

**PENERAPAN *ARTIFICIAL NEURAL NETWORK*
PADA PROSES KLASIFIKASI CURAH HUJAN
DI JAWA TENGAH**

SKRIPSI

**OLEH:
NUR KHOLIDA MAULIDIA
NIM. 19610055**



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

**PENERAPAN *ARTIFICIAL NEURAL NETWORK*
PADA PROSES KLASIFIKASI CURAH HUJAN
DI JAWA TENGAH**

SKRIPSI

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

**Oleh
NUR KHOLIDA MAULIDIA
NIM. 19610055**

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

**PENERAPAN *ARTIFICIAL NEURAL NETWORK*
PADA PROSES KLASIFIKASI CURAH HUJAN
DI JAWA TENGAH**

SKRIPSI

**Oleh
Nur Kholida Maulidia
NIM. 19610055**

**Telah Disetujui Untuk Diuji
Malang, 6 November 2023**

Dosen Pembimbing I

Dosen Pembimbing II



**Prof. Dr. Hj. Sri Harini, M.Si.
NIP. 19731014 200112 2 002**



**Juhari, M.Si.
NIP. 198840209 202321 1 010**

**Mengetahui,
Ketua Program Studi Matematika**



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

**PENERAPAN *ARTIFICIAL NEURAL NETWORK*
PADA PROSES KLASIFIKASI CURAH HUJAN
DI JAWA TENGAH**

SKRIPSI

Oleh
Nur Kholida Maulidia
NIM. 19610055

Telah Dipertahankan di Depan Dewan Penguji
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)
Tanggal, 20 November 2023

Ketua Penguji : Abdul Aziz, M.Si.

Anggota Penguji 1 : Angga Dwi Mulyanto, M.Si.

Anggota Penguji 2 : Prof. Dr. Hj. Sri Harini, M.Si.

Anggota Penguji 3 : Juhari, M.Si.



.....
.....
.....

Mengetahui,
Ketua Program Studi Matematika

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



PERNYATAAN KEASLIAN TULISAN

Saya yang bertandatangan di bawah ini:

Nama : Nur Kholida Maulidia
NIM : 19610055
Program Studi : Matematika
Fakultas : Sains dan Teknologi
Judul Skripsi : Penerapan *Artificial Neural Netwrok* pada Proses
Klasifikasi Curah Hujan di Jawa Tengah

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

Malang, 20 November 2023

Yang membuat pernyataan,


Nur Kholida Maulidia
NIM.19610055

MOTO

“Never let your emotions overpower your intelligence”

PERSEMBAHAN

Skripsi ini penulis persembahkan untuk:

Ayahanda Ahmad Nur Halim dan ibunda Siti Zubaidah yang senantiasa ikhlas mendoakan dan memberikan dukungan secara emosional dan finansial dalam pengerjaan skripsi ini, sahabat penulis Ariendha Lintang Fatichah yang selalu ada menjadi teman diskusi penulis, dan penulis sendiri yang mau dan bisa bertahan untuk menyelesaikan penulisan skripsi ini.

KATA PENGANTAR

Assalamu 'alaikum Warahmatullahi Wabarakatuh

Segala puji hanya milik Allah SWT, Tuhan semesta alam yang Maha Pengasih dan Maha Penyayang. *Shalawat* serta salam semoga selalu terlimpah kepada Rasulullah SAW, nabi dan utusan terakhir yang membawa rahmat bagi seluruh alam. Penulisan hasil skripsi ini dilandasi oleh keyakinan peneliti akan kebesaran dan kekuasaan Allah SWT yang selalu menuntun dan memberikan jalan keluar dalam setiap persoalan yang dihadapi.

Peneliti juga menyadari bahwa proses penulisan ini tidak mungkin terwujud tanpa adanya bimbingan dari berbagai pihak. Oleh karena itu, peneliti mengucapkan terima kasih kepada Allah SWT yang telah memberikan karunia-Nya dan melimpahkan bimbingan-Nya kepada peneliti. Selain itu, ucapan terima kasih juga ditujukan kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Elly Susanti, M.Sc., selaku ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Prof. Dr. Hj. Sri Harini, M.Si., selaku dosen pembimbing I yang selalu memberikan arahan, bimbingan, dan berbagi ilmu kepada peneliti agar dapat menyelesaikan proposal skripsi ini.
5. Juhari, M.Si., selaku dosen pembimbing II yang selalu memberikan arahan, bimbingan, dan berbagi ilmu kepada peneliti agar dapat menyelesaikan proposal skripsi ini.
6. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
7. Ucapan terima kasih yang tak terhingga kepada Ayah Halim dan Bunda Ida serta keluarga yang senantiasa memberikan doa, dukungan, dan motivasi selama peneliti menyelesaikan skripsi ini. Tanpa doa dan dukungan dari keluarga, peneliti tidak akan mampu mencapai tahap ini.

8. Seluruh mahasiswa angkatan 2019 yang telah menjadi bagian dari perjalanan pendidikan ini. Terima kasih atas dukungan, motivasi, dan kerja sama yang diberikan selama peneliti menempuh pendidikan di Universitas Islam Negeri Maulana Malik Ibrahim Malang.
9. Para sahabat seperjuangan, yang telah berbagi ilmu, pengalaman, dan juga tawa dalam perjalanan kuliah ini. Semoga Allah SWT senantiasa memberikan keberkahan dan kebahagiaan untuk kita semua.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 20 November 2023

Peneliti

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTO	vi
PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
مستخلص البحث	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	7
1.3 Tujuan Penelitian.....	7
1.4 Manfaat Penelitian.....	7
1.5 Batasan Masalah.....	8
1.6 Definisi Istilah	9
BAB II KAJIAN TEORI	11
2.1 Teori Pendukung	11
2.1.1 <i>Time Series</i>	11
2.1.2 Peramalan (<i>Forecasting</i>).....	12
2.1.3 <i>Artificial Neural Network</i>	13
2.1.3.1 Struktur ANN.....	14
2.1.3.2 Fungsi Aktivasi	16
2.1.3.3 <i>Loss Function</i>	17
2.1.3.4 Algoritma <i>Backpropagation Neural Network</i> (BPNN)	18
2.1.3.5 <i>Encoding Data</i>	24
2.1.3.6 Normalisasi Data	25
2.1.3.7 <i>Decoding Data</i>	25
2.1.3.8 Evaluasi Model	26
2.1.4 Curah Hujan	29
2.2 Kajian Integrasi Topik dengan Al-Quran/Hadits	33
BAB III METODE PENELITIAN	35
3.1 Jenis Penelitian	35
3.2 Data dan Sumber Data.....	35
3.3 Teknik Pengumpulan Data	36
3.4 Langkah Penelitian	36
3.5 <i>Flowchart</i>	39
BAB IV PEMBAHASAN	40
4.1 Analisis Deskriptif.....	40
4.2 Analisis Data	42

4.2.1	<i>Encoding</i> Data Kategorik.....	42
4.2.2	Normalisasi Data Numerik.....	43
4.2.3	Model ANN.....	44
4.2.3.1	Model Pertama.....	44
4.2.3.2	Model Kedua	56
4.2.3.3	Model Ketiga	66
4.2.4	Evaluasi Model ANN.....	78
4.2.4.1	Model Pertama.....	78
4.2.4.2	Model Kedua	81
4.2.4.3	Model Ketiga	84
4.2.5	Prediksi Hasil Klasifikasi Curah Hujan di Jawa Tengah	87
4.2.5.1	Model Pertama.....	87
4.2.5.2	Model Kedua	89
4.2.5.3	Model Ketiga	90
4.3	Integrasi Nilai-Nilai Agama dengan Hasil Penelitian.....	92
BAB V PENUTUP.....		94
5.1	Kesimpulan.....	94
5.2	Saran.....	95
DAFTAR PUSTAKA		96
LAMPIRAN.....		99
RIWAYAT HIDUP		150

