

**IMPLEMENTASI JARINGAN SYARAF TIRUAN
BACKPROPAGATION UNTUK MENDIAGNOSIS PENYAKIT
KULIT PADA ANAK**

SKRIPSI

Untuk memenuhi persyaratan memperoleh gelas Sarjana Komputer

Disusun oleh:
Rokky Septian Suhartanto
135150201111063



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2017**

PENGESAHAN

IMPLEMENTASI JARINGAN SYARAF TIRUAN BACKPROPAGATION UNTUK
MENDIAGNOSIS PENYAKIT KULIT PADA ANAK

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
ROKKY SEPTIAN SUHARTANTO
NIM: 135150201111063

Skripsi ini telah diuji dan dinyatakan lulus pada
24 Mei 2017

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Candra Dewi, S.Kom., M.Sc
NIP: 1977114 200312 2 001

Lailil Muflikhah, S.Kom., M.Sc
NIP: 19741113 200501 2 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 17 April 2017

Rokky Septian Suhartanto

NIM: 135150201111063



KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah SWT, karena dengan rahmat dan karunia-Nya penulis dapat menyelesaikan tugas akhir yang berjudul **“Implementasi Jaringan Syaraf Tiruan Backpropagation Untuk Mendiagnosis Penyakit Kulit Pada Anak”** dengan baik.

Shalawat dan salam selalu penulis haturkan untuk nabi besar Nabi Muhammad SAW, yang dengan semangat juangnya terus memberikan tauladan untuk tetap bersemangat dan tidak berputus asa dengan rahmat-Nya. Maka tanpa itu semua, niscaya penulis tidak dapat menyelesaikan tugas akhir dengan baik dan tepat waktu.

Penulisan dan penyusunan laporan skripsi ini dapat terlaksana dengan baik karena adanya bantuan secara langsung maupun tidak langsung dari pihak tertentu diantaranya:

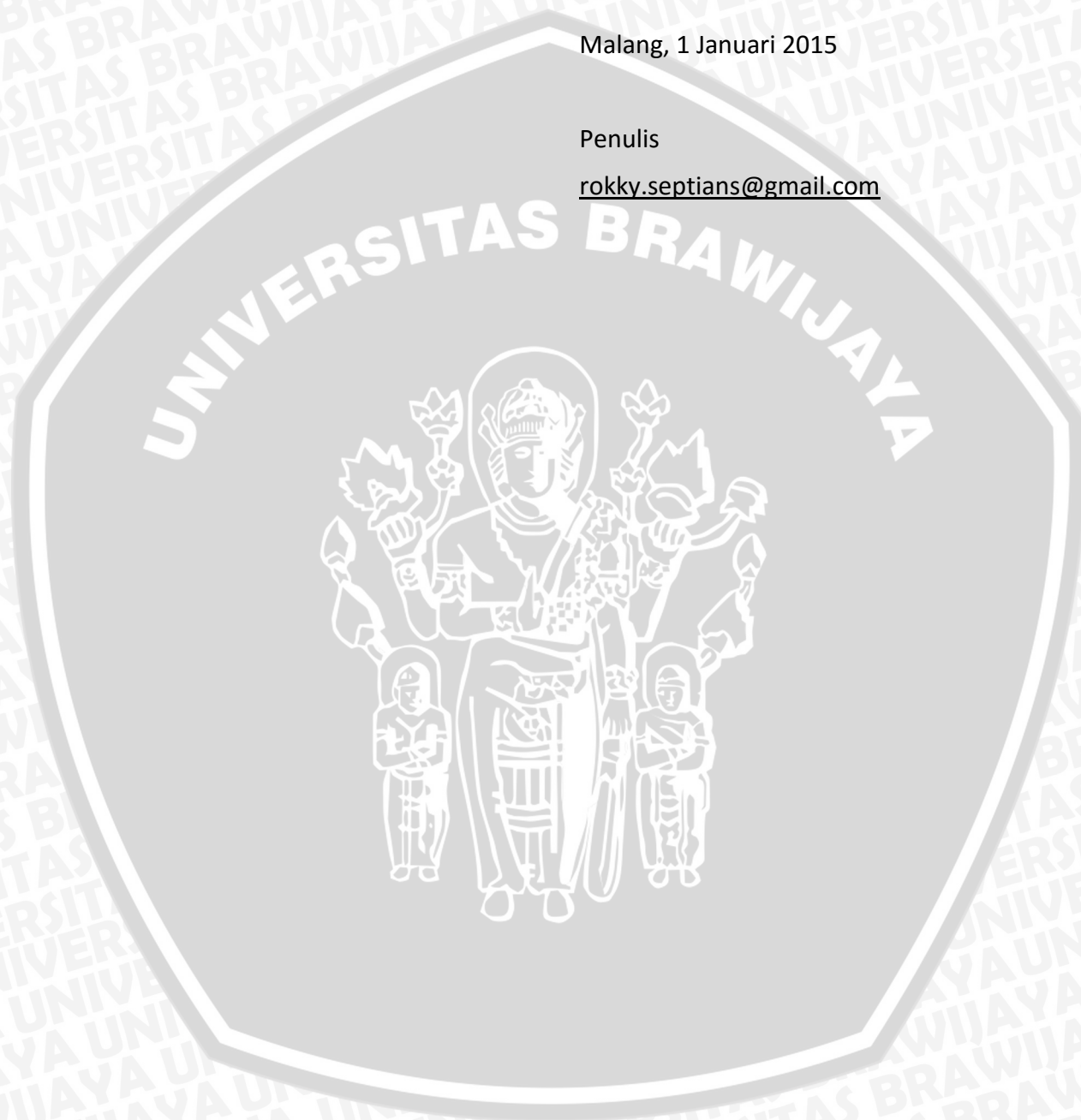
1. Kedua orang tua, Bapak Rubadi & Ibu Sri Hartini yang memberikan motivasi, kasih sayang, doa, serta dukungannya selama ini.
2. Ibu Candra Dewi, S.Kom.,M.Sc. selaku dosen pembimbing I yang telah banyak memberikan ilmu dan saran dalam menyelesaikan laporan ini.
3. Ibu Lailil Muflikhah, S.Kom.,M.Sc. selaku dosen pembimbing II yang juga banyak memberikan ilmu dan saran dalam menyelesaikan laporan ini.
4. Ibu Dian Eka Ratnawati, S.Si., M.Kom. selaku dosen pembimbing akademik yang selalu memberikan kritik dan saran serta bimbingan dalam pelaksanaan kuliah.
5. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
6. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
7. Putri Nur Laily yang tidak pernah lelah memberi doa, support dan mendorong penulis untuk semangat dalam menyelesaikan laporan ini.
8. D' Kontrakan, Indra, Arab, Ipung, Aldi, terimakasih untuk support dan semangatnya selama masa kuliah ini.
9. Ambaru yang menjadi keluarga kedua selama berada di Malang.
10. Semua teman-teman FILKOM, khususnya Informatika 2013 terima kasih atas segala bantuan dan dukungannya selama ini.
11. Segenap dosen dan karyawan FILKOM Universitas Brawijaya yang telah membantu pelaksanaan skripsi ini.
12. Semua pihak yang tidak dapat disebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas ini

Semoga jasa dan amal baiknya mendapatkan balasan dari Allah SWT. Dengan segala kerendahan hati, penulis menyadari sepenuhnya bahwa skripsi ini masih jauh dari sempurna karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Akhirnya semoga skripsi ini dapat bermanfaat dan berguna bagi pembaca terutama mahasiswa FILKOM Universitas Brawijaya.

Malang, 1 Januari 2015

Penulis

rokky.septians@gmail.com

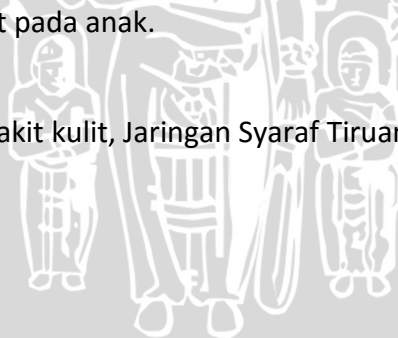


ABSTRAK

Suhartanto Rokky Septian. 2017. Implementasi Jaringan Syaraf Tiruan Backpropagation Untuk Mendiagnosis Penyakit Kulit Pada Anak. Fakultas Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing: Candra Dewi S.Kom., M.Sc. dan Lailil Muflikhah S.Kom., M.Sc.

Sistem imun yang dimiliki oleh anak-anak yang lebih lemah dibandingkan orang dewasa membuat anak-anak lebih gampang terserang penyakit. Penyakit kulit adalah salah satunya, hal ini dikarenakan kulit merupakan indra peraba bagi manusia. Adanya kemiripan gejala dari setiap penyakit kulit membuat orang awam sulit membedakan penyakit yang di derita. Padahal setiap jenis penyakit memiliki penanganan dan pengobatan yang berbeda-beda. Pada penelitian ini mengimplementasikan metode jaringan syaraf tiruan *backpropagation* untuk mempelajari data yang lampau agar dapat mendiagnosis penyakit kulit pada anak. Masukan yang digunakan berupa gejala dari semua penyakit yang berjumlah 19 kemudian di representasikan kedalam biner 0 dan 1 dimana nilai akan bernilai 1 jika mengalami gejala tersebut dan sebaliknya. Fungsi aktivasi yang digunakan adalah *sigmoid biner*. Bobot awal didapat menggunakan Nguyen-Widrow yang kemudian akan dilakukan pembelajaran secara berulang-ulang sehingga dihasilkan jaringan yang memberi tanggapan benar terhadap masukannya Berdasarkan hasil pengujian yang telah dilakukan didapatkan parameter yang optimal yaitu pada *hidden neuron* berjumlah 4 , *learning rate* 0.4 dan *epoch* maksimum 300000 dan hasil akurasi dari penelitian mencapai 87.22 % yang menunjukkan bahwa metode *backpropagation* ini dapat digunakan dalam mendiagnosis penyakit kulit pada anak.

Kata Kunci: Diagnosis penyakit kulit, Jaringan Syaraf Tiruan, *Backpropagation*.



ABSTRACT

Suhartanto Rokky Septian. 2017. *Implementation of Neural Network Backpropagation To Diagnose Skin Diseases in Children*. Faculty of Computer Science, University of Brawijaya, Malang. Advisor: Candra Dewi S.Kom., M.Sc. dan Lailil Muflikhah S.Kom., M.Sc.

Immune systems owned by children who are weaker than adults make children more susceptible to disease. Skin disease is one of them, this is because the skin is the sense of touch for humans. The similarity of symptoms of any skin disease makes the layman difficult to distinguish the illness in suffering whereas every type of disease has a different treatment. In this study implements artificial neural network method backpropagation to study the past data in order to diagnose skin diseases in children. The input used in the form of symptoms of all diseases amounted to 19 then represented into binary 0 and 1 where the value will be worth 1 if experiencing the symptoms and vice versa. The activation function used is sigmoid binner. The initial weights are obtained using Nguyen-Widrow which will then be done by repeatedly learning so that the result of the network that gives the correct response to the input. Based on the result of the test, the optimal parameters are 4 hidden learning neurons, learning rate 0.4 and epoch maximum 300000 and The results of the accuracy of the study reached 87.22% which indicates that this backpropagation method can be used in diagnosing skin diseases in children.

Keywords: *Diagnosis of skin diseases, Neural Network, Backpropagation.*



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah.....	3
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.1 Dasar Teori	6
2.1.1 Penyakit Kulit Anak.....	6
2.2.2 Jaringan Syaraf Tiruan (JST)	9
BAB 3 METODOLOGI	17
3.1 Studi Literatur	17
3.2 Pengumpulan data.....	18
3.3 Perancangan Sistem.....	18
3.3.1 Analisa Lingkungan Pengujian.....	18
3.4 Implementasi Sistem	19
3.5 Pengujian dan Analisis	19
3.6 Kesimpulan dan Saran	19
BAB 4 Perancangan	20
4.1 Perhitungan <i>Backpropagation</i>	20
4.1.1 <i>Feed Forward</i>	20

4.1.2 <i>Feed Backward</i>	20
4.1.3 Diagram Alir <i>Backpropagation</i>	20
4.1.4 Perhitungan Manual.....	25
4.2 Implementasi	73
4.3 Pengujian dan Analisis	73
4.4 Pengambilan Kesimpulan Dan Saran	75
BAB 5 IMPLEMENTASI	76
5.1 Spesifikasi Sistem	76
5.1.1 Spesifikasi Perangkat Keras.....	76
5.1.2 Spesifikasi perangkat lunak.....	76
5.2 Batasan Implementasi	76
5.3 Implementasi Algoritma	77
5.3.1 Algoritma Inisialisasi Bobot dengan Nguyen – Widrow.....	77
5.3.2 Algoritma Feed Forward	79
5.3.3 Algoritma Feed Backward	80
BAB 6 PENGUJIAN DAN ANALISIS.....	83
6.1 Pengujian dan Analisis	83
6.1.1 Pengujian dan Analisis Perbedaan Jumlah <i>Hidden Neuron</i>	83
6.1.2 Pengujian dan Analisis Perbedaan Nilai <i>Learning Rate</i>	85
6.1.3 Pengujian dan Analisis Perbedaan Nilai <i>Epoch</i> Maksimum	88
6.1.4 Pengujian dan Analisis Akurasi Sistem	90
BAB 7 PENUTUP	91
7.1 Kesimpulan.....	91
7.2 Saran	91
DAFTAR PUSTAKA.....	92
LAMPIRAN 1	94
LAMPIRAN 2	101

DAFTAR TABEL

Table 3.1 Data Pasien	18
Table 4.1 Data latihan manualisasi yang akan digunakan	27
Table 4.2 Bobot awal acak antara <i>input layer</i> dengan <i>Hidden Layer</i> (V_{ij})	28
Table 4.3 Bobot awal acak antara <i>Hidden Layer</i> dengan <i>output layer</i> (W_{ij})	29
Table 4.4 Hasil perhitungan $ V_{ij} $	29
Table 4.5 Hasil Perhitungan $ W_{ij} $	29
Table 4.6 Hasil perhitungan bobot awal (V_{ij}) <i>Nguyen-Widrow</i>	30
Table 4.7 Hasil perhitungan bobot bias dengan <i>Nguyen-Widrow</i>	31
Table 4.8 Hasil perhitungan bobot W_{ij} dan bias dengan <i>Nguyen-Widrow</i>	31
Table 4.9 Hasil perhitungan Z_{in}	32
Table 4.10 Hasil perhitungan Z_j	32
Table 4.11 Hasil Perhitungan Y_{in}	33
Table 4.12 Hasil Perhitungan Y_{out}	33
Table 4.13 hasil perhitungan δ_k	33
Table 4.14 Hasil perhitungan koreksi bobot ΔW_{ij}	34
Table 4.15 Hasil perhitungan $\delta_{in,j}$	35
Table 4.16 Hasil Perhitungan δ_j	35
Table 4.17 Hasil Perhitungan koreksi bobot ΔV_{ij} & bias	36
Table 4.18 Hasil perhitungan bobot baru V_{ij}	37
Table 4.19 Hasil perhitungan bobot baru W_{ij}	38
Table 4.20 Hasil perhitungan Z_{in} Anak2 Iterasi 1	38
Table 4.21 Hasil perhitungan Z_j Anak2 Iterasi 1	39
Table 4.22 Hasil Perhitungan Y_{in} Anak2 Iterasi 1	39
Table 4.23 Hasil Perhitungan Y_{out} Anak2 Iterasi 1	40
Table 4.24 nilai δ_k Anak2 Iterasi 1	40
Table 4.25 Hasil perhitungan koreksi bobot ΔW_{ij} Anak2 Iterasi 1	41
Table 4.26 Hasil perhitungan $\delta_{in,j}$ Anak2 Iterasi 1	41
Table 4.27 Hasil Perhitungan δ_j Anak2 Iterasi 1	42
Table 4.28 Hasil Perhitungan koreksi bobot ΔV_{ij} & bias Anak2 Iterasi 1	42
Table 4.29 Hasil perhitungan bobot baru V_{ij} Anak 2 Iterasi 1	43
Table 4.30 Hasil perhitungan bobot baru W_{ij} Anak2 Iterasi 1	44

Table 4.31 Hasil perhitungan Z_{in} Anak3 Iterasi 1.....	45
Table 4.32 Hasil perhitungan Z_j Anak3 Iterasi 1.....	45
Table 4.33 Hasil Perhitungan Y_{in} Anak3 Iterasi 1	46
Table 4.34 Hasil Perhitungan Y_{out} Anak3 Iterasi 1	46
Table 4.35 nilai δ_k Anak3 Iterasi 1.....	47
Table 4.36 Hasil perhitungan koreksi bobot ΔW_{ij} Anak3 Iterasi 1.....	47
Table 4.37 Hasil perhitungan δ_{in_j} Anak3 Iterasi 1.....	48
Table 4.38 Hasil Perhitungan δ_j Anak3 Iterasi 1.....	48
Table 4.39 Hasil Perhitungan koreksi bobot ΔV_{ij} & bias Anak3 Iterasi 1	49
Table 4.40 Hasil perhitungan bobot baru V_{ij} Anak 3 Iterasi 1	50
Table 4.41 Hasil perhitungan bobot baru W_{ij} Anak3 Iterasi 1.....	51
Table 4.42 Hasil perhitungan Z_{in} Anak4 Iterasi 1.....	51
Table 4.43 Hasil perhitungan Z_j Anak4 Iterasi 1.....	52
Table 4.44 Hasil Perhitungan Y_{in} Anak4 Iterasi 1	52
Table 4.45 Hasil Perhitungan Y_{out} Anak4 Iterasi 1	53
Table 4.46 nilai δ_k Anak4 Iterasi 1.....	53
Table 4.47 Hasil perhitungan koreksi bobot ΔW_{ij} Anak4 Iterasi 1.....	54
Table 4.48 Hasil perhitungan δ_{in_j} Anak4 Iterasi 1.....	54
Table 4.49 Hasil Perhitungan δ_j Anak4 Iterasi 1.....	55
Table 4.50 Hasil Perhitungan koreksi bobot ΔV_{ij} & bias Anak4 Iterasi 1	55
Table 4.51 Hasil perhitungan bobot baru V_{ij} Anak 4 Iterasi 1	56
Table 4.52 Hasil perhitungan bobot baru W_{ij} Anak4 Iterasi 1.....	57
Table 4.53 Hasil perhitungan Z_{in} Anak5 Iterasi 1.....	58
Table 4.54 Hasil perhitungan Z_j Anak5 Iterasi 1.....	58
Table 4.55 Hasil Perhitungan Y_{in} Anak5 Iterasi 1	59
Table 4.56 Hasil Perhitungan Y_{out} Anak5 Iterasi 1	59
Table 4.57 nilai δ_k Anak5 Iterasi 1.....	60
Table 4.58 Hasil perhitungan koreksi bobot ΔW_{ij} Anak5 Iterasi 1.....	60
Table 4.59 Hasil perhitungan δ_{in_j} Anak5 Iterasi 1.....	61
Table 4.60 Hasil Perhitungan δ_j Anak5 Iterasi 1.....	61
Table 4.61 Hasil Perhitungan koreksi bobot ΔV_{ij} & bias Anak5 Iterasi 1	62
Table 4.62 Hasil perhitungan bobot baru V_{ij} Anak 5 Iterasi 1	63

Table 4.63 Hasil perhitungan bobot baru W_{ij} Anak5 Iterasi 1.....	64
Table 4.64 Hasil perhitungan Z_{in} Anak6 Iterasi 1.....	64
Table 4.65 Hasil perhitungan Z_j Anak6 Iterasi 1.....	65
Table 4.66 Hasil Perhitungan Y_{in} Anak6 Iterasi 1	65
Table 4.67 Hasil Perhitungan Y_{out} Anak6 Iterasi 1	66
Table 4.68 nilai δ_k Anak6 Iterasi 1.....	66
Table 4.69 Hasil perhitungan koreksi bobot ΔW_{ij} Anak6 Iterasi 1.....	67
Table 4.70 Hasil perhitungan $\delta_{in,j}$ Anak6 Iterasi 1.....	67
Table 4.71 Hasil Perhitungan δ_j Anak6 Iterasi 1.....	68
Table 4.72 Hasil Perhitungan koreksi bobot ΔV_{ij} & bias Anak6 Iterasi 1.....	68
Table 4.73 Hasil perhitungan bobot baru V_{ij} Anak6 Iterasi 1	69
Table 4.74 Hasil perhitungan bobot baru W_{ij} Anak6 Iterasi 1.....	70
Table 4.75 Data Uji.....	71
Table 4.76 Hasil perhitungan Z_{in} Data Uji.....	71
Table 4.77 Hasil perhitungan Z_j Data Uji.....	72
Table 4.78 Hasil Perhitungan Y_{in} Data Uji.....	72
Table 4.79 Hasil Perhitungan Y_{out} Data Uji	73
Table 4.80 Y_{out} yang telah di bulatkan	73
Table 4.81 Tabel Pengujian Perbedaan Jumlah <i>Hidden Neuron</i>	74
Table 4.82 Tabel Pengujian Nilai <i>Learning rate</i>	74
Table 4.83 Tabel Pengujian Nilai Maksimum Epoch	74
Table 4.84 Tabel Pengujian Akurasi	75
Table 5.1 Spesifikasi Perangkat Keras	76
Table 5.2 Spesifikasi Perangkat Lunak	76
Table 6.1 Hasil Pengujian Perbedaan Jumlah <i>Hidden Neuron</i>	83
Table 6.2 Hasil Pengujian Perbedaan Nilai <i>Learning Rate</i>	86
Table 6.3 Hasil Pengujian Perbedaan Nilai <i>Epoch</i> Maksimum.....	89
Table 6.4 Table Hasil Pengujian Akurasi.....	90

DAFTAR GAMBAR

Gambar 2.1 Cacar Air	6
Gambar 2.2 Abses	7
Gambar 2.3 Campak.....	7
Gambar 2.4 Herpes	8
Gambar 2.5 Dermatitis.....	8
Gambar 2.6 Scabies.....	9
Gambar 2.7 Proses Komunikasi antar neuron	10
Gambar 2.8 Single Layer Network	10
Gambar 3.1 Diagram Blok Metodologi penelitian	17
Gambar 4.1 Diagram Alir <i>Backpropagation</i>	21
Gambar 4.2 Diagram Alir Proses <i>Nguyen-Widrow</i>	22
Gambar 4.3 Diagram Alir Sistem Fungsi <i>Feed Forward</i>	24
Gambar 4.4 Diagram Alir Sistem Fungsi <i>Feed Backward</i>	25
Gambar 4.5 Struktur JST untuk Diagnosis penyakit kulit pada anak	26
Gambar 6.1 Grafik Pengujian Perbedaan Jumlah <i>Hidden Neuron</i>	85
Gambar 6.2 Grafik Pengujian Perbedaan Nilai <i>Learning Rate</i>	88
Gambar 6.3 Grafik Pengujian Perbedaan Jumlah <i>Epoch</i> Maksimum	89
Gambar 6.4 Grafik Pengujian Akurasi	90

BAB 1 PENDAHULUAN

1.1 Latar belakang

Kulit ialah salah satu penunjang kehidupan manusia yang digunakan sebagai indra peraba. Pada manusia kulit bisa terjangkit bermacam-macam penyakit, mulai penyakit yang ringan yang menyebabkan gatal-gatal ataupun penyakit berat yang dapat mengakibatkan kematian. Sistem Imun pada anak yang berbeda dengan orang dewasa membuat Anak – anak lebih mudah untuk terserang beragam penyakit karena infeksi bakteri, virus, parasit, dan jamur (Annisa & Destiani & Dhami, 2012). Kurangnya pengetahuan terhadap gejala dari sebuah penyakit membuat para orang tua kebingungan. Para orang tua diharapkan untuk selalu waspada dan cepat tanggap dalam menangani apa bila anak terserang penyakit agar bisa di tangani sejak dini.

Ada berbagai macam penyakit kulit yang dapat di derita anak – anak. Penyakit campak merupakan salah satu penyakit kulit yang bisa menyerang anak – anak. di Indonesia angka kejadian penyakit campak ini dari tahun 1990 hingga 2002 masih cukup tinggi sekitar 3000-4000 per tahunnya. Pasien yang sedang mengidap penyakit campak terbanyak berumur <12 bulan, diikuti kelompok umur 1-4 dan 5-14 tahun (Antonius, 2009). Seiring perkembangan zaman gejala yang terdapat di suatu penyakit kulit memiliki gejala yang hampir mirip dengan gejala penyakit kulit lainnya padahal pengobatan dari setiap penyakit berbeda-beda. Sehingga untuk orang awam susah untuk membedakan terutama orang tua. Padahal hal ini sangat penting agar orang tua bisa memberikan pertolongan pertama kepada anak dengan segera. Masalah karena sulit membedakan penyakit yang diderita karena adanya kemiripan gejala dapat diselesaikan dengan metode yang memberikan rekomendasi dengan pembelajaran pada data yang telah ada. Di era yang serba teknologi ini banyak peneliti yang mengembangkan sebuah sistem yang dapat meringankan pekerjaan manusia. Dengan teknologi suatu pekerjaan yang harusnya dikerjakan dengan manual dan memerlukan waktu yang lebih lama. Akan di lakukan dengan lebih efektif dan efisien.

Jaringan syaraf tiruan ialah salah satu sistem pemrosesan suatu informasi yang dirancang dengan meniru cara kerja manusia untuk menyelesaikan suatu masalah dengan menjalankan proses pembelajaran dengan perubahan bobot sinapsisnya. Pada jaringan syaraf tiruan data masalalu akan di *learning* sehingga memiliki kemampuan untuk memberikan keputusan terhadap data yang belum di pelajari (Agustin & Prahasto, 2012). JST ditentukan oleh tiga hal, yaitu: pola hubungan antara *neuron* (arsitektur jaringan), metode untuk menentukan bobot penghubung (metode training/learning/algoritma) dan fungsi aktivasinya (Jong Jek Siang, 2015). *Backpropagation* adalah sebuah metode yang ada pada Jaringan Syaraf tiruan yang sering digunakan. membutuhkan banyak waktu untuk berkumpul di sebuah solusi yang dapat diterima. Backpropagation ialah

algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya (Sri Kusumadewi, 2004). Pelatihan akan dilakukan berulang-ulang sehingga dapat dihasilkan jaringan yang memberikan tanggapan yang benar terhadap semua masukannya. Maria Agustin (2012), telah melakukan penelitian menggunakan metode JST *backpropagation*. Pada penelitian ini peneliti fokus terhadap proses JST yang digunakan untuk menghasilkan output yang telah dipelajari dari data yang telah ada sebelumnya, dimana hasil output dari penelitian ini merupakan mahasiswa baru yang diterima pada jurusan Teknik Komputer di Politeknik Negeri Sriwijaya. Data calon mahasiswa yang digunakan yaitu data selama 3 tahun, pada 2008, 2009 sebagai data pelatihan dan 2010 untuk data pengujian. Pada penelitian kali ini peneliti menggunakan fungsi aktivasi *tansig*. Pengujian dilakukan dengan membedakan jumlah neuron dan *Hidden Layer*. Pada pengujian dengan jumlah neuron 35, 3 *Hidden Layer*, dan jumlah iterasi 5000 didapatkan nilai regresi paling tinggi yaitu 0.8563 yang memberikan hasil analisa bahwa metode JST *backpropagation* cukup efektif untuk di gunakan dalam seleksi penerimaan mahasiswa baru.

Dengan metode *backpropagation* diharapkan dapat mendiagnosis jenis penyakit kulit pada anak dengan cara melakukan pembelajaran terhadap data yang sudah ada, sehingga dapat mengetahui jenis penyakit yang sedang diderita dengan menggunakan data latih yang di dapat dari penderita sebelumnya. Berdasarkan uraian di atas, penulis tertarik melakukan penelitian “ Implementasi Jaringan Syaraf Tiruan *Backpropagation* Untuk Diagnosis Penyakit Kulit Pada Anak” sebagai solusi agar memudahkan mencari solusi dalam mendiagnosis penyakit yang di derita pada anak.

1.2 Rumusan masalah

Berdasarkan latar belakang di atas, dapat dirumuskan masalah yang ada yaitu:

1. Bagaimana mengimplementasikan *Algoritma Backpropagation* sehingga dapat digunakan untuk diagnosis penyakit kulit pada anak?
2. Berapa tingkat akurasi sistem setelah dilakukan pengolahan bobot pada sistem serta pengaruh data uji terhadap tingkat akurasi?
3. Bagaimana pengaruh variasi jumlah *Hidden Neuron* pada implementasi *backpropagation* dalam mendiagnosis penyakit kulit pada anak?
4. Bagaimana pengaruh perbedaan nilai *learning rate* yang digunakan pada implementasi *backpropagation* dalam mendiagnosis penyakit kulit pada anak?
5. Bagaimana pengaruh jumlah *epoch* yang digunakan pada implementasi *backpropagation* dalam mendiagnosis penyakit kulit pada anak?

1.3 Tujuan

Untuk menghindari adanya kemungkinan semakin berkembangnya masalah yang akan diteliti, maka penulisan laporan hanya menitik beratkan permasalahan pada beberapa hal :

1. Menganalisis *Algoritma Backpropagation* pada aplikasi pendiagnosis penyakit kulit pada anak.
2. Merancang dan mengimplementasikan *Algoritma backpropagation* agar dapat mendiagnosis penyakit kulit pada anak.
3. Menganalisis akurasi hasil sistem dengan data yang sudah didapat.
4. Menganalisis parameter yang optimal pada sistem diagnosis penyakit kulit pada anak

1.4 Manfaat

Berdasarkan latar belakang, rumusan masalah dan tujuan yang telah di jabarkan penelitian ini memiliki manfaat sebagai berikut:

1. Menerapkan *Algoritma Backpropagation* untuk mendiagnosis penyakit kulit pada anak.
2. Memudahkan para dokter dan petugas medis dalam mendeteksi gejala penyakit kulit pada anak
3. Dapat memberi tambahan pengetahuan mengenai penyakit kulit pada anak
4. Membantu masyarakat agar dapat mengenali jenis dan gejala penyakit kulit pada anak
5. Membantu para orang tua agar dapat melakukan penanganan dini dengan tepat dan cepat

1.5 Batasan masalah

Berdasarkan latar belakang dan rumusan masalah yang telah di jabarkan, maka penelitian ini memiliki batasan-batasan masalah:

1. Diagnosis pada sistem diperuntukkan pada anak yang berusia dibawah 12 tahun.
2. Data yang di gunakan berasal dari poliklinik Sang Timur Sumenep.
3. Data yang didapat dari penelitian sebelumnya yang dilakukan oleh Ika muwaddamah
4. Terdapat 19 macam gejala yang ada pada penelitian ini, yaitu muncul gelembung air merata, demam, nyeri kepala, gatal-gatal, lesi seluruh tubuh, mata merah, batuk, pilek, meradang, nyeri tekan, gelembung nanah, perih, bersisik, melepuh, panas, bengkak, lesi diarea tertentu,benjolan nanah, dan gelembung air di area tertentu.
5. Penyakit yang akan di menjadi hasil diagnosis dalam penelitian ini ada 6 macam yaitu cacar air, scabies, campak, dermatitis, herpes dan abses.
6. Fungsi aktivasi yang digunakan dalam penelitian ini ialah fungsi aktivasi *sigmoid biner*.

7. Lingkungan pengujian menggunakan bahasa pemrograman JAVA.
8. Jumlah hidden layer yang digunakan ialah 1 hidden layer.
9. Masukan yang digunakan harus terdapat minimal 3 gejala yang diderita pasien.

1.6 Sistematika pembahasan

Sistematika penulisan penelitian ditunjukkan guna memberi gambaran dan uraian dari pembuatan dan penyusunan tugas akhir secara garis besar yang meliputi beberapa bab, sebagai berikut.

BAB 1 : Pendahuluan

Menguraikan latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB 2 : Landasan Kepustakaan

Menguraikan teori – teori yang digunakan dan menjadi referensi dalam pelaksanaan penelitian skripsi.

BAB 3 : Metodologi Penelitian

Metodologi ini menjelaskan bagaimana strategi melakukan penelitian dan membahas tentang analisis masalah, analisis sumber pengetahuan, analisis penyakit dan gejala, analisa table keputusan.

BAB 4 : Perancangan

Pada bab ini akan dibahas tentang langkah – langkah implementasi, analisis, dan pengujian sistem.

BAB 5 : Implementasi

pada bab implementasi akan dijelaskan penerapan metode *backpropagation* untuk diagnosis penyakit kulit pada anak yang dibuat.

BAB 6 : Hasil dan Pengujian

Memuat proses hasil pengujian terhadap sistem yang sudah di implementasikan

BAB 7 : Penutup

Berisi kesimpulan serta saran dari penelliti untuk penelitian selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Miss Ankeeta R. Patel dan Maulin M. Joshi sudah melakukan penelitian menggunakan *Neural Network* dengan topik penyakit jantung pada tahun 2013. Pada penelitian ini peneliti akan mengklasifikasikan outputan menjadi 4 yaitu normal, stroke 1, stroke 2, dan stroke 3 yang berarti akan mengakhiri kehidupan pasien. Inputan data didasarkan pada tes pasien yang dilakukan oleh para ahli medis. Terdapat 13 atribut yang digunakan dalam inputan. Data yang digunakan pada penelitian ini berasal dari database standar Cheveland dimana pada data ini merupakan data pasien nyata dan telah digunakan secara luas untuk pengujian di berbagai mesin teknik pembelajaran. Peneliti menggunakan fungsi aktivasi tansig. Pengujian dilakukan dengan dataset yang berbeda. Dimana hasil dari pengujian ini pelatihan yang dilakukan dengan data Cheveland MSE untuk data chevelad ini 0.0013 tingkat keberhasilan mencapai 100% tetapi ketika diuji dengan data lain dengan mengubah parameter non critical MSE 3.64e-20 tingkat keberhasilannya berkurang menjadi 95.833% berdasarkan hasil penelitian menunjukan *Neural Network* cukup bagus digunakan dalam mendiagnosis penyakit jantung.

Penelitian terkait selanjutnya yang telah dilakukan Ika Srimuddawamah pada tahun 2015 tentang sistem pakar dengan topik penyakit kulit pada anak menggunakan metode *naive bayes*. pada penelitian ini bertujuan untuk membantu dalam mendiagnosis penyakit kulit pada anak. Dimana peneliti menggunakan masukan gejala penyakit kulit yang menyerang pada anak dan menghitung nilai posterior (probabilitas terakhir) masing-masing penyakit dan penyakit yang memiliki posterior paling tinggi akan dijadikan hasil diagnosis sistem. Pada penelitian ini menghasilkan rata-rata akurasi sistem sebesar 85,4%.

