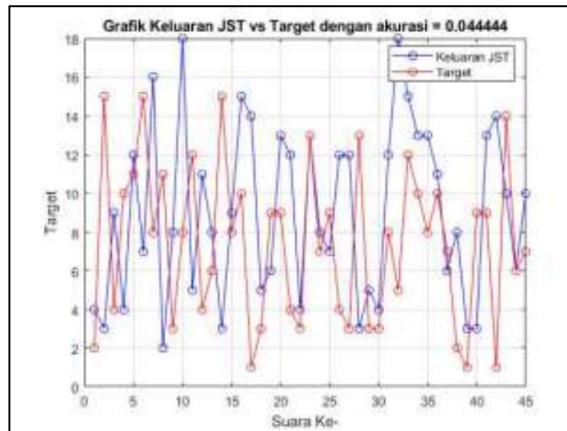
Gambar 4.15 *Testing* pada *Fold 1*

Percobaan *training* kedua pada Gambar 4.16 didapatkan nilai $MSE = 0,055737$

dan *testing* kedua pada Gambar 4.17 didapatkan akurasi 0,044444.

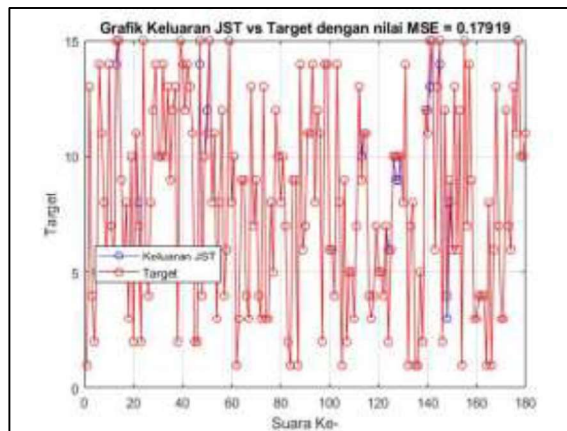


Gambar 4.16 *Training* pada *Fold 2*

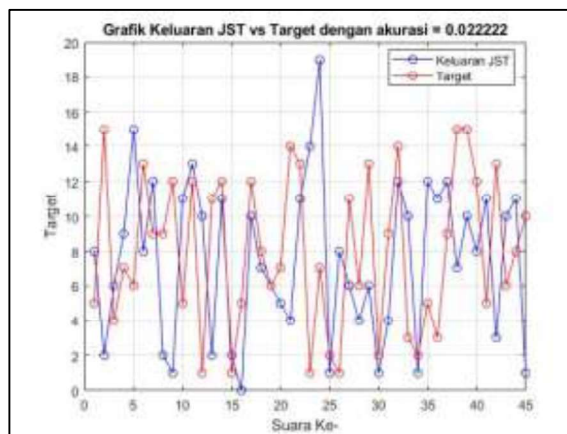


Gambar 4.17 *Testing* pada *Fold 2*

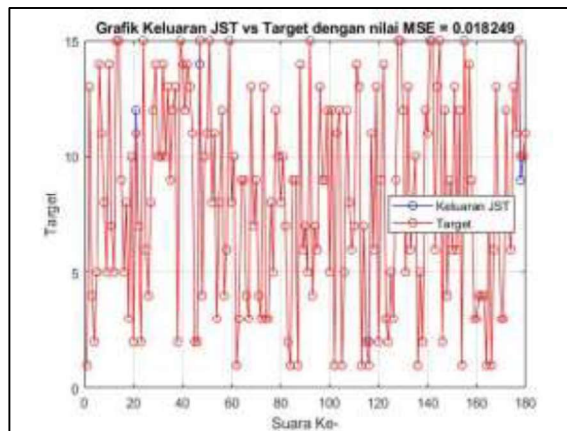
Percobaan *training* ketiga pada Gambar 4.18 didapatkan nilai $MSE = 0,17919$ dan *testing* ketiga pada Gambar 4.19 didapatkan akurasi 0,022222.



Gambar 4.18 *Training* pada *Fold 3*

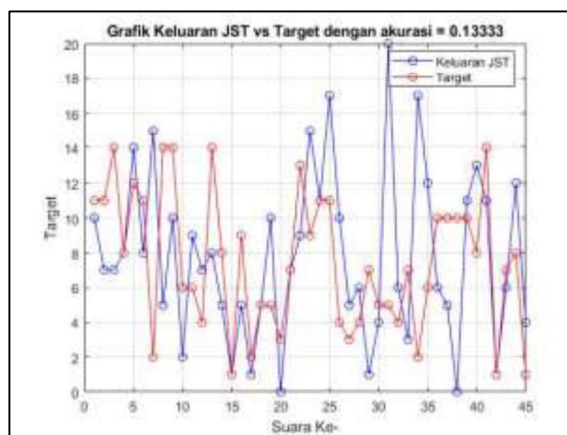


Gambar 4.19 *Testing* pada *Fold 3*

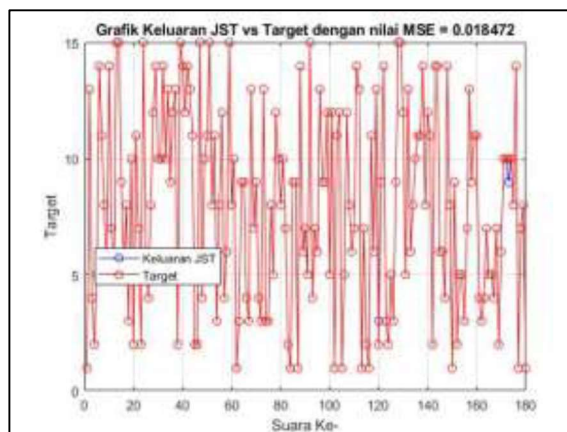


Gambar 4.20 Training pada Fold 4

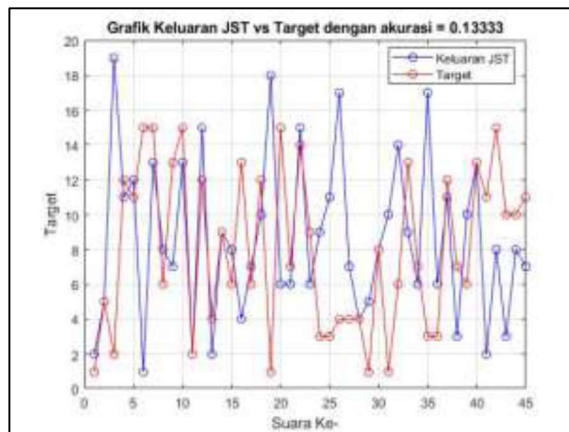
Percobaan *training* keempat pada Gambar 4.20 didapatkan nilai $MSE = 0,018249$ dan *testing* keempat pada Gambar 4.21 didapatkan akurasi 0,13333.