DAFTAR GAMBAR

Gambar 2.1	<i>Single Layer Perceptron</i>	15
Gambar 2.2	<i>Multilayer Perceptron</i>	16
Gambar 3.1	<i>Flowchart Tahapan Prediksi Menggunakan ANN dengan Algoritma Backpropagation</i>	39
Gambar 4.1	Plot Suhu pada Tahun 2019-2022	40
Gambar 4.2	Plot Kecepatan Angin pada Tahun 2019-2022.....	40
Gambar 4.3	Plot Kelembaban pada Tahun 2019-2022	41
Gambar 4.4	Plot Tekanan Udara pada Tahun 2019-2022	41
Gambar 4.5	Plot Penyinaran Matahari pada Tahun 2019-2022	41
Gambar 4.6	Plot Hujan pada Tahun 2019-2022.....	42
Gambar 4.7	Plot Curah Hujan pada Tahun 2019-2022	42
Gambar 4.8	Model Jaringan E dengan Bobot Akhir	45
Gambar 4.9	Model Jaringan AQ dengan Bobot Akhir.....	57
Gambar 4.10	Model Jaringan DQ dengan Bobot Akhir.....	67
Gambar 4.11	<i>Confusion Matrix</i> Model E Menggunakan Data <i>Training</i>	78
Gambar 4.12	<i>Confusion Matrix</i> Model E Menggunakan Data <i>Testing</i>	80
Gambar 4.13	<i>Confusion Matrix</i> Model AQ Menggunakan Data <i>Training</i>	81
Gambar 4.14	<i>Confusion Matrix</i> Model AQ Menggunakan Data <i>Testing</i>	83
Gambar 4.15	<i>Confusion Matrix</i> Model DQ Menggunakan Data <i>Training</i>	84
Gambar 4.16	<i>Confusion Matrix</i> Model DQ Menggunakan Data <i>Testing</i>	86

DAFTAR TABEL

Tabel 2.1	<i>Confusion Matrix</i>	26
Tabel 3.1	Variabel Penelitian	35
Tabel 3.2	Kelas Curah Hujan	36
Tabel 3.3	Data Ilustrasi.....	37
Tabel 4.1	<i>Encoding</i> Data Kategorik	43
Tabel 4.2	Hasil <i>Hidden Layer</i> Sebelum Aktivasi.....	47
Tabel 4.3	Hasil <i>Hidden Layer</i> Setelah Aktivasi	48
Tabel 4.4	Hasil <i>Hidden Layer</i> 1 Sebelum Aktivasi	59
Tabel 4.5	Hasil <i>Hidden Layer</i> 1 Setelah Aktivasi	60
Tabel 4.6	Hasil <i>Hidden Layer</i> 2 Sebelum Aktivasi	61
Tabel 4.7	Hasil <i>Hidden Layer</i> 2 Setelah Aktivasi	62
Tabel 4.8	Hasil <i>Hidden Layer</i> 1 Sebelum Aktivasi	68
Tabel 4.9	Hasil <i>Hidden Layer</i> 1 Setelah Aktivasi	69
Tabel 4.10	Hasil <i>Hidden Layer</i> 2 Sebelum Aktivasi	70
Tabel 4.11	Hasil <i>Hidden Layer</i> 2 Setelah Aktivasi	71
Tabel 4.12	Hasil <i>Hidden Layer</i> 3 Sebelum Aktivasi	72
Tabel 4.13	Hasil <i>Hidden Layer</i> 3 Setelah Aktivasi	73
Tabel 4.14	Hasil Perulangan Model E Menggunakan Data <i>Training</i>	79
Tabel 4.15	Hasil Perulangan Model E Menggunakan Data <i>Testing</i>	81
Tabel 4.16	Hasil Perulangan Model AQ Menggunakan Data <i>Training</i>	82
Tabel 4.17	Hasil Perulangan Model AQ Menggunakan Data <i>Testing</i>	84
Tabel 4.18	Hasil Perulangan Model DQ Menggunakan Data <i>Training</i>	85
Tabel 4.19	Hasil Perulangan Model DQ Menggunakan Data <i>Testing</i>	87
Tabel 4.20	Hasil Prediksi Curah Hujan di Jawa Tengah Menggunakan Model E	88
Tabel 4.21	Hasil Prediksi Curah Hujan di Jawa Tengah Menggunakan Model AQ	89
Tabel 4.22	Hasil Prediksi Curah Hujan di Jawa Tengah Menggunakan Model DQ	91

ABSTRAK

Maulidia, Nur Kholida, 2023. **Penerapan *Artificial Neural Network* pada Proses Klasifikasi Curah Hujan di Jawa Tengah.** Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: I) Prof. Dr. Hj. Sri Harini, M.Si., II) Juhari, M.Si.

Kata Kunci: *Artificial Neural Network* (ANN), Klasifikasi, Curah Hujan, ReLu, *Softmax*.

Artificial Neural Network (ANN) adalah metode klasifikasi dan prediksi dimana memiliki cara kerja yang terinspirasi dari struktur dan fungsi sistem saraf manusia. Penelitian ini bertujuan untuk mengklasifikasikan curah hujan di wilayah Jawa Tengah menggunakan metode ANN, dengan mempertimbangkan kompleksitas pola cuaca dan faktor lingkungan yang mempengaruhi curah hujan. Pendekatan penelitian yaitu penerapan metode ANN yang mampu memodelkan hubungan kompleks dan *non-linier* antara variabel-variabel terkait curah hujan. Fungsi aktivasi ReLu digunakan pada *input* dan *hidden layer*, sementara fungsi aktivasi *softmax* digunakan pada *output layer*. Data dibagi menjadi 80% data *training* dan 20% data *testing*. Hasil penelitian menunjukkan bahwa model E, AQ, dan DQ merupakan model yang bisa digunakan untuk klasifikasi curah hujan. Model E dengan satu *hidden layer* memiliki arsitektur jaringan 8-20-4. Model AQ dengan dua *hidden layer* memiliki arsitektur 8-28-32-4. Model DQ dengan tiga *hidden layer* memiliki arsitektur jaringan 8-12-20-32-4. Evaluasi pada data *testing* dari ketiga model tersebut menghasilkan nilai MAE yang sama yaitu 0.198. Penelitian ini menyimpulkan bahwa metode ANN mampu mengklasifikasikan curah hujan di Jawa Tengah dengan tingkat akurasi tinggi. Kontribusi penting penelitian ini ada dalam pemahaman dan klasifikasi curah hujan di wilayah Jawa Tengah. Hasilnya memiliki implikasi praktis dalam berbagai sektor kehidupan, termasuk pertanian, ketersediaan air, dan manajemen risiko bencana alam di wilayah tersebut.

ABSTRACT

Maulidia, Nur Kholida, 2023. **Implementation of Artificial Neural Network for Rainfall Classification Process in Central Java**. Thesis. Department of Mathematics, Faculty of Science and Technology, State Islamic University Maulana Malik Ibrahim Malang. Advisors: I) Prof. Dr. Hj. Sri Harini, M.Si., II) Juhari, M.Si.

Keywords: Artificial Neural Network, Classification, Rainfall, ReLu, *Softmax*.

Artificial Neural Network (ANN) is a classification and prediction method that has a way of working inspired by the structure and function of the human nervous system. This study aims to classify rainfall in the Central Java region using the ANN method, taking into account the complexity of weather patterns and environmental factors that affect rainfall. The research approach is the application of the ANN method which is able to model complex and non-linear relationships between rainfall-related variables. The ReLu activation function is used on the input and hidden layers, while the softmax activation function is used on the output layer. Data is divided into 80% training data and 20% testing data. The results showed that the E, AQ, and DQ models are models that can be used for rainfall classification. Model E with one hidden layer has an 8-20-4 network architecture. The AQ model with two hidden layers has an 8-28-32-4 architecture. The DQ model with three hidden layers has a network architecture of 8-12-20-32-4. Evaluation of the testing data from the three models resulted in the same MAE value of 0.198. This study concluded that the ANN method was able to classify rainfall in Central Java with a high degree of accuracy. An important contribution of this research is in the understanding and classification of rainfall in the Central Java region. The results have practical implications in various sectors of life, including agriculture, water availability, and natural disaster risk management in the region.