Penelitian terkait selanjutnya dilakukan Sri Yastita, Yohana Dewi Lulu, dan Rika Perdana Sari sudah melakukan penelitian tentang sistem sistem pakar dengan topik penyakit kulit pada anak pada tahun 2012. Dalam penelitian ini peneliti menggunakan metode *Certainty Factor*. Penelitian ini bertujuan untuk membantu dalam mendiagnosis penyakit kulit pada manusia. Kriteria yang di gunakan dalam penelitian ini ialah gejala – gejala dari setiap penyakit yang sudah di beri bobot oleh seorang ahli pakar yang kemudian akan di hitung dengan metode *Certainty Factor* sehingga mendapatkan nilai CF yang paling tinggi dari masing – masing penyakit dan menjadikan penyakit tersebut sebagai outputan dari sistem. Penelitian yang sudah di lakukan ini memiliki akurasi sebesar 73.15%.

Selanjutnya Maria Agustin telah melakukan penelitian menggunakan metode JST pada tahun 2012. Pada penelitian ini peneliti fokus terhadap proses JST yang digunakan untuk menghasilkan output yang telah dipelajari dari data yang telah ada sebelumnya, dimana hasil output dari penelitian ini merupakan mahasiswa baru yang diterima pada jurusan Teknik Komputer di Politeknik Negeri Sriwijaya. Data calon mahasiswa yang digunakan yaitu data selama 3 tahun, pada 2008, 2009 sebagai data pelatihan dan 2010 untuk data pengujian. Pada penelitian kali ini peneliti menggunakan fungsi aktivasi *tansig*. Pengujian dilakukan dengan

membedakan jumlah neuron dan *Hidden Layer*. Pada pengujian dengan jumlah neuron 35, 3 *Hidden Layer*, dan jumlah iterasi 5000 didapatkan nilai regresi paling tinggi yaitu 0.8563 yang memberikan hasil analisa bahwa metode JST *backpropagation* cukup efektif untuk di gunakan dalam seleksi penerimaan mahasiswa baru.

Berdasarkan penelitian diatas, diketahui bahwa *Algoritma Backpropagation* yang ada pada Jaringan Syaraf Tiruan dapat digunakan untuk penentuan suatu data terhadap kelas tertentu dengan memberikan hasil learning yang baik pada objek peneletian. Oleh karena itu diagnosis penyakit kulit pada anak ini menggunakan *Algoritma Backpropagation*.

2.1 Dasar Teori

Pada sub-bab ini akan menjelaskan tentang penyakit kulit dan metode *backpropagation*.

2.1.1 Penyakit Kulit Anak

Penyakit kulit adalah penyakit yang bisa menyerang siapapun. Laki – laki, perempuan, anak – anak, bayi bahkan orang dewasa. Kulit anak – anak sangat berbeda dengan kulit yang dimiliki orang dewasa sehingga anak – anak merupakan kelompok usia yang paling rentan terhadap penyakit kulit. Hal – hal yang menyebabkan penyakit kulit adalah alergi kulit, perubahan cuaca, alergi kulit, virus dan jamur. Tanda – tanda awal dari penyakit kulit pada umumnya ialah ruam merah, gatal – gatal atau merasa sakit pada kulit yang terinfeksi penyakit kulit tersebut.

Berikut adalah penyakit kulit yang sering menyerang pada anak – anak disertai gejala, ciri-ciri, penyebab & cara mengatasinya.

2.2.1.1 Cacar Air



Gambar 2.1 Cacar Air

Sumber: Jamaludin, 2014

Penyakit cacar air merupakan penyakit menular akut. Penularan dapat terjadi melalui kontak langsung dengan lesi, tetapi terutama udara (Siti Aisah Boediardja, 2005). Penyakit cacar ini diakibatkan oleh virus yang bernama *varicella zoster*. Penyakit ini menyerang 90% pada anak-anak usia kurang dari 10 tahun.

Gejala-gejala yang muncul pada cacar air ialah kulit meradang, muncul gelembung berisi air yang tersebar di seluruh tubuh dan gatal.

2.2.1.2 Abses



Gambar 2.2 Abses

Sumber: Jamaludin, 2014

Abses ialah radang folikel yang diawali oleh folikulitis superfisial yang meluas. Jika terdapat lebih dari satu disebut juga furunkolosis. Sering didapatkan pada anak-anak yang lebih dewasa dan umumnya berkembang dari folikulitis profunda (Siti Aisah Boediardja, 2005). Gejala pada penyakit ini ialah munculnya benjolan berisi nanah, gatal dipermukaan kulit, demam dan nyeri kepala.

2.2.1.3 Campak



Gambar 2.3 Campak

Sumber: Jamaludin, 2014

Penyakit campak merupakan penyakit menular dimana gejala yang ditimbulkan oleh penyakit ini ialah adanya ruam di seluruh tubuh, dan bisa disertai mata merah dan batuk pilek. Penularan penyakit ini dengan cara kontak langsung dengan orang yang sedang terinfeksi. Penyebab dari penyakit ini ialah virus *Ribonucleid Acid* (Mursinah, 2010)

2.2.1.4 Herpes



Gambar 2.4 Herpes

Sumber: Jamaludin, 2014

Herpes ialah penyakit yang disebabkan oleh virus anggota family hepertoviridae. Sehingga apabila seseorang menderita penyakit ini akan merdapat ruam bintik – bintik nanah yang berkelompok. Virus ini menyerang pada bagian kulit dan selaput lendir. Penyakit ini akan timbul jika penderita pernah terkena varisela. Penyakit ini menyerang 5-10% pada anak-anak. Berbeda dengan varisela, penyakit ini tidak memberikan kekebalan terhadap virus yang sama. Gejala yang ditimbulkan adalah gatal, muncul gelembung air, tidak menyebar diseluruh tubuh.

2.2.1.5 Dermatitis



Gambar 2.5 Dermatitis

Sumber: Jamaludin, 2014

Dermatitis atau bisa disebut juga eksema adalah penyakit kulit yang menyebabkan munculnya rasa gatal-gatal. Kondisi untuk penyakit ini adalah kulit yang memerah, kering, dan pecah – pecah. Dermatitis merupakan peradangan pada lapisan kulit bagian atas yang menyebabkan kerusakan kulit. Penyakit ini sering dijumpai pada anak-anak dan orang dewasa. Kurang lebih 80% bayi pernah mengalami penyakit ini pada bulan pertama kehidupan.

2.2.1.6 Scabies



Gambar 2.6 Scabies

Sumber : Jamaludin, 2014

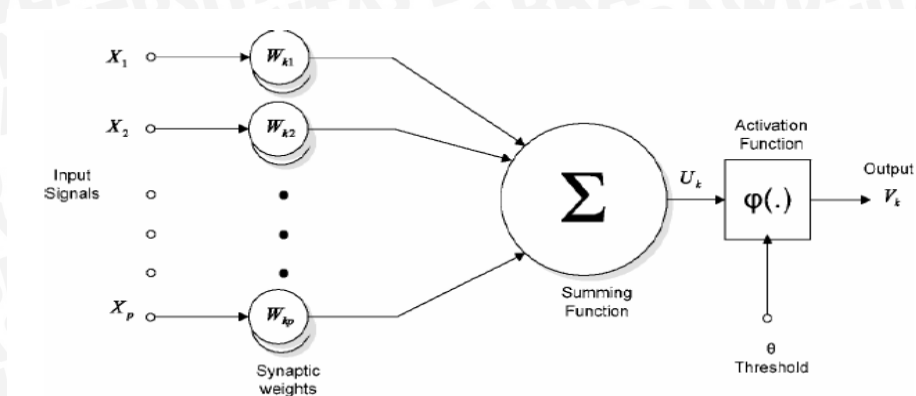
Scabies ialah penyakit gatal pada kulit yang menular diakibatkan oleh binatang yang disebut tungau atau kutu yang bernama *sarcoptes scabiei*, ditandai dengan keluhan gatal, terutama di malam hari dan gampang menular melalui kontak secara langsung maupun tidak langsung. Penyakit ini banyak dijumpai pada anak-anak dan orang dewasa.

2.2.2 Jaringan Syaraf Tiruan (JST)

Jaringan Syaraf Tiruan (JST) merupakan salah satu metode yang berdasarkan pada cara kerja jaringan syaraf pada manusia. Metode ini ialah sebuah sistem pemrosesan informasi yang memiliki karakteristik seperti jaringan syaraf manusia (Maharani & Irawan, 2012).). Jaringan syaraf tiruan merupakan sistem pemrosesan informasi yang mempunyai karakteristik serupa dengan jaringan syaraf biologi, dimana Jaringan Syaraf Tiruan (JST) menyerupai otak manusia dalam mendapatkan pengetahuan dengan proses *learning* (belajar) dan menyimpan pengetahuan yang didapat didalam kekuatan koneksi antar *neuron* (Islam, 2009). Jaringan Syaraf Tiruan melakukan pembelajaran dari pola – pola pengalaman yang sudah ada sebelumnya, sehingga setiap sinyal masukan akan dilakukan pembelajaran untuk mendapatkan keluaran atau kesimpulan yang sesuai.

2.2.2.6 Konsep Dasar

Pada jaringan Syaraf tiruan Terdapat *neuron* seperti halnya pada jaringan syaraf biologis manusia. Dimana setiap *neuron* terhubung dan mengalirkan sinyal, sehingga sel syaraf akan menggunakan fungsi aktivasi guna menentukan sinyal output yang seharusnya. Gambar 2.2 adalah gambar dari proses komunikasi antar *neuron*.



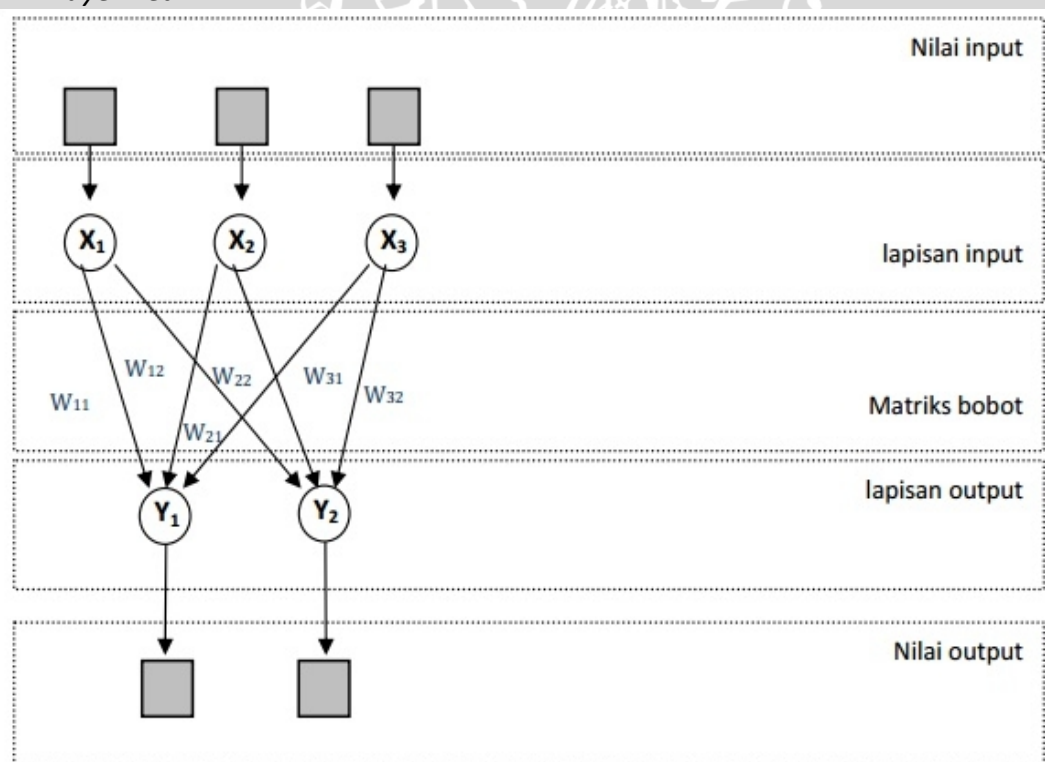
Gambar 2.7 Proses Komunikasi antar neuron

Sumber: (Komang, 2012)

Ada tiga macam arsitektur dalam Jaringan Syaraf Tiruan (JST) yaitu (Maharani & Irawan, 2012):

1. *Single Layer Network* (Jaringan dengan lapisan Tunggal).

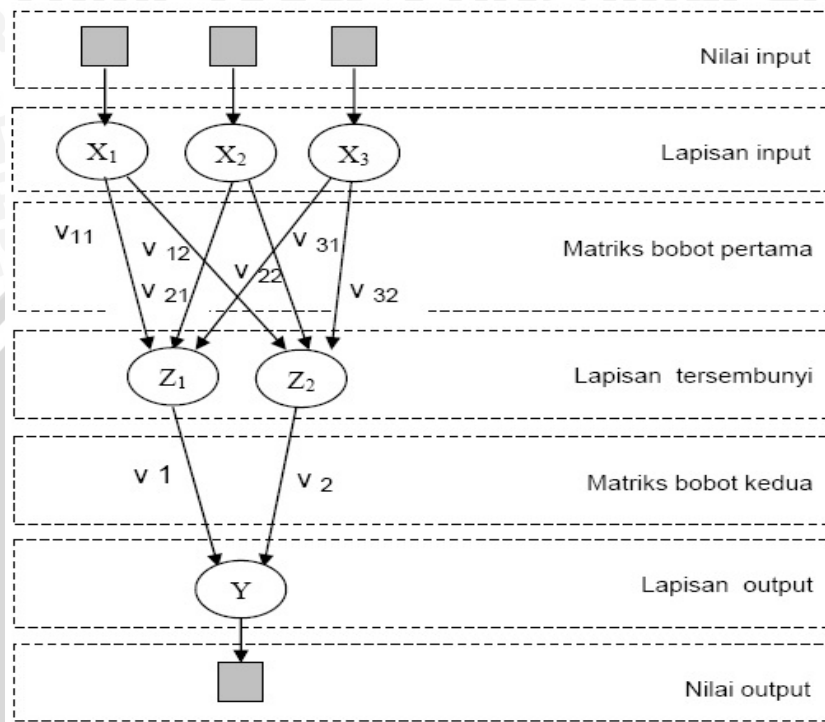
Jaringan ini menerima masukan dan mengolahnya, kemudian menghasilkan keluaran yang diinginkan tanpa melewati lapisan – lapisan tersembunyi. Karena itulah proses ini disebut *single layer network* yang hanya mempunyai satu layer atau lapisan. Gambar 2.3 berikut ini merupakan gambaran *single layer net*.



Gambar 2.8 Single Layer Network

Sumber: (Maharani & Irawan, 2012)

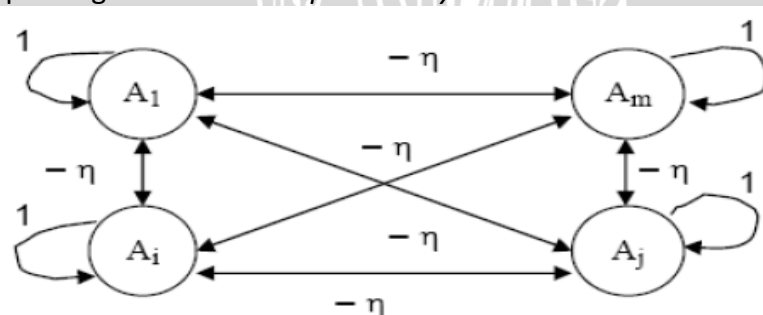
2. **Multi Layer Network** (Jaringan dengan banyak lapisan).
 Pada jaringan ini mempunyai beberapa *layer* diantara *layer* masukan dan keluaran. Penggunaan jenis jaringan ini ialah untuk permasalahan-permasalahan yang lebih sulit dibandingkan dengan menggunakan *Single Layer Network*, karena pembelajarannya yang jauh lebih rumit. Gambar 2.4 berikut merupakan gambaran dari *multi layer network*.



Gambar 2.1 Multi Layer Network

Sumber: (Maharani & Irawan, 2012)

3. **Competitive Layer Network** (Jaringan dengan lapisan kompetitif).
Competitive layer network ini adalah dimana setiap *layer neuron* bersaing untuk mendapatkan banyak hak menjadi *neuron* yang aktif. Gambar 2.5 merupakan gambar dari *competitive layer network*.



Gambar 2.2 Kompetitif Layer Network

Sumber: (Maharani & Irawan, 2012)

2.2.2.7 Backpropagation

Backpropagation merupakan salah satu model yang terdapat pada JST yang menggunakan *supervised learning*. Algoritma ini sering digunakan untuk menyelesaikan suatu masalah yang rumit. Hal ini dikarenakan algoritma ini dilatih menggunakan metode pembelajaran. Pada jaringan ini diberikan pola yang sepasang yang terdiri atas pola masukan dan pola yang dikehendaki (Arizona, 2013). *Backpropagation* memiliki proses pembelajaran maju dan perbaikan kesalahan secara mundur. Model jaringan ini sering digunakan untuk proses prediksi, pengenalan dan peramalan (Dewi & Muslikh, 2013).

Pada *backpropagation*, setiap masukan yang di inputkan ke dalam jaringan akan di proses, kemudian selisih antara keluaran dan target merupakan kesalahan yang terjadi. Target berguna untuk melatih jaringan agar menghasilkan bobot yang dikehendaki, karena terjadi kesalahan, sehingga jaringan akan memodifikasi bobot sesuai kesalahan tersebut.

Metode ini mempunyai tiga fase dalam melakukan pelatihan *backpropagation*, yaitu *feed forward*, *backpropagation*, dan fase untuk memodifikasi bobot. Ketiga fase ini akan selalu dijalankan sampai kondisi penghentian terpenuhi.

2.2.2.8 Fungsi Aktivasi

Fungsi aktivasi merupakan penentuan keluaran suatu neuron yang mendefinisikan nilai output dari sebuah neuron. Ada beberapa jenis fungsi aktivasi diantaranya (Ratna, 2013):

1. Hard Limit (Tangga Biner)

Fungsi aktivasi ini digunakan pada *Single Layer Network*, output dari fungsi ini berupa bilangan biner (0 atau 1).

Rumus fungsi aktivasi ini adalah:

$$y = \begin{cases} 0 & \text{jika } x \leq 0 \\ 1 & \text{jika } x > 0 \end{cases} \dots\dots\dots (2-1)$$

2. Symetric Hard Limit (Bipolar)

fungsi *Symetric hard limit* ini merupakan pengembangan dari fungsi *hard limit*, namun memiliki keluaran antara 1, 0 atau -1 .

Rumus fungsi aktivasi ini adalah:

$$y = \begin{cases} 1, & \text{jika } x > 0 \\ 0, & \text{jika } x = 0 \\ -1, & \text{jika } x < 0 \end{cases} \dots\dots\dots (2-2)$$

3. Threshold

Fungsi ini juga merupakan hasil pengembangan dari fungsi aktivasi *Hard Limit*, namun dengan menambahkan nilai nilai *threshold* (θ).

Rumus fungsi aktivasi ini adalah:

$$y = \begin{cases} 0 & \text{jika } x < \theta \\ 1 & \text{jika } x \geq \theta \end{cases} \dots\dots\dots (2-3)$$

4. Bipolar Threshold

Fungsi ini merupakan pengembangan dari fungsi aktivasi *threshold*, fungsi ini memiliki keluaran berupa nilai 1 atau -1.

Rumus fungsi aktivasi ini adalah:

$$y = \begin{cases} 1 & \text{jika } x \geq \theta \\ -1 & \text{jika } x < \theta \end{cases} \dots\dots\dots (2-4)$$

5. *Linier* (Identitas)

Fungsi aktivasi ini memiliki keluaran atau output yang sama dengan nilai masukannya.

Rumus Fungsi aktivasi ini adalah:

$$y = x \dots\dots\dots (2-5)$$

6. *Sigmoid* Biner

Fungsi ini memiliki *range* nilai antara 0 sampai 1.

Rumus fungsi aktivasi ini adalah:

$$y = f(x) = \frac{1}{1 + e^{-x}} \dots\dots\dots (2-6)$$

7. *Sigmoid* Bipolar (*Symetric Sigmoid*)

Sigmoid bipolar atau fungsi aktivasi *tansig* ini berbeda dengan fungsi akticasi *sigmoid* biner, fungsi ini memiliki nilai range antara -1 sampai 1.

Rumus fungsi aktivasi ini adalah:

$$y = f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \dots\dots\dots (2-7)$$

2.2.2.9 Learning Rate

Learning rate atau bisa juga disebut laju pembelajaran ini di simbolkan dengan α . Koefisien *learning rate* ini juga berpengaruh terhadap proses pembelajaran sistem juga dengan efektivitas dan kecepayan dalam mencapai konvergensi dari pembelajaran tersebut. Nilai optimum *learning rate* tergantung dari masalah yang akan diselesaikan, prinsipnya pilih nilai *learning rate* yang nantinya akan mencapai konvergensi optimal pada proses *learning*. Pada umumnya nilai untuk *learning rate* antara 0 sampai 1 (Inggit, 2013).

2.2.2.10 Langkah – langkah *Backpropagation*

Ada tiga fase dalam langkah – langkah algoritma *backpropagation* menurut (Sri Kusumadewi, 2004):

Langkah 0

Hal pertama yang harus dilakukan adalah dengan melakukan inisialisasi bobot, dimana mengambil bobot awal dengan nilai random yang cukup.

- Menetapkan nilai *epoch* maksimum, *target error*, dan *learning rate*(α).
- Inisialisasi : Epoch = 0, MSE = 1.

Langkah 1

Melakukan langkah-langkah selanjutnya dengan kondisi selama ($epoch < epoch$ maksimum) dan ($MSE > target\ error$), dimana nilai epoch akan terus bertambah setiap iterasi ($epoch = epoch + 1$).

Langkah 2

Untuk tiap – tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan langkah – langkah berikutnya.

FASE I : *Feed Forward*

Selama fase *feed forward*, sinyal masukan (X_i) dipropagasikan di layer tersembunyi menggunakan fungsi aktivasi yang telah ditentukan. Outputan (Z_j) tersebut kemudian dipropagasikan maju lagi ke layer atasnya. Demikian terus sehingga menghasilkan keluaran jaringan (Y_k). Keluaran jaringan selanjutnya di bandingkan dengan target awal yang harus dicapai, selisih dari keluaran dan target adalah kesalahan yang terjadi, jika kesalahan tersebut lebih kecil dari kesalahan maksimum yang telah di tentukan, maka iterasi di hentikan, jika tidak iterasi di teruskan dengan memodifikasi bobot dalam jaringan.

Langkah 3

Tiap – tiap unit input ($X_i, i=1,2,...,n$) mendapatkan sinyal X_i dan meneruskan ke semua unit pada lapisan tersembunyi.

Langkah 4

Tiap – tiap unit pada lapisan tersembunyi ($Z_j, j=1,2,...,m$) menjumlahkan sinyal – sinyal input terbobot:

$$Z_{inj} = v_{jo} + \sum_{i=1}^n x_i v_{ji} \dots\dots\dots (2-8)$$

Dengan:

Z_{inj} = total sinyal masukan pada *hidden* unit j

X_i = nilai masukan pada unit i

V_{ji} = bobot antara *input* unit dan *hidden* unit j

Lalu gunakan fungsi aktivasi untuk menghitung sinyal output:

$$Z_j = f(z_{netj}) = \frac{1}{1+e^{-z_{netj}}} \dots\dots\dots (2-9)$$

Dengan :

Z_j = keluaran pada *hidden* unit j

Z_{netj} = total sinyal masukan pada *hidden* unit j

Kemudian sinyal tersebut dikirimkan ke semua unit lapisan atasnya.

Langkah 5

Tiap – tiap unit output $Y_k (k=1,2,...,p)$ menjumlahkan sinyal –sinyal input terbobot:

$$Y_{ink} = W_{ko} + \sum_{j=1}^p Z_j W_{kj} \dots\dots\dots (2-10)$$

Dengan :

Y_{ink} = total sinyal masukan pada *output* unit k

Z_j = nilai masukan pada *hidden* unit j

W_{kj} = bobot antara *hidden* unit j dan *output* unit k

Gunakan fungsi aktifasi untuk menghitung sinyal output :

$$y_{out} = f(y_{ink}) = \frac{1}{\alpha + e^{-y_{ink}}} \dots\dots\dots (2-11)$$

Kemudian dikirimkan sinyal tersebut kesemua unit layer atasnya.

FASE II : *Feed Backward*

Langkah 6

- Tiap –tiap unit output Y_{out} ($k=1,2,...,p$) menerima target pola yang berhubungan dengan pola inputan pelatihan, hitung informasi erornya:

$$\delta_k = (t_k - y_{out})y_{out}(1 - y_{out}) \dots\dots\dots (2-12)$$

Dengan :

δ_k = faktor kesalahan pada output unit k

Y_{out} = keluaran pada output unit k

- Kemudian hitung koreksi bobot (yang akan digunakan untuk memperbaiki bobot baru)

$$\Delta W_{kj} = \alpha \delta_k z_j \dots\dots\dots (2-13)$$

Langkah 7

Tiap – tiap unit tersembunyi Z_j ($j=1,2,...,m$)

- Menjumlahkan delta inputnya

$$\delta_{in_j} = \sum_{k=1}^m \delta_k W_{kj} \dots\dots\dots (2-14)$$

- Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi error:

$$\delta_j = \delta_{in_j} z_j Z(1 - z_j) \dots\dots\dots (2-15)$$

- Hitung koreksi bobot yang akan digunakan untuk memperbaiki bobot baru :

$$\Delta v_{ji} = \alpha \delta_j x_i \dots\dots\dots (2-16)$$

Langkah 8

- Tiap – tiap unit *output* Y_k ($k=1,2,...,p$) memperbaiki bias dan bobotnya ($j=0,1,2,...,m$):

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \dots\dots\dots (2-17)$$

- Tiap – tiap unit tersembunyi Z_j ($j=1,2,...,p$) memperbaiki bias dan bobotnya ($i=1,2,...,n$)

$$V_{ij}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij} \dots\dots\dots (2-18)$$

Langkah 9

Hitung MSE pada tiap epoch

$$MSE = \frac{1}{n_{Pola}} \sum_k^{n_{Pola}} (t_k - y_{out})^2 \dots\dots\dots (2-19)$$

Dimana :

t_k = target pada output unit k

Y_{out} = keluaran pada output unit k

n = jumlah pola

2.2.2.11 Algoritma *Nguyen – Widrow*

Nguyen – Widrow merupakan algoritma yang digunakan untuk melakukan inisialisasi bobot awal pada jst untuk mengurangi waktu pelatihan. Algoritma *Nguyen – Widrow* adalah sebagai berikut (Adrian. 2014):

1. Inisialisasi semua bobot dengan *range* antara (-0,5) – 0,5.

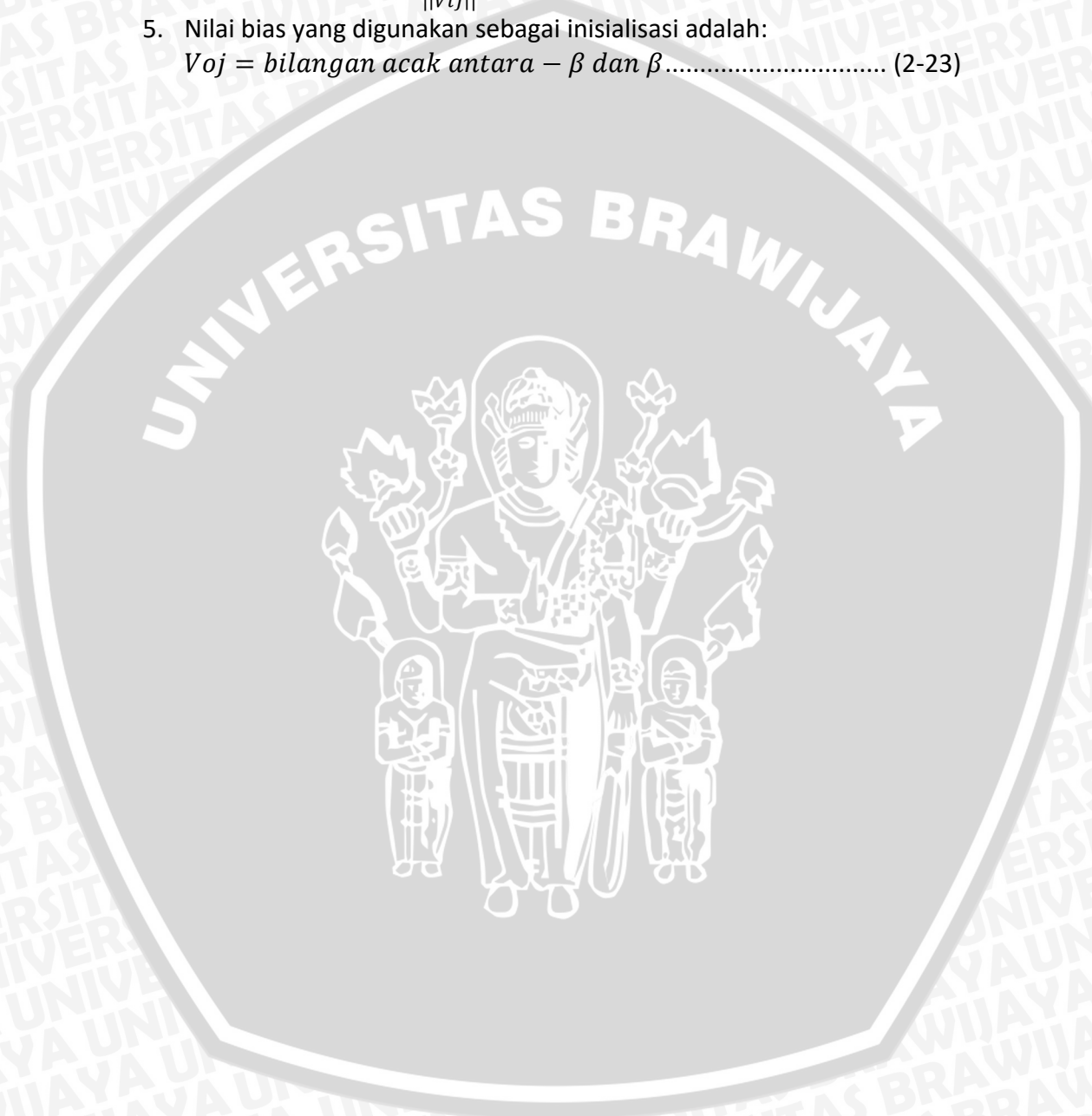
$$2. \text{ Hitung nilai } ||V_{ji}|| = \sqrt{V_1^2 j + V_2^2 j + \dots + V_n^2 j} \dots\dots\dots (2-20)$$

$$3. \text{ Hitung } \beta = \text{faktor skala} = 0,7 (p)^{1/n} = 0,7 \sqrt[n]{p} \dots\dots\dots (2-21)$$

n= jumlah unit masukan
p= jumlah unit tersembunyi

$$4. \text{ Hitung nilai } V_{ij} = \frac{\beta V_{ij}(\text{lama})}{||V_{ij}||} \dots\dots\dots (2-22)$$

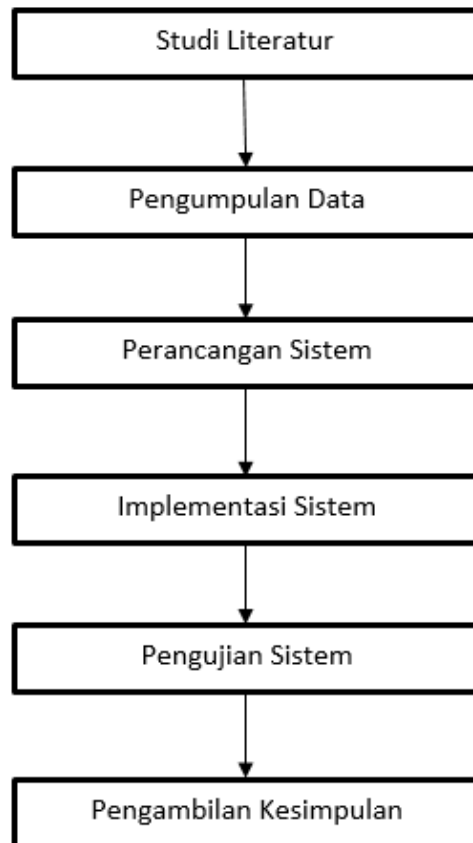
$$5. \text{ Nilai bias yang digunakan sebagai inisialisasi adalah:} \\ V_{oj} = \text{bilangan acak antara } -\beta \text{ dan } \beta \dots\dots\dots (2-23)$$



BAB 3 METODOLOGI

Bab tiga ini berisi tentang metode penelitian yang akan dilakukan pada perancangan dan implementasi metode *Backpropagation* pada diagnosis penyakit kulit pada anak.

Metode penelitian ini meliputi studi literature, pengumpulan data analisis kebutuhan, perancangan sistem, implementasi sistem, pengambilan kesimpulan.



Gambar 3.1 Diagram Blok Metodologi penelitian

3.1 Studi Literatur

Studi literatur penelitian ini bertujuan untuk mempelajari konsep yang akan digunakan dalam penelitian ini. Sumber ilmu pengetahuan dari penelitian ini terdiri dari data penyakit beserta defenisi, cara pengobatan, pencegahan, penyebab dan gejala yang diperoleh dari berbagai sumber informasi seperti Buku, penelitian sebelumnya, jurnal, *e-book* maupun informasi yang ada di internet. Beberapa hal yang dilakukan untuk mendukung penelitian ini yaitu:

1. Penyakit Kulit pada Anak sebagai kasus dan kriteria yang digunakan dalam diagnosis penyakit.
2. Pemahaman tentang cara kerja *Algoritma Backpropagation*.
3. Pemahaman tentang JST meliputi konsep dasar dan fungsi aktivasi.

4. Pemahaman tentang implementasi *backpropagation* menggunakan bahasa Java.

3.2 Pengumpulan data

Data yang digunakan dalam penelitian ini adalah data tentang pasien yang mengalami penyakit kulit yang bersumber dari poliklinik Sang Timur Sumenep yang didapat dari penelitian yang dilakukan Ika Srimuddawamah (Srimuddawamah,2015). Data tersebut meliputi gejala dan penyakit yang dialami. Data yang di gunakan berupa jumlah anak-anak sakit, penyakit kulit yang diderita anak-anak dan gejala-gejala yang diderita pada anak-anak. Sedangkan output dari sistem ialah keputusan untuk menentukan penyakit apa yang diderita anak tersebut. Table data pasien yang diperlukan seperti table 3.1.

NO	Nama Pasien	Usia	Gejala	Penyakit
1				
2				
...				
n				

Table 3.1 Data Pasien

3.3 Perancangan Sistem

Perancangan akan membahas tentang bagaimana langkah-langkah yang akan dilakukan untuk melakukan implementasi metode *backpropagation*. Langkah – langkah yang dilakukan dalam perancangan meliputi analisa kebutuhan sistem, arsitektur sistem dan perancangan basis data.

3.3.1 Analisa Lingkungan Pengujian

analisis kebutuhan dilakukan untuk memberikan definisi tentang hal – hal yang diperlukan dalam perancangan dan implementasi sistem yang dibutuhkan.