Gambar 4.21 Testing pada Fold 4



Gambar 4.22 Training pada Fold 5



Gambar 4.23 Testing pada *Fold* 5

Percobaan *training* kelima pada Gambar 4.22 didapatkan nilai $MSE = 0,018472$ dan *testing* kelima pada Gambar 4.23 didapatkan akurasi 0,13333.

Berdasarkan hasil proses *K-fold cross validation* didapatkan nilai akurasi pada tiap *fold*. Hasil akurasi tiap *fold* diambil rata-rata keseluruhan pada Tabel 4.30.

Tabel 4.30 Hasil Proses *Cross Validation*

<i>Fold</i>	Akurasi
F1	0,11111
F2	0,04444
F3	0,02222
F4	0,13333
F5	0,13333
Rata-rata	0,088886

Berdasarkan Tabel 4.30, didapatkan nilai akurasi rata-rata sebesar 0,088886. Hal ini mengindikasikan bahwa model memiliki tingkat akurasi yang kecil dalam proses identifikasi. Terdapat beberapa faktor yang mempengaruhi nilai akurasi yang kecil sehingga perlu ditingkatkan untuk mencapai performa yang lebih baik.

4.6 Implementasi Metode JST *Backpropagation* Dalam Pandangan Islam

Suara merupakan salah satu hal yang dapat dipelajari dan diteliti, Banyak orang yang telah meneliti tentang suara. Dengan adanya penelitian tentang suara tersebut memunculkan berbagai macam teknologi dengan menggunakan suara.

Salah satu metode yang dapat digunakan dalam penelitian tentang suara yakni metode Jaringan Saraf Tiruan (JST) *Backpropagation*. Metode JST dibuat berdasarkan sistem kerja dari otak manusia dalam melakukan proses belajar dan mengidentifikasi masalah melalui perubahan bobot sinapsis. Pada algoritma *backpropagation* dilatih untuk dapat mengenali pola *input* suatu data untuk mendapatkan tingkat akurasi yang cukup baik.

Suara mempunyai bentuk gelombang sinyal dan pada gelombang sinyal tersebut terdapat pola yang berbeda-beda untuk tiap suara. Pola pada suara didapatkan dengan melihat dan mengambil ekstraksi fitur atau ciri suara. Hasil ekstraksi fitur suara tersebut digunakan dalam proses identifikasi pada pengenalan suara yang dapat diterapkan pada berbagai aplikasi. Dapat terlihat bahwa terdapat perbedaan antara suara manusia satu dengan manusia yang lain sehingga menjadikan beragamnya jenis suara manusia.

Allah SWT. menciptakan manusia dengan sangat detail dari anggota tubuh hingga suara. Hal tersebut menjadikan manusia sebagai makhluk ciptaan Allah SWT. yang sebaik-baiknya. Kuasa Allah SWT. menciptakan manusia dengan memberi anugerah berupa suara yang beragam. Keberagaman suara manusia dalam hal ini memiliki karakter atau ciri khas yang berbeda-beda pada masing-masing individu. Sekilas memiliki bunyi yang sama, akan tetapi apabila diamati dengan teliti terdapat perbedaan pada suara tersebut. Keberagaman suara manusia menjadikan salah satu bukti bahwa Allah SWT. menciptakan manusia dengan sebaik-baiknya. Perihal tersebut tercantum dalam Al-Qur'an surah At-Tiin [95] ayat 4.

Artinya: “Sesungguhnya Kami telah menciptakan manusia dalam bentuk yang sebaik-baiknya.” (Q.S. At-Tiin: 4).

Sebagai makhluk ciptaan yang sebaik-baiknya, menjadikan manusia untuk menggunakan segala yang diberikan Allah SWT. dalam mempelajari dan mengkaji berbagai aspek kehidupan. Berbagai macam aspek yang dapat dipelajari dan diteliti oleh manusia dalam mengagumi akan kesempurnaan Allah SWT. Mempelajari tentang struktur suara merupakan salah satu cara memuji keagungan Allah SWT. sebagai Sang Pencipta serta selalu berdzikir dan mengingat Allah SWT. Hal tersebut tersirat dalam Al-Qur'an Surah Ali-Imran [2] ayat 190-191.

Artinya: “Sesungguhnya dalam penciptaan langit dan bumi, dan silih bergantinya malam dan siang terdapat tanda-tanda bagi orang-orang yang berakal. (Yaitu) orang-orang yang mengingat Allah sambil berdiri atau duduk atau dalam keadaan berbaring dan mereka memikirkan tentang penciptaan langit dan bumi (seraya berkata): “Ya Rabb, Kami tiadalah Engkau menciptakan ini dengan sia-sia, Mahasuci Engkau, maka peliharalah kami dari siksa Neraka.” (Q.S. Ali-Imran: 190-191).

Berdasarkan ayat di atas bahwa terdapat sebab dan alasan di setiap ciptaan Allah SWT. Alasan dibalik seluruh penciptaan Allah SWT. dapat menjadikan manusia untuk belajar, berpikir, mengamati dan meneliti segala sesuatu yang ada di alam semesta. Berpikir dengan tujuan agar senantiasa bersyukur atas segala yang telah diciptakan Allah SWT. serta memuji akan kuasa Allah SWT.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, didapatkan kesimpulan bahwa metode jaringan saraf tiruan *backpropagation* yang diimplementasikan pada suara manusia diperoleh hasil berikut:

1. Didapatkan model arsitektur jaringan saraf tiruan 64-15-1-1, dimana 64 data masukan, dua *hidden layer* di mana fungsi aktivasi *sigmoid* dengan neuron 15 serta fungsi aktivasi *purelin* dengan neuron 1 dan satu *output layer*. Parameter yang digunakan yakni *error* = 0,01, *learning rate* (α) = 0,01, *epoch* maksimal yang digunakan 3000. Diperoleh waktu pelatihan yakni 15 detik dengan 3000 *epoch*. Nilai MSE yang diperoleh untuk mendekati *error* target 0,01 adalah 0,025776.
2. Menggunakan pengujian akurasi *K-fold cross validation*, didapatkan akurasi model arsitektur jaringan saraf tiruan terhadap data *training* sebesar 0,95 dan terhadap data *testing* sebesar 0,088886. Dapat disimpulkan bahwa model jaringan saraf tiruan berjalan sangat baik terhadap data *training* dan kurang baik terhadap data *testing*.