مستخلص البحث

مولدية، نورخالدة، ٢٠٢٠. تنفيذ الشبكة العصبية الاصطناعية (*ANN*) في عملية تصنيف هطول الأمطار في جاوى الوسطى. أطروحة. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانغ. المشرف الأولى : البروفيسور، الدكتور، سري هاريني ، الماجستير، الحاجة، المشرف الثاني : جوهرى، الماجستير.

الكلمات المفتاحية: الشبكة العصبية الاصطناعية ، التصنيف ، هطول الأمطار ، *Softmax* ، *ReLU*

الشبكة العصبية الاصطناعية (*ANN*) هي طريقة تصنيف وتنبؤ لها طريقة عمل مستوحاة من بنية ووظيفة الجهاز العصبي البشري. تهدف هذه الدراسة إلى تصنيف هطول الأمطار في منطقة جاوة الوسطى باستخدام طريقة *ANN* ، مع مراعاة تعقيد أنماط الطقس والعوامل البيئية التي تؤثر على هطول الأمطار. منهج البحث هو تطبيق طريقة *ANN* القادرة على نمذجة العلاقات المعقدة وغير الخطية بين المتغيرات المتعلقة بهطول الأمطار. يتم استخدام وظيفة تنشيط *ReLU* على طبقات الإدخال والمخفية ، بينما يتم استخدام وظيفة تنشيط *softmax* في طبقة الإخراج. تنقسم البيانات إلى ٨٠٪ من بيانات التدريب و ٢٠٪ من بيانات الاختبار. أظهرت النتائج أن نماذج E و AQ و DQ هي نماذج يمكن استخدامها لتصنيف هطول الأمطار. يحتوي النموذج E بطبقة مخفية واحدة على بنية شبكة ٨-٢٠-٤. يحتوي نموذج AQ مع طبقتين مخفيتين على بنية ٨-٢٨-٣٢-٤. يحتوي نموذج DQ بثلاث طبقات مخفية على بنية شبكة ٨-١٢-٢٠-٣٢-٤. أدى تقييم بيانات الاختبار من النماذج الثلاثة إلى نفس قيمة *MAE* البالغة ٠,١٩٨. خلصت هذه الدراسة إلى أن طريقة *ANN* كانت قادرة على تصنيف هطول الأمطار في جاوة الوسطى بدرجة عالية من الدقة. مساهمة مهمة لهذا البحث هي في فهم وتصنيف هطول الأمطار في منطقة جاوة الوسطى. وللتأثيرات آثار عملية في مختلف قطاعات الحياة، بما في ذلك الزراعة وتوافر المياه وإدارة مخاطر الكوارث الطبيعية في المنطقة.

BAB I PENDAHULUAN

1.1 Latar Belakang

Artificial Neural Network (ANN) atau sering disebut sebagai jaringan saraf tiruan adalah suatu metode dalam bidang kecerdasan buatan atau *Artificial Intelligence* (AI) yang menjadi bagian dari *Machine Learning* (ML) dan *Deep Learning* (DL). ANN adalah model komputasi yang terinspirasi dari cara kerja otak manusia dalam memproses informasi (Haykin, 2010). ANN terdiri dari beberapa lapisan *neuron* yang saling terhubung dan dapat belajar dari data untuk melakukan tugas-tugas seperti klasifikasi, regresi, pengenalan pola, dan prediksi (Haykin, 2009). Fungsi utama ANN adalah untuk memproses data masukan melalui serangkaian operasi matematis yang kompleks dan menghasilkan keluaran yang dihasilkan dari proses pembelajaran yang telah dilakukan (Goodfellow, 2016). ANN memiliki kemampuan untuk mengenali pola dalam data, melakukan pengklasifikasian objek atau data, serta melakukan prediksi hasil berdasarkan data yang telah diberikan (Bishop & Nasrabadi, 2006). Kelebihan ANN antara lain kemampuan untuk memproses data yang kompleks, adaptif dalam melakukan pembelajaran, serta dapat mengenali pola-pola yang sulit dikenali oleh manusia atau metode konvensional (Zhang dkk, 2020). ANN juga dapat digunakan untuk mengatasi berbagai tantangan dalam bidang kecerdasan buatan, seperti pengenalan wajah, deteksi anomali, pengenalan suara, dan lain sebagainya. Tugas ANN melibatkan pengenalan pola dalam data, pengklasifikasian objek atau data ke dalam kategori yang telah ditentukan, serta prediksi hasil berdasarkan data yang telah diberikan (Haykin, 2009). Penerapan ANN dapat digunakan dalam berbagai bidang

seperti pengenalan gambar, pengolahan bahasa alami, pengenalan suara, analisis data, dan lain sebagainya.

Tahapan dalam penggunaan ANN meliputi beberapa langkah. Pertama, pemilihan dan preprocessing data, yaitu memilih dataset yang sesuai dan melakukan pengolahan data seperti normalisasi, *encoding*, dan *feature selection* (Bishop & Nasrabadi, 2006). Kedua, desain arsitektur jaringan, yaitu menentukan jumlah lapisan, jumlah neuron dalam setiap lapisan, serta jenis fungsi aktivasi yang digunakan (Goodfellow, 2016). Ketiga, pelatihan jaringan, yaitu proses mengoptimalkan parameter jaringan melalui algoritma pembelajaran dengan menggunakan data latih (Haykin, 2009). Terakhir, evaluasi kinerja jaringan, yaitu menguji kinerja jaringan dengan menggunakan data uji, melakukan analisis hasil, dan melakukan tuning jika diperlukan (Zhang dkk, 2020)

Algoritma yang akan digunakan dalam penelitian ini yaitu *Artificial Neural Network (ANN) Backpropagation*. *Backpropagation* atau propagasi *error* adalah metode umum dari pembelajaran Jaringan Syaraf Tiruan bagaimana menyelesaikan suatu tugas pembelajaran termonitor dan merupakan implementasi dari *delta rule* (Vamsidhar & Veeravalli, 2010). Hal ini juga menyediakan metode komputasi yang efisien untuk perubahan bobot dalam jaringan umpan maju (*feed forward*) dengan unit-unit fungsi aktivasi terdiferensial untuk pembelajaran sebuah set pola *input output* (Abiodun, 2019). Dengan kata lain, *backpropagation* merupakan algoritma pembelajaran untuk meminimalisasikan nilai kesalahan pada hasil prediksi dengan cara membandingkan *output* dan target yang diharapkan (Bilgili, 2010).

Metode ANN memiliki potensi yang menarik dan sesuai untuk digunakan dalam bidang meteorologi, terutama dalam klasifikasi curah hujan. Dalam bidang

meteorologi, ANN dapat digunakan untuk memprediksi atau mengklasifikasikan curah hujan berdasarkan data historis cuaca, seperti suhu, kelembaban udara, tekanan atmosfer, dan penyinaran matahari (Paras dkk, 2009). Metode ANN dapat membantu dalam pengambilan keputusan dalam berbagai aplikasi meteorologi, seperti perencanaan sumber daya air, manajemen bencana, dan pengelolaan sumber daya alam (Logar dkk, 2017). Tujuan utama dari metode ANN adalah untuk mengenali pola yang kompleks atau non-linear dalam data dan melakukan prediksi atau klasifikasi berdasarkan pola-pola ini. Metode ANN memodelkan dan mensimulasikan cara kerja otak manusia dalam mengenali pola dan mempelajari hubungan antara data masukan dan keluaran (Erdil & Arcaklioglu, 2013).