3.3.1.1 Lingkungan Perangkat Lunak

Lingkungan perangkat lunak yang diperlukan dalam perancangan dan implementasi sistem adalah:

1. Sistem Operasi Windows 10
2. Netbeans 8.0.2
3. Microsoft Excel 2013.

3.3.1.2 Lingkungan Perangkat Keras

Kebutuhan perangkat keras yang diperlukan dalam perancangan dan implementasi adalah:

1. Laptop *processor* AMD Radeon 1.95 GHz.

2. Memori RAM 4 GB DDR3
3. Hardisk 500GB.

3.4 Implementasi Sistem

Pada bagian implementasi sistem akan dibahas bagaimana fase-fase membangun suatu sistem yang mengacu pada perancangan sistem dan menerapkan hal – hal yang didapat dalam proses studi literatur . Fase-fase yang ada dalam implementasi antara lain:

1. Implementasi interface menggunakan software Netbeans 8.0.2.
2. Implementasi database menggunakan file berbentuk text dokumen yang bertujuan untuk memudahkan dalam penyimpanan data dan meringankan dalam proses komputasi.
3. Implementasi komputasi *learning* pada metode *Backpropagation* kedalam bahasa pemrograman JAVA.
4. Implementasi dilakukan agar dapat melakukan diagnosis penyakit kulit pada anak-anak.

3.5 Pengujian dan Analisis

Bagian ini menjelaskan tentang pengujian data latih terhadap data uji. Peneliti menggunakan data yang akan melalui proses pengujian sehingga memperoleh nilai akurasi yang berbeda pada setiap skenarionya. Berikut beberapa pengujian yang akan dilakukan:

1. Pengujian perbedaan jumlah *Hidden Neuron*.
2. Pengujian perbedaan nilai learning rate
3. Pengujian perbedaan nilai maksimum epoch/ iterasi.
4. Pengujian akurasi dengan membandingkan jumlah data latih.

3.6 Kesimpulan dan Saran

Pengambilan kesimpulan akan dilakukan ketika semua tahapan perancangan, implementasi dan pengujian sistem sudah selesai dilakukan. Kesimpulan diambil guna menjawab rumusan masalah yang telah ditetapkan sebelumnya. Tahap terakhir dari penelitian ini adalah berupa saran berkenaan dengan hasil yang telah dicapai untuk memperbaiki kesalahan-kesalahan yang terjadi. Serta untuk memberikan pertimbangan pada pengembangan sistem selanjutnya.

BAB 4 PERANCANGAN

Pada bab perancangan ini akan dibahas siklus penyelesaian masalah menggunakan metode *backpropagation* dan perancangan uji coba yang akan digunakan dalam pembuatan aplikasi implementasi jaringan syaraf tiruan *backpropagation* dalam mendiagnosis penyakit kulit pada anak.

4.1 Perhitungan *Backpropagation*

Pada *Implementasi Jaringan Syaraf Tiruan Backpropagation Untuk Diagnosis Penyakit Kulit Pada Anak* ini, terdapat beberapa variable yang bisa di ubah – ubah, yaitu *error* maksimum, dan *learning rate* (α). Arsitekur Jaringan Syaraf Tiruan (JST) yang di gunakan dalam penelitian ini adalah jaringan *multi layer*. Terdiri dari tiga jenis *layer* yaitu *input layer*, *Hidden Layer*, dan *output layer*.

4.1.1 *Feed Forward*

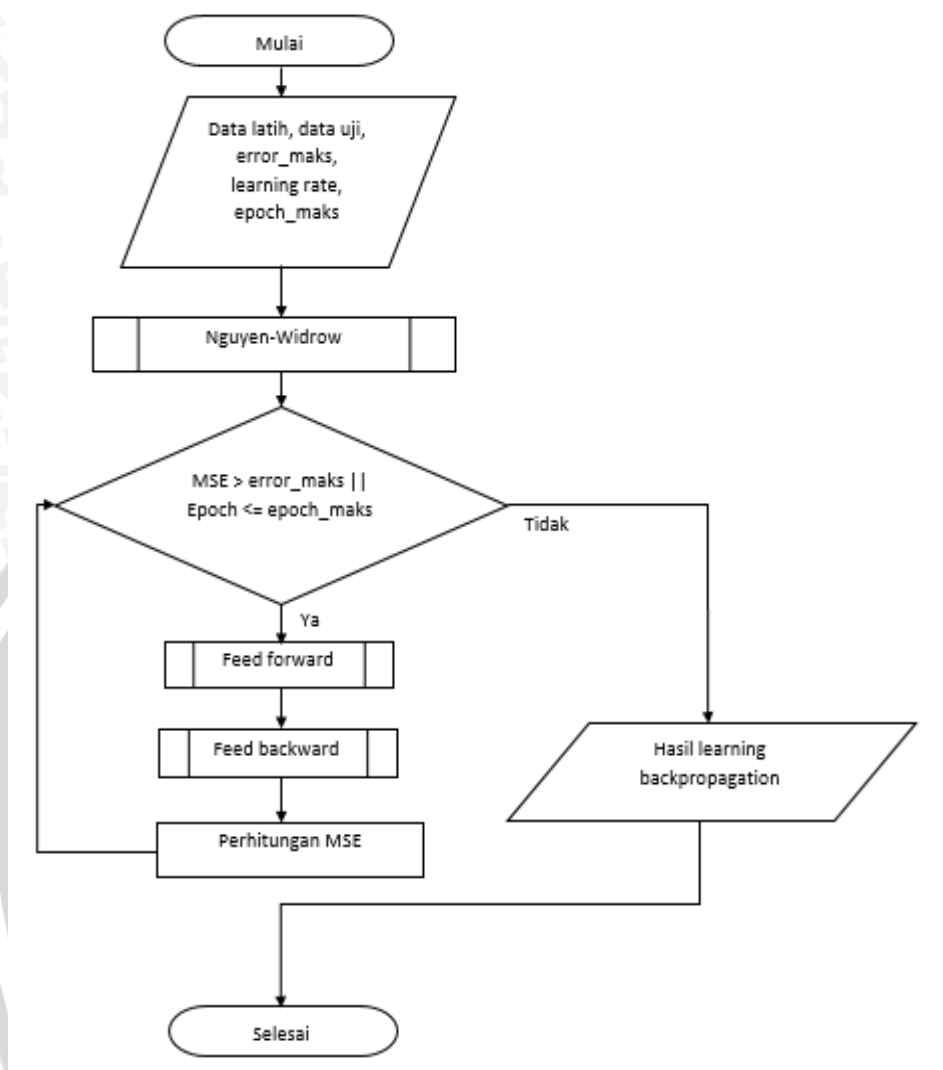
Pada fase ini inputan akan dilakukan perhitungan dengan bobot yang ada yang kemudian akan di teruskan di *Hidden Layer*. Lalu dihitung dengan fungsi aktivasi *sigmoid* yang akan di gunakan nilai sel – sel yang akan masuk ke *output layer*. Setelah mengetahui nilai sel – sel yang masuk, maka akan di hitung nilai *output* pada *output layer* dengan menggunakan fungsi aktivasi yang sama yaitu dengan fungsi aktivasi *sigmoid*.

4.1.2 *Feed Backward*

Pada fase *feed backward* akan dilakukan perhitungan nilai *error* secara mundur dari *output layer* menuju *Hidden Layer* dan dari *Hidden Layer* menuju *input layer*. Setelah *error* selesai di hitung maka koreksi bobot dapat dihitung untuk mendapatkan bobot yang baru yang akan di gunakan di iterasi selanjutnya.

4.1.3 Diagram Alir *Backpropagation*

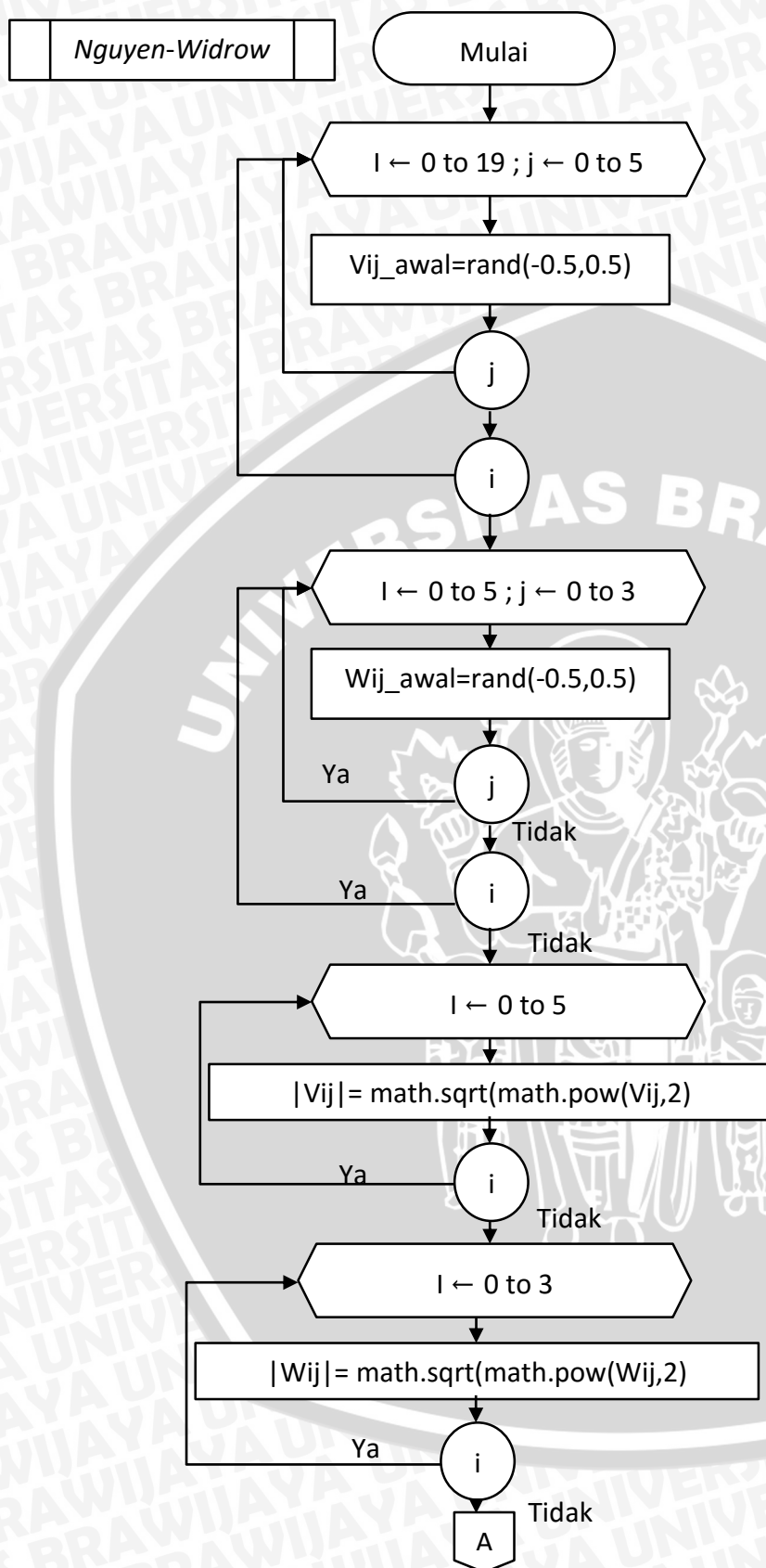
Backpropagation merupakan algoritma yang digunakan untuk proses pelatihan terdapat data yang sudah disediakan. Ada beberapa proses yang dilakukan pada algoritma ini yaitu menginisialisasi parameter dan bobot untuk setiap inputan yang selanjutnya akan diproses untuk menghitung keluaran (*feed foward*) dan memperbaiki bobot (*feed backward*) dilakukan berulang-ulang sampai didapatkan pola dari data yang telah disediakan. Gambar 4.1 merupakan diagram alir algoritma *backpropagation*.



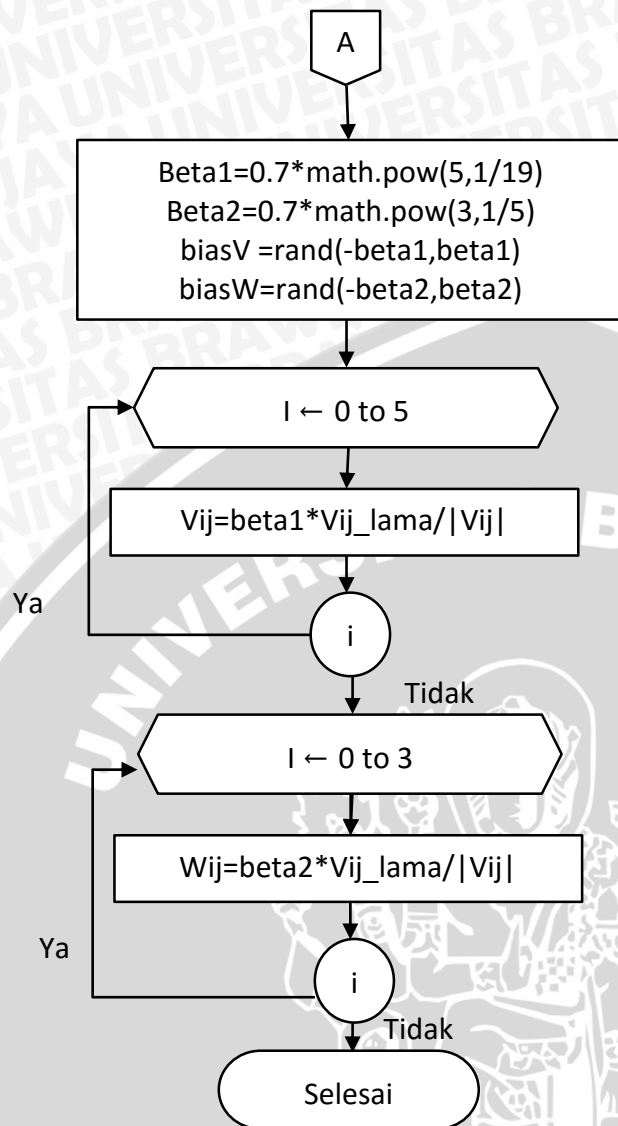
Gambar 4.1 Diagram Alir Backpropagation

4.1.3.1 Nguyen-Widrow

Nguyen Widrow adalah proses inisialisasi bobot masukan dari *input layer* ke *hidden layer*, *hidden layer* ke *output layer* serta biasanya yang dilakukan secara random yang hasilnya digunakan untuk proses berikutnya. Pertama menginisialisasi bobot masukan antara *input layer* dan *hidden layer* dan bobot masukan antara *hidden layer* dan *output layer*, selanjutnya menghitung nilai beta dan bobot optimal yang nanti akan digunakan untuk proses selanjutnya. Gambar 4.2 merupakan diagram alir proses nguyen-widrow.



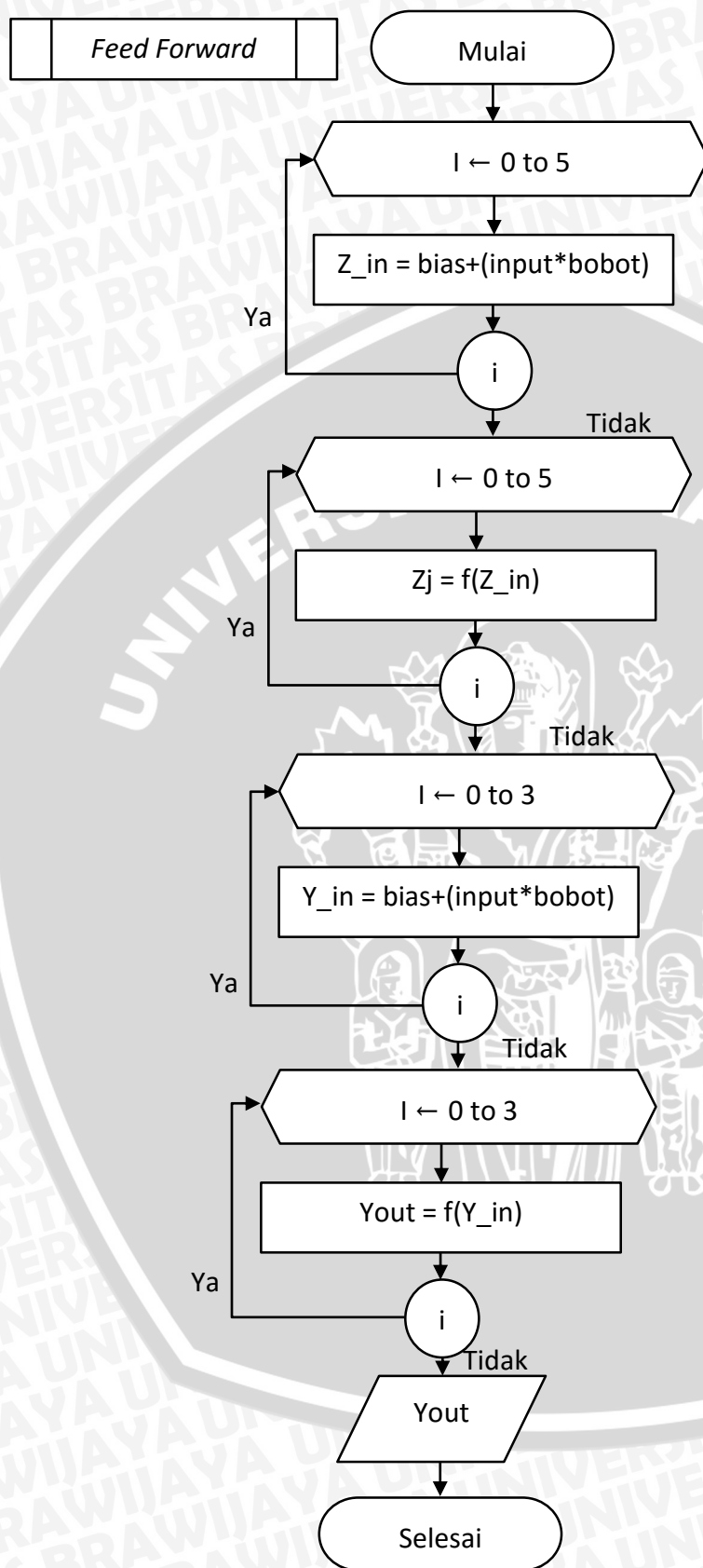
Gambar 4.2 Diagram Alir Proses Nguyen-Widrow



Gambar 4.2 Diagram Alir Proses Nguyen-Widrow

4.1.3.2 Feed Forward

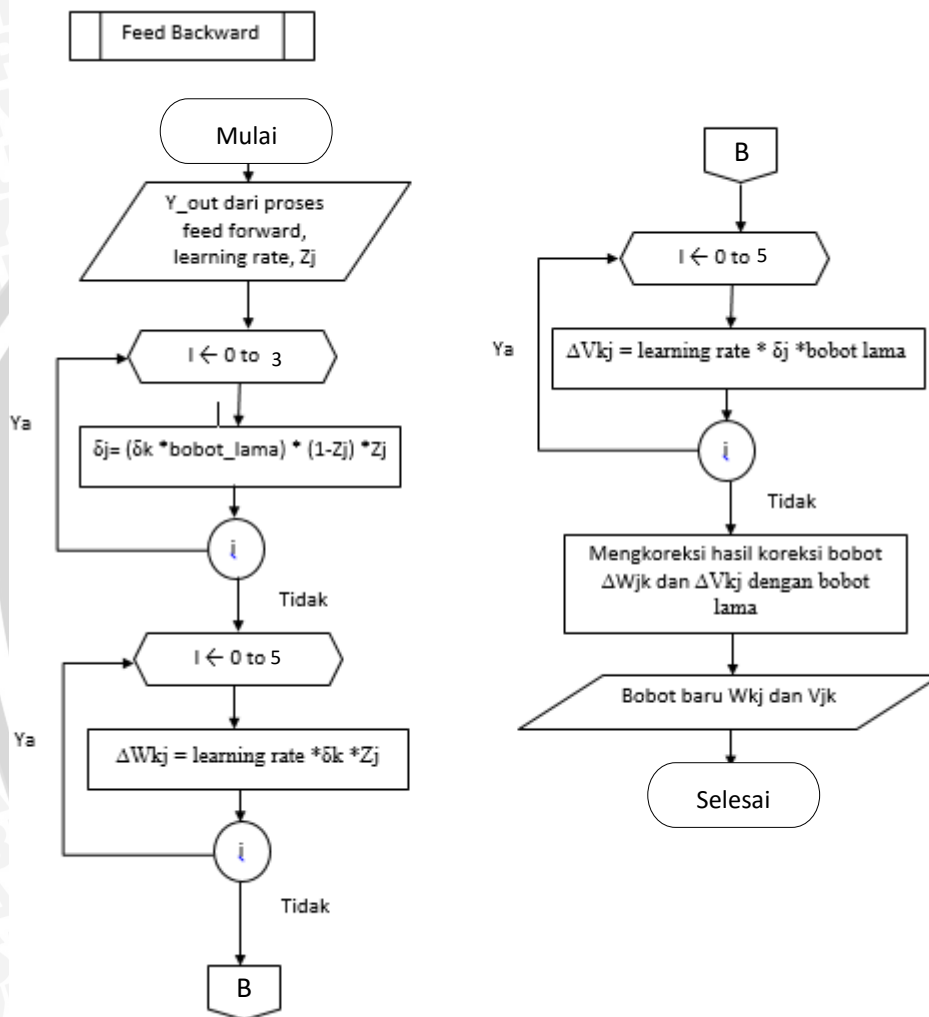
Feed forward adalah fase awal dalam proses *learning backpropagation*. Dalam fase ini *input layer* dihubungkan dengan *Hidden Layer* dan dihitung nilai keluarannya berdasarkan nilai inputan dan bobot yang telah didapat pada proses sebelumnya menggunakan fungsi aktivasi *sigmoid biner* yang telah ditentukan yang nantinya unit input akan menerima sinyal dan akan diteruskan ke *Hidden Layer*. Dan akan dilakukan perhitungan yang sama untuk menghitung nilai keluaran pada *output layer*. Gambar 4.3 merupakan diagram alir fase *feed forward*.



Gambar 4.3 Diagram Alir Sistem Fungsi Feed Forward

4.1.3.3 Feed Backward

Fase kedua ialah *feed backward* yang merupakan lanjutan dari fase *feed forward*, dimana pada fase ini melakukan perhitungan dari hasil *output Layer* dan *hidden layer*. Pada fase ini dilakukan koreksi factor error dan perbaikan bobot bias yang akan di gunakan untuk proses selanjutnya. Gambar 4.4 merupakan diagram alir dari sistem fungsi *feed backward*.

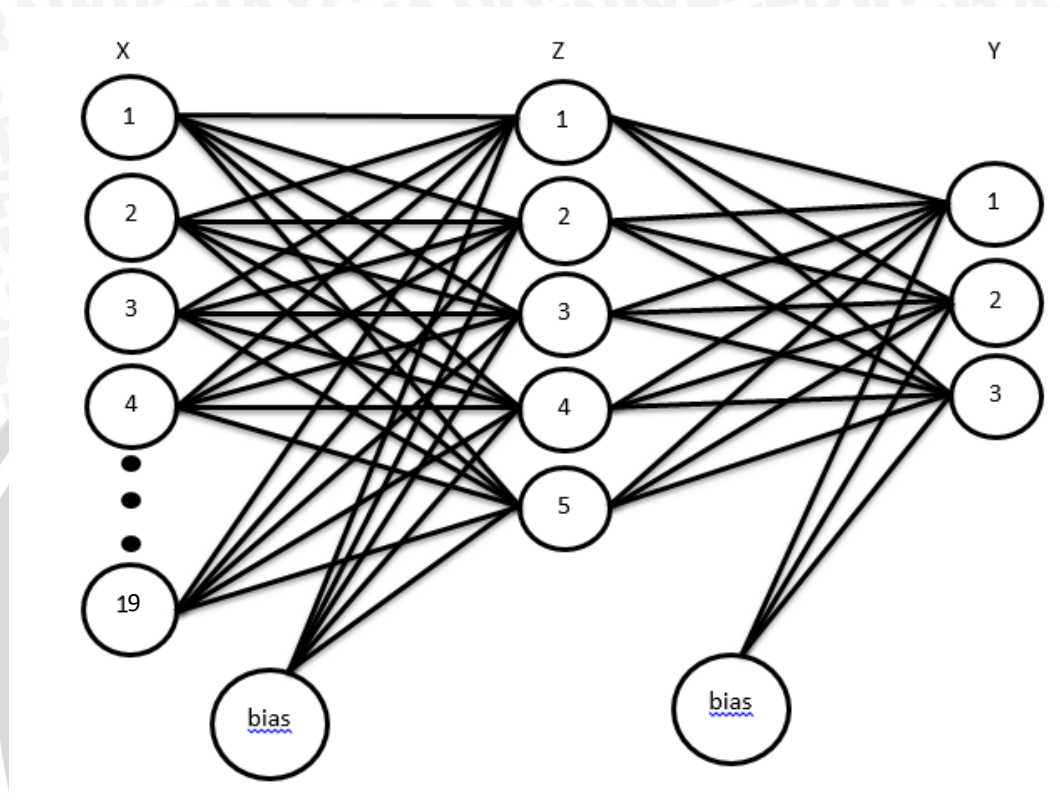


Gambar 4.4 Diagram Alir Sistem Fungsi *Feed Backward*

4.1.4 Perhitungan Manual

Perhitungan manual berfungsi sebagai gambaran umum perancangan sistem. Dengan melakukan perhitungan manual dapat diketahui apakah perhitungan yang nantinya akan di lakukan sistem benar atau tidak. Pada perhitungan manual ini menggunakan 6 data latih dan 1 data uji. Pada *input layer* terdapat 19 node yang merupakan gejala-gejala yang ada pada penyakit kulit seperti gelembung air merata, demam, nyeri kepala, gatal, lesi di seluruh tubuh,

mata merah, batuk, pilek, meradang, nyeri tekan, gelembung nanah, perih, bersisik, melepuh, panas, bengkak, lesi di area tertentu, benjolan pus/nanah dan gelembung air di area tertentu. Sedangkan pada *Hidden Layer* terdapat 5 node dan pada *Hidden Layer* terdapat 3 node. Struktire JST yang digunakan pada perhitungan manual ini adalah seperti gambar 4.4 sebagai berikut:



Gambar 4.5 Struktur JST untuk Diagnosis penyakit kulit pada anak

Keterangan: X adalah input layer, Y adalah Hidden layer, Z adalah Output layer

X_1 = Gelembung Air Merata

X_2 = Demam

X_3 = Nyeri Kepala

X_4 = Gatal

X_5 = Lesi Seluruh tubuh

X_6 = Mata merah

X_7 = Batuk

X_8 = Pilek

X_9 = Meradang

X_{10} = Nyeri tekan

X_{11} = Gelembung Nanah

X_{12} = Perih

X_{13} = Bersisik

X_{14} = Melepuh

X_{15} = Panas

X_{16} = Bengkak

X_{17} = Lesi di area tertentu

X_{18} = Benjolan pus/ nanah

X_{19} = Gelembung air di area tertentu

Variable lain yang ditentukan pada manualisasi ini adalah sebagai berikut:

- *Learning rate*(α) = 0,3
- Maximum epoch = 1
- Fungsi aktivasi = *sigmoid biner*

Pada perhitungan manual ini digunakan data latih sebanyak 5 data.

Table 4.1 Data latih manualisasi yang akan digunakan

		Data Latih																			
ID	NA MA	X 1	X 2	X 3	X 4	X 5	X 6	X 7	X 8	X 9	X1 0	X1 1	X1 2	X1 3	X1 4	X1 5	X1 6	X1 7	X1 8	X1 9	Target
1	Ana k1	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1 1 1
2	Ana k2	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1 1 1
3	Ana k3	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1 1 1
4	Ana k4	0	0	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	1 1 0
5	Ana k5	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	1 1 0
6	Ana k6	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1 1 0

Pada Tabel 4.1 terdapat 19 kolom nilai yang penilaiannya seperti dibawah ini:

- Pada kolom x1 sampai x18 merupakan gejala yang di derita:
1 = anak mengalami gejala tersebut
0 = anak tidak mengalami gejala
- Pada kolom target merupakan penyakit yang di derita

111 = Cacar Air

110 = Skabies

101 = Campak

100 = Dermatitis

011 = Herpes

010 = Abses

Berikut ini merupakan tahap-tahap perhitungan manual dengan metode *backpropagation* :

1. Menentukan bobot awal menggunakan algoritma *Nguyen-Widrow*.

a) Bobot ditentukan acak dengan range $(-0,5) - 0,5$ seperti pada Tabel 4.2 dibawah ini:

Table 4.2 Bobot awal acak antara *input layer* dengan *Hidden Layer* (Vij)

Vij	z1	z2	z3	z4	z5
X1	0.5	-0.1	0.1	-0.4	0.5
X2	0.2	-0.4	-0.2	-0.2	0.4
X3	0.5	0.3	0.3	-0.5	-0.1
X4	0.1	-0.4	-0.5	0.3	0.1
X5	-0.5	0.3	0.3	0.2	-0.4
X6	-0.1	-0.5	0.4	-0.4	0.2
X7	0.3	0.3	-0.1	-0.3	-0.4
X8	-0.1	-0.5	-0.1	-0.2	0.4
X9	0.1	-0.5	-0.1	0.4	-0.2
X10	0.4	0.1	0.4	0.5	0.1
X11	0.4	0.3	0.5	-0.3	0.1
X12	0.1	0.5	0.5	-0.2	0.1
X13	-0.1	0.3	-0.1	0.1	0.4
X14	-0.1	0.4	-0.4	0.2	0.3
X15	-0.2	-0.3	0.2	-0.3	-0.4
X16	-0.5	-0.3	-0.3	-0.2	-0.4
X17	-0.4	-0.3	-0.1	0.4	0.5
X18	-0.4	-0.4	0.2	-0.5	-0.5
X19	-0.1	0.3	-0.2	0.2	-0.3
Bias	0.1	-0.3	-0.4	-0.3	0.4

Table 4.3 Bobot awal acak antara *Hidden Layer* dengan *output layer* (*Wij*)

Wij	y1	y2	y3
z1	-0.5	-0.1	-0.1
z2	0.3	-0.5	0.2
z3	-0.4	-0.4	0.4
z4	0.1	-0.2	0.1
z5	0.2	0.3	-0.2
z6	0.5	-0.4	0.5

- b) Menghitung nilai $||V_{ij}||$ dengan bobot V_{ij} acak sebelumnya pada tabel 4.2 sesuai persamaan (2-22)

$$\begin{aligned}
 ||V_{i1}|| &= \sqrt{V_{11}^2 + V_{21}^2 + V_{31}^2 + V_{41}^2 + V_{51}^2 + V_{61}^2 + V_{71}^2 + V_{81}^2 + V_{91}^2 +} \\
 &\quad \sqrt{V_{101}^2 + V_{111}^2 + V_{121}^2 + V_{131}^2 + V_{141}^2 + V_{151}^2 + V_{161}^2 + V_{171}^2 + V_{181}^2} \\
 &= \sqrt{0.5^2 + 0.2^2 + 0.5^2 + 0.1^2 + (-0.5)^2 + (-0.1)^2 + 0.3^2 + (-0.1)^2} \\
 &\quad + 0.1^2 + 0.4^2 + 0.4^2 + 0.1^2 + (-0.1)^2 + (-0.1)^2 + (-0.2)^2 \\
 &\quad + (-0.5)^2 + (-0.4)^2 + (-0.4)^2} \\
 &= 1.37
 \end{aligned}$$

Perhitungan di atas dilakukan hingga $||V_{i5}||$ selesai dihitung, sehingga didapatkan hasil perhitungan seluruh $||V_{ij}||$ pada tabel 4.4 dibawah ini

Table 4.4 Hasil perhitungan $||V_{ij}||$

	z1	z2	z3	z4	z5
$ V_{ij} $	1.37	1.57	1.31	1.43	1.48

- c) Menghitung nilai $||W_{ij}||$ dengan bobot W_{ij} acak sebelumnya pada tabel 4.3

$$\begin{aligned}
 ||W_{i1}|| &= \sqrt{W_{11}^2 + W_{21}^2 + W_{31}^2 + W_{41}^2 + W_{51}^2} \\
 &= \sqrt{(-0.5)^2 + 0.3^2 + (-0.4)^2 + 0.1^2 + 0.2^2} \\
 &= 0.742
 \end{aligned}$$

Perhitungan diatas dilakukan hingga $||W_{i3}||$ selesai dihitung, sehingga didapatkan hasil perhitungan seluruh $||W_{ij}||$ pada tabel 4.5 dibawah ini

Table 4.5 Hasil Perhitungan $||W_{ij}||$

	y1	y2	y3
$ W_{ij} $	0.742	0.742	0.510

- d) Menghitung nilai β antara *input layer* dengan *Hidden Layer* sesuai persamaan (2-23).

n = jumlah unit masukan = 19 , p = jumlah unit keluaran = 5

$$\beta_1 = 0.7 \sqrt[n]{p} = 0.7 \sqrt[19]{5} = 0.76$$

- e) Menghitung nilai β antara *Hidden Layer* dengan *output layer* sesuai persamaan (2-23).

n = jumlah unit masukan = 5 , p = jumlah unit keluaran = 3

$$\beta_2 = 0.7 \sqrt[n]{p} = 0.7 \sqrt[5]{3} = 0.87$$

- f) Menghitung nilai bobot V_{ij} dengan menggunakan bobot acak V_{ij} lama pada tabel 4.2 sesuai persamaan (2-24). Dibawah ini merupakan perhitungan bobot V_{ij} pada *Hidden Layer* pertama:

$$V_{11} = \frac{\beta_1 V_{11}(\text{lama})}{||V_{i1}||} = \frac{0.77 \times 0.5}{1.43} = 0.28$$

Perhitungan di atas dilakukan hingga V_{195} selesai dihitung, sehingga didapatn hasil perhitungan seluruh V_{ij} pada tabel 4.6 dibawah ini.