5.2 Saran

Berdasarkan hasil penelitian ini, terdapat beberapa saran yang diberikan penulis untuk penelitian selanjutnya sebagai berikut:

1. Menggunakan metode ekstraksi fitur yang lain seperti *Pitch-Sincronous Zero-crossing Peak-amplitude* (PS-ZCPA), *Mel-frequency Cepstrum* (MFCC) agar dapat meningkatkan nilai akurasi menjadi lebih baik.
2. Menambah variasi pengujian tentang persentase pembagian data *training* dan data *testing* agar lebih banyak memberikan peluang yang lebih besar.
3. Melakukan perubahan data suara yang bervariasi pada proses *training* dan *testing*.
4. Menggunakan variasi pengucapan kata selain kata “Hai”.

DAFTAR PUSTAKA

- Adinugraha, T. A. C. (2016). *Prediksi Jumlah Pendapatan Asli Daerah Kabupaten Boyolali dengan Metode Jaringan Syaraf Tiruan Backpropagation*. Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- Agustin, M., & Prahasto, T. (2012). Penggunaan Jaringan Syaraf Tiruan Backpropagation Untuk Seleksi Penerimaan Mahasiswa Baru Pada Jurusan Teknik Komputer di Politeknik Negeri Sriwijaya. *Jurnal Sistem Informatika Bisnis*, 02, 89–97.
- Alwi, A. A., Adikara, P. P., & Indriati. (2020). Pengenalan Jenis Kelamin dan Rentang Umur berdasarkan Suara menggunakan Metode Backpropagation Neural Network. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 4(7), 2083–2093.
- Arini, U. D. (2018). *Macam-macam Metode JST*. Garuda Cyber Indonesia. <https://garudacyber.co.id/artikel/1411-macam-macam-metode-jst>
- Chalik, R. (2016). *Anatomi Fisiologi Manusia* (Sunarti & A. Sutisna (eds.)). Kementerian Kesehatan Republik Indonesia.
- Dinuriyati, I. S. (2019). *Klasifikasi Pengenal Suara Kicau Burung Menggunakan Metode Linear Predictive Coding (LPC) dan Nearest Neighbor*. Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- Effendi, A. (2013). *Penggunaan Artificial Neural Network Untuk Mendeteksi Kelainan Mata Miopi pada Manusia dengan Metode Backpropagation*. Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- Faradiba. (2017). Pengenalan Pola Sinyal Suara Manusia Menggunakan Metode Back Propagation Neural Network. *Jurnal EduMatSains*, 2(1), 1–15.
- Fatman, Y., & Islamiyati. (2020). Pengenalan Suku Kata Bahasa Indonesia Menggunakan Metode LPC dan Backpropagation Neural Network. *JOINTECS (Journal of Information Technology and Computer Science)*, 5(3), 155–166.
- Hapsari, J. P. (2014). *Aplikasi Pengenalan Suara Dalam Pengaksesan Sistem Informasi Akademik* [Universitas Diponegoro]. <http://eprints.undip.ac.id/25592/>
- Harum, L. H. (2018). *Implementasi Metode Extreme Learning Machine (ELM) Untuk Memprediksikan Penjualan Roti (Studi Kasus: Harum Bakery)*. Universitas Brawijaya.
- Humaini, Q. (2015). Jaringan Syaraf Tiruan Extreme Learning Machine (Elm)

- Untuk Memprediksi Kondisi Cuaca. *Skripsi*, 1–86.
- Kemenag, Q. (2019). *Qur'an Kemenag*. Kemenag. <https://quran.kemenag.go.id/>
- Kustaman, R. (2017). Bunyi dan Manusia. *ProTVF*, 1(2), 117–124. <https://doi.org/10.24198/ptvf.v1i2.19871>
- Kusuma, D. T. (2021). Fast Fourier Transform (FFT) Dalam Transformasi Sinyal Frekuensi Suara Sebagai Upaya Perolehan Average Energy (AE) Musik. *PETIR: Jurnal Pengkajian Dan Penerapan Teknik Informatika*, 14(1), 28–35. <https://doi.org/10.33322/petir.v14i1.1022>
- Kusumadewi, S. (2003). Artificial Intelligence (Teknik dan Aplikasinya). In *Artificial Intelligence (Teknik dan Aplikasinya)* (1st ed.). Graha Ilmu.
- Loppies, S. H. D. (2018). Implementasi Jaringan Saraf Tiruan Backpropagation Untuk Deteksi Wajah Dalam Citra Digital. *Musamus Journal of Technology & Information*, 1(1), 1–7. <https://doi.org/10.35724/mjti.v1i1.991>
- Nurdiawan, O. (2018). Pengenalan Sandi Morse dengan Menggunakan Jaringan Saraf Tiruan Metode Backpropagation. *Jurnal Ilmiah Manajemen Informatika Dan Komputer*, 02(02), 50–61.
- Pradipta, N. R., Tasripan, T., & Kusuma, H. (2019). Perancangan Perangkat Antarmuka Berbasis Pengenalan Suara pada Purwarupa Mesin Cetak Huruf Braille ITS. *Jurnal Teknik ITS*, 7(2), A401–A407. <https://doi.org/10.12962/j23373539.v7i2.30923>
- Pratiwi, S. W., Mubarak, M. S., & Trisnawati, I. (2021). PEMAKNAAN KONSEP AHSAN TAQWIM (Suatu Kajian Tahlili Pada Q.S al-Tin/95:4). *El-Maqra'*, 1(2), 121–137.
- Ramadhanty, A., Cholissodin, I., & Dewi, C. (2017). Prediksi Jumlah Permintaan Koran Menggunakan Metode Extreme Learning Machine. *Repository.Ub.Ac.Id*, 1–7.
- Seikel, J. A., King, D. W., & Drumright, D. G. (2009). *Anatomy and Physiology dor Speech, Language, and Hearing* (D. Garza (ed.); 4th ed.). Delmar Cengage Learning.
- Setiawan, P. (2021). *Pengertian, Jenis Dan Struktur Neuron (Sel Saraf)*. Gurupendidikan.Com. https://www.gurupendidikan.co.id/neuron-sel-saraf/#Struktur_dan_Bagian_Neuron_Sel_Saraf
- Siang, J. J. (2005). Jaringan Saraf Tiruan dan Pemograman Menggunakan Matlab. In ANDI (Ed.), *ANDI Yogyakarta*. ANDI OFFSET.
- Sofia, W. N. (2021). Interpretasi Imam Al-Maraghi dan Ibnu Katsir Terhadap QS.