Penelitian sebelumnya yaitu membandingkan antara dua metode yaitu ANN dengan SVM pada kasus klasifikasi biner pada curah hujan harian menggunakan variabel iklim sebagai variabel prediktor dimana pada penelitian tersebut menunjukkan bahwa akurasi klasifikasi menggunakan algoritma *Support Vector Machine* (SVM) dan *Artificial Neural Network* (ANN) masing-masing mencapai 82.1% dan 82.8%. Hal tersebut mengindikasikan bahwa model ANN memiliki hasil yang lebih baik dalam hal parameter kinerja seperti tingkat kesalahan klasifikasi dan skor F1 dibandingkan dengan model SVM yang memiliki parameter yang lebih sederhana (Hudnurkar & Rayavarapu, 2022). Penelitian lain juga dilakukan yaitu melakukan klasifikasi pada curah hujan menjadi dua kelas dengan memebandingkan jumlah variabel *input* yang akan digunakan. Hasil dari penelitian tersebut menunjukkan bahwa dengan menggunakan jumlah variabel *input* yang lebih banyak akan memiliki tingkat akurasi yang lebih besar (Shofura dkk., 2021). Pada penelitian lain yaitu memprediksi curah hujan dengan menggunakan metode

ANN *backpropagation* dengan membandingkan antara data harian dan data bulanan. Pada penelitian tersebut menunjukkan bahwa dengan menggunakan data bulanan akan memberikan hasil yang lebih optimal dibandingkan dengan menggunakan data harian (Nguyen dkk., 2021). Selain itu penelitian lain juga pernah dilakukan dengan kasus memprediksi curah hujan untuk satu tahun selanjutnya dimana penelitian tersebut menggunakan data historis yang diambil dari beberapa stasiun pada wilayah tersebut. Hasil penelitian tersebut menunjukkan nilai kesalahan prediksi yang sangat kecil yaitu kurang dari 5% (Nabila dkk., 2023).

Hujan memegang peranan penting dalam kehidupan manusia, terutama dalam bidang pertanian, penerbangan, kelistrikan, penampungan air, pariwisata, dan sektor lainnya. Adanya perubahan cuaca yang ekstrem belakangan ini menjadi pengingat bagi manusia akan firman Allah SWT dalam surah Al-Mu'minun ayat 18 yang artinya (Kemenag, 2023a):

“Dan Kami turunkan air dari langit menurut kadar yang ditentukan, kemudian Kami pancarkan air itu dalam bumi, dan sesungguhnya Kami benar-benar kuasa mengambilnya kembali (menjadikannya hilang).”

Q.S. Al-Mu'minun ayat 18 menjelaskan bahwa Allah SWT menurunkan air dari langit dalam jumlah dan waktu yang ditentukan, yang kemudian menyerap ke dalam tanah dan mengalir di dalamnya membentuk sungai-sungai dan mata air. Ayat ini juga menyatakan bahwa Allah SWT memiliki kuasa penuh untuk mengambil kembali air yang telah diturunkan tersebut, misalnya dengan mematikan hujan atau menimbulkan kekeringan. Ayat ini mengandung makna yang dalam, bahwa segala yang terjadi di alam semesta ini adalah hasil dari kehendak dan kuasa Allah SWT Yang Maha Sempurna. Air yang turun dari langit merupakan salah satu contoh nyata dari keajaiban ciptaan Allah SWT. Tanpa air, kehidupan di bumi tidak mungkin terjadi dan manusia tidak akan dapat bertahan hidup. Selain itu, ayat ini

juga mengajarkan manusia untuk bersyukur dan berterima kasih atas nikmat air yang diberikan oleh Allah SWT. Manusia harus menjaga air agar tetap terjaga dan terpelihara keberadaannya, tidak hanya untuk kebutuhan manusia saat ini, tetapi juga untuk generasi yang akan datang (Shihab, 2002).

Jawa Tengah memiliki variasi curah hujan yang tinggi, dengan wilayah pegunungan dan dataran rendah memiliki pola curah hujan yang berbeda. Prediksi curah hujan yang akurat sangat penting untuk perencanaan sumber daya air, pengelolaan banjir, serta keberlanjutan sektor pertanian di wilayah ini. Namun, prediksi curah hujan di Jawa Tengah kompleks karena dipengaruhi oleh banyak faktor seperti iklim regional, topografi, dan penggunaan lahan. Oleh karena itu, data curah hujan yang akurat dan dapat diandalkan menjadi hal yang krusial untuk mendukung pengelolaan sumber daya air dan mitigasi risiko bencana alam di wilayah ini.

Dalam penelitian ini, data curah hujan yang dikumpulkan dari Badan Pusat Statistik di Jawa Tengah akan digunakan sebagai data pelatihan dan pengujian untuk mengembangkan model ANN. Dengan menggunakan metode ANN, diharapkan dapat dikembangkan model klasifikasi curah hujan yang akurat dan dapat diandalkan untuk Jawa Tengah. Model ini dapat digunakan sebagai alat prediktif untuk membantu perencanaan dan pengelolaan sumber daya air, pengaturan irigasi, serta pengendalian banjir di wilayah tersebut. Selain itu, penelitian ini juga diharapkan dapat meningkatkan pemahaman tentang hubungan antara faktor iklim regional dengan curah hujan di Jawa Tengah, serta memberikan kontribusi dalam pengembangan metode prediksi curah hujan yang lebih baik dan lebih akurat.

Pentingnya penerapan metode ANN dalam klasifikasi curah hujan di Jawa Tengah dikarenakan beberapa alasan. Pertama, terjadinya hujan atau tidak dapat memiliki dampak signifikan pada sektor pertanian di wilayah tersebut. Prediksi yang akurat tentang curah hujan dapat membantu petani dalam pengambilan keputusan terkait pemilihan varietas tanaman, pengaturan pola tanam, serta pengelolaan irigasi yang efisien. Kedua, curah hujan yang ekstrem juga dapat menyebabkan bencana alam seperti banjir dan longsor. Oleh karena itu, kemampuan untuk mengklasifikasikan curah hujan secara akurat dapat membantu dalam upaya mitigasi dan pengelolaan risiko bencana di Jawa Tengah.

Berdasarkan permasalahan tersebut maka akan dilakukan penelitian mengenai klasifikasi curah hujan di Jawa Tengah menggunakan metode ANN. Pada penelitian ini akan dilakukan pengembangan model yaitu melakukan klasifikasi curah hujan di Jawa Tengah menjadi empat kelas yaitu: hujan ringan, sedang, lebat, dan sangat lebat. Metode *Artificial Neural Networks* (ANN) akan digunakan untuk mengembangkan model klasifikasi curah hujan yang akurat dan dapat diandalkan untuk Jawa Tengah (Bishop & Nasrabadi, 2006). Data yang digunakan dalam penelitian ini merupakan data-data yang berkaitan dengan kondisi cuaca pada kecamatan, seperti suhu, kelembapan, kecepatan angin, tekanan udara, penyinaran matahari, tinggi rata-rata ibukota, luas wilayah, jumlah hari hujan, dan jumlah curah hujan selama periode 2019-2022. Untuk itu, akan dibentuk arsitektur jaringan yang terdiri dari *input layer*, *hidden layer*, dan *output layer*. Fungsi aktivasi ReLU (*Rectified Linear Unit*) akan digunakan pada *input* dan *hidden layer*, karena fungsi aktivasi tersebut dapat bekerja dengan baik pada data *non-linear*. Sedangkan pada *output layer*, akan digunakan fungsi aktivasi *softmax* untuk melakukan klasifikasi

(Haykin, 2010). Pendekatan pengolahan data yang cermat, serta pemilihan dan optimasi parameter dalam model ANN, termasuk metode pelatihan, akan digunakan dalam proses penelitian ini.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang, rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimana model ANN dalam mengklasifikasi curah hujan di Jawa Tengah?
2. Bagaimana tingkat akurasi yang dihasilkan ANN pada hasil klasifikasi curah hujan di Jawa Tengah?
3. Bagaimana hasil klasifikasi curah hujan di Jawa Tengah pada tahun 2023?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah, tujuan dari penelitian ini adalah sebagai berikut:

1. Mendapatkan model klasifikasi yang optimal untuk mengklasifikasikan curah hujan di Jawa Tengah dengan menggunakan metode ANN.
2. Mendapatkan tingkat akurasi yang dihasilkan ANN pada hasil klasifikasi curah hujan di Jawa Tengah.
3. Mendapatkan hasil klasifikasi curah hujan di Jawa Tengah pada tahun 2023.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Secara Teoritis

Penelitian ini dapat membawa kontribusi dalam pengembangan ilmu pengetahuan terkait dengan klasifikasi curah hujan menggunakan metode *Artificial Neural Network* (ANN). Teori di balik penerapan ANN dalam penelitian ini mendasarkan pada kemampuan jaringan saraf tiruan untuk mempelajari pola-pola kompleks dalam data curah hujan. Dengan mengoptimalkan model ANN, penelitian ini menawarkan kontribusi penting dalam meningkatkan akurasi prediksi cuaca, khususnya terkait dengan curah hujan di wilayah Jawa Tengah.

2. Secara Praktis

Manfaat praktis dari penelitian ini sangat relevan untuk program studi kampus dan instansi pemerintah di Jawa Tengah. Program Studi dapat menggunakan penelitian ini sebagai referensi tambahan tentang penerapan metode *Artificial Neural Network* (ANN) dalam proses klasifikasi curah hujan. Selain itu, instansi pemerintah di Jawa Tengah dapat memanfaatkan penelitian ini untuk meningkatkan perencanaan dan penanganan keadaan darurat terkait curah hujan, yang pada akhirnya akan mendukung upaya-upaya untuk mengurangi dampak negatif dari perubahan cuaca ekstrem di wilayah tersebut.

1.5 Batasan Masalah

Batasan masalah pada penelitian yang dilakukan yaitu sebagai berikut:

1. Variabel indikator atau variabel *input* yang digunakan ada 8, yaitu: suhu, kelembapan, kecepatan angin, tekanan udara, hari hujan, penyinaran

matahari, tinggi wilayah, dan luas wilayah. Sedangkan variabel *output* yang akan digunakan yaitu curah hujan.

2. Data *input* yang digunakan pada penelitian ini hanya 11 dari 35 kabupaten/kota di Jawa Tengah pada tahun 2019 hingga tahun 2022 yang diambil pada *website* resmi Badan Pusat Statistik (BPS) (<https://jateng.bps.go.id/> yang diakses pada 12 April 2023), dimana data yang digunakan yaitu data bulanan.
3. Skenario model pada penelitian ini akan dilakukan penambahan jumlah *layer* pada *hidden layer* yaitu 3, dimana setiap *layer* pada *hidden layer* nya akan dilakukan kombinasi jumlah neuron yaitu 4, 8, 12, 16, 20, 24, 28, 32.

1.6 Definisi Istilah

Klasifikasi	: Pengelompokan hal-hal tertentu berdasarkan ciri-ciri atau karakteristiknya
Prediksi	: Ramalan dengan perhitungan yang objektif dengan menggunakan data-data masa lalu untuk mendapatkan penentuan di masa depan
Kompleks	: Himpunan kesatuan atau mengandung beberapa unsur yang rumit atau sulit
Linier	: Saling berkaitan atau masih dalam satu garis
Nonlinier	: Sistem yang sifatnya tidak tetap, mudah berubah, sulit dikontrol, dan sulit diprediksi
<i>Input layer</i>	: Lapisan yang membawa masuk data ke dalam sistem untuk kemudian dilakukan pemrosesan pada <i>layer</i> selanjutnya

- Output layer* : Lapisan terakhir pada *neuron* yang menghasilkan *output* sistem
- Hidden layer* : Dalam jaringan saraf tiruan, *hidden layer* merupakan lapisan di antara *input layer* dan *output layer*, di mana terdapat *artificial neuron* yang memiliki kumpulan bobot *input* dan prosedur untuk menghasilkan *output neuron* melalui fungsi aktivasi
- ReLU : Fungsi aktivasi yang sering digunakan dalam jaringan saraf tiruan, mengaktifkan *neuron* jika *inputnya* lebih besar dari nol dan mengabaikan nilai negatif.
- Softmax* : Fungsi yang umum digunakan dalam jaringan saraf untuk masalah klasifikasi multikelas.

BAB II KAJIAN TEORI

2.1 Teori Pendukung

2.1.1 *Time Series*

Time series atau deret waktu adalah kumpulan data yang diperoleh dari pengamatan pada suatu peristiwa, gejala, atau perubahan yang terjadi dalam suatu periode waktu tertentu (Hanke & Wichern, 2005). Data yang dikumpulkan bersifat kronologis dan teratur, misalnya dalam periode waktu harian, mingguan, bulanan, tahunan, atau semesteran. Dengan melihat pola pergerakan data *time series*, maka dapat dilakukan analisis terhadap data pada periode waktu sebelumnya sebagai acuan untuk melakukan peramalan terhadap nilai atau kejadian pada periode waktu selanjutnya atau di masa depan. Hal ini penting dalam peramalan bisnis dan dalam berbagai bidang lainnya.

Data *time series* digunakan untuk menganalisis, memodelkan, dan meramalkan pergerakan variabel-variabel tertentu dalam periode waktu tertentu, seperti suhu, harga saham, volume penjualan, dan sebagainya (Enders, 2004). Menurut teori, analisis *time series* melibatkan tiga komponen utama, yaitu (Montgomery dkk., 2012):

1. Tren (*trend*) merupakan perubahan jangka panjang dari suatu variabel atau data *time series*.
2. Siklus (*cycle*) merupakan fluktuasi jangka menengah yang terjadi dalam data *time series*.
3. Variasi musiman (*seasonal variation*) merupakan fluktuasi jangka pendek yang terjadi dalam data *time series* dan biasanya berkaitan dengan musim atau bulan dalam tahun.

Dalam analisis *time series*, terdapat beberapa metode dan teknik, seperti *moving average*, *exponential smoothing*, ARIMA (*Autoregressive Integrated Moving Average*), dan sebagainya.

2.1.2 Peramalan (*Forecasting*)

Peramalan (*forecasting*) secara umum menjelaskan sebagai salah satu cara memprediksi suatu kejadian atau nilai yang kemungkinan akan terjadi pada masa yang akan datang dengan berdasarkan pada data yang sudah pernah ada sebelumnya. Teknik peramalan dibagi menjadi dua golongan utama yaitu golongan peramalan berdasarkan metode kualitatif dan kuantitatif (Hanke & Wichern, 2005). Peramalan yang berdasarkan metode kualitatif merupakan metode peramalan yang berdasarkan pada naluri, pengetahuan, *experience* dan pertimbangan dari orang yang melakukan peramalan. Metode kuantitatif merupakan metode dengan sifat yang objektif dikarenakan berdasar pada keadaan data asli yang diolah dengan menggunakan metode tertentu. Peramalan deret waktu (*time series prediction*) dan peramalan kausan merupakan salah satu definisi dari metode peramalan kuantitatif. Metode peramalan yang digunakan dan periode waktu peramalan sangat berpengaruh dalam keakurasian pada peramalan (Hanke & Wichern, 2005).

Permalan yang didasarkan pada periode waktu, digolongkan menjadi tiga jenis sebagai berikut (Montgomery dkk, 2015):

1. Jangka Pendek (*Short Term*)

Jangka pendek (*short term*) merupakan peramalan yang didasarkan pada jarak waktu yang singkat seperti harian, mingguan, atau bulanan. Peramalan pada jenis ini adalah peramalan yang mempunyai tingkat keakurasian yang masih tinggi.