Table 4.6 Hasil perhitungan bobot awal (V_{ij}) Nguyen-Widrow

V_{ij}	z_1	z_2	z_3	z_4	z_5
X1	0.28	-0.05	0.06	-0.21	0.26
X2	0.11	-0.19	-0.12	-0.11	0.21
X3	0.28	0.15	0.17	-0.27	-0.05
X4	0.06	-0.19	-0.29	0.16	0.05
X5	-0.28	0.15	0.17	0.11	-0.21
X6	-0.06	-0.24	0.23	-0.21	0.10
X7	0.17	0.15	-0.06	-0.16	-0.21
X8	-0.06	-0.24	-0.06	-0.11	0.21
X9	0.06	-0.24	-0.06	0.21	-0.10
X10	0.22	0.05	0.23	0.27	0.05
X11	0.22	0.15	0.29	-0.16	0.05
X12	0.06	0.24	0.29	-0.11	0.05
X13	-0.06	0.15	-0.06	0.05	0.21
X14	-0.06	0.19	-0.23	0.11	0.15
X15	-0.11	-0.15	0.12	-0.16	-0.21
X16	-0.28	-0.15	-0.17	-0.11	-0.21
X17	-0.22	-0.15	-0.06	0.21	0.26

Table 4.6 Hasil Perhitungan bobot awal (V_{ij}) Nguyen-Widrow

Vij	z1	z2	z3	z4	z5
X18	-0.22	-0.19	0.12	-0.27	-0.26
X19	-0.06	0.15	-0.12	0.11	-0.15

Untuk perhitungan bobot bias dipakai bilangan acak antara range $-\beta_1$ dan β_1 yaitu antara $(-0.76) - 0.76$. tabel 4.7 merupakan hasil perhitungan bobot bias

Table 4.7 Hasil perhitungan bobot bias dengan Nguyen-Widrow

	z1	z2	z3	z4	z5
Bias	0.09	-0.43	-0.48	-0.34	0.12

Menghitung nilai bobot Wij dengan menggunakan bobot acak Wij lama pada table 3.4 sesuai persamaan (2-24) dibawah ini merupakan perhitungan bobot Wij:

$$W_{11} = \frac{\beta_2 W_{11}(\text{lama})}{||W_{i1}||} = \frac{0.87x(-0.5)}{0.742} = (-0.59)$$

Perhitungan di atas di lakukan hingga W_{53} selesai dihitung, sehingga didapatkn hasil perhitungan seluruh W_{ij} . Untuk perhitungan bobot bias dipakai bilangan acak antara range $-\beta_2$ dan β_2 yaitu antara $(-0.87) - 0.87$. tabel 4.8 merupakan tabel hasil perhitungan Wij dan bias.

Table 4.8 Hasil perhitungan bobot Wij dan bias dengan Nguyen-Widrow

Wij	y1	y2	y3
z1	-0.59	-0.12	-0.17
z2	0.35	-0.59	0.34
z3	-0.47	-0.47	0.68
z4	0.12	-0.24	0.17
z5	0.24	0.35	-0.34
Bias	-0.35	-0.29	0.46

Iterasi 1

Anak1, ID 1

1. Menghitung neuron yang masuk ke *Hidden Layer* dari *input layar* ($Z_{in,j}$) dengan persaman (2-8). Tiap nilai input (X_1) Anak 1 dengan ID 1 pada tabel 4.1 dikalikan dengan bobot (V_{ij}) pada tabel 4.6 dan kemudian dijumlahkan dengan bobot biasnya.

Perhitungan *input* pada *Hidden Layer* pertama ($Z_{in,1}$):

$$\begin{aligned}
 Z_{in_1} &= \text{bobot bias}(V_{71}) + (X_1 * V_{11}) + (X_2 * V_{21}) + (X_3 * V_{31}) + (X_4 * V_{41}) + (X_5 * V_{51}) + \\
 &\quad (X_6 * V_{61}) + (X_7 * V_{71}) + (X_8 * V_{81}) + (X_9 * V_{91}) + (X_{10} * V_{101}) + (X_{11} * V_{111}) + \\
 &\quad (X_{12} * V_{121}) + (X_{13} * V_{131}) + (X_{14} * V_{141}) + (X_{15} * V_{151}) + (X_{16} * V_{161}) + (X_{17} * V_{171}) + \\
 &\quad (X_{18} * V_{181}) + (X_{19} * V_{191}) \\
 &= 0.09 + (1 * 0.28) + (1 * 0.11) + (0 * 0.28) + (0 * 0.06) + (1 * (-0.28)) + (0 * (-0.06)) + \\
 &\quad (0 * 0.17) + (0 * (-0.06)) + (0 * 0.06) + (1 * 0.22) + (0 * 0.22) + (0 * 0.06) + (0 * (-0.06)) + \\
 &\quad (0 * (-0.06)) + (0 * (-0.11)) + (0 * (-0.28)) + (0 * (-0.22)) + (0 * (-0.22)) + (0 * (-0.06)) \\
 &= 0.42
 \end{aligned}$$

Perhitungan diatas dilakukan hingga Z_{in_5} selesai di hitung, sehingga didapatkan hasil perhitungan seluruh Z_{in} pada tabel 4.9 dibawah ini :

Table 4.9 Hasil perhitungan Z_{in}

Z_{in_1}	Z_{in_2}	Z_{in_3}	Z_{in_4}	Z_{in_5}
0.42	-0.48	-0.13	-0.29	0.43

- Menghitung neuron yang keluar dari *Hidden Layer* menuju *output layer* (Z_j). Z_j dihitung dengan menggunakan fungsi aktivasi sigmoid biner dengan persamaan (2-9).

Perhitungan *output* dari *Hidden Layer* pertama Z_1 :

$$Z_1 = \frac{1}{(1 + e^{-Z_{in1}})} = \frac{1}{(1 + e^{-0.42})} = 0.60$$

Perhitungan di lakukan hingga Z_5 . tabel 4.10 merupakan hasil perhitungan Z_j

Table 4.10 Hasil perhitungan Z_j

Z_1	Z_2	Z_3	Z_4	Z_5
0.60	0.38	0.47	0.43	0.61

- Menghitung neuron yang masuk ke *output layer* dari *Hidden Layer* Y_{in} . Y_{in} dihitung dengan menggunakan persamaan (2-10). Tiap neuron yang keluar dari *Hidden Layer* (Z_j) pada tabel 4.10 dikalikan dengan bobot antara *Hidden Layer* dan *output layer* (W_{ij}) pada tabel 4.8 dan kemudian dijumlahkan dengan bobot biasnya

Perhitungan Y_{in} :

$$\begin{aligned}
 Y_{in_1} &= \text{bobot bias}(W_{61}) + (Z_1 * W_{11}) + (Z_2 * W_{21}) + (Z_3 * W_{31}) + (Z_4 * W_{41}) + \\
 &\quad (Z_5 * W_{51}) \\
 &= (-0.35) + (0.60 * (-0.59)) + (0.38 * 0.35) + (0.47 * (-0.47)) + (0.43 * 0.12) + \\
 &\quad (0.61 * 0.24) \\
 &= (-0.5977)
 \end{aligned}$$

Perhitungan dilakukan terus hingga Y_{in_3} didapatkan pada tabel 4.11 dibawah ini :

Table 4.11 Hasil Perhitungan Y_{in}

Y_{in_1}	Y_{in_2}	Y_{in_3}
-0.5977	-0.6928	0.6733

4. Menghitung neuron keluaran dari *output layer* (Y_{out}). Y_{out} dihitung dengan menggunakan fungsi aktivasi *sigmoid biner* dengan persamaan (2-11).

Perhitungan Y_{out_1} :

$$\begin{aligned} Y_{out_1} &= 1/(1+e^{Y_{in_j}}) \\ &= 1/(1+e^{-(-0.5977)}) \\ &= 0.355 \end{aligned}$$

Perhitungan dilakukan terus hingga Y_{out_3} didapatkan terlihat pada tabel 4.12 dibawah ini:

Table 4.12 Hasil Perhitungan Y_{out}

Y_{out_1}	Y_{out_2}	Y_{out_3}
0.355	0.333	0.662

5. Menghitung nilai *error* antara *Hidden Layer* dan *input layer* (δ_k). δ_k dihitung menggunakan perkalian antara selisih target di table 4.1 dan keluaran (Y_{out}) seperti pada persamaan (2-12).

Perhitungan δ_k pada Anak 1, ID 1:

$$\begin{aligned} \delta_{k_1} &= (\text{target} - Y_{out_1}) * (1 - Y_{out_1}) * Y_{out_1} \\ &= (1 - 0.355) * (1 - 0.355) * 0.355 \\ &= 0.148 \end{aligned}$$

Perhitungan di lakukan hingga δ_{k_3} didapatkan. Hasil δ_k dapat dilihat pada tabel 4.13 dibawah ini:

Table 4.13 hasil perhitungan δ_k

δ_{k_1}	δ_{k_2}	δ_{k_3}
0.148	0.148	0.076

6. Menghitung koreksi bobot antara *Hidden Layer* dan *output layer* (ΔW_{ij}). ΔW_{ij} dihitung dengan mengalikan nilai α (*learning rate*) dengan nilai δ_k masing – masing node dan Z_j pada tabel 3.11 seperti persamaan (2-13).

Perhitungan ΔW_{ij} :

$$\begin{aligned}\Delta W_{11} &= \alpha * \delta_{k_1} * Z_1 \\ &= 0.3 * 0.148 * 0.60 \\ &= 0.027\end{aligned}$$

Perhitungan dilakukan hingga kelima koreksi perubahan bobot pada setiap node *output* didapatkan. Sedangkan untuk menghitung nilai perubahan bobot dari bias hanya mengalikan α (*learning rate*) dengan nilai *error* (δ_k), seperti dibawah ini:

$$\begin{aligned}\Delta W_{bias_1} &= \alpha * \delta_{k_1} \\ &= 0.3 * 0.148 \\ &= 0.044\end{aligned}$$

Perhitungan juga dilakukan hingga perubahan bobot biar untuk ketiga node *output* didapatkan. Tabel 4.14 merupakan hasil perhitungan perubahan bobot antara *Hidden Layer* dan *output layer* (ΔW_{ij} & ΔW_{bias_j}).

Table 4.14 Hasil perhitungan koreksi bobot ΔW_{ij}

ΔW_{ij}	δ_1	δ_2	δ_3
Z1	0.027	0.027	0.014
Z2	0.017	0.017	0.009
Z3	0.021	0.021	0.011
Z4	0.019	0.019	0.010
Z5	0.027	0.027	0.014
Bias	0.044	0.044	0.023

7. Menghitung nilai *error* antara *Hidden Layer* dan *input layer* (δ_i) yang akan digunakan untuk menghitung perubahan bobot. δ_j dihitung dengan mengalikan δ_{in_j} , bobot awal antara *Hidden Layer* dan *output layer*, serta Z_j dengan menggunakan persamaan (2-15).

menghitung δ_{in_j} dengan menjumlahkan hasil perkalian dari nilai *error* (δ_k) dikali dengan bobot awal antara *Hidden Layer* dan *output layer* pada tabel 3.9.

Perhitungan δ_{in_j} :

$$\begin{aligned}\delta_{in_1} &= (\delta_{k_1} * W_{11}) + (\delta_{k_2} * W_{12}) + (\delta_{k_3} * W_{13}) \\ &= (0.148 * (-0.59)) + (0.148 * (-0.12)) + (0.076 * (-0.17)) \\ &= (-0.117)\end{aligned}$$

Perhitungan dilakukan hingga δ_{in_5} didapatkan seperti pada tabel 3.16 dibawah ini:

Table 4.15 Hasil perhitungan δ_{in_j}

δ_{in1}	δ_{in2}	δ_{in3}	δ_{in4}	δ_{in5}
-0.117	-0.009	-0.087	-0.005	0.061

Perhitungan δ_j :

$$\begin{aligned}\delta_1 &= \delta_{in_1} * (1-Z_1) * Z_1 \\ &= (-0.117) * (1-0.60) * 0.60 \\ &= (-0.028)\end{aligned}$$

Perhitungan dilakukan hingga δ_5 didapatkan seperti pada tabel 4.16 dibawah ini:

Table 4.16 Hasil Perhitungan δ_j

δ_1	δ_2	δ_3	δ_4	δ_5
-0.028	-0.002	-0.022	-0.001	0.015

8. Menghitung koreksi bobot antara *Hidden Layer* dan *input layer* (ΔV_{ij}). ΔV_{ij} dihitung dengan mengalikan α (*learning rate*) dengan δ_j ditabel 4.16 dan bobot awal X_i (V_{ij}) di tabel 4.6 seperti persamaan (2-18).

Perhitungan ΔV_{ij} :

$$\begin{aligned}\Delta V_{11} &= \alpha * \delta_{j1} * X_{11} \\ &= 0.3 * (-0.028) * 1 \\ &= (-0.0084)\end{aligned}$$

Perhitungan dilakukan hingga semua koreksi bobot antara *Hidden Layer* dan *input layer* didapatkan. Untuk menghitung koreksi bobot bias dengan mengalikan α (*learning rate*) dengan δ_j .

Perhitungan ΔV_{bias_j} :

$$\begin{aligned}\Delta V_{bias1} &= \alpha * \delta_1 \\ &= 0.3 * (-0.028) \\ &= (-0.0084)\end{aligned}$$

Perhitungan dilakukan hingga ΔV_{bias5} didapatkan seperti pada tabel 4.17 dibawah ini:

Table 4.17 Hasil Perhitungan koreksi bobot ΔV_{ij} & bias

ΔV_{ij}	δ_1	δ_2	δ_3	δ_4	δ_5
X1	-0.0084	-0.0006	-0.0065	-0.0003	0.0044
X2	-0.0084	-0.0006	-0.0065	-0.0003	0.0044
X3	0.0000	0.0000	0.0000	0.0000	0.0000
X4	0.0000	0.0000	0.0000	0.0000	0.0000
X5	-0.0084	-0.0006	-0.0065	-0.0003	0.0044
X6	0.0000	0.0000	0.0000	0.0000	0.0000
X7	0.0000	0.0000	0.0000	0.0000	0.0000
X8	0.0000	0.0000	0.0000	0.0000	0.0000
X9	0.0000	0.0000	0.0000	0.0000	0.0000
X10	-0.0084	-0.0006	-0.0065	-0.0003	0.0044
X11	0.0000	0.0000	0.0000	0.0000	0.0000
X12	0.0000	0.0000	0.0000	0.0000	0.0000
X13	0.0000	0.0000	0.0000	0.0000	0.0000
X14	0.0000	0.0000	0.0000	0.0000	0.0000
X15	0.0000	0.0000	0.0000	0.0000	0.0000
X16	0.0000	0.0000	0.0000	0.0000	0.0000
X17	0.0000	0.0000	0.0000	0.0000	0.0000
X18	0.0000	0.0000	0.0000	0.0000	0.0000
X19	0.0000	0.0000	0.0000	0.0000	0.0000
Bias	-0.0084	-0.0006	-0.0065	-0.0003	0.0044

9. Menghitung bobot baru antara *input layer* dan *Hidden Layer* (V_{ij}). V_{ij} dihitung dengan menjumlahkan bobot lama pada tabel 4.6 dengan koreksi bobot ΔV_{ij} pada tabel 3.18 seperti persamaan 2-18.

Perhitungan V_{ij} pertama :

$$\begin{aligned}
 V_{11} &= V_{1i} \text{ (lama)} + \Delta V_{11} \\
 &= 0.28 + (-0.0084) \\
 &= 0.269
 \end{aligned}$$

Perhitungan dilakukan hingga V_{18} 5 didapatkan, sehingga didapat nilai bobot baru secara keseluruhan pada tabel 4.18

Table 4.18 Hasil perhitungan bobot baru V_{ij}

V_{ij}	z_1	z_2	z_3	z_4	z_5
X1	0.269	-0.049	0.052	-0.214	0.262
X2	0.102	-0.195	-0.123	-0.107	0.211
X3	0.277	0.145	0.174	-0.267	-0.052
X4	0.055	-0.194	-0.290	0.160	0.052
X5	-0.285	0.145	0.168	0.106	-0.202
X6	-0.055	-0.242	0.232	-0.213	0.103
X7	0.166	0.145	-0.058	-0.160	-0.206
X8	-0.055	-0.242	-0.058	-0.107	0.206
X9	0.055	-0.242	-0.058	0.213	-0.103
X10	0.213	0.048	0.226	0.266	0.056
X11	0.222	0.145	0.290	-0.160	0.052
X12	0.055	0.242	0.290	-0.107	0.052
X13	-0.055	0.145	-0.058	0.053	0.206
X14	-0.055	0.194	-0.232	0.107	0.155
X15	-0.111	-0.145	0.116	-0.160	-0.206
X16	-0.277	-0.145	-0.174	-0.107	-0.206
X17	-0.222	-0.145	-0.058	0.213	0.258
X18	-0.222	-0.194	0.116	-0.267	-0.258
X19	-0.055	0.145	-0.116	0.107	-0.155
Bias	0.082	-0.431	-0.487	-0.340	0.124

10. Menghitung bobot baru antara *Hidden Layer* dan *output layer* (W_{ij}). W_{ij} dihitung dengan menjumlahkan bobot lama pada tabel 4.8 dengan koreksi bobot ΔW_{ij} pada tabel 4.14 seperti persamaan 2-17.

Perhitungan W_{ij} pertama:

$$\begin{aligned}
 W_{11} &= W_{1i(\text{lama})} + \Delta W_{11} \\
 &= (-0.59) + 0.027 \\
 &= (-0.561)
 \end{aligned}$$

Perhitungan dilakukan hingga W_{53} didapatkan, sehingga didapatkan nilai bobot baru secara keseluruhan pada tabel 4.19

Table 4.19 Hasil perhitungan bobot baru W_{ij}

W_{ij}	y_1	y_2	y_3
z_1	-0.561	-0.091	-0.157
z_2	0.370	-0.571	0.351
z_3	-0.450	-0.450	0.695
z_4	0.137	-0.216	0.181
z_5	0.262	0.380	-0.328
Bias	-0.306	-0.246	0.483

Selanjutnya akan melakukan perhitungan yang sama seperti diatas dengan menggunakan data ke-2.

Anak2 ,ID 2

Pada data Anak2, ID 2 perhitungan akan dilakukan dengan menggunakan hasil bobot baru pada tabel 4.18 dan tabel 4.19 yang di dapat dari perhitungan data pertama.

1. Menghitung neuron yang masuk ke *Hidden Layer* dari *input layar* (Z_{in_j}) dengan persamaan (2-8). Tiap nilai input (X_1) Anak 2 dengan ID 2 pada tabel 4.1 dikalikan dengan bobot (V_{ij}) pada tabel 4.19 dan kemudian dijumlahkan dengan bobot biasnya.

Perhitungan *input* pada *Hidden Layer* pertama (Z_{in_1}):

$$\begin{aligned}
 Z_{in_1} &= \text{bobot bias}(V_{71})+(X_1*V_{11})+ (X_2*V_{21})+ (X_3*V_{31})+ (X_4*V_{41})+ (X_5*V_{51})+ \\
 & (X_6*V_{61})+ (X_7*V_{71})+ (X_8*V_{81})+ (X_9*V_{91})+ (X_{10}*V_{101})+ (X_{11}*V_{111})+ \\
 & (X_{12}*V_{121})+ (X_{13}*V_{131})+ (X_{14}*V_{141})+ (X_{15}*V_{151})+ (X_{16}*V_{161})+ (X_{17}*V_{171})+ \\
 & (X_{18}*V_{181})+(X_{19}*X_{191}) \\
 &= 0.082+(1*0.269)+ (0*0.102)+ (0*0.277)+ (1*0.055)+ (1*(-0.285))+ \\
 & (0*(-0.055))+ (0*0.166)+ (0*(-0.055))+ (0*0.55)+ (1*0.213)+ \\
 & (0*0.222)+ (0*0.055)+ (0*(-0.055))+ (0*(-0.055))+ (0*(-0.111))+ \\
 & (0*(-0.277))+ (0*(-0.222))+ (0*(-0.222))+ (0*(-0.055)) \\
 &= 0.33
 \end{aligned}$$

Perhitungan diatas dilakukan hingga Z_{in_5} selesai di hitung, sehingga didapatkan hasil perhitungan seluruh Z_{in} pada tabel 4.20 dibawah ini :

Table 4.20 Hasil perhitungan Z_{in} Anak2 Iterasi 1

Z_{in_1}	Z_{in_2}	Z_{in_3}	Z_{in_4}	Z_{in_5}
0.33	-0.48	-0.33	-0.02	0.29

2. Menghitung neuron yang keluar dari *Hidden Layer* menuju *output layer* (Z_j). Z_j dihitung dengan menggunakan fungsi aktivasi sigmoid biner dengan persamaan (2-9).

Perhitungan *output* dari *Hidden Layer* pertama Z_1 :

$$Z_1 = \frac{1}{(1+e^{Z_{in1}})} = \frac{1}{(1+e^{0.33})} = 0.583$$

Perhitungan di lakukan hingga Z_5 . tabel 4.21 merupakan hasil perhitungan Z_j

Table 4.21 Hasil perhitungan Z_j Anak2 Iterasi 1

Z1	Z2	Z3	Z4	Z5
0.583	0.382	0.418	0.495	0.573

3. Menghitung neuron yang masuk ke *output layer* dari *Hidden Layer* Y_{in} . Y_{in} dihitung dengan menggunakan persamaan (2-10). Tiap neuron yang keluar dari *Hidden Layer* (Z_j) pada tabel 4.21 dikalikan dengan bobot antara *Hidden Layer* dan *output layer* (W_{ij}) pada tabel 4.19 dan kemudian dijumlahkan dengan bobot biasnya

Perhitungan Y_{in} :

$$\begin{aligned} Y_{in_1} &= \text{bobot bias}(W_{61}) + (Z_1*W_{11})+ (Z_2*W_{21})+ (Z_3*W_{31})+ (Z_4*W_{41})+ \\ &\quad (Z_5*W_{51}) \\ &= (-0.306)+(0.583*(-0.561))+ (0.382*0.370)+ (0.418*(-0.450))+ \\ &\quad (0.495*0.137)+ (0.573*0.262) \\ &= (-0.462) \end{aligned}$$

Perhitungan dilakukan terus hingga Y_{in_3} didapatkan pada tabel 4.22 dibawah ini :

Table 4.22 Hasil Perhitungan Y_{in} Anak2 Iterasi 1

Y_{in_1}	Y_{in_2}	Y_{in_3}
-0.462	-0.594	0.717

4. Menghitung neuron keluaran dari *output layer* (Y_{out}). Y_{out} dihitung dengan menggunakan fungsi aktivasi *sigmoid biner* dengan persamaan (2-11).

Perhitungan Y_{out_1} :

$$\begin{aligned} Y_{out_1} &= 1/(1+e^{Y_{in_j}}) \\ &= 1/(1+e^{-(-0.462)}) \\ &= 0.387 \end{aligned}$$

Perhitungan dilakukan terus hingga Y_{out_3} didapatkan terlihat pada tabel 4.23 dibawah ini:

Table 4.23 Hasil Perhitungan Y_{out} Anak2 Iterasi 1

Yout_1	Yout_2	Yout_3
0.387	0.356	0.672

5. Menghitung nilai *error* antara *Hidden Layer* dan *input layer* (δ_k). δ_k dihitung menggunakan perkalian antara selisih target di table 4.1 dan keluaran (Y_{out}) seperti pada persamaan (2-12).

Perhitungan δ_k pada Anak 2, ID 2:

$$\begin{aligned}\delta_{k_1} &= (\text{target} - Y_{out_1}) * (1 - Y_{out_1}) * Y_{out_1} \\ &= (1 - 0.387) * (1 - 0.387) * 0.387 \\ &= 0.145\end{aligned}$$

Perhitungan di lakukan hingga δ_{k_3} didapatkan. Hasil δ_k dapat dilihat pada tabel 4.24 dibawah ini:

Table 4.24 nilai δ_k Anak2 Iterasi 1

δ_1	δ_2	δ_3
0.145	0.148	0.072

6. Menghitung koreksi bobot antara *Hidden Layer* dan *output layer* (ΔW_{ij}). ΔW_{ij} dihitung dengan mengalikan nilai α (*learning rate*) dengan nilai δ_k masing – masing node dan Z_j pada tabel 4.21 seperti persamaan (2-13).

Perhitungan ΔW_{ij} :

$$\begin{aligned}\Delta W_{11} &= \alpha * \delta_{k_1} * Z_1 \\ &= 0.3 * 0.145 * 0.583 \\ &= 0.025\end{aligned}$$

Perhitungan dilakukan hingga kelima koreksi perubahan bobot pada setiap node *output* didapatkan. Sedangkan untuk menghitung nilai perubahan bobot dari bias hanya mengalikan α (*learning rate*) dengan nilai *error* (δ_k). seperti dibawah ini:

$$\begin{aligned}\Delta W_{bias_1} &= \alpha * \delta_{k_1} \\ &= 0.3 * 0.145 \\ &= 0.044\end{aligned}$$

Perhitungan juga dilakukan hingga perubahab bobot biar untuk ketiga node *output* didapatkan. Tabel 4.25 merupakan hasil perhitungan perubahan bobot antara *Hidden Layer* dan *output layer* (ΔW_{ij} & ΔW_{bias_j}).

Table 4.25 Hasil perhitungan koreksi bobot ΔW_{ij} Anak2 Iterasi 1

ΔW_{ij}	δ_1	δ_2	δ_3
Z1	0.025	0.026	0.013
Z2	0.017	0.017	0.008
Z3	0.018	0.019	0.009
Z4	0.022	0.022	0.011
Z5	0.025	0.025	0.012
Bias	0.044	0.044	0.022

7. Menghitung nilai *error* antara *Hidden Layer* dan *input layer* (δ_j) yang akan digunakan untuk menghitung perubahan bobot. δ_j dihitung dengan mengalikan δ_{in_j} , bobot awal antara *Hidden Layer* dan *output layer*, serta Z_j dengan menggunakan persamaan (2-15).

menghitung δ_{in_j} dengan menjumlahkan hasil perkalian dari nilai *error* (δ_k) dikali dengan bobot awal antara *Hidden Layer* dan *output layer* pada tabel 4.19.

Perhitungan δ_{in_j} :

$$\begin{aligned}\delta_{in_1} &= (\delta_{k_1} * W_{11}) + (\delta_{k_2} * W_{12}) + (\delta_{k_3} * W_{13}) \\ &= (0.145 * (-0.561)) + (0.148 * (-0.091)) + (0.072 * (-0.157)) \\ &= (-0.106)\end{aligned}$$

Perhitungan dilakukan hingga δ_{in_5} didapatkan seperti pada tabel 4.26 dibawah ini:

Table 4.26 Hasil perhitungan δ_{in_j} Anak2 Iterasi 1

δ_{in1}	δ_{in2}	δ_{in3}	δ_{in4}	δ_{in5}
-0.106	-0.005	-0.082	0.001	0.070

Perhitungan δ_j :

$$\begin{aligned}\delta_1 &= \delta_{in_1} * (1 - Z_1) * Z_1 \\ &= (-0.106) * (1 - 0.583) * 0.583 \\ &= (-0.026)\end{aligned}$$

Perhitungan dilakukan hingga δ_5 didapatkan seperti pada tabel 4.27 dibawah ini:

Table 4.27 Hasil Perhitungan δ_j Anak2 Iterasi 1

δ_1	δ_2	δ_3	δ_4	δ_5
-0.026	-0.001	-0.020	0.0003	0.017

8. Menghitung koreksi bobot antara *Hidden Layer* dan *input layer* (ΔV_{ij}). ΔV_{ij} dihitung dengan mengalikan α (*learning rate*) dengan δ_j ditabel 4.16 dan bobot awal X_i (V_{ij}) di tabel 4.18 seperti persamaan (2-18).

Perhitungan ΔV_{ij} :

$$\begin{aligned}\Delta V_{11} &= \alpha * \delta_{j1} * X_{11} \\ &= 0.3 * (-0.026) * 1 \\ &= (-0.0078)\end{aligned}$$

Perhitungan dilakukan hingga semua koreksi bobot antara *Hidden Layer* dan *input layer* didapatkan. Untuk menghitung koreksi bobot bias dengan mengalikan α (*learning rate*) dengan δ_j .

Perhitungan ΔV_{bias_j} :

$$\begin{aligned}\Delta V_{bias1} &= \alpha * \delta_1 \\ &= 0.3 * (-0.026) \\ &= (-0.0078)\end{aligned}$$

Perhitungan dilakukan hingga ΔV_{bias5} didapatkan seperti pada tabel 4.28 dibawah ini:

Table 4.28 Hasil Perhitungan koreksi bobot ΔV_{ij} & bias Anak2 Iterasi 1

ΔV_{ij}	δ_1	δ_2	δ_3	δ_4	δ_5
X1	-0.0078	-0.0004	-0.0059	0.0001	0.0052
X2	0.0000	0.0000	0.0000	0.0000	0.0000
X3	0.0000	0.0000	0.0000	0.0000	0.0000
X4	-0.0078	-0.0004	-0.0059	0.0001	0.0052
X5	-0.0078	-0.0004	-0.0059	0.0001	0.0052
X6	0.0000	0.0000	0.0000	0.0000	0.0000
X7	0.0000	0.0000	0.0000	0.0000	0.0000
X8	0.0000	0.0000	0.0000	0.0000	0.0000
X9	0.0000	0.0000	0.0000	0.0000	0.0000
X10	-0.0078	-0.0004	-0.0059	0.0001	0.0052
X11	0.0000	0.0000	0.0000	0.0000	0.0000
X12	0.0000	0.0000	0.0000	0.0000	0.0000

Gambar 4.28 Hasil Perhitungan koreksi bobot ΔV_{ij} & bias Anak2 Iterasi 1

V_{ij}	z1	z2	z3	z4	z5
X13	0.0000	0.0000	0.0000	0.0000	0.0000
X14	0.0000	0.0000	0.0000	0.0000	0.0000
X15	0.0000	0.0000	0.0000	0.0000	0.0000
X16	0.0000	0.0000	0.0000	0.0000	0.0000
X17	0.0000	0.0000	0.0000	0.0000	0.0000
X18	0.0000	0.0000	0.0000	0.0000	0.0000
X19	0.0000	0.0000	0.0000	0.0000	0.0000
Bias	-0.0078	-0.0004	-0.0059	0.0001	0.0052

9. Menghitung bobot baru antara *input layer* dan *Hidden Layer* (V_{ij}). V_{ij} dihitung dengan menjumlahkan bobot lama pada tabel 4.18 dengan koreksi bobot ΔV_{ij} pada tabel 4.28 seperti persamaan 2-18.

Perhitungan V_{ij} pertama :

$$\begin{aligned}
 V_{11} &= V_{1i} \text{ (lama)} + \Delta V_{11} \\
 &= 0.269 + (-0.0078) \\
 &= 0.261
 \end{aligned}$$

Perhitungan dilakukan hingga V_{18} didapatkan, sehingga didapat nilai bobot baru secara keseluruhan pada tabel 4.29

Table 4.29 Hasil perhitungan bobot baru V_{ij} Anak 2 Iterasi 1

V_{ij}	z1	z2	z3	z4	z5
X1	0.261	-0.049	0.046	-0.214	0.268
X2	0.102	-0.195	-0.123	-0.107	0.211
X3	0.277	0.145	0.174	-0.267	-0.052
X4	0.048	-0.194	-0.296	0.160	0.057
X5	-0.293	0.144	0.162	0.106	-0.197
X6	-0.055	-0.242	0.232	-0.213	0.103
X7	0.166	0.145	-0.058	-0.160	-0.206
X8	-0.055	-0.242	-0.058	-0.107	0.206
X9	0.055	-0.242	-0.058	0.213	-0.103
X10	0.206	0.047	0.220	0.266	0.061
X11	0.222	0.145	0.290	-0.160	0.052
X12	0.055	0.242	0.290	-0.107	0.052

Table 4.29 Hasil Perhitungan bobot baru V_{ij} Anak 2 Iterasi 1

V_{ij}	z_1	z_2	z_3	z_4	z_5
X13	-0.055	0.145	-0.058	0.053	0.206
X14	-0.055	0.194	-0.232	0.107	0.155
X15	-0.111	-0.145	0.116	-0.160	-0.206
X16	-0.277	-0.145	-0.174	-0.107	-0.206
X17	-0.222	-0.145	-0.058	0.213	0.258
X18	-0.222	-0.194	0.116	-0.267	-0.258
X19	-0.055	0.145	-0.116	0.107	-0.155
Bias	0.074	-0.431	-0.492	-0.340	0.130

10. Menghitung bobot baru antara *Hidden Layer* dan *output layer* (W_{ij}). W_{ij} dihitung dengan menjumlahkan bobot lama pada tabel 4.19 dengan koreksi bobot ΔW_{ij} pada tabel 4.25 seperti persamaan 2-17.

Perhitungan W_{ij} pertama:

$$\begin{aligned}
 W_{11} &= W_{1i(\text{lama})} + \Delta W_{11} \\
 &= (-0.561) + 0.025 \\
 &= (-0.536)
 \end{aligned}$$

Perhitungan dilakukan hingga W_{53} didapatkan, sehingga didapatkan nilai bobot baru secara keseluruhan pada tabel 4.30

Table 4.30 Hasil perhitungan bobot baru W_{ij} Anak2 Iterasi 1

W_{ij}	y_1	y_2	y_3
z_1	-0.536	-0.065	-0.145
z_2	0.386	-0.554	0.359
z_3	-0.431	-0.431	0.704
z_4	0.158	-0.194	0.191
z_5	0.287	0.405	-0.316
Bias	-0.262	-0.201	0.504

Selanjutnya akan melakukan perhitungan yang sama seperti diatas dengan menggunakan data ke-3.

Anak3 ,ID 3

Pada data Anak3, ID 3 perhitungan akan dilakukan dengan menggunakan hasil bobot baru pada tabel 4.29 dan tabel 4.30 yang di dapat dari perhitungan data pertama.

1. Menghitung neuron yang masuk ke *Hidden Layer* dari *input layer* (Z_{in_j}) dengan persamaan (2-8). Tiap nilai input (X_1) Anak 3 dengan ID 3 pada tabel 4.1 dikalikan dengan bobot (V_{ij}) pada tabel 4.29 dan kemudian dijumlahkan dengan bobot biasnya.

Perhitungan *input* pada *Hidden Layer* pertama (Z_{in_1}):

$$\begin{aligned} Z_{in_1} &= \text{bobot bias}(V_{71}) + (X_1 * V_{11}) + (X_2 * V_{21}) + (X_3 * V_{31}) + (X_4 * V_{41}) + (X_5 * V_{51}) + \\ & (X_6 * V_{61}) + (X_7 * V_{71}) + (X_8 * V_{81}) + (X_9 * V_{91}) + (X_{10} * V_{101}) + (X_{11} * V_{111}) + \\ & (X_{12} * V_{121}) + (X_{13} * V_{131}) + (X_{14} * V_{141}) + (X_{15} * V_{151}) + (X_{16} * V_{161}) + (X_{17} * V_{171}) + \\ & (X_{18} * V_{181}) + (X_{19} * V_{191}) \\ &= 0.074 + (1 * 0.261) + (1 * 0.102) + (0 * 0.277) + (0 * 0.048) + (1 * (-0.293)) + \\ & (0 * (-0.055)) + (1 * 0.166) + (1 * (-0.055)) + (0 * 0.55) + (0 * 0.206) + \\ & (0 * 0.222) + (0 * 0.055) + (0 * (-0.055)) + (0 * (-0.055)) + (0 * (-0.111)) + \\ & (0 * (-0.277)) + (0 * (-0.222)) + (0 * (-0.222)) + (0 * (-0.055)) \\ &= 0.25 \end{aligned}$$

Perhitungan diatas dilakukan hingga Z_{in_5} selesai di hitung, sehingga didapatkan hasil perhitungan seluruh Z_{in} pada tabel 4.31 dibawah ini :

Table 4.31 Hasil perhitungan Z_{in} Anak3 Iterasi 1

Z_{in_1}	Z_{in_2}	Z_{in_3}	Z_{in_4}	Z_{in_5}
0.25	-0.63	-0.52	-0.82	0.41

2. Menghitung neuron yang keluar dari *Hidden Layer* menuju *output layer* (Z_j). Z_j dihitung dengan menggunakan fungsi aktivasi sigmoid biner dengan persamaan (2-9).