- Ali Imran Ayat 190-191. *Tafkir: Interdisciplinary Journal of Islamic Education*, 2(1), 41–57. <https://doi.org/10.31538/tijie.v2i1.16>
- Solikhun, S., Safii, M., & Trisno, A. (2017). Jaringan Saraf Tiruan Untuk Memprediksi Tingkat Pemahaman Sisiwa Terhadap Matapelajaran Dengan Menggunakan Algoritma Backpropagation. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, 1(1), 24–36. <https://doi.org/10.30645/j-sakti.v1i1.26>
- Supartini, I. A. M., Sukarsa, I. K. G., & Srinadi, I. G. A. M. (2017). Analisis Diskriminan Pada Klasifikasi Desa Di Kabupaten Tabanan Menggunakan Metode K-Fold Cross Validation. *E-Jurnal Matematika*, 6(2), 106. <https://doi.org/10.24843/mtk.2017.v06.i02.p154>
- Yodha, J. W., & Kurniawan, A. W. (2014). Perbandingan Penggunaan Deteksi Tepi Dengan Metode Laplace, Sobel, dan Prewit dan Canny pada Pengenalan Pola. *Techno.COM*, 13(3), 189–197.
- Yolanda, R., TM, J., & Iqbal. (2018). Penggunaan Jaringan Saraf Tiruan dalam Pengenalan Pola Huruf Hijaiyah Tulisan Tangan. *Jurnal TIKA Fakultas Ilmu Komputer Universitas Al Muslim*, 3(3), 1–9.
- Zaitun, Warsito, & Pauzi, G. A. (2015). Sistem Identifikasi dan Pengenalan Pola Citra Tanda-Tangan Menggunakan Sistem Jaringan Saraf Tiruan (Artificial Neural Networks) Dengan Metode Backpropagation. *JURNAL Teori Dan Aplikasi Fisika FMIPA Universitas Lampung*, 03(02), 93–101.

Lampiran 1 *Array Data Suara Asli*

<https://tinyurl.com/ArrayDataSuaraAsli>

Lampiran 2 *Array Filter Suara*

<https://tinyurl.com/ArrayFilterSuara>

Lampiran 3 *Array Preemphasis*

<https://tinyurl.com/ArrayPreemphasis>

Lampiran 4 *Array Frame Blocking*

<https://tinyurl.com/ArrayFrameBlocking>

Lampiran 5 *Array Frame Blocking pada Frame ke-2*

<https://tinyurl.com/2ndFrameBlocks>

Lampiran 6 *Array Windowing pada Frame ke-2*

<https://tinyurl.com/2ndFrameWindowing>

Lampiran 7 *Array Windowing*

<https://tinyurl.com/ArrayWindowing>

Lampiran 8 *Array Autokorelasi*

<https://tinyurl.com/ArrayAutokorelasi>

Lampiran 9 *Array Autokorelasi pada Frame ke-2*

<https://tinyurl.com/2ndFrameAutokorelasi>

Lampiran 10 *Array* Analisa LPC

<https://tinyurl.com/ArrayAnalisaLPC>

Lampiran 11 *Array* FFT Suara

<https://tinyurl.com/ArrayFFT>

Lampiran 12 *Array* Normalisasi Satu Suara

<https://tinyurl.com/ArrayNormalisasi1Suara>

Lampiran 13 *Array* Normalisasi Data Suara

<https://tinyurl.com/NormalisasiDataSuara>

Lampiran 14 Bobot Awal v_{ij}

<https://tinyurl.com/BobotAwalvij>

Lampiran 15 Bobot Awal w_{jk}

<https://tinyurl.com/BobotAwalwjk>

Lampiran 16 Bias Awal v_{0j}

<https://tinyurl.com/BiasAwalv0j>

Lampiran 17 Sinyal *Input* z_{inj} pada *Training*

<https://tinyurl.com/TrainingZinj>

Lampiran 18 Sinyal *Input* Z_j pada *Training*

<https://tinyurl.com/TrainingZj>

Lampiran 19 Perbaikan Bobot W_{jk}

<https://tinyurl.com/PerbaikanBobotWjk>

Lampiran 20 Faktor Koreksi δ_{in_j}

<https://tinyurl.com/FaktorKoreksiDeltainj>

Lampiran 21 Faktor Koreksi δ_j

<https://tinyurl.com/FaktorKoreksiDeltaj>

Lampiran 22 Perbaikan Bobot v_{ij}

<https://tinyurl.com/PerbaikanBobotvij>

Lampiran 23 Faktor Koreksi Δv_{0j}

<https://tinyurl.com/FaktorKoreksiDeltav0j>

Lampiran 24 Bobot Baru v_{ij}

<https://tinyurl.com/BobotBaruvij>

Lampiran 25 Bias Baru v_{0j}

<https://tinyurl.com/BiasBaruv0j>

Lampiran 26 Bobot Baru w_{jk}

<https://tinyurl.com/BobotBaruwjk>

Lampiran 27 Sinyal *Input* z_{in_j} pada *Testing*

<https://tinyurl.com/TestingZinj>

Lampiran 28 Sinyal *Input* Z_j pada *Testing*

<https://tinyurl.com/TestingZj>

Lampiran 29 Hasil Pelatihan Model Terbaik Data Latih

Nama	Target Latih	Hasil Latih	Error	Klasifikasi	Kesimpulan
Ali	1	1.080254	-0.08025	1	Benar
Andri	2	2.041759	-0.04176	2	Benar
Cahya	3	3.002239	-0.00224	3	Benar
Dimas	4	4.052847	-0.05285	4	Benar
Doni	5	4.996148	0.00385	5	Benar
Fitrah	6	6.067328	-0.06733	6	Benar
Ike	7	7.018986	-0.01899	7	Benar
Iqbal	8	8.022499	-0.02250	8	Benar
Lia	9	8.980048	0.01995	9	Benar
Mustofa	10	10.04333	-0.04333	10	Benar
Nana	11	10.98466	0.01534	11	Benar
Sabit	12	12.05086	-0.05086	12	Benar
Sania	13	12.98467	0.01533	13	Benar
Wahyu	14	14.00766	-0.00766	14	Benar
Yuslih	15	14.97535	0.02465	15	Benar
Ali	1	0.960807	0.03919	1	Benar
Andri	2	2.018552	-0.01855	2	Benar
Cahya	3	3.052476	-0.05248	3	Benar
Dimas	4	3.76857	0.23143	4	Benar
Doni	5	4.957343	0.04266	5	Benar
Fitrah	6	5.982294	0.01771	6	Benar
Ike	7	6.979929	0.02007	7	Benar
Iqbal	8	8.047408	-0.04741	8	Benar
Lia	9	9.142064	-0.14206	9	Benar
Mustofa	10	10.01391	-0.01391	10	Benar
Nana	11	10.95789	0.04211	11	Benar
Sabit	12	11.8174	0.18260	12	Benar
Sania	13	12.98138	0.01862	13	Benar
Wahyu	14	14.00517	-0.00517	14	Benar
Yuslih	15	15.08826	-0.08826	15	Benar
Ali	1	1.121277	-0.12128	1	Benar
Andri	2	1.98357	0.01643	2	Benar
Cahya	3	2.988038	0.01196	3	Benar
Dimas	4	4.000071	-0.00007	4	Benar
Doni	5	4.978683	0.02132	5	Benar
Fitrah	6	5.93963	0.06037	6	Benar
Ike	7	7.181394	-0.18139	7	Benar
Iqbal	8	7.976336	0.02366	8	Benar
Lia	9	9.142261	-0.14226	9	Benar
Mustofa	10	9.891349	0.10865	10	Benar
Nana	11	11.09879	-0.09879	11	Benar
Sabit	12	11.99979	0.00021	12	Benar
Sania	13	12.98477	0.01523	13	Benar
Wahyu	14	13.98792	0.01208	14	Benar
Yuslih	15	14.9202	0.07980	15	Benar
Ali	1	1.068619	-0.06862	1	Benar
Andri	2	1.824972	0.17503	2	Benar
Cahya	3	2.971291	0.02871	3	Benar
Dimas	4	4.04397	-0.04397	4	Benar
Doni	5	5.048744	-0.04874	5	Benar
Fitrah	6	7.438936	-1.43894	7	Salah