2. Jangka Menengah (*Medium Term*)

Jangka menengah (*short term*) merupakan peramalan yang didasarkan pada waktu yang musiman seperti triwulan, kuartal, semester, dan sebagainya. Peramalan pada jenis ini adalah peramalan yang mempunyai tingkat keakurasian yang menengah.

3. Jangka Panjang (*Long Term*)

Jangka panjang (*long term*) merupakan peramalan yang didasarkan pada jarak waktu yang lebih dari dua tahun. Peramalan pada jenis ini memiliki tingkat keakurasian yang rendah. Sehingga dibutuhkan data pengamatan dengan periode yang cukup lama dan banyak sehingga didapatkan keakurasian yang optimal.

2.1.3 Artificial Neural Network

Artificial Neural Network (ANN) merupakan model komputasi yang dapat digunakan untuk berbagai masalah. Jaringan Saraf Tiruan dapat digunakan untuk klasifikasi pola, sintesis dan pengenalan ucapan antarmuka adaptif antara manusia dan sistem fisik yang kompleks, perkiraan fungsi, kompresi gambar, memori asosiatif, pengelompokan, peramalan dan prediksi, optimasi kombinatorial, pemodelan sistem nonlinear, dan kontrol. Meskipun dinamakan “saraf”, jaringan ini tidak selalu mereplikasi fenomena saraf atau kognitif biologis secara akurat dan lebih terkait dengan model matematika dan/atau statistik tradisional seperti pengklasifikasi pola nonparametrik, algoritma pengelompokan, filter nonlinier, dan model regresi statistik (Hassoun, 1995). *Artificial Neural Network* (ANN) atau juga disebut sebagai Jaringan Saraf Tiruan (JST) ini merupakan metode klasifikasi dan prediksi yang mempunyai gambaran replikasi proses seperti otak manusia dimana

metode ini memiliki kemampuan memprediksi untuk model yang kompleks pada model yang linier maupun model yang nonlinier (Hadihardaja & Sutikno, 2010).

Neuron dan sinaps merupakan komponen utama dari ANN. Neuron mengandung informasi yang menentukan apakah informasi akan diteruskan ke neuron lain atau tidak, sementara sinaps merepresentasikan hubungan antara satu neuron dengan neuron lainnya. Apakah informasi diteruskan ke suatu neuron atau tidak, bergantung pada pengaturan bobot (pengali) dan bias (penambah). Pada awalnya, bias dan bobot pada ANN diatur secara manual, namun dengan ditemukannya prinsip pembelajaran seperti propagasi balik (*backpropagation*), bobot dan bias diatur berdasarkan data pelatihan yang tersedia. Data pelatihan tersebut memiliki label atau target yang harus dicapai oleh model ANN. Biasanya, terdapat kesalahan antara hasil prediksi model ANN dengan label atau target yang diinginkan. Namun, jika kesalahan tersebut cukup kecil, model yang dihasilkan dari proses pelatihan dapat digunakan untuk memprediksi data lain di luar data pelatihan. Karena prinsip pembelajarannya, ANN dianggap sebagai salah satu metode dalam mesin pembelajaran (*machine learning*). Selain itu, karena ANN berusaha untuk menggali prinsip-prinsip dalam data, maka ANN juga dianggap sebagai salah satu metode dalam penggalian data (*Data Mining*) (Handayanto & Herlawati, 2020).

2.1.3.1 Struktur ANN

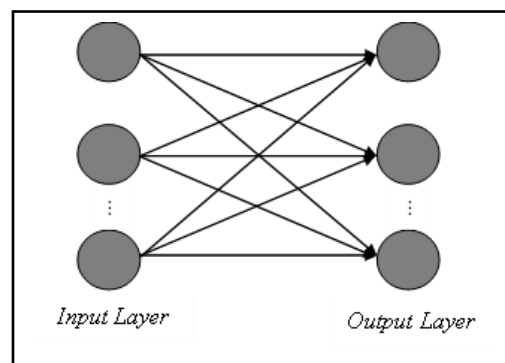
Artificial Neural Network (ANN) atau Jaringan saraf tiruan adalah sebuah teknologi komputasi yang tujuan dasarnya adalah untuk mengenali pola dalam data. Berdasarkan sebuah model komputasi yang mirip dengan dasar struktur otak, ANN berbagi tentang kemampuan otak untuk belajar atau beradaptasi dalam

menanggapi input dari luar. Saat diekspos ke aliran data pelatihan, ANN dapat mengetahui hubungan yang sebelumnya tidak diketahui dan belajar pemetaan nonlinier kompleks dalam data (Back dkk, 1998).

Menurut Haykin (2009), *Artificial Neural Networks* (ANN) memiliki berbagai macam arsitektur yang sering digunakan, sebagai berikut:

1. *Single Layer Network* atau Jaringan dengan Lapisan Tunggal.

Dalam arsitektur ini, terdapat satu lapisan yang memiliki bobot koneksi terbaik. Ketika menerima sinyal dari luar, unit *input* akan menjadi aktif, sedangkan untuk membaca respon jaringan, unit *output* yang berperan aktif. Selain itu, unit *input* terhubung ke seluruh unit *output*, tetapi tidak terhubung ke unit *input* yang lain pada lapisan yang sama. Struktur dari *Single Layer Network* ditunjukkan dalam Gambar 2.1, di mana X merupakan *input layer* dan Y merupakan *output layer*.

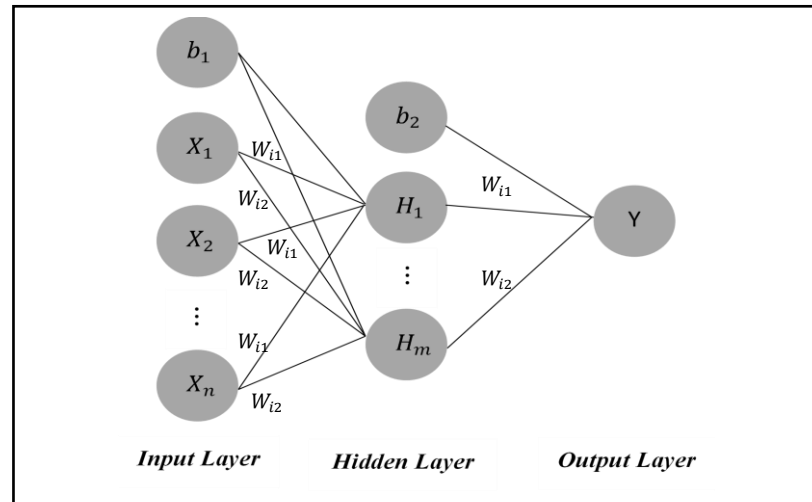


Gambar 2.1 *Single Layer Perceptron*

2. *Multilayer Network* atau Jaringan dengan Lapisan Banyak.

Sebuah jaringan yang terdiri dari beberapa lapisan node yang mana disebut perangkat tersembunyi, di antara unit *input* dan *output* disebut sebagai jaringan *multilayer*. Jaringan ini sering terdiri dari beberapa lapisan bertingkat dan biasanya menggunakan bobot untuk

menghubungkan unit pada setiap lapisannya (termasuk unit *input*, *hidden*, dan *output*) seperti yang terlihat pada Gambar 2.2.



Gambar 2.2 Multilayer Perceptron

Multilayer Perceptron dengan satu lapisan tersembunyi dan satu neuron tersembunyi belajar untuk memodelkan fungsi $f(x)$ yang ditentukan oleh persamaan (Goodfellow dkk, 2016):

$$f(x) = g\left(\sum(x_i W_{ij})\right) + b \quad (2.1)$$

Dimana x_i adalah *input* neuron. W_{ij} adalah bobot dan b adalah bias. g adalah fungsi aktivasi yang diterapkan pada hasil penjumlahan $(x_i W_{ij})$. Fungsi aktivasi ini memberikan non-linearitas pada model dan memungkinkan model untuk mempelajari hubungan yang lebih kompleks antara fitur *input*. $f(x)$ adalah *output* yang dihasilkan untuk satu *layer*.