Perhitungan *output* dari *Hidden Layer* pertama Z_1 :

$$Z_1 = \frac{1}{(1+e^{-Z_{in_1}})} = \frac{1}{(1+e^{-0.25})} = 0.5633$$

Perhitungan di lakukan hingga Z_5 . tabel 4.32 merupakan hasil perhitungan Z_j

Table 4.32 Hasil perhitungan Z_j Anak3 Iterasi 1

Z_1	Z_2	Z_3	Z_4	Z_5
0.5633	0.3481	0.3719	0.3055	0.6013

3. Menghitung neuron yang masuk ke *output layer* dari *Hidden Layer* Y_{in} . Y_{in} dihitung dengan menggunakan persamaan (2-10). Tiap neuron yang keluar dari *Hidden Layer* (Z_j) pada tabel 4.32 dikalikan dengan bobot antara *Hidden Layer* dan *output layer* (W_{ij}) pada tabel 4.30 dan kemudian dijumlahkan dengan bobot biasnya

Perhitungan Y_{in} :

$$\begin{aligned} Y_{in_1} &= \text{bobot bias}(W_{61}) + (Z_1 * W_{11}) + (Z_2 * W_{21}) + (Z_3 * W_{31}) + (Z_4 * W_{41}) + (Z_5 * W_{51}) \\ &= (-0.262) + (0.5633 * (-0.536)) + (0.3481 * 0.386) + (0.3719 * (-0.431)) + (0.3055 * 0.158) + (0.6013 * 0.287) \\ &= (-0.369) \end{aligned}$$

Perhitungan dilakukan terus hingga Y_{in_3} didapatkan pada tabel 4.33 dibawah ini :

Table 4.33 Hasil Perhitungan Y_{in} Anak3 Iterasi 1

Y_{in_1}	Y_{in_2}	Y_{in_3}
-0.369	-0.407	0.678

4. Menghitung neuron keluaran dari *output layer* (Y_{out}). Y_{out} dihitung dengan menggunakan fungsi aktivasi *sigmoid biner* dengan persamaan (2-11).

Perhitungan Y_{out_1} :

$$\begin{aligned} Y_{out_1} &= 1 / (1 + e^{-Y_{in_j}}) \\ &= 1 / (1 + e^{-(-0.369)}) \\ &= 0.409 \end{aligned}$$

Perhitungan dilakukan terus hingga Y_{out_3} didapatkan terlihat pada tabel 4.34 dibawah ini:

Table 4.34 Hasil Perhitungan Y_{out} Anak3 Iterasi 1

Y_{out_1}	Y_{out_2}	Y_{out_3}
0.409	0.400	0.663

5. Menghitung nilai *error* antara *Hidden Layer* dan *input layer* (δ_k). δ_k dihitung menggunakan perkalian antara selisih target di table 4.1 dan keluaran (Y_{out}) seperti pada persamaan (2-12).

Perhitungan δ_k pada Anak 3, ID 3:

$$\begin{aligned} \delta_{k_1} &= (\text{target} - Y_{out_1}) * (1 - Y_{out_1}) * Y_{out_1} \\ &= (1 - 0.409) * (1 - 0.409) * 0.409 \\ &= 0.143 \end{aligned}$$

Perhitungan di lakukan hingga δ_{k_3} didapatkan. Hasil δ_k dapat dilihat pada tabel 4.35 dibawah ini:

Table 4.35 nilai δ_k Anak3 Iterasi 1

δ_1	δ_2	δ_3
0.143	0.144	0.075

6. Menghitung koreksi bobot antara *Hidden Layer* dan *output layer* (ΔW_{ij}). ΔW_{ij} dihitung dengan mengalikan nilai α (*learning rate*) dengan nilai δ_k masing – masing node dan Z_j pada tabel 4.32 seperti persamaan (2-13).

Perhitungan ΔW_{ij} :

$$\begin{aligned}\Delta W_{11} &= \alpha * \delta_{k_1} * Z_1 \\ &= 0.3 * 0.143 * 0.5633 \\ &= 0.024\end{aligned}$$

Perhitungan dilakukan hingga kelima koreksi perubahan bobot pada setiap node *output* didapatkan. Sedangkan untuk menghitung nilai perubahan bobot dari bias hanya mengalikan α (*learning rate*) dengan nilai *error* (δ_k), seperti dibawah ini:

$$\begin{aligned}\Delta W_{bias_1} &= \alpha * \delta_{k_1} \\ &= 0.3 * 0.143 \\ &= 0.043\end{aligned}$$

Perhitungan juga dilakukan hingga perubahab bobot biar untuk ketiga node *output* didapatkan. Tabel 4.36 merupakan hasil perhitungan perubahan bobot antara *Hidden Layer* dan *output layer* (ΔW_{ij} & ΔW_{bias_j}).

Table 4.36 Hasil perhitungan koreksi bobot ΔW_{ij} Anak3 Iterasi 1

ΔW_{ij}	δ_1	δ_2	δ_3
Z1	0.024	0.024	0.013
Z2	0.015	0.015	0.008
Z3	0.016	0.016	0.008
Z4	0.013	0.013	0.007
Z5	0.026	0.026	0.014
Bias	0.043	0.043	0.023

7. Menghitung nilai *error* antara *Hidden Layer* dan *input layer* (δ_j) yang akan digunakan untuk menghitung perubahan bobot. δ_j dihitung dengan mengalikan δ_{in_j} , bobot awal antara *Hidden Layer* dan *output layer*, serta Z_j dengan menggunakan persamaan (2-15).

menghitung δ_{in_j} dengan menjumlahkan hasil perkalian dari nilai *error* (δ_k) dikali dengan bobot awal antara *Hidden Layer* dan *output layer* pada tabel 4.35.

Perhitungan δ_{in_j} :

$$\begin{aligned}\delta_{in_1} &= (\delta_{k_1} * W_{11}) + (\delta_{k_2} * W_{12}) + (\delta_{k_3} * W_{13}) \\ &= (0.143 * (-0.536)) + (0.144 * (-0.065)) + (0.075 * (-0.145)) \\ &= (-0.097)\end{aligned}$$

Perhitungan dilakukan hingga δ_{in_5} didapatkan seperti pada tabel 4.37 dibawah ini:

Table 4.37 Hasil perhitungan δ_{in_j} Anak3 Iterasi 1

δ_{in1}	δ_{in2}	δ_{in3}	δ_{in4}	δ_{in5}
-0.097	0.002	-0.071	0.009	0.076

Perhitungan δ_j :

$$\begin{aligned}\delta_1 &= \delta_{in_1} * (1 - Z_1) * Z_1 \\ &= (-0.097) * (1 - 0.5633) * 0.5633 \\ &= (-0.024)\end{aligned}$$

Perhitungan dilakukan hingga δ_5 didapatkan seperti pada tabel 4.38 dibawah ini:

Table 4.38 Hasil Perhitungan δ_j Anak3 Iterasi 1

δ_1	δ_2	δ_3	δ_4	δ_5
-0.024	0.001	-0.017	0.002	0.018

- Menghitung koreksi bobot antara *Hidden Layer* dan *input layer* (ΔV_{ij}). ΔV_{ij} dihitung dengan mengalikan α (*learning rate*) dengan δ_j ditabel 4.16 dan bobot awal X_i (V_{ij}) di tabel 4.29 seperti persamaan (2-18).

Perhitungan ΔV_{ij} :

$$\begin{aligned}\Delta V_{11} &= \alpha * \delta_{j1} * X_{11} \\ &= 0.3 * (-0.024) * 1 \\ &= (-0.0071)\end{aligned}$$

Perhitungan dilakukan hingga semua koreksi bobot antara *Hidden Layer* dan *input layer* didapatkan. Untuk menghitung koreksi bobot bias dengan mengalikan α (*learning rate*) dengan δ_j .

Perhitungan ΔV_{bias_j} :

$$\begin{aligned}\Delta V_{bias1} &= \alpha * \delta_1 \\ &= 0.3 * (-0.024) \\ &= (-0.0071)\end{aligned}$$

Perhitungan dilakukan hingga ΔV_{bias5} didapatkan seperti pada tabel 4.39 dibawah ini:

Table 4.39 Hasil Perhitungan koreksi bobot ΔV_{ij} & bias Anak3 Iterasi 1

ΔV_{ij}	δ_1	δ_2	δ_3	δ_4	δ_5
X1	-0.0071	0.0002	-0.0050	0.0006	0.0054
X2	-0.0071	0.0002	-0.0050	0.0006	0.0054
X3	0.0000	0.0000	0.0000	0.0000	0.0000
X4	0.0000	0.0000	0.0000	0.0000	0.0000
X5	-0.0071	0.0002	-0.0050	0.0006	0.0054
X6	0.0000	0.0000	0.0000	0.0000	0.0000
X7	-0.0071	0.0002	-0.0050	0.0006	0.0054
X8	-0.0071	0.0002	-0.0050	0.0006	0.0054
X9	0.0000	0.0000	0.0000	0.0000	0.0000
X10	0.0000	0.0000	0.0000	0.0000	0.0000
X11	0.0000	0.0000	0.0000	0.0000	0.0000
X12	0.0000	0.0000	0.0000	0.0000	0.0000
X13	0.0000	0.0000	0.0000	0.0000	0.0000
X14	0.0000	0.0000	0.0000	0.0000	0.0000
X15	0.0000	0.0000	0.0000	0.0000	0.0000
X16	0.0000	0.0000	0.0000	0.0000	0.0000
X17	0.0000	0.0000	0.0000	0.0000	0.0000
X18	0.0000	0.0000	0.0000	0.0000	0.0000
X19	0.0000	0.0000	0.0000	0.0000	0.0000
Bias	-0.0071	0.0002	-0.0050	0.0006	0.0055

- Menghitung bobot baru antara *input layer* dan *Hidden Layer* (V_{ij}). V_{ij} dihitung dengan menjumlahkan bobot lama pada tabel 4.29 dengan koreksi bobot ΔV_{ij} pada tabel 4.39 seperti persamaan 2-18.

Perhitungan V_{ij} pertama :

$$\begin{aligned}V_{11} &= V_{1i}(\text{lama}) + \Delta V_{11} \\ &= 0.261 + (-0.0071)\end{aligned}$$

$$= 0.254$$

Perhitungan dilakukan hingga $V_{18\ 5}$ didapatkan, sehingga didapat nilai bobot baru secara keseluruhan pada tabel 4.40

Table 4.40 Hasil perhitungan bobot baru V_{ij} Anak 3 Iterasi 1

Vij	z1	z2	z3	z4	z5
X1	0.254	-0.049	0.041	-0.213	0.273
X2	0.095	-0.194	-0.128	-0.106	0.216
X3	0.277	0.145	0.174	-0.267	-0.052
X4	0.048	-0.194	-0.296	0.160	0.057
X5	-0.300	0.145	0.157	0.107	-0.191
X6	-0.055	-0.242	0.232	-0.213	0.103
X7	0.159	0.146	-0.063	-0.159	-0.201
X8	-0.063	-0.242	-0.063	-0.106	0.212
X9	0.055	-0.242	-0.058	0.213	-0.103
X10	0.206	0.047	0.220	0.266	0.061
X11	0.222	0.145	0.290	-0.160	0.052
X12	0.055	0.242	0.290	-0.107	0.052
X13	-0.055	0.145	-0.058	0.053	0.206
X14	-0.055	0.194	-0.232	0.107	0.155
X15	-0.111	-0.145	0.116	-0.160	-0.206
X16	-0.277	-0.145	-0.174	-0.107	-0.206
X17	-0.222	-0.145	-0.058	0.213	0.258
X18	-0.222	-0.194	0.116	-0.267	-0.258
X19	-0.055	0.145	-0.116	0.107	-0.155
Bias	0.067	-0.431	-0.498	-0.340	0.135

10. Menghitung bobot baru antara *Hidden Layer* dan *output layer* (W_{ij}). W_{ij} dihitung dengan menjumlahkan bobot lama pada tabel 4.30 dengan koreksi bobot ΔW_{ij} pada tabel 4.36 seperti persamaan 2-17.

Perhitungan W_{ij} pertama:

$$\begin{aligned}
 W_{11} &= W_{1i(\text{lama})} + \Delta W_{11} \\
 &= (-0.536) + 0.024 \\
 &= (-0.512)
 \end{aligned}$$

Perhitungan dilakukan hingga W_{53} didapatkan, sehingga didapatkan nilai bobot baru secara keseluruhan pada tabel 4.41

Table 4.41 Hasil perhitungan bobot baru W_{ij} Anak3 Iterasi 1

W_{ij}	y_1	y_2	y_3
z_1	-0.512	-0.041	-0.132
z_2	0.401	-0.539	0.367
z_3	-0.415	-0.415	0.712
z_4	0.171	-0.181	0.198
z_5	0.313	0.431	-0.302
Bias	-0.219	-0.158	0.527

Selanjutnya akan melakukan perhitungan yang sama seperti diatas dengan menggunakan data ke-4.

Anak4 ,ID 4

Pada data Anak4, ID 4 perhitungan akan dilakukan dengan menggunakan hasil bobot baru pada tabel 4.40 dan tabel 4.41 yang di dapat dari perhitungan data ketiga.

1. Menghitung neuron yang masuk ke *Hidden Layer* dari *input layer* (Z_{in_j}) dengan persamaan (2-8). Tiap nilai input (X_1) Anak 34 dengan ID 4 pada tabel 4.1 dikalikan dengan bobot (V_{ij}) pada tabel 4.40 dan kemudian dijumlahkan dengan bobot biasnya.

Perhitungan *input* pada *Hidden Layer* pertama (Z_{in_1}):

$$\begin{aligned}
 Z_{in_1} &= \text{bobot bias}(V_{71})+(X_1*V_{11})+ (X_2*V_{21})+ (X_3*V_{31})+ (X_4*V_{41})+ (X_5*V_{51})+ \\
 & (X_6*V_{61})+ (X_7*V_{71})+ (X_8*V_{81})+ (X_9*V_{91})+ (X_{10}*V_{101})+ (X_{11}*V_{111})+ \\
 & (X_{12}*V_{121})+ (X_{13}*V_{131})+ (X_{14}*V_{141})+ (X_{15}*V_{151})+ (X_{16}*V_{161})+ (X_{17}*V_{171})+ \\
 & (X_{18}*V_{181})+ (X_{19}*V_{191}) \\
 &= 0.067+(0*0.254)+ (0*0.95)+ (0*0.277)+ (1*0.048)+ (0*(-0.300))+ \\
 & (0*(-0.055))+ (0*0.159)+ (0*(-0.063)+ (1*0.55)+ (0*0.206)+ \\
 & (1*0.222)+ (0*0.055)+ (1*(-0.055))+ (0*(-0.055))+ (0*(-0.111))+ \\
 & (0*(-0.277))+ (0*(-0.222))+ (0*(-0.222))+ (0*(-0.055)) \\
 &= 0.34
 \end{aligned}$$

Perhitungan diatas dilakukan hingga Z_{in_5} selesai di hitung, sehingga didapatkan hasil perhitungan seluruh Z_{in} pada tabel 4.42 dibawah ini :

Table 4.42 Hasil perhitungan Z_{in} Anak4 Iterasi 1

Z_{in_1}	Z_{in_2}	Z_{in_3}	Z_{in_4}	Z_{in_5}
0.34	-0.58	-0.62	-0.07	0.35

2. Menghitung neuron yang keluar dari *Hidden Layer* menuju *output layer* (Z_j). Z_j dihitung dengan menggunakan fungsi aktivasi sigmoid biner dengan persamaan (2-9).

Perhitungan *output* dari *Hidden Layer* pertama Z_1 :

$$Z_1 = \frac{1}{(1+e^{Z_{in1}})} = \frac{1}{(1+e^{0.34})} = 0.5832$$

Perhitungan di lakukan hingga Z_5 . tabel 4.43 merupakan hasil perhitungan Z_j

Table 4.43 Hasil perhitungan Z_j Anak4 Iterasi 1

Z1	Z2	Z3	Z4	Z5
0.5832	0.3597	0.3499	0.4818	0.5858

3. Menghitung neuron yang masuk ke *output layer* dari *Hidden Layer* Y_{in} . Y_{in} dihitung dengan menggunakan persamaan (2-10). Tiap neuron yang keluar dari *Hidden Layer* (Z_j) pada tabel 4.43 dikalikan dengan bobot antara *Hidden Layer* dan *output layer* (W_{ij}) pada tabel 4.41 dan kemudian dijumlahkan dengan bobot biasnya

Perhitungan Y_{in} :

$$\begin{aligned} Y_{in_1} &= \text{bobot bias}(W_{61}) + (Z_1*W_{11})+ (Z_2*W_{21})+ (Z_3*W_{31})+ (Z_4*W_{41})+ \\ &\quad (Z_5*W_{51}) \\ &= (-0.219)+(0.5832*(-0.512))+ (0.3597*0.401)+ (0.3499*(-0.415))+ \\ &\quad (0.4818*0.171)+ (0.5858*0.313) \\ &= (-0.253) \end{aligned}$$

Perhitungan dilakukan terus hingga Y_{in_3} didapatkan pada tabel 4.44 dibawah ini :

Table 4.44 Hasil Perhitungan Y_{in} Anak4 Iterasi 1

Y_{in_1}	Y_{in_2}	Y_{in_3}
-0.253	-0.355	0.750

4. Menghitung neuron keluaran dari *output layer* (Y_{out}). Y_{out} dihitung dengan menggunakan fungsi aktivasi *sigmoid biner* dengan persamaan (2-11).

Perhitungan Y_{out_1} :

$$\begin{aligned} Y_{out_1} &= 1/(1+e^{Y_{in_j}}) \\ &= 1/(1+e^{-(-0.253)}) \\ &= 0.437 \end{aligned}$$

Perhitungan dilakukan terus hingga Y_{out_3} didapatkan terlihat pada tabel 4.45 dibawah ini:

Table 4.45 Hasil Perhitungan Y_{out} Anak4 Iterasi 1

Yout_1	Yout_2	Yout_3
0.437	0.412	0.679

5. Menghitung nilai *error* antara *Hidden Layer* dan *input layer* (δ_k). δ_k dihitung menggunakan perkalian antara selisih target di table 4.1 dan keluaran (Y_{out}) seperti pada persamaan (2-12).

Perhitungan δ_k pada Anak 4, ID 4:

$$\begin{aligned}\delta_{k_1} &= (\text{target} - Y_{out_1}) * (1 - Y_{out_1}) * Y_{out_1} \\ &= (1 - 0.437) * (1 - 0.437) * 0.437 \\ &= 0.138\end{aligned}$$

Perhitungan di lakukan hingga δ_{k_3} didapatkan. Hasil δ_k dapat dilihat pada tabel 4.46 dibawah ini:

Table 4.46 nilai δ_k Anak4 Iterasi 1

δ_1	δ_2	δ_3
0.139	0.142	-0.148

6. Menghitung koreksi bobot antara *Hidden Layer* dan *output layer* (ΔW_{ij}). ΔW_{ij} dihitung dengan mengalikan nilai α (*learning rate*) dengan nilai δ_k masing – masing node dan Z_j pada tabel 4.43 seperti persamaan (2-13).

Perhitungan ΔW_{ij} :

$$\begin{aligned}\Delta W_{11} &= \alpha * \delta_{k_1} * Z_1 \\ &= 0.3 * 0.138 * 0.5832 \\ &= 0.024\end{aligned}$$

Perhitungan dilakukan hingga kelima koreksi perubahan bobot pada setiap node *output* didapatkan. Sedangkan untuk menghitung nilai perubahan bobot dari bias hanya mengalikan α (*learning rate*) dengan nilai *error* (δ_k). seperti dibawah ini:

$$\begin{aligned}\Delta W_{bias_1} &= \alpha * \delta_{k_1} \\ &= 0.3 * 0.138 \\ &= 0.042\end{aligned}$$

Perhitungan juga dilakukan hingga perubahab bobot biar untuk ketiga node *output* didapatkan. Tabel 4.47 merupakan hasil perhitungan perubahan bobot antara *Hidden Layer* dan *output layer* (ΔW_{ij} & ΔW_{bias_j}).

Table 4.47 Hasil perhitungan koreksi bobot ΔW_{ij} Anak4 Iterasi 1

ΔW_{ij}	δ_1	δ_2	δ_3
Z1	0.024	0.025	-0.026
Z2	0.015	0.015	-0.016
Z3	0.015	0.015	-0.016
Z4	0.020	0.021	-0.021
Z5	0.024	0.025	-0.026
Bias	0.042	0.043	-0.044

7. Menghitung nilai *error* antara *Hidden Layer* dan *input layer* (δ_j) yang akan digunakan untuk menghitung perubahan bobot. δ_j dihitung dengan mengalikan δ_{in_j} , bobot awal antara *Hidden Layer* dan *output layer*, serta Z_j dengan menggunakan persamaan (2-15).

menghitung δ_{in_j} dengan menjumlahkan hasil perkalian dari nilai *error* (δ_k) dikali dengan bobot awal antara *Hidden Layer* dan *output layer* pada tabel 4.41.

Perhitungan δ_{in_j} :

$$\begin{aligned}\delta_{in_1} &= (\delta_{k_1} * W_{11}) + (\delta_{k_2} * W_{12}) + (\delta_{k_3} * W_{13}) \\ &= (0.138 * (-0.512)) + (0.142 * (-0.041)) + (0.148 * (-0.132)) \\ &= (-0.057)\end{aligned}$$

Perhitungan dilakukan hingga δ_{in_5} didapatkan seperti pada tabel 4.48 dibawah ini:

Table 4.48 Hasil perhitungan δ_{in_j} Anak4 Iterasi 1

δ_{in1}	δ_{in2}	δ_{in3}	δ_{in4}	δ_{in5}
-0.057	-0.075	-0.222	-0.031	0.149

Perhitungan δ_j :

$$\begin{aligned}\delta_1 &= \delta_{in_1} * (1 - Z_1) * Z_1 \\ &= (-0.057) * (1 - 0.5832) * 0.5832 \\ &= (-0.014)\end{aligned}$$

Perhitungan dilakukan hingga δ_5 didapatkan seperti pada tabel 4.49 dibawah ini:

Table 4.49 Hasil Perhitungan δ_j Anak4 Iterasi 1

δ_1	δ_2	δ_3	δ_4	δ_5
-0.014	-0.017	-0.051	-0.008	0.036

8. Menghitung koreksi bobot antara *Hidden Layer* dan *input layer* (ΔV_{ij}). ΔV_{ij} dihitung dengan mengalikan α (*learning rate*) dengan δ_j ditabel 4.49 dan bobot awal X_i (V_{ij}) di tabel 4.40 seperti persamaan (2-18).

Perhitungan ΔV_{ij} :

$$\begin{aligned}\Delta V_{11} &= \alpha * \delta_{j1} * X_{11} \\ &= 0.3 * (-0.014) * 0 \\ &= (0.0000)\end{aligned}$$

Perhitungan dilakukan hingga semua koreksi bobot antara *Hidden Layer* dan *input layer* didapatkan. Untuk menghitung koreksi bobot bias dengan mengalikan α (*learning rate*) dengan δ_j .

Perhitungan ΔV_{bias_j} :

$$\begin{aligned}\Delta V_{bias1} &= \alpha * \delta_1 \\ &= 0.3 * (-0.014) \\ &= (-0.0042)\end{aligned}$$

Perhitungan dilakukan hingga ΔV_{bias5} didapatkan seperti pada tabel 4.50 dibawah ini:

Table 4.50 Hasil Perhitungan koreksi bobot ΔV_{ij} & bias Anak4 Iterasi 1

ΔV_{ij}	δ_1	δ_2	δ_3	δ_4	δ_5
X1	0.0000	0.0000	0.0000	0.0000	0.0000
X2	0.0000	0.0000	0.0000	0.0000	0.0000
X3	0.0000	0.0000	0.0000	0.0000	0.0000
X4	-0.0042	-0.0052	-0.0152	-0.0024	0.0109
X5	0.0000	0.0000	0.0000	0.0000	0.0000
X6	0.0000	0.0000	0.0000	0.0000	0.0000
X7	0.0000	0.0000	0.0000	0.0000	0.0000
X8	0.0000	0.0000	0.0000	0.0000	0.0000
X9	-0.0042	-0.0052	-0.0152	-0.0024	0.0109
X10	0.0000	0.0000	0.0000	0.0000	0.0000
X11	-0.0042	-0.0052	-0.0152	-0.0024	0.0109
X12	0.0000	0.0000	0.0000	0.0000	0.0000

Table 4.50 Hasil Perhitungan koreksi bobot ΔV_{ij} & bias Anak4 Iterasi 1

ΔV_{ij}	δ_1	δ_2	δ_3	δ_4	δ_5
X13	-0.0042	-0.0052	-0.0152	-0.0024	0.0109
X14	0.0000	0.0000	0.0000	0.0000	0.0000
X15	0.0000	0.0000	0.0000	0.0000	0.0000
X16	0.0000	0.0000	0.0000	0.0000	0.0000
X17	0.0000	0.0000	0.0000	0.0000	0.0000
X18	0.0000	0.0000	0.0000	0.0000	0.0000
X19	0.0000	0.0000	0.0000	0.0000	0.0000
Bias	-0.0042	-0.0052	-0.0152	-0.0024	0.0109

9. Menghitung bobot baru antara *input layer* dan *Hidden Layer* (V_{ij}). V_{ij} dihitung dengan menjumlahkan bobot lama pada tabel 4.40 dengan koreksi bobot ΔV_{ij} pada tabel 4.50 seperti persamaan 2-18.

Perhitungan V_{ij} pertama :

$$\begin{aligned}
 V_{11} &= V_{1i} \text{ (lama)} + \Delta V_{11} \\
 &= 0.254 + (0.0000) \\
 &= 0.254
 \end{aligned}$$

Perhitungan dilakukan hingga $V_{18\ 5}$ didapatkan, sehingga didapat nilai bobot baru secara keseluruhan pada tabel 4.51

Table 4.51 Hasil perhitungan bobot baru V_{ij} Anak 4 Iterasi 1

V_{ij}	z1	z2	z3	z4	z5
X1	0.254	-0.049	0.041	-0.213	0.273
X2	0.095	-0.194	-0.128	-0.106	0.216
X3	0.277	0.145	0.174	-0.267	-0.052
X4	0.043	-0.199	-0.312	0.158	0.068
X5	-0.300	0.145	0.157	0.107	-0.191
X6	-0.055	-0.242	0.232	-0.213	0.103
X7	0.159	0.146	-0.063	-0.159	-0.201
X8	-0.063	-0.242	-0.063	-0.106	0.212
X9	0.051	-0.248	-0.073	0.211	-0.092
X10	0.206	0.047	0.220	0.266	0.061
X11	0.218	0.140	0.275	-0.162	0.062
X12	0.055	0.242	0.290	-0.107	0.052

Table 4.51 Hasil perhitungan bobot baru V_{ij} Anak 4 Iterasi 1

X13	-0.060	0.140	-0.073	0.051	0.217
X14	-0.055	0.194	-0.232	0.107	0.155
X15	-0.111	-0.145	0.116	-0.160	-0.206
X16	-0.277	-0.145	-0.174	-0.107	-0.206
X17	-0.222	-0.145	-0.058	0.213	0.258
X18	-0.222	-0.194	0.116	-0.267	-0.258
X19	-0.055	0.145	-0.116	0.107	-0.155
Bias	0.063	-0.436	-0.513	-0.342	0.146

10. Menghitung bobot baru antara *Hidden Layer* dan *output layer* (W_{ij}). W_{ij} dihitung dengan menjumlahkan bobot lama pada tabel 4.41 dengan koreksi bobot ΔW_{ij} pada tabel 4.47 seperti persamaan 2-17.

Perhitungan W_{ij} pertama:

$$\begin{aligned}
 W_{11} &= W_{1i(\text{lama})} + \Delta W_{11} \\
 &= (-0.512) + 0.024 \\
 &= (-0.487)
 \end{aligned}$$

Perhitungan dilakukan hingga W_{53} didapatkan, sehingga didapatkan nilai bobot baru secara keseluruhan pada tabel 4.52

Table 4.52 Hasil perhitungan bobot baru W_{ij} Anak4 Iterasi 1

W_{ij}	y1	y2	y3
z1	-0.487	-0.016	-0.158
z2	0.416	-0.524	0.351
z3	-0.401	-0.400	0.697
z4	0.191	-0.160	0.177
z5	0.337	0.456	-0.328
Bias	-0.178	-0.115	0.483

Selanjutnya akan melakukan perhitungan yang sama seperti diatas dengan menggunakan data ke-5.

Anak5 ,ID 5

Pada data Anak5, ID 5 perhitungan akan dilakukan dengan menggunakan hasil bobot baru pada tabel 4.51 dan tabel 4.52 yang di dapat dari perhitungan data ketiga.

1. Menghitung neuron yang masuk ke *Hidden Layer* dari *input layer* (Z_{in_j}) dengan persamaan (2-8). Tiap nilai input (X_1) Anak 5 dengan ID 5 pada tabel 4.1 dikalikan dengan bobot (V_{ij}) pada tabel 4.51 dan kemudian dijumlahkan dengan bobot biasnya.

Perhitungan *input* pada *Hidden Layer* pertama (Z_{in_1}):

$$\begin{aligned} Z_{in_1} &= \text{bobot bias}(V_{71}) + (X_1 * V_{11}) + (X_2 * V_{21}) + (X_3 * V_{31}) + (X_4 * V_{41}) + (X_5 * V_{51}) + \\ &\quad (X_6 * V_{61}) + (X_7 * V_{71}) + (X_8 * V_{81}) + (X_9 * V_{91}) + (X_{10} * V_{101}) + (X_{11} * V_{111}) + \\ &\quad (X_{12} * V_{121}) + (X_{13} * V_{131}) + (X_{14} * V_{141}) + (X_{15} * V_{151}) + (X_{16} * V_{161}) + (X_{17} * V_{171}) + \\ &\quad (X_{18} * V_{181}) + (X_{19} * V_{191}) \\ &= 0.063 + (0 * 0.254) + (0 * 0.095) + (0 * 0.277) + (1 * 0.043) + (0 * (-0.300)) + \\ &\quad (0 * (-0.055)) + (0 * 0.159) + (0 * (-0.063)) + (0 * 0.51) + (0 * 0.206) + \\ &\quad (1 * 0.218) + (1 * 0.055) + (0 * (-0.060)) + (0 * (-0.055)) + (1 * (-0.111)) + \\ &\quad (0 * (-0.277)) + (0 * (-0.222)) + (0 * (-0.222)) + (0 * (-0.055)) \\ &= 0.27 \end{aligned}$$

Perhitungan diatas dilakukan hingga Z_{in_5} selesai di hitung, sehingga didapatkan hasil perhitungan seluruh Z_{in} pada tabel 4.53 dibawah ini :

Table 4.53 Hasil perhitungan Z_{in} Anak5 Iterasi 1

Z_{in_1}	Z_{in_2}	Z_{in_3}	Z_{in_4}	Z_{in_5}
0.27	-0.40	-0.14	-0.61	0.12

2. Menghitung neuron yang keluar dari *Hidden Layer* menuju *output layer* (Z_j). Z_j dihitung dengan menggunakan fungsi aktivasi sigmoid biner dengan persamaan (2-9).

Perhitungan *output* dari *Hidden Layer* pertama Z_1 :

$$Z_1 = \frac{1}{(1+e^{Z_{in_1}})} = \frac{1}{(1+e^{0.27})} = 0.5666$$

Perhitungan di lakukan hingga Z_5 . tabel 4.54 merupakan hasil perhitungan Z_j

Table 4.54 Hasil perhitungan Z_j Anak5 Iterasi 1

Z_1	Z_2	Z_3	Z_4	Z_5
0.5666	0.4017	0.4645	0.3513	0.5303

3. Menghitung neuron yang masuk ke *output layer* dari *Hidden Layer* Y_{in} . Y_{in} dihitung dengan menggunakan persamaan (2-10). Tiap neuron yang keluar dari *Hidden Layer* (Z_j) pada tabel 4.54 dikalikan dengan bobot antara *Hidden Layer* dan *output layer* (W_{ij}) pada tabel 4.52 dan kemudian dijumlahkan dengan bobot biasnya

Perhitungan Y_{in} :

$$\begin{aligned}
 Y_{in_1} &= \text{bobot bias}(W_{61}) + (Z_1 * W_{11}) + (Z_2 * W_{21}) + (Z_3 * W_{31}) + (Z_4 * W_{41}) + (Z_5 * W_{51}) \\
 &= (-0.178) + (0.5666 * (-0.487)) + (0.4017 * 0.416) + (0.4645 * (-0.401)) + (0.3513 * 0.191) + (0.5303 * 0.337) \\
 &= (-0.227)
 \end{aligned}$$

Perhitungan dilakukan terus hingga Y_{in_3} didapatkan pada tabel 4.55 dibawah ini :

Table 4.55 Hasil Perhitungan Y_{in} Anak5 Iterasi 1

Y_{in_1}	Y_{in_2}	Y_{in_3}
-0.227	-0.335	0.746

4. Menghitung neuron keluaran dari *output layer* (Y_{out}). Y_{out} dihitung dengan menggunakan fungsi aktivasi *sigmoid biner* dengan persamaan (2-11).

Perhitungan Y_{out_1} :

$$\begin{aligned}
 Y_{out_1} &= 1 / (1 + e^{Y_{in_j}}) \\
 &= 1 / (1 + e^{-(-0.227)}) \\
 &= 0.444
 \end{aligned}$$

Perhitungan dilakukan terus hingga Y_{out_3} didapatkan terlihat pada tabel 4.56 dibawah ini:

Table 4.56 Hasil Perhitungan Y_{out} Anak5 Iterasi 1

Y_{out_1}	Y_{out_2}	Y_{out_3}
0.444	0.417	0.678

5. Menghitung nilai *error* antara *Hidden Layer* dan *input layer* (δ_k). δ_k dihitung menggunakan perkalian antara selisih target di table 4.1 dan keluaran (Y_{out}) seperti pada persamaan (2-12).