Ike	7	6.978052	0.02195	7	Benar
Iqbal	8	8.364705	-0.36470	8	Benar
Lia	9	9.453179	-0.45318	9	Benar
Mustofa	10	10.01597	-0.01597	10	Benar
Nana	11	10.76082	0.23918	11	Benar
Sabit	12	11.989	0.01100	12	Benar
Sania	13	13.01775	-0.01775	13	Benar
Wahyu	14	14.03044	-0.03044	14	Benar
Yuslih	15	15.058	-0.05800	15	Benar
Ali	1	0.990084	0.00992	1	Benar
Andri	2	1.995757	0.00424	2	Benar
Cahya	3	3.012299	-0.01230	3	Benar
Dimas	4	3.985444	0.01456	4	Benar
Doni	5	5.011138	-0.01114	5	Benar
Fitrah	6	5.983878	0.01612	6	Benar
Ike	7	6.981927	0.01807	7	Benar
Iqbal	8	7.928245	0.07175	8	Benar
Lia	9	8.975601	0.02440	9	Benar
Mustofa	10	9.988655	0.01134	10	Benar
Nana	11	10.96712	0.03288	11	Benar
Sabit	12	12.00898	-0.00898	12	Benar
Sania	13	12.99955	0.00045	13	Benar
Wahyu	14	13.98252	0.01748	14	Benar
Yuslih	15	14.91528	0.08472	15	Benar
Ali	1	1.279379	-0.27938	1	Benar
Andri	2	2.003562	-0.00356	2	Benar
Cahya	3	2.736098	0.26390	3	Benar
Dimas	4	3.995557	0.00444	4	Benar
Doni	5	5.002464	-0.00246	5	Benar
Fitrah	6	6.001352	-0.00135	6	Benar
Ike	7	7.033469	-0.03347	7	Benar
Iqbal	8	8.000432	-0.00043	8	Benar
Lia	9	8.957927	0.04207	9	Benar
Mustofa	10	9.976643	0.02336	10	Benar
Nana	11	10.99544	0.00456	11	Benar
Sabit	12	12.03083	-0.03083	12	Benar
Sania	13	12.99955	0.00045	13	Benar
Wahyu	14	13.94971	0.05029	14	Benar
Yuslih	15	15.11256	-0.11256	15	Benar
Ali	1	1.00959	-0.00959	1	Benar
Andri	2	3.192191	-1.19219	3	Salah
Cahya	3	3.001682	-0.00168	3	Benar
Dimas	4	3.961246	0.03875	4	Benar
Doni	5	5.042748	-0.04275	5	Benar
Fitrah	6	5.976589	0.02341	6	Benar
Ike	7	6.991907	0.00809	7	Benar
Iqbal	8	7.991945	0.00805	8	Benar
Lia	9	8.990839	0.00916	9	Benar
Mustofa	10	9.980078	0.01992	10	Benar
Nana	11	10.92392	0.07608	11	Benar
Sabit	12	11.99693	0.00307	12	Benar
Sania	13	12.98908	0.01092	13	Benar
Wahyu	14	14.03452	-0.03452	14	Benar
Yuslih	15	15.00118	-0.00118	15	Benar
Ali	1	0.97623	0.02377	1	Benar

Andri	2	2.041287	-0.04129	2	Benar
Cahya	3	3.030376	-0.03038	3	Benar
Dimas	4	3.972231	0.02777	4	Benar
Doni	5	4.990025	0.00997	5	Benar
Fitrah	6	6.060503	-0.06050	6	Benar
Ike	7	6.946233	0.05377	7	Benar
Iqbal	8	8.020397	-0.02040	8	Benar
Lia	9	9.22214	-0.22214	9	Benar
Mustofa	10	10.00338	-0.00338	10	Benar
Nana	11	10.97983	0.02017	11	Benar
Sabit	12	12.01288	-0.01288	12	Benar
Sania	13	12.55708	0.44292	13	Benar
Wahyu	14	13.99317	0.00683	14	Benar
Yuslih	15	15.11471	-0.11471	15	Benar
Ali	1	1.008249	-0.00825	1	Benar
Andri	2	1.947207	0.05279	2	Benar
Cahya	3	3.068882	-0.06888	3	Benar
Dimas	4	4.008838	-0.00884	4	Benar
Doni	5	5.033094	-0.03309	5	Benar
Fitrah	6	5.972051	0.02795	6	Benar
Ike	7	6.917496	0.08250	7	Benar
Iqbal	8	7.471738	0.52826	7	Salah
Lia	9	7.194086	1.80591	7	Salah
Mustofa	10	9.960619	0.03938	10	Benar
Nana	11	11.03042	-0.03042	11	Benar
Sabit	12	12.03436	-0.03436	12	Benar
Sania	13	12.98503	0.01497	13	Benar
Wahyu	14	14.0491	-0.04910	14	Benar
Yuslih	15	14.9584	0.04160	15	Benar
Ali	1	1.022023	-0.02202	1	Benar
Andri	2	2.022305	-0.02230	2	Benar
Cahya	3	3.018848	-0.01885	3	Benar
Dimas	4	3.982388	0.01761	4	Benar
Doni	5	4.989928	0.01007	5	Benar
Fitrah	6	6.056073	-0.05607	6	Benar
Ike	7	6.830934	0.16907	7	Benar
Iqbal	8	8.0704	-0.07040	8	Benar
Lia	9	8.910209	0.08979	9	Benar
Mustofa	10	9.969153	0.03085	10	Benar
Nana	11	10.95636	0.04364	11	Benar
Sabit	12	12.00519	-0.00519	12	Benar
Sania	13	13.00861	-0.00861	13	Benar
Wahyu	14	13.95787	0.04213	14	Benar
Yuslih	15	14.93479	0.06521	15	Benar
Ali	1	1.317825	-0.31782	1	Benar
Andri	2	1.992336	0.00766	2	Benar
Cahya	3	3.098486	-0.09849	3	Benar
Dimas	4	4.032946	-0.03295	4	Benar
Doni	5	4.705555	0.29444	5	Benar
Fitrah	6	5.839343	0.16066	6	Benar
Ike	7	7.022653	-0.02265	7	Benar
Iqbal	8	8.024476	-0.02448	8	Benar
Lia	9	9.062803	-0.06280	9	Benar
Mustofa	10	9.649253	0.35075	10	Benar
Nana	11	10.87003	0.12997	11	Benar