2.1.3.2 Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi matematis yang diterapkan pada setiap *neuron* yang mana berperan sebagai sinyal untuk menentukan *output* yang diproses ke *neuron* lainnya dalam ANN (Puspitasari, 2006). Fungsi tersebut

sangat berperan penting dalam suatu jaringan dimana penggunaannya sesuai kebutuhan dan target yang diinginkan. Fungsi aktivasi tersebut yang akan menentukan besarnya bobot yang harus digunakan (Sagiroglu dkk, 2003). Berikut adalah beberapa fungsi aktivasi yang sering digunakan pada jaringan saraf tiruan (*Artificial Neural Network (ANN)*):

1. Fungsi ReLU (Rectified Linear Unit)

Fungsi ReLU merupakan fungsi aktivasi yang sering digunakan pada ANN dengan banyak *layers* atau "*deep neural network*". Fungsi ini menghasilkan *output* yang sama dengan *input*-nya jika *input*-nya positif, dan menghasilkan *output* nol jika *input*-nya negatif. Persamaan dari fungsi ReLU adalah sebagai berikut (X. Zhang & LeCun, 2015):

$$f(x) = \max(0, x) \quad (2.2)$$

2. Fungsi *Softmax*

Fungsi *softmax* merupakan fungsi yang umum digunakan dimana fungsi ini mengubah nilai *input* menjadi probabilitas distribusi yang dapat digunakan untuk mengklasifikasikan data kedalam kelas-kelas yang berbeda (Goodfellow dkk, 2016):

$$\text{softmax}(x) = \frac{e^x}{\sum e^x} \quad (2.3)$$

2.1.3.3 *Loss Function*

Loss Function merupakan alat matematis yang mengukur perbedaan antara hasil prediksi model dengan nilai sebenarnya dari data pelatihan. Fungsi ini berperan penting dalam proses pembelajaran jaringan karena memberikan ukuran sejauh mana model berhasil dalam memodelkan data yang ada. Tujuan

utama dari pembelajaran mesin adalah mengoptimalkan parameter-model agar fungsi kerugian mencapai nilai serendah mungkin. Salah satu jenis fungsi kerugian yang digunakan adalah *Categorical Cross-Entropy*. Jenis fungsi kerugian ini digunakan untuk masalah klasifikasi multikelas dimana fungsi ini menilai perbedaan antara distribusi probabilitas hasil prediksi model dengan distribusi data aktual. Berikut rumus dari fungsi kerugian *Categorical Cross-Entropy* (Goodfellow dkk, 2016):

$$\text{Categorical Cross - Entropy} = - \sum_i y_{target_i} (\log(y_i)) \quad (2.4)$$

Pada setiap iterasi pelatihan, ANN membandingkan hasil prediksi dengan data aktual menggunakan fungsi kerugian. Nilai kerugian yang dihasilkan kemudian digunakan oleh algoritma optimasi untuk menghitung gradien dan menyesuaikan parameter model. Tujuan utama adalah mengurangi nilai fungsi kerugian seiring berjalannya pelatihan, sehingga model dapat membuat prediksi yang semakin akurat.

2.1.3.4 Algoritma *Backpropagation Neural Network* (BPNN)

Algoritma *Backpropagation* sangat optimal sebagai algoritma *training* dan dapat digunakan sebagai dasar untuk mencari solusi dari kompleksitas (Mustafidah & Suwarsito, 2016). Algoritma ini menggunakan parameter tingkat pembelajaran dan nilai minimum kesalahan pada proses *training* dan *testing* dengan membangun jaringan menggunakan fungsi aktivasi (Wanto dkk, 2017). Dengan menggunakan bentuk pola, algoritma *backpropagation* dapat menghasilkan nilai kesalahan minimum (Roza dkk, 2018). Algoritma ini membentuk model neuron untuk fungsi kerugian (*loss function*) dan dapat

diimplementasikan dengan melakukan perhitungan, pengujian, dan pelatihan jaringan menggunakan data historis (Hadisuwito & Hassan, 2019). Dalam hal ini, algoritma backpropagation sangat sukses dalam memecahkan masalah pada *Artificial Neural Network* (ANN) dengan menggunakan data historis (Lillicrap dkk., 2020).

Pelatihan *backpropagation* mencakup tiga tahap utama, yaitu (Jumarwanto dkk, 2009):

1. Propagasi Maju

Dalam langkah propagasi maju, sinyal *input* (X_i) dipropagasikan ke *hidden layer* melalui fungsi aktivasi. *Output* dari setiap *node* pada *hidden layer* (Z_{ji}) yang telah diubah melalui fungsi aktivasi, diteruskan ke *node-node* lain dalam *layer* tersebut, akhirnya menghasilkan *output* pada *output layer*. Setelahnya, *output* yang dihasilkan (y_{jk}) dibandingkan dengan nilai target yang diinginkan (y). Perbandingan ini dilakukan dengan menghitung nilai kesalahan ($y - y_{jk}$). Jika kesalahan yang dihitung lebih kecil dari ambang kesalahan yang ditetapkan, pelatihan dihentikan. Namun, jika kesalahan melebihi ambang toleransi, maka bobot pada setiap *node* disesuaikan untuk mengurangi kesalahan tersebut. Proses penyesuaian bobot ini dikenal sebagai propagasi mundur.

2. Propagasi Mundur

Langkah propagasi mundur (*backpropagation*) merupakan prosedur sentral dalam pelatihan jaringan saraf tiruan yang melibatkan perhitungan gradien dari fungsi kesalahan terhadap bobot jaringan $dW_{ji} (i = 1, 2, \dots, n)$, yang memungkinkan penyesuaian berdasarkan arah penurunan kesalahan.

Proses ini melibatkan penghitungan selisih antara jaringan dan target yang diinginkan oleh output yang dihasilkan $(y - y_{jk})$, dilanjutkan dengan perhitungan gradien yang mengukur sensitivitas kesalahan terhadap perubahan kecil pada bobot jaringan. Dengan memanfaatkan gradien ini, bobot pada lapisan-lapisan jaringan diperbarui dengan menggeser mereka dalam arah yang berlawanan dengan gradien, memungkinkan pengurangan kesalahan secara berangsur-angsur. Langkah propagasi mundur ini dilakukan secara iteratif melalui berbagai data *training*, dengan tujuan mengoptimalkan jaringan untuk menghasilkan keluaran yang semakin mendekati hasil yang diinginkan.

3. Perubahan Bobot

Setelah menghitung gradien error maka akan dilakukan proses penyesuaian bobot pada setiap koneksi antar neuron secara individual, dengan perubahan bobot yang ditentukan oleh gradien error terhadap bobot tersebut. Penyesuaian bobot pada *layer* berikutnya bergantung pada gradien error yang dihitung pada *layer* sebelumnya. Langkah-langkah ini berulang dalam iterasi berkelanjutan hingga *error* secara perlahan berkurang dan mencapai ambang batas yang telah ditetapkan untuk konvergensi yang lebih baik dalam pelatihan jaringan saraf tiruan.

Berikut ini adalah algoritma backpropagation untuk satu hidden layer dengan menggunakan fungsi aktivasi ReLu dan softmax (Syukri & Samsuddin, 2019):

Tahap Inisialisasi

Langkah 0 : Menginisialisasi bobot dan bias untuk setiap simpul dengan menggunakan bilangan acak (*random*).

Langkah 1 : Menetapkan nilai maksimum *Epoch*. ($Epoch < Maks\ Epoch$) dan $Error < Target\ Error$.

Langkah 2 : Setiap pasangan elemen akan menjalani proses pembelajaran, dan kemudian dilanjutkan dengan langkah-langkah berikutnya.