Perhitungan δ_k pada Anak 5, ID 5:

$$\begin{aligned}
 \delta_{k_1} &= (\text{target} - Y_{out_1}) * (1 - Y_{out_1}) * Y_{out_1} \\
 &= (1 - 0.444) * (1 - 0.444) * 0.444 \\
 &= 0.137
 \end{aligned}$$

Perhitungan di lakukan hingga δ_{k_3} didapatkan. Hasil δ_k dapat dilihat pada tabel 4.57 dibawah ini:

Table 4.57 nilai δ_k Anak5 Iterasi 1

δ_1	δ_2	δ_3
0.137	0.142	-0.148

6. Menghitung koreksi bobot antara *Hidden Layer* dan *output layer* (ΔW_{ij}). ΔW_{ij} dihitung dengan mengalikan nilai α (*learning rate*) dengan nilai δ_k masing – masing node dan Z_j pada tabel 4.54 seperti persamaan (2-13).

Perhitungan ΔW_{ij} :

$$\begin{aligned}\Delta W_{11} &= \alpha * \delta_{k_1} * Z_1 \\ &= 0.3 * 0.137 * 0.5666 \\ &= 0.023\end{aligned}$$

Perhitungan dilakukan hingga kelima koreksi perubahan bobot pada setiap node *output* didapatkan. Sedangkan untuk menghitung nilai perubahan bobot dari bias hanya mengalikan α (*learning rate*) dengan nilai *error* (δ_k), seperti dibawah ini:

$$\begin{aligned}\Delta W_{bias_1} &= \alpha * \delta_{k_1} \\ &= 0.3 * 0.138 \\ &= 0.041\end{aligned}$$

Perhitungan juga dilakukan hingga perubahab bobot biar untuk ketiga node *output* didapatkan. Tabel 4.58 merupakan hasil perhitungan perubahan bobot antara *Hidden Layer* dan *output layer* (ΔW_{ij} & ΔW_{bias_j}).

Table 4.58 Hasil perhitungan koreksi bobot ΔW_{ij} Anak5 Iterasi 1

ΔW_{ij}	δ_1	δ_2	δ_3
Z1	0.023	0.024	-0.025
Z2	0.017	0.017	-0.018
Z3	0.019	0.020	-0.021
Z4	0.014	0.015	-0.016
Z5	0.022	0.023	-0.024
Bias	0.041	0.043	-0.044

7. Menghitung nilai *error* antara *Hidden Layer* dan *input layer* (δ_j) yang akan digunakan untuk menghitung perubahan bobot. δ_j dihitung dengan mengalikan δ_{in_j} , bobot awal antara *Hidden Layer* dan *output layer*, serta Z_j dengan menggunakan persamaan (2-15).

menghitung δ_{in_j} dengan menjumlahkan hasil perkalian dari nilai *error* (δ_k) dikali dengan bobot awal antara *Hidden Layer* dan *output layer* pada tabel 4.52.

Perhitungan δ_{in_j} :

$$\begin{aligned}\delta_{in_1} &= (\delta_{k_1} * W_{11}) + (\delta_{k_2} * W_{12}) + (\delta_{k_3} * W_{13}) \\ &= (0.137 * (-0.487)) + (0.142 * (-0.016)) + (0.148 * (-0.158)) \\ &= (-0.046)\end{aligned}$$

Perhitungan dilakukan hingga δ_{in_5} didapatkan seperti pada tabel 4.59 dibawah ini:

Table 4.59 Hasil perhitungan δ_{in_j} Anak5 Iterasi 1

δ_{in1}	δ_{in2}	δ_{in3}	δ_{in4}	δ_{in5}
-0.046	-0.069	-0.215	-0.023	0.160

Perhitungan δ_j :

$$\begin{aligned}\delta_1 &= \delta_{in_1} * (1 - Z_1) * Z_1 \\ &= (-0.046) * (1 - 0.5666) * 0.5666 \\ &= (-0.011)\end{aligned}$$

Perhitungan dilakukan hingga δ_5 didapatkan seperti pada tabel 4.60 dibawah ini:

Table 4.60 Hasil Perhitungan δ_j Anak5 Iterasi 1

δ_1	δ_2	δ_3	δ_4	δ_5
-0.011	-0.017	-0.053	-0.005	0.040

8. Menghitung koreksi bobot antara *Hidden Layer* dan *input layer* (ΔV_{ij}). ΔV_{ij} dihitung dengan mengalikan α (*learning rate*) dengan δ_j ditabel 4.60 dan bobot awal X_i (V_{ij}) di tabel 4.51 seperti persamaan (2-18).

Perhitungan ΔV_{ij} :

$$\begin{aligned}\Delta V_{11} &= \alpha * \delta_{j1} * X_{11} \\ &= 0.3 * (-0.011) * 0 \\ &= (0.0000)\end{aligned}$$

Perhitungan dilakukan hingga semua koreksi bobot antara *Hidden Layer* dan *input layer* didapatkan. Untuk menghitung koreksi bobot bias dengan mengalikan α (*learning rate*) dengan δ_j .

Perhitungan ΔV_{bias_j} :

$$\begin{aligned}\Delta V_{bias1} &= \alpha * \delta_1 \\ &= 0.3 * (-0.011) \\ &= (-0.0034)\end{aligned}$$

Perhitungan dilakukan hingga ΔV_{bias5} didapatkan seperti pada tabel 4.61 dibawah ini:

Table 4.61 Hasil Perhitungan koreksi bobot ΔV_{ij} & bias Anak5 Iterasi 1

ΔV_{ij}	δ_1	δ_2	δ_3	δ_4	δ_5
X1	0.0000	0.0000	0.0000	0.0000	0.0000
X2	0.0000	0.0000	0.0000	0.0000	0.0000
X3	0.0000	0.0000	0.0000	0.0000	0.0000
X4	-0.0034	-0.0050	-0.0160	-0.0015	0.0119
X5	0.0000	0.0000	0.0000	0.0000	0.0000
X6	0.0000	0.0000	0.0000	0.0000	0.0000
X7	0.0000	0.0000	0.0000	0.0000	0.0000
X8	0.0000	0.0000	0.0000	0.0000	0.0000
X9	0.0000	0.0000	0.0000	0.0000	0.0000
X10	0.0000	0.0000	0.0000	0.0000	0.0000
X11	-0.0034	-0.0050	-0.0160	-0.0015	0.0119
X12	-0.0034	-0.0050	-0.0160	-0.0015	0.0119
X13	0.0000	0.0000	0.0000	0.0000	0.0000
X14	0.0000	0.0000	0.0000	0.0000	0.0000
X15	-0.0034	-0.0050	-0.0160	-0.0015	0.0119
X16	0.0000	0.0000	0.0000	0.0000	0.0000
X17	0.0000	0.0000	0.0000	0.0000	0.0000
X18	0.0000	0.0000	0.0000	0.0000	0.0000
X19	0.0000	0.0000	0.0000	0.0000	0.0000
Bias	-0.0034	-0.0050	-0.0160	-0.0015	0.0119

- Menghitung bobot baru antara *input layer* dan *Hidden Layer* (V_{ij}). V_{ij} dihitung dengan menjumlahkan bobot lama pada tabel 4.51 dengan koreksi bobot ΔV_{ij} pada tabel 4.61 seperti persamaan 2-18.

Perhitungan V_{ij} pertama :

$$\begin{aligned}V_{11} &= V_{1i} \text{ (lama)} + \Delta V_{11} \\ &= 0.254 + (0.0000)\end{aligned}$$

$$= 0.254$$

Perhitungan dilakukan hingga V_{18} 5 didapatkan, sehingga didapat nilai bobot baru secara keseluruhan pada tabel 4.62

Table 4.62 Hasil perhitungan bobot baru V_{ij} Anak 5 Iterasi 1

V_{ij}	z_1	z_2	z_3	z_4	z_5
X1	0.254	-0.049	0.041	-0.213	0.273
X2	0.095	-0.194	-0.128	-0.106	0.216
X3	0.277	0.145	0.174	-0.267	-0.052
X4	0.040	-0.204	-0.328	0.156	0.080
X5	-0.300	0.145	0.157	0.107	-0.191
X6	-0.055	-0.242	0.232	-0.213	0.103
X7	0.159	0.146	-0.063	-0.159	-0.201
X8	-0.063	-0.242	-0.063	-0.106	0.212
X9	0.051	-0.248	-0.073	0.211	-0.092
X10	0.206	0.047	0.220	0.266	0.061
X11	0.214	0.135	0.259	-0.164	0.074
X12	0.052	0.237	0.274	-0.108	0.064
X13	-0.060	0.140	-0.073	0.051	0.217
X14	-0.055	0.194	-0.232	0.107	0.155
X15	-0.114	-0.150	0.100	-0.162	-0.194
X16	-0.277	-0.145	-0.174	-0.107	-0.206
X17	-0.222	-0.145	-0.058	0.213	0.258
X18	-0.222	-0.194	0.116	-0.267	-0.258
X19	-0.055	0.145	-0.116	0.107	-0.155
Bias	0.059	-0.441	-0.529	-0.344	0.158

10. Menghitung bobot baru antara *Hidden Layer* dan *output layer* (W_{ij}). W_{ij} dihitung dengan menjumlahkan bobot lama pada tabel 4.52 dengan koreksi bobot ΔW_{ij} pada tabel 4.58 seperti persamaan 2-17.

Perhitungan W_{ij} pertama:

$$\begin{aligned}
 W_{11} &= W_{1i(\text{lama})} + \Delta W_{11} \\
 &= (-0.487) + 0.023 \\
 &= (-0.464)
 \end{aligned}$$

Perhitungan dilakukan hingga W_{53} didapatkan, sehingga didapatkan nilai bobot baru secara keseluruhan pada tabel 4.63

Table 4.63 Hasil perhitungan bobot baru W_{ij} Anak5 Iterasi 1

Wij	y1	y2	y3
z1	-0.464	0.008	-0.183
z2	0.433	-0.506	0.333
z3	-0.382	-0.380	0.676
z4	0.206	-0.145	0.161
z5	0.359	0.479	-0.352
Bias	-0.136	-0.073	0.438

Selanjutnya akan melakukan perhitungan yang sama seperti diatas dengan menggunakan data ke-6.

Anak6 ,ID 6

Pada data Anak6, ID 6 perhitungan akan dilakukan dengan menggunakan hasil bobot baru pada tabel 4.62 dan tabel 4.63 yang di dapat dari perhitungan data ketiga.

1. Menghitung neuron yang masuk ke *Hidden Layer* dari *input layer* (Z_{in_j}) dengan persamaan (2-8). Tiap nilai input (X_1) Anak 6 dengan ID 6 pada tabel 4.1 dikalikan dengan bobot (V_{ij}) pada tabel 4.62 dan kemudian dijumlahkan dengan bobot biasnya.

Perhitungan *input* pada *Hidden Layer* pertama (Z_{in_1}):

$$\begin{aligned}
 Z_{in_1} &= \text{bobot bias}(V_{71})+(X_1*V_{11})+ (X_2*V_{21})+ (X_3*V_{31})+ (X_4*V_{41})+ (X_5*V_{51})+ \\
 & (X_6*V_{61})+ (X_7*V_{71})+ (X_8*V_{81})+ (X_9*V_{91})+ (X_{10}*V_{101})+ (X_{11}*V_{111})+ \\
 & (X_{12}*V_{121})+ (X_{13}*V_{131})+ (X_{14}*V_{141})+ (X_{15}*V_{151})+ (X_{16}*V_{161})+ (X_{17}*V_{171})+ \\
 & (X_{18}*V_{181})+ (X_{19}*V_{191}) \\
 &= 0.059+(0*0.254)+ (0*0.095)+ (0*0.277)+ (1*0.040)+ (0*(-0.300))+ \\
 & (0*(-0.055))+ (0*0.159)+ (0*(-0.063)+ (0*0.51)+ (0*0.206)+ \\
 & (1*0.214)+ (1*0.052)+ (0*(-0.060))+ (0*(-0.055))+ (1*(-0.114))+ \\
 & (0*(-0.277))+ (0*(-0.222))+ (0*(-0.222))+ (0*(-0.055)) \\
 &= 0.42
 \end{aligned}$$

Perhitungan diatas dilakukan hingga Z_{in_5} selesai di hitung, sehingga didapatkan hasil perhitungan seluruh Z_{in} pada tabel 4.64 dibawah ini :

Table 4.64 Hasil perhitungan Z_{in} Anak6 Iterasi 1

Zin_1	Zin_2	Zin_3	Zin_4	Zin_5
0.42	-0.52	-0.40	-0.25	0.28

2. Menghitung neuron yang keluar dari *Hidden Layer* menuju *output layer* (Z_j). Z_j dihitung dengan menggunakan fungsi aktivasi sigmoid biner dengan persamaan (2-9).

Perhitungan *output* dari *Hidden Layer* pertama Z_1 :

$$Z_1 = \frac{1}{(1+e^{-Z_{in1}})} = \frac{1}{(1+e^{-0.42})} = 0.6027$$

Perhitungan di lakukan hingga Z_5 . tabel 4.65 merupakan hasil perhitungan Z_j

Table 4.65 Hasil perhitungan Z_j Anak6 Iterasi 1

Z1	Z2	Z3	Z4	Z5
0.6027	0.3727	0.4023	0.4382	0.5703

3. Menghitung neuron yang masuk ke *output layer* dari *Hidden Layer* Y_{in} . Y_{in} dihitung dengan menggunakan persamaan (2-10). Tiap neuron yang keluar dari *Hidden Layer* (Z_j) pada tabel 4.65 dikalikan dengan bobot antara *Hidden Layer* dan *output layer* (W_{ij}) pada tabel 4.62 dan kemudian dijumlahkan dengan bobot biasnya

Perhitungan Y_{in} :

$$\begin{aligned} Y_{in_1} &= \text{bobot bias}(W_{61}) + (Z_1*W_{11})+ (Z_2*W_{21})+ (Z_3*W_{31})+ (Z_4*W_{41})+ \\ &\quad (Z_5*W_{51}) \\ &= (-0.136)+(0.6027*(-0.464))+ (0.3727*0.433)+ (0.4023*(-0.382))+ \\ &\quad (0.4382*0.206)+ (0.5703*0.359) \\ &= (-0.114) \end{aligned}$$

Perhitungan dilakukan terus hingga Y_{in_3} didapatkan pada tabel 4.66 dibawah ini :

Table 4.66 Hasil Perhitungan Y_{in} Anak6 Iterasi 1

Y_{in_1}	Y_{in_2}	Y_{in_3}
-0.114	-0.200	0.594

4. Menghitung neuron keluaran dari *output layer* (Y_{out}). Y_{out} dihitung dengan menggunakan fungsi aktivasi *sigmoid biner* dengan persamaan (2-11).

Perhitungan Y_{out_1} :

$$\begin{aligned} Y_{out_1} &= 1/(1+e^{-Y_{in_j}}) \\ &= 1/(1+e^{-(-0.114)}) \\ &= 0.472 \end{aligned}$$

Perhitungan dilakukan terus hingga Y_{out_3} didapatkan terlihat pada tabel 4.67 dibawah ini:

Table 4.67 Hasil Perhitungan Y_{out} Anak6 Iterasi 1

Yout_1	Yout_2	Yout_3
0.472	0.450	0.644

5. Menghitung nilai *error* antara *Hidden Layer* dan *input layer* (δ_k). δ_k dihitung menggunakan perkalian antara selisih target di table 4.1 dan keluaran (Y_{out}) seperti pada persamaan (2-12).

Perhitungan δ_k pada Anak 6, ID 6:

$$\begin{aligned}\delta_{k_1} &= (\text{target} - Y_{out_1}) * (1 - Y_{out_1}) * Y_{out_1} \\ &= (1 - 0.472) * (1 - 0.472) * 0.472 \\ &= 0.132\end{aligned}$$

Perhitungan di lakukan hingga δ_{k_3} didapatkan. Hasil δ_k dapat dilihat pada tabel 4.68 dibawah ini:

Table 4.68 nilai δ_k Anak6 Iterasi 1

δ_1	δ_2	δ_3
0.132	0.136	-0.148

6. Menghitung koreksi bobot antara *Hidden Layer* dan *output layer* (ΔW_{ij}). ΔW_{ij} dihitung dengan mengalikan nilai α (*learning rate*) dengan nilai δ_k masing – masing node dan Z_j pada tabel 4.65 seperti persamaan (2-13).

Perhitungan ΔW_{ij} :

$$\begin{aligned}\Delta W_{11} &= \alpha * \delta_{k_1} * Z_1 \\ &= 0.3 * 0.132 * 0.6027 \\ &= 0.024\end{aligned}$$

Perhitungan dilakukan hingga kelima koreksi perubahan bobot pada setiap node *output* didapatkan. Sedangkan untuk menghitung nilai perubahan bobot dari bias hanya mengalikan α (*learning rate*) dengan nilai *error* (δ_k). seperti dibawah ini:

$$\begin{aligned}\Delta W_{bias_1} &= \alpha * \delta_{k_1} \\ &= 0.3 * 0.132 \\ &= 0.039\end{aligned}$$

Perhitungan juga dilakukan hingga perubahan bobot biar untuk ketiga node *output* didapatkan. Tabel 4.69 merupakan hasil perhitungan perubahan bobot antara *Hidden Layer* dan *output layer* (ΔW_{ij} & ΔW_{bias_j}).

Table 4.69 Hasil perhitungan koreksi bobot ΔW_{ij} Anak6 Iterasi 1

ΔW_{ij}	δ_1	δ_2	δ_3
Z1	0.024	0.025	-0.027
Z2	0.015	0.015	-0.017
Z3	0.016	0.016	-0.018
Z4	0.017	0.018	-0.019
Z5	0.023	0.023	-0.025
Bias	0.039	0.041	-0.044

- Menghitung nilai *error* antara *Hidden Layer* dan *input layer* (δ_j) yang akan digunakan untuk menghitung perubahan bobot. δ_j dihitung dengan mengalikan δ_{in_j} , bobot awal antara *Hidden Layer* dan *output layer*, serta Z_j dengan menggunakan persamaan (2-15).

menghitung δ_{in_j} dengan menjumlahkan hasil perkalian dari nilai *error* (δ_k) dikali dengan bobot awal antara *Hidden Layer* dan *output layer* pada tabel 4.63.

Perhitungan δ_{in_j} :

$$\begin{aligned}\delta_{in_1} &= (\delta_{k_1} * W_{11}) + (\delta_{k_2} * W_{12}) + (\delta_{k_3} * W_{13}) \\ &= (0.132 * (-0.464)) + (0.136 * (0.008)) + (0.148 * (-0.183)) \\ &= (-0.033)\end{aligned}$$

Perhitungan dilakukan hingga δ_{in_5} didapatkan seperti pada tabel 4.70 dibawah ini:

Table 4.70 Hasil perhitungan δ_{in_j} Anak6 Iterasi 1

δ_{in1}	δ_{in2}	δ_{in3}	δ_{in4}	δ_{in5}
-0.033	-0.061	-0.202	-0.017	0.164

Perhitungan δ_j :

$$\begin{aligned}\delta_1 &= \delta_{in_1} * (1 - Z_1) * Z_1 \\ &= (-0.033) * (1 - 0.6027) * 0.6027 \\ &= (-0.008)\end{aligned}$$

Perhitungan dilakukan hingga δ_5 didapatkan seperti pada tabel 4.71 dibawah ini:

Table 4.71 Hasil Perhitungan δ_j Anak6 Iterasi 1

δ_1	δ_2	δ_3	δ_4	δ_5
-0.008	-0.014	-0.049	-0.004	0.041

8. Menghitung koreksi bobot antara *Hidden Layer* dan *input layer* (ΔV_{ij}). ΔV_{ij} dihitung dengan mengalikan α (*learning rate*) dengan δ_j ditabel 4.71 dan bobot awal X_i (V_{ij}) di tabel 4.62 seperti persamaan (2-18).

Perhitungan ΔV_{ij} :

$$\begin{aligned}\Delta V_{11} &= \alpha * \delta_{j1} * X_{11} \\ &= 0.3 * (-0.008) * 0 \\ &= (0.0000)\end{aligned}$$

Perhitungan dilakukan hingga semua koreksi bobot antara *Hidden Layer* dan *input layer* didapatkan. Untuk menghitung koreksi bobot bias dengan mengalikan α (*learning rate*) dengan δ_j .

Perhitungan $\Delta V_{bias,j}$:

$$\begin{aligned}\Delta V_{bias1} &= \alpha * \delta_1 \\ &= 0.3 * (-0.008) \\ &= (-0.0024)\end{aligned}$$

Perhitungan dilakukan hingga ΔV_{bias5} didapatkan seperti pada tabel 4.72 dibawah ini:

Table 4.72 Hasil Perhitungan koreksi bobot ΔV_{ij} & bias Anak6 Iterasi 1

ΔV_{ij}	δ_1	δ_2	δ_3	δ_4	δ_5
X1	0.0000	0.0000	0.0000	0.0000	0.0000
X2	0.0000	0.0000	0.0000	0.0000	0.0000
X3	0.0000	0.0000	0.0000	0.0000	0.0000
X4	-0.0024	-0.0043	-0.0146	-0.0012	0.0121
X5	0.0000	0.0000	0.0000	0.0000	0.0000
X6	0.0000	0.0000	0.0000	0.0000	0.0000
X7	0.0000	0.0000	0.0000	0.0000	0.0000
X8	0.0000	0.0000	0.0000	0.0000	0.0000
X9	-0.0024	-0.0043	-0.0146	-0.0012	0.0121
X10	0.0000	0.0000	0.0000	0.0000	0.0000
X11	-0.0024	-0.0043	-0.0146	-0.0012	0.0121

Table 4.72 Hasil Perhitungan koreksi bobot ΔV_{ij} & bias Anak6 Iterasi 1

ΔV_{ij}	δ_1	δ_2	δ_3	δ_4	δ_5
X12	-0.0024	-0.0043	-0.0146	-0.0012	0.0121
X13	0.0000	0.0000	0.0000	0.0000	0.0000
X14	0.0000	0.0000	0.0000	0.0000	0.0000
X15	0.0000	0.0000	0.0000	0.0000	0.0000
X16	0.0000	0.0000	0.0000	0.0000	0.0000
X17	0.0000	0.0000	0.0000	0.0000	0.0000
X18	0.0000	0.0000	0.0000	0.0000	0.0000
X19	0.0000	0.0000	0.0000	0.0000	0.0000
Bias	-0.0024	-0.0043	-0.0148	-0.0012	0.0122

9. Menghitung bobot baru antara *input layer* dan *Hidden Layer* (V_{ij}). V_{ij} dihitung dengan menjumlahkan bobot lama pada tabel 4.62 dengan koreksi bobot ΔV_{ij} pada tabel 4.72 seperti persamaan 2-18.

Perhitungan V_{ij} pertama :

$$\begin{aligned}
 V_{11} &= V_{1i} \text{ (lama)} + \Delta V_{11} \\
 &= 0.254 + (0.0000) \\
 &= 0.254
 \end{aligned}$$

Perhitungan dilakukan hingga $V_{18 \ 5}$ didapatkan, sehingga didapat nilai bobot baru secara keseluruhan pada tabel 4.73

Table 4.73 Hasil perhitungan bobot baru V_{ij} Anak6 Iterasi 1

V_{ij}	z_1	z_2	z_3	z_4	z_5
X1	0.254	-0.049	0.041	-0.213	0.273
X2	0.095	-0.194	-0.128	-0.106	0.216
X3	0.277	0.145	0.174	-0.267	-0.052
X4	0.038	-0.209	-0.342	0.155	0.092
X5	-0.300	0.145	0.157	0.107	-0.191
X6	-0.055	-0.242	0.232	-0.213	0.103
X7	0.159	0.146	-0.063	-0.159	-0.201
X8	-0.063	-0.242	-0.063	-0.106	0.212
X9	0.049	-0.252	-0.088	0.210	-0.080
X10	0.206	0.047	0.220	0.266	0.061
X11	0.212	0.131	0.245	-0.165	0.086

Table 4.73 Hasil perhitungan bobot baru V_{ij} Anak6 Iterasi 1

X12	0.050	0.233	0.260	-0.109	0.076
X13	-0.060	0.140	-0.073	0.051	0.217
X14	-0.055	0.194	-0.232	0.107	0.155
X15	-0.114	-0.150	0.100	-0.162	-0.194
X16	-0.277	-0.145	-0.174	-0.107	-0.206
X17	-0.222	-0.145	-0.058	0.213	0.258
X18	-0.222	-0.194	0.116	-0.267	-0.258
X19	-0.055	0.145	-0.116	0.107	-0.155
Bias	0.057	-0.445	-0.544	-0.345	0.170

10. Menghitung bobot baru antara *Hidden Layer* dan *output layer* (W_{ij}). W_{ij} dihitung dengan menjumlahkan bobot lama pada tabel 4.63 dengan koreksi bobot ΔW_{ij} pada tabel 4.69 seperti persamaan 2-17.

Perhitungan W_{ij} pertama:

$$\begin{aligned}
 W_{11} &= W_{11(lama)} + \Delta W_{11} \\
 &= (-0.464) + 0.024 \\
 &= (-0.440)
 \end{aligned}$$

Perhitungan dilakukan hingga W_{53} didapatkan, sehingga didapatkan nilai bobot baru secara keseluruhan pada tabel 4.74

Table 4.74 Hasil perhitungan bobot baru W_{ij} Anak6 Iterasi 1

W_{ij}	y1	y2	y3
z1	-0.440	0.033	-0.210
z2	0.448	-0.491	0.317
z3	-0.366	-0.364	0.658
z4	0.223	-0.128	0.142
z5	0.381	0.502	-0.377
Bias	-0.097	-0.032	0.394

11. Menghitung nilai MSE error pada akhir iterasi dengan mengurangi nilai target pada tabel 4.1 dengan nilai Y_{out} data terakhir.

Perhitungan nilai *MSE error* pada Anak 6, ID 6:

$$MSE\ error = \frac{1}{nNodeoutput} \sum_k^{Nnodeoutput} (target_k - Y_{out})^2$$

70

$$MSE\ error = \frac{1}{3} \times ((1 - 0.472) + (1 - 0.450) + (0 - 0.644))^2$$

$$MSE\ error = 0.062782$$

Perhitungan Data Uji

Setelah proses learning dari 6 data latih selesai di lakukan, selanjutnya ialah melakukan perhitungan untuk 1 data uji pada tabel 4.75 dengan menggunakan bobot akhir dari proses learning yang telah di dapat pada tabel 4.73 dan 4.74.

Table 4.75 Data Uji

ID	NAMA	X 1	X 2	X 3	X 4	X 5	X 6	X 7	X 8	X 9	X 10	X 11	X 12	X 13	X 14	X 15	X 16	X 17	X 18	X 19
10	Anak 10	0	0	0	1	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0

Selanjutnya di lakukan perhitungan hingga didapatkan Y_{out} nya.

1. Menghitung neuron yang masuk ke *Hidden Layer* dari *input layer* (Z_{in_j}) dengan persamaan (2-8). Tiap nilai input (X_1) Anak 10 dengan ID 10 pada tabel 4.72 dikalikan dengan bobot (V_{ij}) pada tabel 4.73 dan kemudian dijumlahkan dengan bobot biasnya.

Perhitungan *input* pada *Hidden Layer* pertama (Z_{in_1}):

$$\begin{aligned} Z_{in_1} &= \text{bobot bias}(V_{71}) + (X_1 * V_{11}) + (X_2 * V_{21}) + (X_3 * V_{31}) + (X_4 * V_{41}) + (X_5 * V_{51}) + \\ & (X_6 * V_{61}) + (X_7 * V_{71}) + (X_8 * V_{81}) + (X_9 * V_{91}) + (X_{10} * V_{101}) + (X_{11} * V_{111}) + \\ & (X_{12} * V_{121}) + (X_{13} * V_{131}) + (X_{14} * V_{141}) + (X_{15} * V_{151}) + (X_{16} * V_{161}) + (X_{17} * V_{171}) + \\ & (X_{18} * V_{181}) + (X_{19} * V_{191}) \\ &= 0.057 + (0 * 0.254) + (0 * 0.095) + (0 * 0.277) + (1 * 0.038) + (0 * (-0.300)) + \\ & (0 * (-0.055)) + (1 * 0.159) + (1 * (-0.063)) + (1 * 0.49) + (0 * 0.206) + \\ & (0 * 0.212) + (0 * 0.050) + (0 * (-0.060)) + (1 * (-0.055)) + (0 * (-0.114)) + \\ & (0 * (-0.277)) + (0 * (-0.222)) + (0 * (-0.222)) + (0 * (-0.055)) \\ &= 0.18 \end{aligned}$$

Perhitungan diatas dilakukan hingga Z_{in_5} selesai di hitung, sehingga didapatkan hasil perhitungan seluruh Z_{in} pada tabel 4.76 dibawah ini :

Table 4.76 Hasil perhitungan Z_{in} Data Uji

Z_{in_1}	Z_{in_2}	Z_{in_3}	Z_{in_4}	Z_{in_5}
0.18	-0.81	-1.33	-0.14	0.35

2. Menghitung neuron yang keluar dari *Hidden Layer* menuju *output layer* (Z_j). Z_j dihitung dengan menggunakan fungsi aktivasi sigmoid biner dengan persamaan (2-9).

Perhitungan *output* dari *Hidden Layer* pertama Z_1 :

$$Z_1 = \frac{1}{(1+e^{-Z_{in1}})} = \frac{1}{(1+18)} = 0.5460$$

Perhitungan di lakukan hingga Z_5 . tabel 4.77 merupakan hasil perhitungan Z_j

Table 4.77 Hasil perhitungan Z_j Data Uji

Z1	Z2	Z3	Z4	Z5
0.5460	0.3082	0.2089	0.4653	0.5859

3. Menghitung neuron yang masuk ke *output layer* dari *Hidden Layer* Y_{in} . Y_{in} dihitung dengan menggunakan persamaan (2-10). Tiap neuron yang keluar dari *Hidden Layer* (Z_j) pada tabel 4.77 dikalikan dengan bobot antara *Hidden Layer* dan *output layer* (W_{ij}) pada tabel 4.74 dan kemudian dijumlahkan dengan bobot biasnya

Perhitungan Y_{in} :

$$\begin{aligned} Y_{in_1} &= \text{bobot bias}(W_{61}) + (Z_1 * W_{11}) + (Z_2 * W_{21}) + (Z_3 * W_{31}) + (Z_4 * W_{41}) + (Z_5 * W_{51}) \\ &= (-0.097) + (0.5460 * (-0.440)) + (0.3082 * 0.448) + (0.2089 * (-0.366)) + (0.4653 * 0.223) + (0.5859 * 0.381) \\ &= (0.051) \end{aligned}$$

Perhitungan dilakukan terus hingga Y_{in_3} didapatkan pada tabel 4.78 dibawah ini :

Table 4.78 Hasil Perhitungan Y_{in} Data Uji

Y_{in_1}	Y_{in_2}	Y_{in_3}
0.051	-0.007	0.359

4. Menghitung neuron keluaran dari *output layer* (Y_{out}). Y_{out} dihitung dengan menggunakan fungsi aktivasi *sigmoid biner* dengan persamaan (2-11).

Perhitungan Y_{out_1} :

$$\begin{aligned} Y_{out_1} &= 1 / (1 + e^{-Y_{in_j}}) \\ &= 1 / (1 + e^{-(0.051)}) \\ &= 0.5129 \end{aligned}$$

Perhitungan dilakukan terus hingga Y_{out_3} didapatkan terlihat pada tabel 4.79 dibawah ini:

Table 4.79 Hasil Perhitungan Y_{out} Data Uji

Yout_1	Yout_2	Yout_3
0.5129	0.4983	0.5889

Y_{out} pada tabel 4.79 di bulatkan sehingga menjadi angka biner seperti tabel 4.80.

Table 4.80 Y_{out} yang telah di bulatkan

Yout_1	Yout_2	Yout_3
1	0	1

Selanjutnya dilakukan seleksi sebagai berikut :

- 111 = Maka Terdiagnosa Cacar Air
- 110 = Maka Terdiagnosa Skabies
- 101 = Maka Terdiagnosa Campak
- 100 = Maka Terdiagnosa Dermatitis
- 011 = Maka Terdiagnosa Herpes
- 010 = Maka Terdiagnosa Abses

Hasil output data uji adalah 101 maka anak 10 terdiagnosis penyakit Campak.

4.2 Implementasi

Implementasi diterapkan sesuai perancangan yang telah disusun ditahap perancangan. Tahap-tahapan perancangan akan di implementasikan dengan menggunakan bahasa pemrograman java.

4.3 Pengujian dan Analisis

Pengujian dan analisis dilakukan untuk mengetahui dan menganalisa hasil learning Jaringan Syaraf Tiruan yang dilakukan oleh sistem yang akan mendiagnosis penyakit kulit pada anak. Beberapa pengujian akan dilakukan untuk mengetahui keberhasilan sistem dengan menggunakan algoritma *backpropagation*. Ada 4 pengujian yang akan dilakukan guna mengetahui bagaimana kemampuan sistem yang telah di implementasi. Pengujian yang akan dilakukan sebagai berikut:

1. Pengujian untuk pengaruh variasi jumlah *Hidden Neuron*. Tabel 4.81 merupakan table yang akan digunakan dalam pengujian pengaruh variasi jumlah *Hidden Neuron*.

Table 4.81 Tabel Pengujian Perbedaan Jumlah *Hidden Neuron*

<i>Jumlah Hidden Neuron</i>	Percobaan ke-	MSE	Waktu (millisecond)	Rata – Rata MSE

2. Pengujian untuk mengetahui bagaimana kemampuan *learning* sistem terhadap variasi nilai *learning rate* yang di gunakan. Tabel 4.82 merupakan table yang akan digunakan dalam pengujian perbedaan *learning rate*.

Table 4.82 Tabel Pengujian Nilai *Learning rate*

<i>Learning Rate</i>	Percobaan ke-	<i>Hidden Layer</i>	MSE	Waktu (millisecond)	Rata rata MSE

3. Pengujian untuk mengetahui kemampuan *learning* terhadap jumlah *epoch* yang berbeda. Tabel 4.83 merupakan table yang akan digunakan dalam pengujian Perbedaan nilai maksimum epoch.

Table 4.83 Tabel Pengujian Nilai Maksimum Epoch

<i>Jumlah Hidden Neuron</i>	<i>Learning Rate</i>	<i>Epoch</i>	MSE	Waktu (millisecond)

4. Pengujian akurasi sistem dengan membandingkan jumlah data training yang digunakan. Tabel 4.84 merupakan table yang akan digunakan dalam pengujian akurasi.

Table 4.84 Tabel Pengujian Akurasi

Data ke-	Target node output (1,2,3)	Keluaran node output (1,2,3)	Error	Hasil Sistem	Hasil Poliklinik

4.4 Pengambilan Kesimpulan Dan Saran

Pengambilan kesimpulan dan saran dilakukan setelah tahap perancangan, implementasi, dan pengujian pada sistem telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian sistem dan analisis. Saran merupakan hal yang bertujuan untuk memperbaiki dan menyempurnakan penelitian agar lebih baik untuk penelitian selanjutnya.