Sabit	12	12.03129	-0.03129	12	Benar
Sania	13	13.03409	-0.03409	13	Benar
Wahyu	14	13.98495	0.01505	14	Benar
Yuslih	15	14.37941	0.62059	14	Salah
Ali	1	1.079343	-0.07934	1	Benar
Andri	2	1.986447	0.01355	2	Benar
Cahya	3	2.999693	0.00031	3	Benar
Dimas	4	4.034955	-0.03495	4	Benar
Doni	5	5.013781	-0.01378	5	Benar
Fitrah	6	6.022625	-0.02263	6	Benar
Ike	7	7.715778	-0.71578	7	Benar
Iqbal	8	8.11233	-0.11233	8	Benar
Lia	9	8.963257	0.03674	9	Benar
Mustofa	10	10.16111	-0.16111	10	Benar
Nana	11	11.05661	-0.05661	11	Benar
Sabit	12	11.98977	0.01023	12	Benar
Sania	13	12.96307	0.03693	13	Benar
Wahyu	14	14.02775	-0.02775	14	Benar
Yuslih	15	14.99093	0.00907	15	Benar

Lampiran 30 Hasil Pengujian Model Arsitektur Jaringan Saraf Tiruan

Nama	Target Uji	Hasil Uji	Error	Klasifikasi	Kesimpulan
Ali	1	3.964138	-2.96414	4	Salah
Andri	2	8.173155	-6.17315	8	Salah
Cahya	3	5.628644	-2.62864	6	Salah
Dimas	4	11.45412	-7.45412	11	Salah
Doni	5	15.05236	-10.0524	15	Salah
Fitrah	6	5.877068	0.122932	6	Benar
Ike	7	6.539166	0.460834	7	Benar
Iqbal	8	3.8498	4.1502	4	Salah
Lia	9	8.573824	0.426176	9	Benar
Mustofa	10	4.365619	5.634381	4	Salah
Nana	11	11.37249	-0.37249	11	Benar
Sabit	12	9.546364	2.453636	10	Salah
Sania	13	7.906119	5.093881	8	Salah
Wahyu	14	2.745269	11.25473	3	Salah
Yuslih	15	4.988046	10.01195	5	Salah
Ali	1	7.109795	-6.10979	7	Salah
Andri	2	8.41736	-6.41736	8	Salah
Cahya	3	8.194733	-5.19473	8	Salah
Dimas	4	7.140672	-3.14067	7	Salah
Doni	5	9.896058	-4.89606	10	Salah
Fitrah	6	0.852159	5.147841	1	Salah
Ike	7	10.39351	-3.39351	10	Salah
Iqbal	8	13.30904	-5.30904	13	Salah
Lia	9	12.78224	-3.78224	13	Salah
Mustofa	10	6.175036	3.824964	6	Salah
Nana	11	11.99657	-0.99657	12	Salah
Sabit	12	8.177959	3.822041	8	Salah
Sania	13	16.82811	-3.82811	17	Salah
Wahyu	14	2.779729	11.22027	3	Salah
Yuslih	15	7.171684	7.828316	7	Salah
Ali	1	8.392156	-7.39216	8	Salah
Andri	2	5.474951	-3.47495	5	Salah
Cahya	3	9.539906	-6.53991	10	Salah
Dimas	4	9.519883	-5.51988	10	Salah
Doni	5	15.84187	-10.8419	16	Salah
Fitrah	6	5.415372	0.584628	5	Salah
Ike	7	4.658975	2.341025	5	Salah
Iqbal	8	9.164561	-1.16456	9	Salah
Lia	9	11.06138	-2.06138	11	Salah
Mustofa	10	12.17525	-2.17525	12	Salah
Nana	11	7.232387	3.767613	7	Salah
Sabit	12	6.873937	5.126063	7	Salah
Sania	13	2.773439	10.22656	3	Salah
Wahyu	14	8.95259	5.04741	9	Salah
Yuslih	15	7.219208	7.780792	7	Salah

Lampiran 31 Script Code MATLAB

Filtering Data

```
% Clearing Process
clc;clear all;close all;

% Filtering Data
path_directory = 'datasetSuara';
path_save = 'save';
original_files = dir([path_directory '/*,wav']);
for k = 1:length(original_files)
    filename = [path_directory '/' original_files(k),name];
    % Read audio
    [signal, sample_rate] = audioread(filename);

    % Preprocessing sinyal
    % Filter sinyal
    bandpass_freq = [100, 4000]; % jarak frekuensi bandpass
    dalam Hz
    filter_order = 10; % order filter

    nyquist_freq = sample_rate/2; %frekuensi nyquist (sebagian
    sampling rate)as
    normalized_bandpass_freq = bandpass_freq/nyquist_freq;

    filter_coeffs = firl(filter_order, normalized_bandpass_freq,
    'bandpass');
    filtered_signal = filter(filter_coeffs, 1, signal);
    file_save = [path_save '/filter' original_files(k),name];
    audiowrite(file_save, signal, sample_rate);
end
```

Ekstraksi Fitur

```
% Clearing Process
clc;clear all;close all;

path_directory = 'save';
original_files = dir([path_directory '/*,wav']);
for u = 1:length(original_files)
    filename = [path_directory '/' original_files(u),name];
    % Read audio
    [signal, sample_rate] = audioread(filename);

    % Preemphasis sinyal
    preemphasis_coeff = 0,95; % Koefisien preemphasis (rata-
    rata antara 0,9 and 0,98)

    preemphasized_signal = filter([1, -preemphasis_coeff], 1,
    signal)';

    % Frame blocking sinyal
    % Inisialisasi panjang tiap frame dan pergeseran tiap frame
    M = 256; % Panjang tiap frame
    N = 256; % Ukuran pergeseran tiap frame yang berurutan
```

```

    Fs = floor((length(preemphasized_signal) - M) / N) + 1; %
Menentukan banyaknya frame

    frame_blocks = zeros(M, Fs);

    % Mengisi tiap frame
    frame_blocks(1) = 1;
    for i = 1:Fs
        frame_start = (i - 1) * N + 1;
        frame_end = frame_start + M - 1;
        frame_blocks(:, i) =
preemphasized_signal(frame_start:frame_end);
    end

    % Windowing sinyal
    % Inisialisasi panjang window
    N = 256; % Panjang window

    window = hamming(N); % Menggunakan metode hamming window

    windowed_signal = bsxfun(@times, frame_blocks, window); %
Perhitungan windowing

    % Analisa autokorelasi
    % Inisialisasi lag dan matriks hasil autokorelasi
    N = length(windowed_signal(1,:));
    % Lag
    autokorelasi = zeros(1, N); % Matriks untuk menyimpan hasil
autokorelasi

    % Perhitungan autokorelasi
    for i = 1:size(windowed_signal, 1) % Iterasi setiap saluran
suara
        for k = 0:N-1 % Iterasi setiap lag
            sum = 0;
            for j = 1:N-k % Iterasi setiap elemen dalam lag
                sum = sum + windowed_signal(i, j) *
windowed_signal(i, j+k);
            end
            autokorelasi(i, k+1) = sum;
        end
    end

    % Analisa LPC
    % Inisialisasi order LPC, matriks dan error prediksi awal
    order = 32; % Order LPC
    a = zeros(order, 1); % Matriks koefisien LPC
    E = autokorelasi(1); % Error prediksi awal

    % Iterasi Levinson-Durbin
    for k = 2:order
        lambda = autokorelasi(k);
        for j = 1:k-1
            lambda = lambda - a(j+1)*autokorelasi(k-j+1);
        end
        lambda = lambda / E;

        a_temp = a;

```

```

        for j = 1:k-1
            a(j+1) = a_temp(j+1) - lambda*a_temp(k-j+1);
        end
        a(k) = lambda;

        E = (1 - lambda^2) * E;
    end

    % Fast fourier transform
    % Inisialisasi input data, panjang sinyal dan array FFT
    x = a; % Sinyal dengan dua komponen frekuensi
    N = length(x); % Panjang sinyal
    X = zeros(1, N); % Array hasil FFT

    % Looping untuk menghitung FFT
    for k = 0:N-1
        for n = 0:N-1
            X(k+1) = X(k+1) + x(n+1) * exp(-1i * 2 * pi * k * n
/ N);
        end
    end

    X = abs(X);

    % Normalisasi data
    % Mencari nilai minimum dan maksimum data
    min_val = min(X);
    max_val = max(X);

    % Menghitung rentang data
    range = max_val - min_val;

    % Melakukan normalisasi data ke rentang -1 hingga 1
    normalized_data = ((X - min_val) / range) * 2 - 1;

    % Save fitur suara
    save_fitur(u, :) = normalized_data;
end

% Export ke excel
xlswrite('input_data', save_fitur);

```

Pelatihan dan Pengujian Jaringan

```

% Clearing Process
clc;clear;close all;

% Proses membaca data latih dari excel
filenamel = 'input_data3,xls';
sheet1 = 'Training';
xlRangel = 'A1:BN180';

Data = xlsread(filenamel, sheet1, xlRangel);
data_latih = Data(:,1:64)';
target_latih = Data(:,65)';
[m,n] = size(data_latih);

```

```

% Pembuatan JST
net = newff(minmax(data_latih),[15 1],{'logsig',
'purelin'},'traingdx');

% Memberikan nilai untuk mempengaruhi proses pelatihan
net,performFcn = 'mse';
net,trainParam,goal = 0,01;
net,trainParam,show = 20;
net,trainParam,epochs = 3000;
net,trainParam,mc = 0,95;
net,trainParam,lr = 0,01;

% Proses training
[net_keluaran,tr,Y,E] = train(net,data_latih,target_latih);

% Hasil setelah pelatihan
bobot_hidden = net_keluaran,IW{1,1};
bobot_keluaran = net_keluaran,LW{2,1};
bias_hidden = net_keluaran,b{1,1};
bias_keluaran = net_keluaran,b{2,1};
jumlah_iterasi = tr,num_epochs;
nilai_keluaran = Y;
nilai_error = E;
error_MSE = (1/n)*sum(nilai_error,^2);

% Performa pelatihan jaringan
hasil_latih = round(sim(net_keluaran,data_latih));
hasil_latih = abs(hasil_latih);

save net,mat net_keluaran

% Proses membaca data uji dari excel
filename2 = 'input_data3,xls';
sheet2 = 'Testing';
xlRange2 = 'A1:BN45';

Data2 = xlsread(filename2, sheet2, xlRange2);
data_uji = Data2(:,1:64)';
target_uji = Data2(:,65)';
[m,n] = size(data_uji);

% Performa pengujian jaringan
hasil_uji = round(sim(net_keluaran,data_uji));
hasil_uji = abs(hasil_uji);

% Plot hasil pelatihan dengan target latih
figure,
plot(hasil_latih,'bo-')
hold on
plot(target_latih,'ro-')
hold off
grid on
title(strcat(['Grafik Keluaran JST vs Target dengan nilai MSE = ',
',,,,
num2str(error_MSE)]))
xlabel('Suara Ke-')
ylabel('Target')
legend('Keluaran JST','Target','Location','Best')

```

```

% Plot hasil pengujian dengan target uji
figure,
plot(hasil_uji,'bo-')
hold on
plot(target_uji,'ro-')
hold off
grid on
title(strcat(['Grafik Keluaran JST vs Target dengan nilai MSE = ',
',,,,,
num2str(error_MSE)]))
xlabel('Suara Ke-')
ylabel('Target')
legend('Keluaran JST','Target','Location','Best')