BAB 5 IMPLEMENTASI

Pada bab implementasi ini akan membahas tentang implementasi sistem yang telah dibuat. Pembahasan implementasi meliputi spesifikasi sistem, batasan implementasi implementasi algoritma dan implementasi antarmuka.

5.1 Spesifikasi Sistem

Pada spesifikasi sistem akan membahas spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam pembuatan sistem. Spesifikasi sistem akan dibahas secara detail supaya implementasi sesuai dengan tujuan dengan perancangan.

5.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras dijelaskan pada tabel 5.1 dibawah ini:

Table 5.1 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Prosesor	Laptop <i>processor</i> AMD Radeon 1.95 GHz.
Memori (RAM)	Memori RAM 4 GB DDR3
Hardisk	Hardisk 500GB.

5.1.2 Spesifikasi perangkat lunak

Spesifikasi perangkat lunak dijelaskan pada tabel 5.2 dibawah ini

Table 5.2 Spesifikasi Perangkat Lunak

Sistem Operasi	Sistem Operasi Windows 10 64-bit
Bahasa Pemrograman	Java
Tools Pemrograman	Netbeans 8.0.2

5.2 Batasan Implementasi

Pada sub-bab ini akan dibahas batasan-batasan fungsi yang dapat dilakukan oleh sitem untuk memfasilitasi pengguna dalam mengakses sistem. Berikut merupakan batasan-batasan sistem yaitu:

- a. Masukan yang di terima oleh program ialah:
 - Jumlah node yang terdapat pada *Hidden Layer*
 - *hidden layer* yang digunakan berjumlah 1

- Nilai *learning rate* dengan rentang angka 0-1
- Nilai *epoch* maksimum.
- Nilai minimum eror 0.000000001

b. Keluaran yang dihasilkan berupa hasil diagnosa penyakit kulit berdasarkan hasil learning *backpropagation* dengan inisialisasi bobot Nguyen-Widrow.

5.3 Implementasi Algoritma

Pada sub-bab ini akan membahas implementasi algoritma yang digunakan pada sistem.

5.3.1 Algoritma Inisialisasi Bobot dengan Nguyen – Widrow

Proses inisialisasi bobot dilakukan dengan menentukan secara acak nilai bobot dengan range $(-0.5) - 0.5$ oleh sistem. Nilai ini digunakan untuk melanjutkan perhitungan Nguyen – Widrow. *Source code* 5.1 merupakan *source code* inisialisasi bobot Nguyen-Widrow.

```

1 public void initBobot() {
2     double beta1;
3     double beta2;
4     beta1 =
5     0.7*Math.pow(jumlahNode, (1/(double) jumlahInput));
6     beta2 =
7     0.7*Math.pow(jumlahOutput, (1/(double) jumlahNode));
8     //bobot random
9     System.out.println(" BOBOT RANDOM : ");
10    for (int i = 0; i < jumlahNode; i++) {
11        for (int j = 0; j < jumlahInput; j++) {
12            vAwal[i][j] = ((float) (Math.random() * 10) -
13            5)/10;
14            // System.out.print(v[i][j] + "\t");
15            //zn[j] = 0.0;
16        }
17        // System.out.println();
18    }
19
20    //NILAI V optimasi
21    for (int i = 0; i < jumlahNode; i++) {
22        for (int j = 0; j < jumlahInput; j++) {
23            zn[i] += (Math.pow(vAwal[i][j], 2));
24        }
25        zn[i] = Math.sqrt(zn[i]);
26    }
27
28    System.out.println();
29
30    System.out.println(" NILAI ||V|| : ");
31    for (int i = 0; i < jumlahNode; i++) {
32        System.out.print(zn[i] + "\t");
33    }
34
35    //BOBOT BARU

```

```

36      System.out.println();
37      System.out.println();
38      System.out.println(" BOBOT BARU Input Ke Hidden
39 node : ");
40      for (int i = 0; i < jumlahNode; i++) {
41          for (int j = 0; j < jumlahInput; j++) {
42              v[i][j] = (beta1 * vAwal[i][j]) / zn[i];
43              System.out.print(v[i][j] + "\t");
44          }
45          System.out.println();
46      }
47      System.out.println("");
48      //NILAI BIAS
49      System.out.println("Bias Node : ");
50      for (int i = 0; i < jumlahNode; i++) {
51          biasNode[i] = (Math.random() * (beta1-(-
52 beta1))) + (-beta1);
53          System.out.println(biasNode[i]);
54      }
55
56      //BOBOT NODE
57      // System.out.println("Bobot Node : ");
58      for (int i = 0; i < jumlahOutput; i++) {
59          for (int j = 0; j < jumlahNode; j++) {
60              W[i][j] = ((float)(Math.random() * 10)-
61 5)/10;
62              //System.out.print(W[i][j]+" \t");
63          }
64          // System.out.println("");
65      }
66
67      //NILAI W optimasi
68      for (int i = 0; i < jumlahOutput; i++) {
69          for (int j = 0; j < jumlahNode; j++) {
70              yn[i] += (Math.pow(W[i][j], 2));
71          }
72          yn[i] = Math.sqrt(yn[i]);
73      }
74
75      System.out.println();
76
77      System.out.println(" NILAI ||W|| : ");
78      for (int i = 0; i < jumlahOutput; i++) {
79          System.out.print(yn[i] + "\t");
80      }
81
82      //BOBOT BARU
83      System.out.println();
84      System.out.println();
85      System.out.println(" BOBOT BARU Hidden ke Output:
86 ");
87      for (int i = 0; i < jumlahOutput; i++) {
88          for (int j = 0; j < jumlahNode; j++) {
89              wNode[i][j] = (beta2 * W[i][j]) / yn[i];
90              System.out.print(wNode[i][j] + "\t");
91          }
92          System.out.println();
93      }
94      System.out.println("");

```

```

95 //NILAI BIAS
96 System.out.println("Bias Output : ");
97 for (int i = 0; i < jumlahOutput; i++) {
98     biasOutput[i] = (Math.random() * (beta2-(-
99     beta2))) + (-beta2);
100     System.out.println(biasOutput[i]);
101 }
102
103 //BIAS OUTPUT
104 // System.out.println("Bias Output : ");
105 // biasOutput = Math.random() * 1;
106 // System.out.println(biasOutput);
107 }

```

Source code 5.1 Source code Inisialisasi Bobot Nguyen-Widrow

5.3.2 Algoritma Feed Forward

Pada proses *feed forward* ini diawali dengan menghitung nilai masukan dari nilai bobot yang telah dicari hingga nilai keluaran dari *output layer*.

5.3.2.1 Algoritma Perhitungan Inputan pada *Hidden Layer*

Hasil dari inisialisasi bobot dan nilai inputan dikalikan untuk selanjutnya dijumlahkan. Source code 5.2 merupakan source code proses perhitungan masukan pada *Hidden Layer*.

```

1 //hitung Zin
2 int col=0;
3 for (int j = 0; j < jumlahNode; j++) {
4     zin[j]=biasNode[j]+(input[x][col]* v[j][col])
5     +(input[x][col+1]* v[j][col+1])
6     +(input[x][col+2]* v[j][col+2])
7     +(input[x][col+3]* v[j][col+3])
8     +(input[x][col+4]* v[j][col+4])
9     +(input[x][col+5]* v[j][col+5])
10    +(input[x][col+6]* v[j][col+6])
11    +(input[x][col+7]* v[j][col+7])
12    +(input[x][col+8]* v[j][col+8])
13    +(input[x][col+9]* v[j][col+9])
14    +(input[x][col+10]* v[j][col+10])
15    +(input[x][col+11]* v[j][col+11])
16    +(input[x][col+12]* v[j][col+12])
17    +(input[x][col+13]* v[j][col+13])
18    +(input[x][col+14]* v[j][col+14])
19    +(input[x][col+15]* v[j][col+15])
20    +(input[x][col+16]* v[j][col+16])
21    +(input[x][col+17]* v[j][col+17]);
22 }

```

Source code 5.2 Source code Perhitungan masukan *Hidden Layer*

5.3.2.2 Algoritma Perhitungan Output pada *Hidden Layer*

Setelah hasil nilai masukan didapatkan selanjutnya akan dilakukan perhitungan outputan dari hasil nilai masukan dengan fungsi aktivasi sigmoid biner. *Source Code 5.3* merupakan perhitungan keluaran pada *Hidden Layer*.

```
1 //hitung Zout
2 for (int i = 0; i < jumlahNode; i++) {
3     z[i] = (1 / (1 + Math.exp(-zin[i])));
4 }
```

Source code 5.3 Source code Algoritma Perhitungan Outputan *Hidden Layer*

5.3.2.3 Algoritma Perhitungan Inputan pada *Output Layer*

Hasil dari outputan *Hidden Layer* kemudian akan digunakan untuk perhitungan masukan pada *output layer*. *Source code 5.4* merupakan source code proses perhitungan masukan pada *output layer*.

```
1 //hitung Yin
2 for (int i = 0; i < jumlahNode; i++) {
3     for (int j = 0; j < jumlahOutput; j++) {
4         tampung[j] += z[i] * wNode[j][i];
5     }
6 }
7 for (int i = 0; i < jumlahOutput; i++) {
8     yin[i] += biasOutput[i] + tampung[i];
9 }
```

Source code 5.4 Source Code Algoritma Perhitungan Inputan *Output Layer*

5.3.2.4 Algoritma Perhitungan Keluaran pada *Output Layer*

Perhitungan keluaran pada layer ini menggunakan hasil masukan output layer dengan fungsi aktivasi sigmoid biner. *Source code 5.5* merupakan source code proses perhitungan keluaran pada output layer.

```
1 // hitung Yout
2 for (int i = 0; i < jumlahOutput; i++) {
3     y[i] = (1 / (1 + Math.exp(-yin[i])));
4 }
```

Source code 5.5 Source code Algoritma Perhitungan Outputan *Output Layer*

5.3.3 Algoritma Feed Backward

Pada fase *feed forward* ini dilakukan dengan perhitungan nilai error antar layer dan perbaikan bobot untuk perhitungan *learning* selanjutnya.

5.3.3.1 Algoritma Error pada *Output Layer*

Perhitungan error di layer ini dilakukan dengan menggunakan nilai target dan nilai keluaran pada output layer. *Source code 5.6* merupakan source code perhitungan error output layer.

```
1 // untuk mencari ohmk
2 for (int i = 0; i < jumlahOutput; i++) {
3     fAksen[i] = y[i] * (1 - y[i]);
4 }
5 for (int i = 0; i < jumlahOutput; i++) {
```

6	ohmk[i] = (target[x][i] - y[i]) * fAksen[i];
7	}

Source code 5.6 Source code Algoritma Error Output Layer

5.3.3.2 Algoritma Error Hidden Layer

Perhitungan *error* pada *layer* ini dilakukan dengan menggunakan hasil perhitungan *error* pada output layer. Source code 5.7 merupakan source code perhitungan nilai *error* pada *Hidden Layer*.

1	//nilai ohm_inj
2	for (int i = 0; i < jumlahNode; i++) {
3	for (int j = 0; j < jumlahOutput; j++) {
4	ohm_in[i] += ohmk[j] * wNode[j][i];
5	}
6	}
7	//
8	//aktifasi nilai ohm j
9	for (int i = 0; i < jumlahNode; i++) {
10	fAksenZ[i] = z[i] * (1 - z[i]);
11	ohmj[i] = ohm_in[i] * fAksenZ[i];
12	}

Source code 5.7 Source code Algoritma error Hidden Layer

5.3.3.3 Algoritma Perbaikan Bobot

Perhitungan perbaikan bobot antar layer dilakukan dengan hasil *error* antar layer dengan nilai *learning rate* yang telah di inputkan. Source code 5.8 merupakan source code proses perhitungan perbaikan bobot.

1	//delta Vij dan deltaVBias
2	for (int i = 0; i < jumlahNode; i++) {
3	for (int j = 0; j < jumlahInput; j++) {
4	deltaV[i][j] = alpha*ohmj[i]*v[i][j];
5	deltaVBias[i] = alpha * ohmj[i];
6	}
7	}
8	//delta W
9	for (int i = 0; i < jumlahOutput; i++) {
10	for (int j = 0; j < jumlahNode; j++) {
11	deltaWNode[i][j] = alpha * ohmk[i] * z[j];
12	}
13	}
14	for (int i = 0; i < jumlahOutput; i++) {
15	deltaWBias[i] = alpha * ohmk[i];
16	}

Source code 5.8 Source code Algoritma Perbaikan Bobot

4.3.3.4 Algoritma Bobot Baru

Proses perhitungan bobot baru dilakukan dengan menambahkan nilai bobot lama dengan hasil nilai perbaikan bobot. Source code 5.9 merupakan source code proses perhitungan bobot baru.

1	// update bobot v
2	for (int i = 0; i < jumlahNode; i++) {
3	for (int j = 0; j < jumlahInput; j++) {

```

4         v[i][j] = v[i][j] + deltaV[i][j];
5     }
6 }
7
8 //update bobot bias pada Hidden Layer
9 for (int i = 0; i < jumlahNode; i++) {
10     biasNode[i] = biasNode[i] + deltaVBias[i];
11 }
12
13 //update bobot W
14 //System.out.println("Bobot Node Baru : ");
15 for (int i = 0; i < jumlahOutput; i++) {
16     for (int j = 0; j < jumlahNode; j++) {
17         wNode[i][j] = wNode[i][j] + deltaWNode[i][j];
18     }
19 }
20
21 // update bobot bias ourput layer
22 for (int i = 0; i < jumlahOutput; i++) {
23     biasOutput[i] = biasOutput[i] + deltaWBias[i];
24 }

```

Source code 5.9 Source code Algoritma Bobot Baru.



BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas tentang hasil dan pembahasan dari implementasi yang sebelumnya telah dilakukan.

6.1 Pengujian dan Analisis

Hasil implementasi Jaringan Syaraf Tiruan dengan metode *Backpropagation* pada Diagnosis Penyakit Kulit Pada Anak merupakan sebuah sistem yang dibangun berdasarkan perancangan pada bab sebelumnya. Ada 4 pengujian yang dilakukan pada penelitian ini, yaitu pengujian dengan perbedaan jumlah *Hidden Neuron*, pengujian perbedaan nilai *learning rate*, pengujian perbedaan nilai *epoch* maksimum dan pengujian akurasi. Pengujian jumlah *Hidden Neuron*, nilai *learning rate* dan nilai *epoch* maksimum akan dilakukan terlebih dahulu dengan data yang berjumlah 214 data yang kemudian dibagi menjadi 154 data latih dan 60 data uji. Hasil pengujian terbaik dari ketiganya akan digunakan untuk melakukan pengujian akurasi.

6.1.1 Pengujian dan Analisis Perbedaan Jumlah *Hidden Neuron*

Pengujian pertama merupakan pengujian dengan membedakan jumlah node yang ada pada *Hidden Layer* yaitu dengan jumlah 1, 2, 3, 4 dan 5 dengan nilai *learning rate* 0.1 dilakukan percobaan sebanyak 5 kali.

Table 6.1 Hasil Pengujian Perbedaan Jumlah *Hidden Neuron*

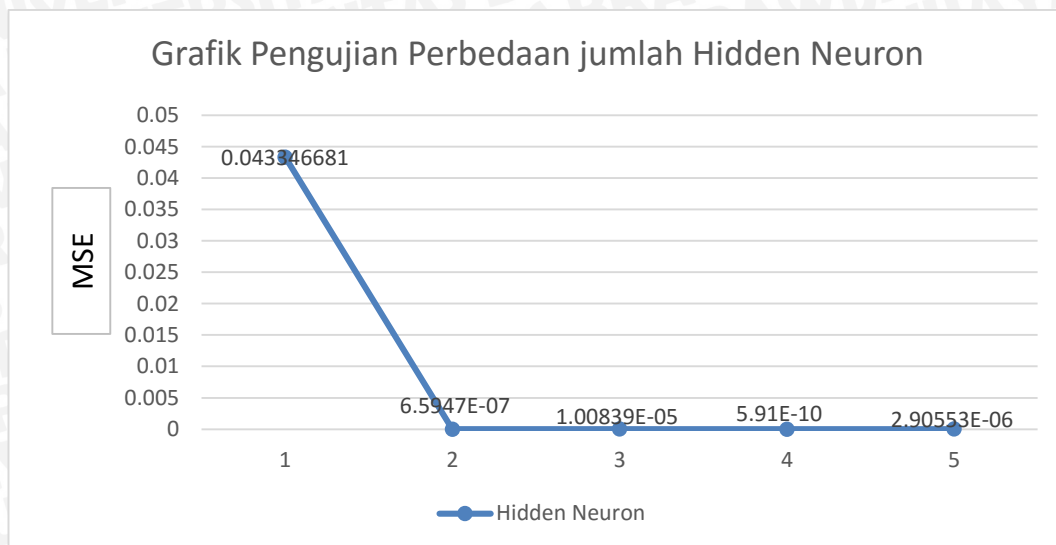
<i>Jumlah Hidden Neuron</i>	Percobaan ke-	MSE	Waktu (millisecond)	Rata – Rata MSE
1	1	0.0034836132180	66958.0	0.043346680564
	2	0.0600369712008	74572.0	
	3	0.0382473747982	77275.0	
	4	0.0629660849065	60830.0	
	5	0.0519993586942	73086.0	
2	1	0.0000018255324	81266.0	0.000000659470
	2	0.0000000959866	119700.0	
	3	0.0000013260711	72379.0	
	4	0.0000000465438	101218.0	
	5	0.0000000032180	99581.0	

Table 6.1 Hasil Pengujian Perbedaan Jumlah *Hidden Neuron*

<i>Jumlah Hidden Neuron</i>	Percobaan ke-	MSE	Waktu (millisecond)	Rata – Rata MSE
3	1	0.0000000009841	3173.0	0.000010083932
	2	0.0000002897957	128414.0	
	3	0.0000244238669	119712.0	
	4	0.0000038804313	130364.0	
	5	0.0000218245841	133050.0	
4	1	0.0000000000422	839.0	0.000000000591
	2	0.0000000009860	6207.0	
	3	0.0000000000092	402.0	
	4	0.0000000009275	8074.0	
	5	0.0000000009905	809.0	
5	1	0.0000108636406	178254.0	0.000002905531
	2	0.0000000000080	573.0	
	3	0.0000000000212	1292.0	
	4	0.0000036631380	185957.0	
	5	0.0000000008481	9292.0	

Dari hasil pengujian diatas, didapatkan nilai rata-rata MSE terbaik yaitu 0.000000000591 dengan waktu rata-rata *learning* 3266.2 *millisecond* didapat dari *learning* dengan jumlah *Hidden Neuron* sebanyak 4.

Dari hasil nilai rata-rata MSE di setiap pengujian didapatkan grafik sebagai berikut.



Gambar 6.1 Grafik Pengujian Perbedaan Jumlah *Hidden Neuron*

Berdasarkan gambar 6.1 *hidden neuron* yang berjumlah 4 menunjukkan rata-rata MSE terkecil. Dimana pada *hidden neuron* 4 didapat rata-rata MSE sebesar 0.000000000591 dengan waktu rata-rata *learning* 3266.2 *millisecond*. Dari 5 kali percobaan ini selalu mendapat nilai MSE yang lebih kecil dibandingkan nilai toleransi yang telah ditentukan yaitu 0.000000001 sebelum jumlah maksimum epoch yang telah ditentukan. Sedangkan nilai MSE pada *hidden neuron* yang berjumlah 1, 2, 3 dan 5 mengalami kenaikan. Pada kasus ini tidak selalu pada *hidden layer* paling banyak lah yang mendapatkan nilai MSE terbaik, dimana pada *hidden neuron* 4 didapatkan nilai MSE terkecil. Hal ini dikarenakan dengan menggunakan *hidden neuron* yang banyak, maka variable yang digunakan dalam proses *learning* juga semakin banyak sehingga menyebabkan nilai MSE yang di dapat tidak stabil, begitu juga sebaliknya. Semakin sedikit *hidden neuron* yang digunakan membuat sedikitnya variable yang digunakan pada proses *learning* yang mengakibatkan proses *learning* menjadi tidak maksimal. Sehingga pada kasus ini *hidden neuron* yang berjumlah 4 merupakan yang terbaik. Selain itu, banyaknya *hidden neuron* juga mempengaruhi waktu *learning* dikarenakan semakin banyak jumlah *hidden neuron* semakin banyak pula variable yang di proses.

6.1.2 Pengujian dan Analisis Perbedaan Nilai *Learning Rate*

Pengujian perbedaan nilai *learning rate* dilakukan dengan menggunakan jumlah *Hidden Neuron* terbaik dengan MSE terkecil yang didapat pada pengujian perbedaan jumlah *Hidden Neuron* yang telah dilakukan sebelumnya. Pengujian dilakukan dengan menggunakan variasi input nilai *learning rate* 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 dan 0.9 dengan jumlah *Hidden Neuron* 4 dilakukan percobaan sebanyak 5 kali yang selanjutnya akan di hitung rata-rata MSE. Tabel 6.2 merupakan tabel hasil keluaran sistem dengan jumlah *Hidden Neuron* sebanyak 3 dengan variasi nilai *learning rate*.

Table 6.2 Hasil Pengujian Perbedaan Nilai *Learning Rate*

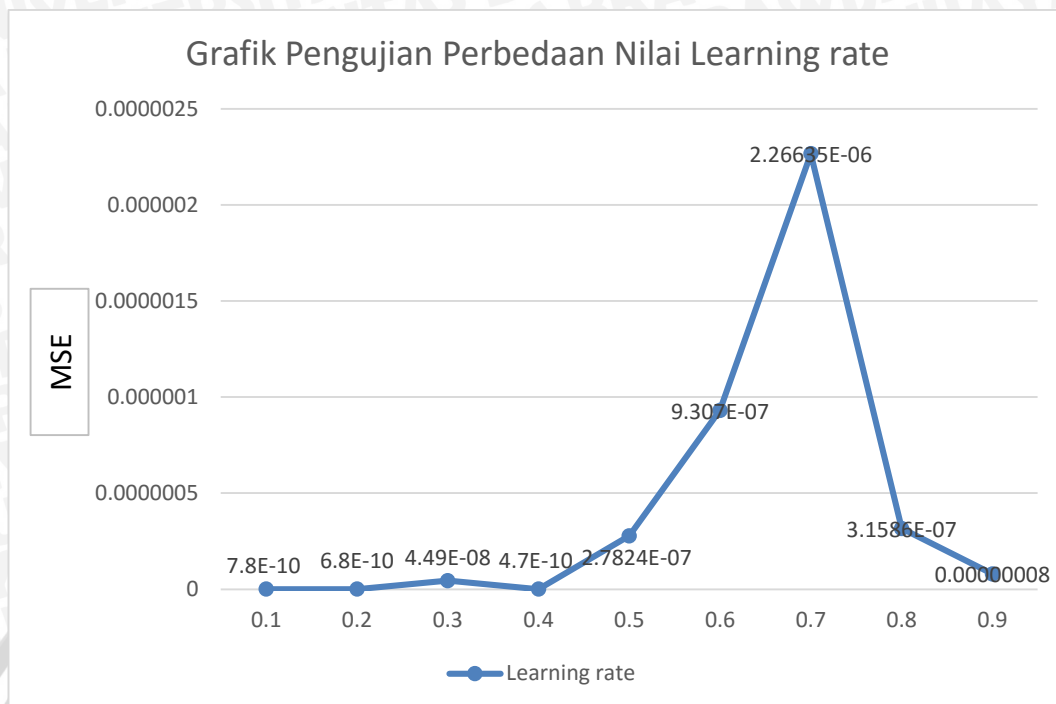
<i>Learnin g Rate</i>	Percobaan ke-	<i>Hidd en Laye r</i>	MSE	Waktu (millisecond)	Rata rata MSE
0.1	1	4	0.0000000000284	350.0	0.00000000078
	2	4	0.0000000009189	4702.0	
	3	4	0.0000000009986	36546.0	
	4	4	0.0000000009779	600.0	
	5	4	0.0000000009781	321.0	
0.2	1	4	0.0000000009996	3817.0	0.00000000068
	2	4	0.0000000005155	390.0	
	3	4	0.0000000009255	2858.0	
	4	4	0.0000000001033	4262.0	
	5	4	0.0000000008540	1251.0	
0.3	1	4	0.0000002225753	164481.0	0.0000000449
	2	4	0.0000000004338	1936.0	
	3	4	0.0000000000922	510.0	
	4	4	0.0000000005699	823.0	
	5	4	0.0000000009712	5010.0	
0.4	1	4	0.0000000004248	236.0	0.00000000047
	2	4	0.0000000000130	525.0	
	3	4	0.0000000009976	27324.0	
	4	4	0.0000000002029	645.0	
	5	4	0.0000000007088	3733.0	
0.5	1	4	0.0000000317519	145976.0	0.00000027824
	2	4	0.0000000007277	1958.0	
	3	4	0.0000000003556	1409.0	
	4	4	0.0000000209706	150972.0	
	5	4	0.0000013373837	156929.0	

Table 6.2 Hasil Pengujian Perbedaan Nilai *Learning Rate*

<i>Learnin g Rate</i>	Percobaan ke-	<i>Hidd en Laye r</i>	MSE	Waktu (millisecond)	Rata rata MSE
0.6	1	4	0.0000005173726	135881.0	0.0000009307
	2	4	0.0000041338818	154000.0	
	3	4	0.0000000009985	52070.0	
	4	4	0.0000000004324	625.0	
	5	4	0.0000000009981	20936.0	
0.7	1	4	0.0000000000231	162.0	0.00000226635
	2	4	0.0000000009886	1674.0	
	3	4	0.0000000009841	6491.0	
	4	4	0.0000113014470	142083.0	
	5	4	0.0000000283056	159367.0	
0.8	1	4	0.0000000008405	800.0	0.00000031586
	2	4	0.0000006532576	141930.0	
	3	4	0.0000000007456	12833.0	
	4	4	0.0000008571199	152589.0	
	5	4	0.0000000673381	151085.0	
0.9	1	4	0.0000000009863	3388.0	0.0000000800
	2	4	0.0000000007181	281.0	
	3	4	0.0000003972600	154885.0	
	4	4	0.0000000009377	9728.0	
	5	4	0.0000000002500	203.0	

Dari hasil pengujian diatas didapatkan nilai rata-rata MSE terkecil yaitu 0.00000000047 pada nilai learning rate 0.4 dengan waktu rata-rata 6432.6 *millisecond*.

Dari hasil nilai rata-rata MSE di pengujian ini didapatkan grafik sebagai berikut.



Gambar 6.2 Grafik Pengujian Perbedaan Nilai *Learning Rate*

Hasil dari gambar 6.2 menunjukkan nilai rata-rata MSE terbaik didapat pada *learning rate* 0.4 dikarenakan pada nilai *learning rate* 0.6 didapatkan rata-rata MSE terkecil dibandingkan *learning rate* 0.1, 0.2, 0.3, 0.5, 0.6, 0.7, 0.8 dan 0.9. Pada pengujian dengan *learning rate* 0.4 didapatkan rata-rata MSE 0.00000000047 yang merupakan rata-rata MSE terkecil di pengujian ini. Hal ini dikarenakan pada pengujian nilai *learning rate* 0.4 mendapatkan nilai MSE yang nilainya lebih kecil dibandingkan nilai toleransi yang ditentukan sebelum nilai *epoch* maksimal yang merupakan *stopping condition*. Berbeda dengan pengujian dengan nilai *learning rate* lainnya yang menghasilkan nilai MSE yang tidak stabil. Hal ini dikarenakan semakin besar nilai *learning rate* dapat mempercepat proses *learning* namun akan menghasilkan perubahan yang tidak stabil. Sedangkan jika menggunakan nilai *learning rate* yang terlalu kecil akan mengakibatkan proses *learning* menjadi lebih lama karena iterasi yang di butuhkan akan semakin besar. Sehingga pada kasus ini nilai *learning rate* 0.4 merupakan yang terbaik.

6.1.3 Pengujian dan Analisis Perbedaan Nilai *Epoch* Maksimum

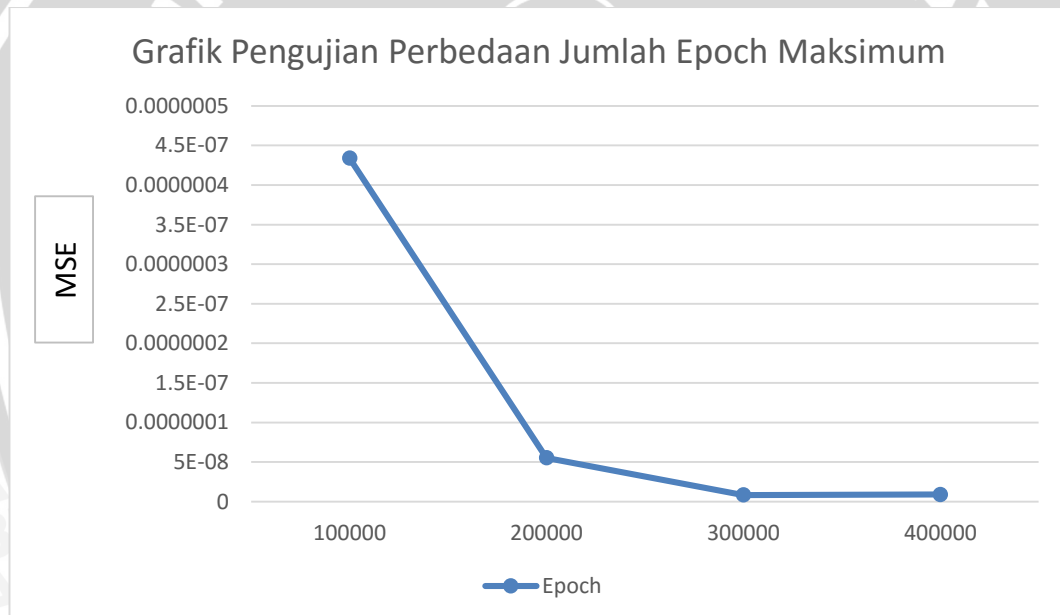
Pengujian ini membandingkan MSE sistem dengan jumlah variasi nilai *epoch* maksimum yang akan digunakan yaitu 100000, 200000, 300000 dan 400000 dengan nilai *learning rate* dan jumlah *Hidden Neuron* terbaik dari pengujian sebelumnya yaitu 0.4 dan jumlah *Hidden Neuron* 4. Tabel 6.3 merupakan hasil pengujian perbedaan nilai *epoch* maksimum.

Table 6.3 Hasil Pengujian Perbedaan Nilai *Epoch* Maksimum

Jumlah Hidden Neuron	Learning Rate	Epoch	MSE	Waktu (millisecond)
4	0.4	100000	0.0000004338	96930
4	0.4	200000	0.0000000552	212653
4	0.4	300000	0.0000000084	340264
4	0.4	400000	0.0000000090	414311

Dari hasil pengujian dengan nilai *epoch* maksimum yang berbeda-beda didapatkan nilai MSE terkecil yaitu 0.0000000084 pada *epoch* maksimum 300000 dengan waktu 340264.0 *millisecond*.

Dari hasil nilai rata-rata MSE di Table 6.3 didapatkan grafik sebagai berikut.



Gambar 6.3 Grafik Pengujian Perbedaan Jumlah *Epoch* Maksimum

Hasil pengujian perbedaan jumlah *epoch* maksimum yang telah dilakukan digambarkan pada grafik gambar 6.3, pada kasus ini memberikan hasil bahwa semakin besar *epoch* yang di gunakan maka akan membuat proses learning semakin lama tetapi akan menghasilkan nilai MSE yang semakin bagus dan mendekati nilai error yang telah ditentukan. Sebaliknya semakin kecil nilai *epoch* maksimum yang ditentukan maka pelatihan akan lebih singkat dan belum memenuhi nilai *error* yang ditentukan. Pada kasus ini nilai MSE terkecil yaitu 0.0000000084 yang didapat pada nilai *epoch* maksimum 300000 Sehingga pada kasus ini didapatkan nilai *epoch* maksimum yang baik adalah berjumlah 300000.

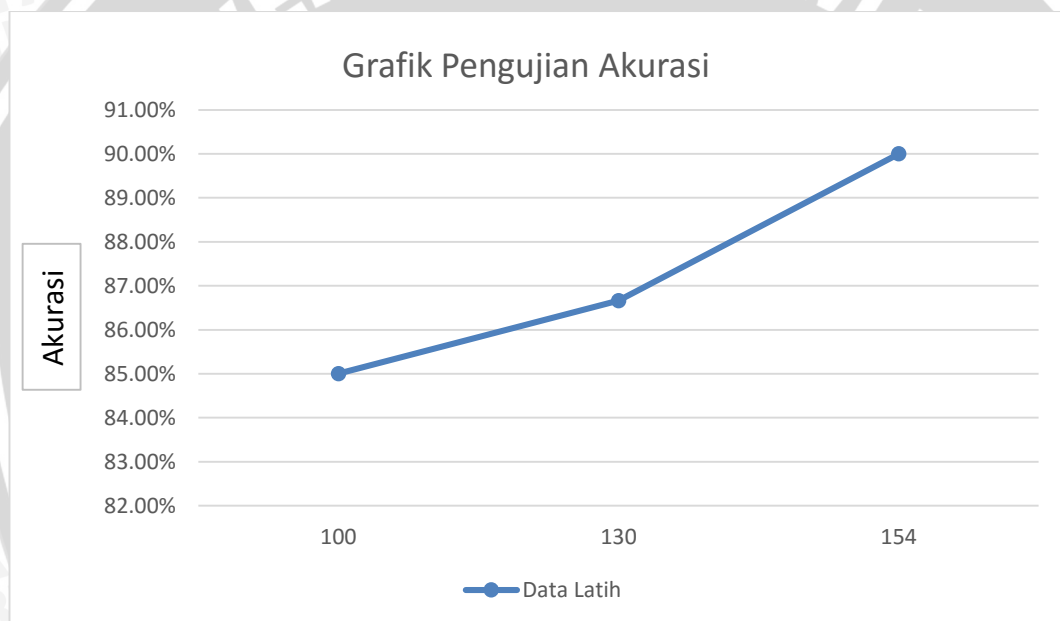
6.1.4 Pengujian dan Analisis Akurasi Sistem

Pengujian ini dilakukan dengan membandingkan hasil keluaran sistem dengan perbandingan data latih sebanyak 50%,60% dan 70% dari total jumlah data dan diuji dengan 60 data uji . Table 6.4 merupakan table hasil pengujian Akurasi

Table 6.4 Table Hasil Pengujian Akurasi

Jumlah Data Latih	MSE error learning	Akurasi
50%(100 data latih)	0.152449363558594	85.00%
60%(130 data latih)	0.000000006900194	86.67%
70%(154 data latih)	0.00000000099710	90.00%

Berdasarkan table 6.4 didapatkan grafik seperti gambar



Gambar 6.4 Grafik Pengujian Akurasi

Gambar 6.4 merupakan grafik hasil pengujian akurasi yang dilakukan dengan perbandingan data latih sebanyak 65%,85% dan 100%, didapatkan bahwa semakin banyak data latih membuat proses learning semakin bagus dan akan mempengaruhi tingkat akurasi. Akurasi sistem ini juga dipengaruhi oleh data uji yang memiliki gejala sama tetapi hasil diagnosisnya berbeda sehingga dari sistem ini tidak bisa maksimal. Pada kasus ini rata-rata nilai akurasi yang didapatkan ketiganya ialah 87.22%. sehingga berdasarkan akurasi yang di dapat pada kasus ini metode yang digunakan sangat bagus untuk menyelesaikan kasus ini. hal ini dikarenakan metode ini dapat mempelajari data yang sudah ada dan dapat memberikan hasil diagnosis yang sesuai.