```

K-fold cross validation

```

clc;clear all;close all;
% Proses membaca data latih dari excel
filename1 = 'input_data3.xls';
sheet1 = 'Original';
xlRangel = 'B1:BO225';
Data = xlsread(filename1, sheet1, xlRangel);

% Split data train dan data test
[row,col] = size(Data);
% Acak indeks terlebih dahulu
randomIndexes = randperm(row);
for i=1:1:col
    % Kemudian Shuffle data berdasarkan indeks yang sudah diacak
    sebelumnya
    dataNew(:,i) = Data(randomIndexes,i);
end
dataNew;

% k-fold Cross validation
k = 5; % Contoh, kita ambil k=5
numTest = row/k;

for j=1:1:k
    idx_test=(j-1)*numTest+1:numTest*j;
    idx_train1=1:numTest*(j-1);
    idx_train2=j*numTest+1:row;
    idx_train=[idx_train1 idx_train2];

    for i=1:1:col
        testing(:,i) = dataNew(idx_test,i);
        training(:,i)= dataNew(idx_train,i);
    end
    testing;
    training;

    training_data = training(:,1:64)';
    target_training = training(:,65)';
    [m,n] = size(training_data);

```

```

% Pembuatan JST
net = newff(minmax(training_data),[15 1],{'logsig',
'purelin'},'traingdx');

% Memberikan nilai untuk mempengaruhi proses pelatihan
net.performFcn = 'mse';
net.trainParam.goal = 0.01;
net.trainParam.show = 20;
net.trainParam.epochs = 3000;
net.trainParam.mc = 0.95;
net.trainParam.lr = 0.01;

% Proses training
[net_keluaran,tr,Y,E] =
train(net,training_data,target_training);

% Hasil setelah pelatihan
bobot_hidden = net_keluaran.IW{1,1};
bobot_keluaran = net_keluaran.LW{2,1};
bias_keluaran = net_keluaran.b{2,1};
bias_hidden = net_keluaran.b{1,1};
jumlah_iterasi = tr.num_epochs;
nilai_keluaran = Y;
nilai_error = E;
error_MSE = (1/n)*sum(nilai_error.^2);

% Performa pelatihan jaringan
hasil_latih = round(sim(net_keluaran,training_data));
hasil_latih = abs(hasil_latih);

save net.mat net_keluaran

% Proses membaca data uji dari excel
testing_data = testing(:,1:64)';
target_test = testing(:,65)';
[m,n] = size(testing_data);

% Performa pengujian jaringan
hasil_uji = round(sim(net_keluaran,testing_data));
hasil_uji = abs(hasil_uji);

predictedLabels = round(hasil_uji);
accuracy = sum(predictedLabels == target_test) /
length(target_test);
evaluationResults(j) = accuracy;

% Plot hasil pelatihan dengan target latih
figure,
plot(hasil_latih,'bo-');
hold on;
plot(target_training,'ro-');
hold off;
grid on;
title(strcat(['Grafik Keluaran JST vs Target dengan nilai
MSE = ',...
num2str(error_MSE)]));
xlabel('Suara Ke-');
ylabel('Target');
legend('Keluaran JST','Target','Location','Best');

```

```
    % Plot hasil pengujian dengan target uji
    figure,
    plot(hasil_uji,'bo-');
    hold on;
    plot(target_test,'ro-');
    hold off;
    grid on;
    title(strcat(['Grafik Keluaran JST vs Target dengan akurasi
= ',...
    num2str(evaluationResults(j))]);
    xlabel('Suara Ke-');
    ylabel('Target');
    legend('Keluaran JST','Target','Location','Best');

end

averageEvaluation = mean(evaluationResults);
disp("Rata-rata Evaluasi: " + averageEvaluation);
```

RIWAYAT HIDUP



Mohammad Bagus Dimas Prayugo atau lebih dikenal Dimas, Lahir di Kabupaten Mojokerto, 24 September 2000. Penulis merupakan anak kedua dari dua bersaudara. Putra dari pasangan Bapak Ridwan dan Ibu Nuryati yang bertempat tinggal di Dusun Tlagan RT/RW 001/006, Desa Pohjejer, Kecamatan Gondang, Kabupaten Mojokerto. Pertama kali penulis menempuh pendidikan di MI Pancasila Pohjejer pada tahun 2007 dan tamat pada tahun 2013 kemudian melanjutkan pendidikan ke MTs Pancasila dan tamat pada tahun 2016. Setelah tamat dari MTs, penulis melanjutkan pendidikan di MA Miftahul Ulum Gondang dan tamat pada tahun 2019. Pada tahun yang sama, penulis terdaftar sebagai mahasiswa Program Studi Matematika, Fakultas Sains dan Teknologi di Universitas Islam Negeri Maulana Malik Ibrahim Malang melalui jalur SNMPTN. Selama menempuh pendidikan, penulis mengikuti berbagai organisasi, diantaranya IPNU dan HMJ “Integral” Matematika selama satu periode. Penulis mengikuti organisasi IPNU pada tahun 2017 sampai 2019 sebagai ketua PAC IPNU Kecamatan Gondang dan HMJ “Integral” Matematika pada tahun 2021 sebagai anggota Divisi Penerbitan dan Jurnalistik. Penulis dapat dihubungi melalui *e-mail*: namikazedimas2@gmail.com.



**KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

BUKTI KONSULTASI SKRIPSI

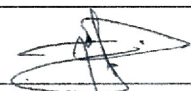
Nama : Mohammad Bagus Dimas Prayugo
NIM : 19610001
Fakultas / Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Implementasi Metode Jaringan Saraf Tiruan
Backpropagation Pada Pengenalan Suara Manusia
Pembimbing I : Hisyam Fahmi, M.Kom.
Pembimbing II : Achmad Nashichuddin, M.A.

No	Tanggal	Hal	Tanda Tangan
1.	24 Februari 2023	Konsultasi Bab I, II, dan III	1.
2.	2 Maret 2023	Konsultasi Kajian Agama Bab I dan II	2.
3.	3 Maret 2023	Konsultasi Bab I, II, dan III	3.
4.	13 Maret 2023	Konsultasi Kajian Agama Bab I dan II	4.
5.	14 Maret 2023	ACC Bab I, II, dan III	5.
6.	15 Maret 2023	ACC Kajian Agama Bab I dan II	6.
7.	6 April 2023	ACC Seminar Proposal	7.
8.	30 Mei 2023	Konsultasi Revisi Seminar Proposal dan Bab IV dan V	8.
9.	30 Mei 2023	Konsultasi Kajian Agama Bab IV	9.
10.	31 Mei 2023	ACC Bab IV dan V	10.
11.	5 Juni 2023	ACC Kajian Agama Bab IV	11.
12.	14 Juni 2023	ACC Seminar Hasil	12.
13.	15 Juni 2023	Konsultasi Revisi Seminar Hasil	13.
14.	21 Juni 2023	ACC Sidang Skripsi	14.



**KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

15.	23 Juni 2023	ACC Keseluruhan	15. 
-----	--------------	-----------------	---

Malang, 23 Juni 2023

Mengetahui,
Ketua Program Studi Matematika




Dr. Elly Susanti, M.Sc

NIP. 19741129 200012 2 005