BAB 7 PENUTUP

Bab ini mengemukakan kesimpulan dan saran dari penelitian yang telah dilakukan berdasarkan hasil perancangan, implementasi dan pengujian sistem.

7.1 Kesimpulan

Hal – hal yang dapat disimpulkan pada penelitian ini sebagai berikut:

1. Penerapan metode *backpropagation* pada diagnosis penyakit kulit pada anak dilakukan dengan melakukan normalisasi data secara manual dengan nilai 0 dan 1, lalu melakukan pembobotan dengan algoritma *Nguyen-Widrow* dan selanjutnya masuk pada proses *learning* dengan metode *Backpropagation*.
2. Nilai akurasi yang dihasilkan oleh sistem dengan menggunakan metode *backpropagation* adalah 87.22% sehingga sesuai untuk di implementasikan pada diagnosis penyakit kulit pada anak.
3. Pengaruh jumlah *hidden neuron* terhadap implementasi jaringan syaraf tiruan *backpropagation* untuk mendiagnosis penyakit kulit pada anak dengan ini didapatkan bahwa semakin banyak jumlah *hidden neuron* yang digunakan maka waktu *learning* akan semakin lama.
4. Pada penelitian ini didapatkan nilai *learning rate* yang optimal, yaitu 0.4 yang memiliki hasil nilai MSE paling kecil dibandingkan dengan nilai *learning rate* 0.1, 0.2, 0.3, 0.5, 0.6, 0.7, 0.8 dan 0.9.
5. Jumlah *epoch* maksimum yang digunakan sangat berpengaruh terhadap *learning* yang dilakukan oleh sistem. Jumlah *epoch* maksimum merupakan salah satu *stopping condition* untuk mengetahui apakah sebelum mencapai *epoch* maksimum data sudah konvergen.

7.2 Saran

Saran yang dapat diberikan oleh peneliti untuk penelitian selanjutnya ialah:

1. Pada penelitian selanjutnya dapat menggunakan data yang memiliki gejala lebih banyak agar dapat membedakan penyakit yang diderita dan meningkatkan akurasi sistem yang sudah ada dan data yang lebih banyak agar proses *learning* lebih akurat .
2. Pada penelitian selanjutnya dapat dilakukan modifikasi pada diagnosis penyakit kulit pada anak ini, agar proses *learning* yang dilakukan tidak memerlukan *epoch* yang besar untuk mencapai konvergensi.

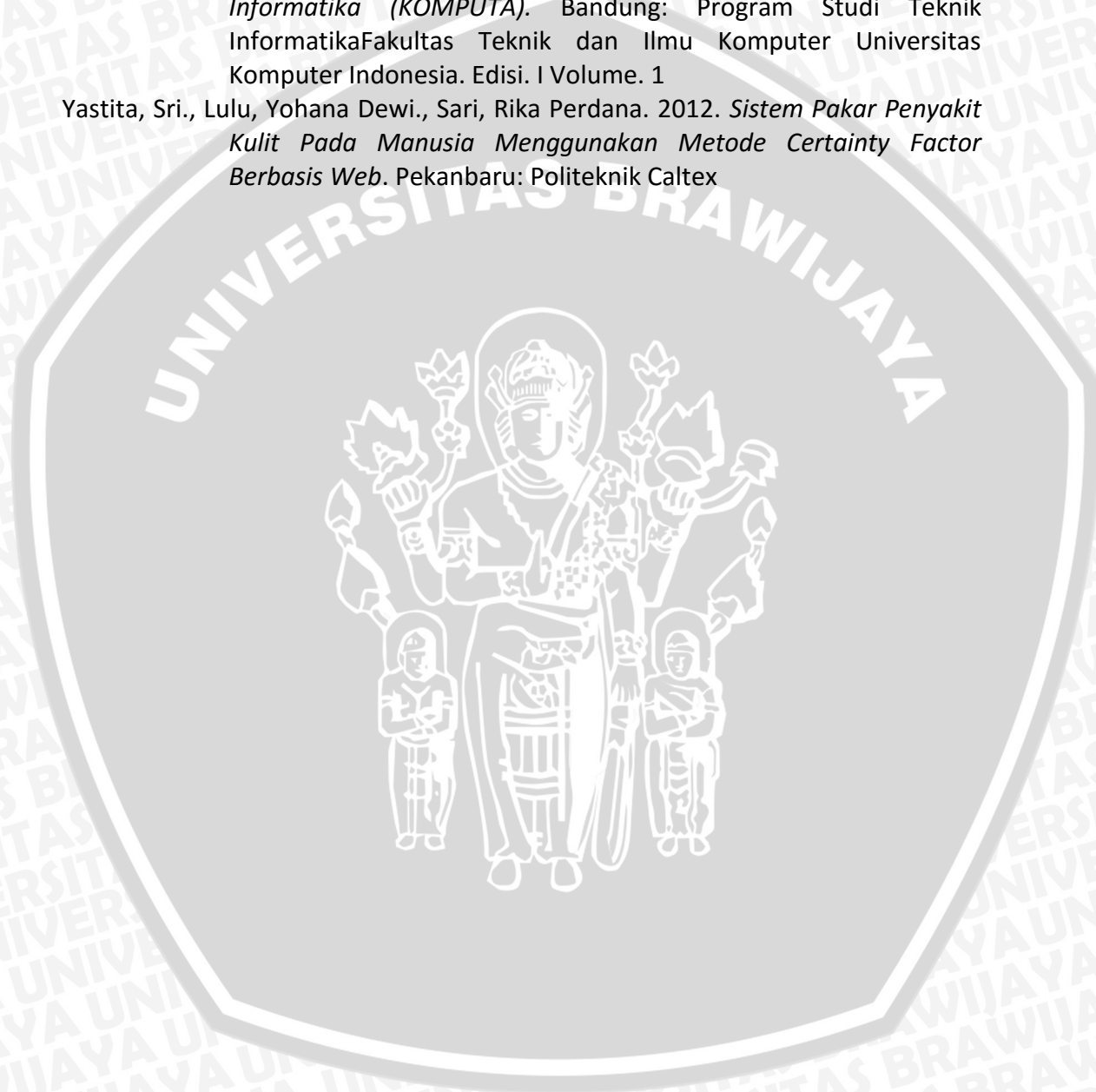
DAFTAR PUSTAKA

- Adrian Yudhy, M.Rhifky. 2014. *Analisis Algoritma Inisialisasi Nguyen Widrow Pada Proses Prediksi Curah Hujan Kota Medan Menggunakan Metode Backpropagation Neural Network*. Medan: STIMIK Potensi Utama.
- Agustin, Maria., Prahasto, Toni. 2012. *Penggunaan Jaringan Syaraf Tiruan Backpropagation Untuk Seleksi Penerimaan Mahasiswa Baru Pada Jurusan Teknik Komputer Di Politeknik Negeri Sriwijaya*. Palembang: Politeknik Negeri Sriwijaya
- Aryasa, Komang. 2012. *Expert Sistem Diagnosa Jenis Penyakit Gigi Menggunakan Jaringan Syaraf Tiruan Backpropagation*. Makasar: STIMIK Dipanegara.
- Boediardja, Siti Aisah., dkk. 2005. *Infeksi Kulit pada Bayi dan Anak*. Fakultas Kedokteran Universitas Indonesia. Jakarta.
- Dewi, Candra., Muslikh, M. 2013. *Perbandingan Akurasi Backpropagation Neural Network dan ANFIS untuk Memprediksi Cuaca*. Program Studi Matematika Universitas Brawijaya. Malang
- Fadilah, Annisa Nurul., Destiani, Dini., Dhamiri, Dhami Johar. 2012. *Perancangan Aplikasi Sistem Pakar Penyakit Kulit Pada Anak Dengan Metode Expert System Developmen Life Cycle*. Garut: STT Garut.
- Hartati, Sri., Iswanti, Sari. 2008. *Sistem Pakar dan Pengembangannya*. Graha Ilmu. Yogyakarta.
- Jamaludin. 2014. *Penyakit kulit dan cara pengobatannya*. (online)<https://penyakitkulitdanpengobatannya.wordpress.com/>. Diakses pada tanggal 3 18 november 2016).
- Kusumadewi, Sri. 2004. *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK*. Yogyakarta: Graha Ilmu.
- Matondang, Zekson Arizona. 2013. *Jaringan Syaraf Tiruan Dengan Algoritma Backpropagation Untuk Penentuan Kelulusan Sidang Skripsi*. *Jurnal Pelita Informatika Budi Darma*. Medan: STIMIK Budi Darma. Volume IV. Nomor 1.
- Mursinah. 2010. *Pengaruh Usia dan Waktu Pengambilan Sampel pada Surveilans Campak Berbasis Kasus (CBMD) di Pulau Sumatera dan DKI Jakarta Tahun 2009*. Kementrian Kesehatan, Jakarta
- Patel, Miss Ankeeta R., Joshi, Maulin M. 2013. *Heart diseases diagnosis using Neural Network*. India: EC Departement
- Pudjiadi, Antonius H. dkk. *Pedoman Pelayanan Medis Ikatan Dokter Anak Indonesia*. 2009.
- Srimuddawamah, Ika., 2015. *Sistem Pakar Diagnosa Penyakit Kulit Pada Anak Menggunakan Metode Naive Bayes*. Fakultas Ilmu Komputer Universitas Brawijaya. Malang.
- Sutoji, T., Mulyanto, Edy., Suhartono, Vincent. 2011. *Kecerdasan Buatan*. C.V Andi Offset (Penerbit ANDI). Yogyakarta.

Wiranti, Ratna Dwi. 2013. Metode Pembelajaran dan Fungsi Aktivasi JST. (Online) (http://r_d_w-fst10.web.unair.ac.id/artikel_detail-76032-Umum-metode%20Pembelajaran%20dan%20Fungsi%20Aktivasi%20JST.html). diakses tanggal 12 desember 2016)

Wuryandari, Maharani Dessy., Afrianto, Irawan. 2012. Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation dan Learning Vector Quantization Pada Pengenalan Wajah. *Jurnal Komputer dan Informatika (KOMPUTA)*. Bandung: Program Studi Teknik Informatika Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia. Edisi. I Volume. 1

Yastita, Sri., Lulu, Yohana Dewi., Sari, Rika Perdana. 2012. *Sistem Pakar Penyakit Kulit Pada Manusia Menggunakan Metode Certainty Factor Berbasis Web*. Pekanbaru: Politeknik Caltex



LAMPIRAN 1

Pengujian dengan 65% data latih dari 154 data latih:

Pada table dibawah ini merupakan pengujian dengan 60 data uji dengan menggunakan hasil bobot akhir dari learning 100 data latih yang telah di lakukan. Pengujian ini menggunakan hasil dari pengujian terbaik sebelumnya, yaitu jumlah *hidden neuron* 4, nilai *learning rate* 0.4, dan jumlah maksimum epoch 300000.

Table Pengujian dengan Data Latih 100

Data ke	target node output (1,2,3)	keluaran node output (1,2,3)	Error	Hasil Sistem	Hasil Poliklinik
1	1,0,1	0,1,1	0.001118623196631	Herpes	Campak
2	1,0,1	1,0,1	0.000000526079388	Campak	Campak
3	1,0,1	0,1,1	0.000107207084329	Herpes	Campak
4	1,0,0	0,1,1	0.333122142646307	Herpes	Dermatitis
5	1,0,0	1,0,0	0.000000719032508	Dermatitis	Dermatitis
6	1,0,0	1,0,0	0.000005004048013	Dermatitis	Dermatitis
7	1,0,0	0,1,1	0.373071868434416	Herpes	Dermatitis
8	1,0,0	0,1,1	0.330863405485442	Herpes	Dermatitis
9	0,1,1	0,1,1	0.000000085166775	Herpes	Herpes
10	0,1,1	0,1,1	0.000002502942126	Herpes	Herpes
11	0,1,1	0,1,1	0.000000000330360	Herpes	Herpes
12	0,1,1	0,1,1	0.000000002552303	Herpes	Herpes
13	0,1,1	0,1,1	0.000000056904568	Herpes	Herpes
14	0,1,1	0,1,1	0.000001393821013	Herpes	Herpes
15	0,1,1	1,0,0	0.100376246983804	Dermatitis	Herpes
16	0,1,1	0,1,1	0.000030798789624	Herpes	Herpes
17	1,1,1	1,1,1	0.000000180374452	Cacar Air	Cacar Air
18	1,1,1	1,1,1	0.000000176274815	Cacar Air	Cacar Air
19	1,1,1	1,1,1	0.000001138522226	Cacar Air	Cacar Air
20	1,1,1	1,1,1	0.000000820744194	Cacar Air	Cacar Air
21	1,1,1	1,1,1	0.000000346147137	Cacar Air	Cacar Air
22	1,1,0	1,1,0	0.000000895276617	Skabies	Skabies
23	1,1,0	1,1,0	0.000000895276617	Skabies	Skabies

24	1,1,0	1,1,0	0.000000000018699	Skabies	Skabies
25	1,1,0	1,1,0	0.000000000018699	Skabies	Skabies
26	1,1,0	1,1,0	0.000000000018699	Skabies	Skabies
27	1,0,1	1,0,1	0.000000108865156	Campak	Campak
28	1,0,1	1,0,0	0.278919239282509	Dermatitis	Campak
29	1,0,1	1,0,1	0.000000035340506	Campak	Campak
30	1,0,1	1,0,1	0.000000526079388	Campak	Campak
31	1,0,1	1,0,1	0.000000016429275	Campak	Campak
32	1,0,1	1,0,1	0.000000526079388	Campak	Campak
33	1,0,0	0,1,1	0.324690669222861	Herpes	Dermatitis
34	1,0,0	1,0,0	0.000000000612488	Dermatitis	Dermatitis
35	1,0,0	1,0,0	0.037726069872421	Dermatitis	Dermatitis
36	1,0,0	1,0,0	0.037726069872421	Dermatitis	Dermatitis
37	0,1,1	0,1,1	0.000000021205910	Herpes	Herpes
38	0,1,1	1,0,0	0.100376246983804	Dermatitis	Herpes
39	0,1,1	0,1,1	0.000000000104000	Herpes	Herpes
40	0,1,1	0,1,1	0.000000005480612	Herpes	Herpes
41	0,1,0	0,1,0	0.000000104946785	Abses	Abses
42	0,1,0	0,1,0	0.000001444309250	Abses	Abses
43	0,1,0	0,1,0	0.000000340167550	Abses	Abses
44	0,1,0	0,1,0	0.000000000543418	Abses	Abses
45	1,1,1	1,1,1	0.000000199372084	Cacar Air	Cacar Air
46	1,1,1	1,1,1	0.000000231486395	Cacar Air	Cacar Air
47	1,1,1	1,1,1	0.000000570270838	Cacar Air	Cacar Air
48	1,1,1	1,1,1	0.000000242799483	Cacar Air	Cacar Air
49	1,1,0	1,1,0	0.000000012866865	Skabies	Skabies
50	1,1,0	1,1,0	0.000000895276617	Skabies	Skabies
51	1,1,0	1,1,0	0.000000012866865	Skabies	Skabies
52	1,1,0	1,1,0	0.000000895276617	Skabies	Skabies
53	1,0,1	1,0,1	0.000000270173162	Campak	Campak
54	0,1,0	0,1,0	0.000003308735392	Abses	Abses
55	0,1,0	0,1,0	0.000000340167550	Abses	Abses
56	0,1,0	0,1,0	0.000000246819022	Abses	Abses

57	1,0,0	1,0,0	0.000000000612488	Dermatitis	Dermatitis
58	1,0,0	1,0,0	0.000000000612488	Dermatitis	Dermatitis
59	1,0,0	1,0,0	0.000000000612488	Dermatitis	Dermatitis
60	0,1,0	0,1,0	0.000000096947447	Abses	Abses

Dari hasil diatas didapatkan data keluaran sistem yang tidak sesuai dengan target yang seharusnya berjumlah 9 dan yang sesuai dengan target yang seharusnya berjumlah 51. Dari hasil tersebut kemudian dilakukan perhitungan nilai akurasi:

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Jumlah data yang sesuai}}{\text{Jumlah data uji keseluruhan}} \times 100\% \\
 &= \frac{51}{60} \times 100\% \\
 &= 85.00\%
 \end{aligned}$$

Pengujian dengan 85% data latih dari 154 data:

Pada table dibawah ini merupakan pengujian dengan 60 data uji dengan menggunakan hasil bobot akhir dari learning 130 data latih yang telah di lakukan. Pengujian ini menggunakan hasil dari pengujian terbaik sebelumnya, yaitu jumlah *hidden neuron* 4, nilai *learning rate* 0.4, dan jumlah maksimum epoch 300000.

Table Pengujian dengan Data Latih 130

Data ke	target node output (1,2,3)	keluaran node output (1,2,3)	Error	Hasil Sistem	Hasil Poliklinik
1	1,0,1	1,0,1	0.000001254877189	Campak	Campak
2	1,0,1	1,0,1	0.000000046524680	Campak	Campak
3	1,0,1	0,1,1	0.000000653675892	Herpes	Campak
4	1,0,0	1,1,1	1.329532945831860	Cacar Air	Dermatitis
5	1,0,0	1,0,0	0.000000016680886	Dermatitis	Dermatitis
6	1,0,0	1,0,0	0.000000016252209	Dermatitis	Dermatitis
7	1,0,0	1,0,1	0.332041077644122	Campak	Dermatitis
8	1,0,0	1,1,1	0.432003943586456	Cacar Air	Dermatitis
9	0,1,1	0,1,1	0.000000016309957	Herpes	Herpes
10	0,1,1	0,1,1	0.000000026643162	Herpes	Herpes
11	0,1,1	0,1,1	0.000000050309953	Herpes	Herpes
12	0,1,1	0,1,1	0.000000014358132	Herpes	Herpes
13	0,1,1	0,1,1	0.000000014060549	Herpes	Herpes

14	0,1,1	0,1,1	0.000000393580506	Herpes	Herpes
15	0,1,1	1,0,0	0.178670835387101	Dermatitis	Herpes
16	0,1,1	0,1,1	0.000000045575445	Herpes	Herpes
17	1,1,1	1,1,1	0.000000012210985	Cacar Air	Cacar Air
18	1,1,1	1,1,1	0.000000002821111	Cacar Air	Cacar Air
19	1,1,1	1,1,1	0.000000013882992	Cacar Air	Cacar Air
20	1,1,1	1,1,1	0.0000000248658617	Cacar Air	Cacar Air
21	1,1,1	1,1,1	0.0000000211877643	Cacar Air	Cacar Air
22	1,1,0	1,1,0	0.0000000571967823	Skabies	Skabies
23	1,1,0	1,1,0	0.0000000571967823	Skabies	Skabies
24	1,1,0	1,1,0	0.000000003640281	Skabies	Skabies
25	1,1,0	1,1,0	0.000000003640281	Skabies	Skabies
26	1,1,0	1,1,0	0.000000003640281	Skabies	Skabies
27	1,0,1	1,0,1	0.000000018617246	Campak	Campak
28	1,0,1	1,0,0	0.319685075758848	Dermatitis	Campak
29	1,0,1	1,0,1	0.000000034836623	Campak	Campak
30	1,0,1	1,0,1	0.000000046524680	Campak	Campak
31	1,0,1	1,0,1	0.000000041119137	Campak	Campak
32	1,0,1	1,0,1	0.000000046524680	Campak	Campak
33	1,0,0	0,1,1	0.119296740452479	Herpes	Dermatitis
34	1,0,0	1,0,0	0.0000000268293437	Dermatitis	Dermatitis
35	1,0,0	1,0,0	0.039589905638860	Dermatitis	Dermatitis
36	1,0,0	1,0,0	0.039589905638860	Dermatitis	Dermatitis
37	0,1,1	0,1,1	0.000000006196773	Herpes	Herpes
38	0,1,1	1,0,0	0.178670835387101	Dermatitis	Herpes
39	0,1,1	0,1,1	0.000000007580468	Herpes	Herpes
40	0,1,1	0,1,1	0.000000032791536	Herpes	Herpes
41	0,1,0	0,1,0	0.000000061241207	Abses	Abses
42	0,1,0	0,1,0	0.0000000101964291	Abses	Abses
43	0,1,0	0,1,0	0.0000000314440817	Abses	Abses
44	0,1,0	0,1,0	0.000000033617863	Abses	Abses
45	1,1,1	1,1,1	0.000000014268942	Cacar Air	Cacar Air
46	1,1,1	1,1,1	0.000000047126710	Cacar Air	Cacar Air

47	1,1,1	1,1,1	0.000000002614840	Cacar Air	Cacar Air
48	1,1,1	1,1,1	0.000000005393837	Cacar Air	Cacar Air
49	1,1,0	1,1,0	0.000000002341846	Skabies	Skabies
50	1,1,0	1,1,0	0.0000000571967823	Skabies	Skabies
51	1,1,0	1,1,0	0.000000002341846	Skabies	Skabies
52	1,1,0	1,1,0	0.0000000571967823	Skabies	Skabies
53	1,0,1	1,0,1	0.000000042176734	Campak	Campak
54	0,1,0	0,1,0	0.000001071542560	Abses	Abses
55	0,1,0	0,1,0	0.000000314440817	Abses	Abses
56	0,1,0	0,1,0	0.000000105810522	Abses	Abses
57	1,0,0	1,0,0	0.000000268293437	Dermatitis	Dermatitis
58	1,0,0	1,0,0	0.000000268293437	Dermatitis	Dermatitis
59	1,0,0	1,0,0	0.000000268293437	Dermatitis	Dermatitis
60	0,1,0	0,1,0	0.000000816830512	Abses	Abses

Dari hasil diatas didapatkan data keluaran sistem yang tidak sesuai dengan target yang seharusnya berjumlah 8 dan yang sesuai dengan target yang seharusnya berjumlah 52. Dari hasil tersebut kemudian dilakukan perhitungan nilai akurasi:

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Jumlah data yang sesuai}}{\text{Jumlah data uji keseluruhan}} \times 100\% \\
 &= \frac{52}{60} \times 100\% \\
 &= 86.66\%
 \end{aligned}$$

Pengujian dengan 100% data latih dari 154 data:

Pada table dibawah ini merupakan pengujian dengan 60 data uji dengan menggunakan hasil bobot akhir dari learning 154 data latih yang telah di lakukan. Pengujian ini menggunakan hasil dari pengujian terbaik sebelumnya, yaitu jumlah *hidden neuron* 4, nilai *learning rate* 0.4, dan jumlah maksimum epoch 300000.

Table Pengujian dengan Data Latih 154

Data ke	target node output (1,2,3)	keluaran node output (1,2,3)	Error	Hasil Sistem	Hasil Poliklinik
1	1,0,1	0,1,1	0.000050283755702	Herpes	Campak
2	1,0,1	1,0,1	0.000000003122131	Campak	Campak
3	1,0,1	0,1,1	0.000001756046484	Herpes	Campak

4	1,0,0	1,1,0	0.440367738620087	Skabies	Dermatitis
5	1,0,0	1,0,0	0.000003331588870	Dermatitis	Dermatitis
6	1,0,0	1,0,0	0.000004672302405	Dermatitis	Dermatitis
7	1,0,0	0,1,1	0.325195515501824	Herpes	Dermatitis
8	1,0,0	1,0,0	0.160567641552109	Dermatitis	Dermatitis
9	0,1,1	0,1,1	0.000000338027357	Herpes	Herpes
10	0,1,1	0,1,1	0.000000371661632	Herpes	Herpes
11	0,1,1	0,1,1	0.000001253372725	Herpes	Herpes
12	0,1,1	0,1,1	0.000000317284096	Herpes	Herpes
13	0,1,1	0,1,1	0.000000316604845	Herpes	Herpes
14	0,1,1	0,1,1	0.000022620802259	Herpes	Herpes
15	0,1,1	0,1,1	0.044055391001333	Herpes	Herpes
16	0,1,1	0,1,1	0.000011672931719	Herpes	Herpes
17	1,1,1	1,1,1	0.000006260202949	Cacar Air	Cacar Air
18	1,1,1	1,1,1	0.000003950125707	Cacar Air	Cacar Air
19	1,1,1	1,1,1	0.000004777661515	Cacar Air	Cacar Air
20	1,1,1	1,1,1	0.000024465658692	Cacar Air	Cacar Air
21	1,1,1	1,1,1	0.000003345838168	Cacar Air	Cacar Air
22	1,1,0	1,1,0	0.000000304982696	Skabies	Skabies
23	1,1,0	1,1,0	0.000000304982696	Skabies	Skabies
24	1,1,0	1,1,0	0.000012224506522	Skabies	Skabies
25	1,1,0	1,1,0	0.000012224506522	Skabies	Skabies
26	1,1,0	1,1,0	0.000012224506522	Skabies	Skabies
27	1,0,1	1,0,1	0.000000386996673	Campak	Campak
28	1,0,1	1,0,0	0.330607231251365	Dermatitis	Campak
29	1,0,1	1,0,1	0.000000000000013	Campak	Campak
30	1,0,1	1,0,1	0.000000003122131	Campak	Campak
31	1,0,1	1,0,1	0.000011411850706	Campak	Campak
32	1,0,1	1,0,1	0.000000003122131	Campak	Campak
33	1,0,0	1,1,0	0.265671459842441	Skabies	Dermatitis
34	1,0,0	1,0,0	0.000023219456801	Dermatitis	Dermatitis
35	1,0,0	1,0,0	0.037489774809384	Dermatitis	Dermatitis
36	1,0,0	1,0,0	0.037489774809384	Dermatitis	Dermatitis

37	0,1,1	0,1,1	0.026413396984159	Herpes	Herpes
38	0,1,1	0,1,1	0.044055391001333	Herpes	Herpes
39	0,1,1	0,1,1	0.000000306245342	Herpes	Herpes
40	0,1,1	0,1,1	0.000000441701273	Herpes	Herpes
41	0,1,0	0,1,0	0.000000005858945	Abses	Abses
42	0,1,0	0,1,0	0.000000172310102	Abses	Abses
43	0,1,0	0,1,0	0.000003100140972	Abses	Abses
44	0,1,0	0,1,0	0.000000018005092	Abses	Abses
45	1,1,1	1,1,1	0.000004294065504	Cacar Air	Cacar Air
46	1,1,1	1,1,1	0.000002630692271	Cacar Air	Cacar Air
47	1,1,1	1,1,1	0.000007217274367	Cacar Air	Cacar Air
48	1,1,1	1,1,1	0.000003493476207	Cacar Air	Cacar Air
49	1,1,0	1,1,0	0.000002776602688	Skabies	Skabies
50	1,1,0	1,1,0	0.000000304982696	Skabies	Skabies
51	1,1,0	1,1,0	0.000002776602688	Skabies	Skabies
52	1,1,0	1,1,0	0.000000304982696	Skabies	Skabies
53	1,0,1	1,0,1	0.000043393414063	Campak	Campak
54	0,1,0	0,1,0	0.000199293143574	Abses	Abses
55	0,1,0	0,1,0	0.000003100140972	Abses	Abses
56	0,1,0	0,1,0	0.000000494935322	Abses	Abses
57	1,0,0	1,0,0	0.000023219456801	Dermatitis	Dermatitis
58	1,0,0	1,0,0	0.000023219456801	Dermatitis	Dermatitis
59	1,0,0	1,0,0	0.000023219456801	Dermatitis	Dermatitis
60	0,1,0	0,1,0	0.000000007389187	Abses	Abses

Dari hasil diatas didapatkan data keluaran sistem yang tidak sesuai dengan target yang seharusnya berjumlah 6 dan yang sesuai dengan target yang seharusnya berjumlah 54. Dari hasil tersebut kemudian dilakukan perhitungan nilai akurasi:

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Jumlah data yang sesuai}}{\text{Jumlah data uji keseluruhan}} \times 100\% \\
 &= \frac{54}{60} \times 100\% \\
 &= 90.00\%
 \end{aligned}$$

LAMPIRAN 2

DATA LATIH DAN DATA UJI PENYAKIT KULIT PADA ANAK



Tabel Data Latih Penyakit Kulit Pada Anak (Poliklinik Sang Timur Sumenep)

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	Target
1	1	1	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	Cacar Air
2	1	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	Cacar Air
3	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	Cacar Air
4	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	Skabies
5	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	Skabies
6	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	Campak
7	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	Campak
8	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Campak
9	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	Campak
10	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Campak
11	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	Dermatitis
12	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	Dermatitis
13	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	Dermatitis
14	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	Herpes
15	0	0	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	Herpes
16	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	Herpes
17	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	Herpes
18	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	Herpes
19	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	Herpes
20	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	Abses
21	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	Abses
22	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	Abses
23	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1	0	Abses
24	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	Skabies
25	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	Skabies

26	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	Skabies
27	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	Skabies
28	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	Dermatitis
29	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	Dermatitis
30	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	Dermatitis
31	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	Dermatitis
32	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	Dermatitis
33	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	Cacar Air
34	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	Cacar Air
35	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	Cacar Air
36	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	Cacar Air
37	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	Abses
38	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	Abses
39	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	Abses
40	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	Abses
41	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	Herpes
42	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	Herpes
43	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	Herpes
44	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	Herpes
45	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	Herpes
46	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	Campak
47	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	Campak
48	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	Campak
49	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	Campak
50	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Campak
51	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	Campak
52	0	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	Campak

53	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
54	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	Cacar Air
55	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
56	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	Cacar Air
57	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
58	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	Skabies
59	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	Skabies
60	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	Skabies
61	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
62	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	Herpes
63	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	Herpes
64	0	1	0	1	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	Herpes
65	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	Dermatitis
66	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	Dermatitis
67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	Dermatitis
68	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	Dermatitis
69	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	Dermatitis
70	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	Abses
71	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	Abses
72	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	Abses
73	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	Abses
74	1	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
75	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
76	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
77	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	Abses
78	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	Abses
79	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	Abses

80	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	Abses
81	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	Abses
82	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	Campak
83	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	Campak
84	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	Campak
85	0	1	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	Campak
86	0	0	0	1	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	Skabies
87	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	Skabies
88	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	Skabies
89	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	Skabies
90	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
91	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
92	1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Cacar Air
93	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
94	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	Skabies
95	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	Skabies
96	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	Skabies
97	0	0	0	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	Herpes
98	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	Herpes
99	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	Herpes
100	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	Herpes
101	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	Herpes
102	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	Abses
103	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	Abses
104	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	Abses
105	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	Abses
106	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	Herpes

107	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	Herpes
108	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Campak
109	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Campak
110	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Campak
111	1	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	Cacar Air
112	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
113	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	Cacar Air
114	1	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
115	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	Skabies
116	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	Skabies
117	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	Skabies
118	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	Skabies
119	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
120	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
121	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	Cacar Air
122	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
123	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	Campak
124	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Campak
125	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	Campak
126	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	Dermatitis
127	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	Dermatitis
128	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	Dermatitis
129	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	Dermatitis
130	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	Dermatitis
131	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	Herpes
132	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	Herpes
133	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	Herpes

134	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	Dermatitis
135	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	Dermatitis
136	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	Dermatitis
137	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	Dermatitis
138	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	Herpes
139	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	Herpes
140	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	Herpes
141	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	Herpes
142	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	Herpes
143	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	Herpes
144	0	0	0	1	0	0	1	1	1	1	0	0	0	1	0	0	0	0	0	0	Abses
145	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	Abses
146	0	1	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	Abses
147	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	Cacar Air
148	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
149	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
150	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
151	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
152	0	0	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	Campak
153	0	1	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	Campak
154	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	Campak
155	0	1	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	Campak
156	0	0	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	Campak
157	0	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	Campak
158	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	Dermatitis
159	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	Dermatitis
160	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	Dermatitis

161	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	Dermatitis
162	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	Dermatitis
163	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0	Herpes
164	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	Herpes
165	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	Herpes
166	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	Herpes
167	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	Herpes
168	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	Herpes
169	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	Herpes
170	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	Herpes
171	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	Cacar Air
172	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
173	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
174	1	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	Cacar Air
175	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	Cacar Air
176	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	Skabies
177	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	Skabies
178	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	Skabies
179	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	Skabies
180	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	Skabies
181	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Campak
182	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	Campak
183	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	Campak
184	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	Campak
185	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	Campak
186	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	Campak
187	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	Dermatitis

184	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	Campak
185	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	Campak
186	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	Campak
187	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	Dermatitis
188	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	Dermatitis
189	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	Dermatitis
190	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	Dermatitis
191	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1 Herpes
192	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1 Herpes
193	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1 Herpes
194	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1 Herpes
195	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0 Abses
196	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0 Abses
197	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0 Abses
198	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0 Abses
199	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Cacar Air
200	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0 Cacar Air
201	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Cacar Air
202	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0 Cacar Air
203	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0 Skabies
204	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0 Skabies
205	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0 Skabies
206	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0 Skabies
207	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0 Campak
208	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0 Abses
209	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0 Abses
210	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0 Abses

211	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	Dermatitis
212	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	Dermatitis
213	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	Dermatitis
214	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	Abses

KODE TARGET PENYAKIT

No	Nama Penyakit	Kode Penyakit	Jumlah data
1.	Cacar Air	111	19.63% (42 data)
2.	Scabies	110	13.55% (29 data)
3.	Campak	101	16.36% (35 data)
4.	Dermatitis	100	15.89% (34 data)
5.	Herpes	011	19.63% (42 data)
6.	Abses	010	14.95% (32 data)
Total			214 data

GEJALA-GEJALA PENYAKIT KULIT

GEJALA YANG DI DAPAT BERDASARKAN DATA LATIH							
kode	Gejala	Cacar air	Scabies	campak	dermatitis	herpes	abses
G1	Gelembung air merata	√					
G2	Demam	√		√	√	√	√

G3	Nyeri kepala	√		√		√	√
G4	Gatal	√	√	√	√	√	√
G5	Lesi seluruh tubuh	√					
G6	Mata merah	√		√	√		
G7	Batuk	√	√	√			
G8	Pilek	√	√	√			
G9	Meradang	√	√	√	√	√	√
G10	Nyeri tekan	√			√	√	√
G11	Gelembung nanah		√				
G12	Perih		√		√	√	
G13	bersisik			√	√		
G14	Melepuh				√	√	
G15	Panas				√		
G16	Bengkak				√		√
G17	Lesi di area tertentu					√	
G18	Benjolan nanah/pus						√
G19	Gelembung air di area tertentu					√	