Fase I Feedforward

Langkah 3 : Setiap unit *input* ($x_i, i = 1, 2, 3, \dots, n$) menerima sinyal masukan x_i dan mengirimkannya ke semua unit di atasnya dalam susunan yang tersembunyi.

Langkah 4 : Setiap unit *hidden layer* ($Z_j, j = 1, 2, 3, \dots, p$) memproses sinyal-sinyal *input* yang berbobot,

$$Z_{net_j} = \left(\sum_{i=1}^n X_i V_{ji} \right) + V_{j0} \quad (2.5)$$

Fungsi aktivasi yang digunakan untuk memproses sinyal *output*nya,

$$Z_j = f(Z_{net_j}) = \max(0, Z_{net_j}) \quad (2.6)$$

Langkah 5 : Setiap unit *output layer* ($y_k, k = 1, 2, 3, \dots, n$) memproses sinyal-sinyal *output* yang berbobot,

$$y_{net_k} = \left(\sum_{i=1}^p Z_i W_{kj} \right) + W_{k0} \quad (2.7)$$

W_{k0} merupakan bobot bias ke 0 pada *hidden layer* dan *output layer* ke k , Z_j *output hidden layer* ke j dan W_{kj} merupakan bobot ke j pada *hidden layer* dan *output layer* ke k . Kemudian nilai *output* sebelum diaktivasi (y_{net_k}) akan diaktivasi dengan fungsi aktivasi *softmax* untuk memproses sinyal *output*,

$$y_k = \text{softmax}(y_{net_k}) = \frac{e^{y_{net_k}}}{\sum e^{y_{net_j}}} \quad (2.8)$$

Fase II Backward

Langkah 6 : Menghitung fungsi kesalahan pada proses klasifikasi dengan metode

Categorical Cross-Entropy. Berikut adalah rumusnya:

$$Loss_t = - \sum_t (y(\log(y_{jk}))) \quad (2.9)$$

dimana setiap unit y_k menerima pola y_{target_k} untuk menghitung eror gradien pada *output* dengan $l = 1, 2, 3, 4$.

Langkah 7 : Menghitung *gradient loss* terhadap bobot pada *output layer* dan memperbarui bobot untuk mengurangi *loss*.

Rumus untuk *gradient loss* terhadap *output*:

$$dy_{jk} = - \left(\frac{y}{y_{jk}} \right) \quad (2.10)$$

Rumus untuk *gradient output* terhadap hasil penjumlahan sebelum fungsi aktivasi:

$$dy_{net_{jk}} = y_{jk} - y \quad (2.11)$$

Rumus untuk *gradient* terhadap bobot W_{jk} :

$$dW_{jk} = Z_{net_{ji}}(dy_{net_{jk}}) \quad (2.12)$$

Rumus untuk *gradient* terhadap bias b_{jk} :

$$db_{jk} = \frac{1}{m} \sum_{i=1}^m dy_{net_{jk}} \quad (2.13)$$

Update Bobot dan Bias

Langkah 8 : *Update* bobot dan bias dengan menggunakan algoritma

stochastic optimizer Adam dengan mengupdate momen pertama dan kedua:

$$m_{t+1} = \beta_1(m_t) + (1 - \beta_1)(dW_{ji}) \quad (2.14)$$

$$v_{t+1} = \beta_2(v_t) + (1 - \beta_2)(dW_{ji})^2 \quad (2.15)$$

dengan menginisialisasi momen pertama ($m = 0$), momen kedua ($v = 0$), dan iterasi ($t = 1$) dimana β_1 dan β_2 adalah *hyperparameter* yang mengontrol laju pembaruan momen pertama dan kedua.

Selanjutnya adalah mengkoreksi momen pertama dan kedua untuk mengurangi bias pada awal iterasi:

$$\hat{m}_{t+1} = \frac{m_{t+1}}{1 - \beta_1^{t+1}} \quad (2.16)$$

$$\hat{v}_{t+1} = \frac{v_{t+1}}{1 - \beta_2^{t+1}} \quad (2.17)$$

Selanjutnya yaitu menghitung perubahan bobot ($\Delta W_{out,1}$) dan bias (b_i):

$$\Delta W_{out,1} = -\frac{\alpha}{\sqrt{\hat{v}_{t+1}} + \epsilon} (\hat{m}_{t+1}) \quad (2.18)$$

$$\Delta b_{out,1} = -\frac{\alpha}{\sqrt{\hat{v}_{t+1}} + \epsilon} (\hat{m}_{t+1}) \quad (2.19)$$

Kemudian melakukan *update* bobot *output layer* dan *hidden layer*

(W_{ij}) bias (b_i):

$$W_{ij}(\text{baru}) = W_{ij}(\text{lama}) + \Delta W_{out,1} \quad (2.20)$$

$$b_{i(\text{baru})} = b_{i(\text{lama})} + \Delta b_{out,1} \quad (2.21)$$

dimana η merupakan laju pembelajaran (*learning rate*) yang mengontrol seberapa besar perubahan dan ϵ adalah bilangan kecil untuk mencegah pembagian dengan nol.

Jika mengadopsi lebih dari satu *hidden layer*, langkah-langkah 4 hingga 8 dilakukan penambahan perhitungan dan penyesuaian yang sesuai untuk setiap *hidden layer* tambahan. Setiap *hidden layer* perlu dilakukan perhitungan gradien dan penyesuaian bobot secara berurutan, dimulai dari *output layer* dan bergerak mundur melalui setiap *hidden layer* menuju *input layer*.

2.1.3.5 Encoding Data

Encoding Data adalah proses penting dalam pemrosesan data di mana data diubah dari representasi aslinya ke bentuk yang dapat dimengerti dan diolah oleh komputer atau algoritma pemrosesan data. Representasi asli data mungkin datang dalam berbagai format, seperti teks, kategori, tanggal, atau numerik. Encoding bertujuan untuk mengubah data ke dalam representasi numerik agar dapat digunakan oleh algoritma pemrosesan data dan pembelajaran mesin (Raschka & Mirjalili, 2019).

One-Hot Encoding adalah teknik khusus dalam *encoding* data kategorikal menjadi bentuk numerik yang lebih cocok untuk analisis. Teknik ini berguna ketika kita memiliki data kategori yang tidak memiliki urutan atau hubungan ordinal antara kategori tersebut. Dalam *One-Hot Encoding*, setiap kategori diubah menjadi vektor biner di mana hanya satu elemen bernilai 1 yang mana mewakili kategori yang sesuai, dan sisanya bernilai 0. Ini menciptakan kolom-kolom baru yang merepresentasikan setiap kategori (Brownlee, 2017).

2.1.3.6 Normalisasi Data

Normalisasi data adalah suatu proses untuk mengubah skala data *input* yang memiliki rentang nilai yang berbeda-beda menjadi skala yang seragam. Hal ini diperlukan untuk mempercepat konvergensi dan meningkatkan akurasi dari model ANN. Salah satu fungsi aktivasi yang umum digunakan pada ANN adalah sigmoid, yang menghasilkan *output* antara 0 dan 1. Normalisasi data dilakukan agar nilai *input* bagi setiap *neuron* pada *input layer* memiliki rentang antara 0 hingga 1. Salah satu metode normalisasi yang umum digunakan pada ANN adalah *min-max normalization*. Pada metode ini, data *input* diubah ke dalam rentang nilai antara 0 hingga 1 dengan menggunakan rumus (Haykin, 2009):

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2.22)$$

dengan:

- x_{norm} : Data hasil normalisasi
- x : Data *input*
- x_{min} : Nilai minimum dari data *input*
- x_{max} : Nilai maksimum dari data *input*

Dengan normalisasi data, setiap variabel *input* akan memiliki pengaruh yang sama terhadap keluaran (*output*) yang dihasilkan oleh ANN.

2.1.3.7 Decoding Data

Decoding merupakan langkah mengembalikan data *output* yang telah mengalami *encoding* ke bentuk data yang sesuai dengan data asli atau *input* awal. Dalam tugas klasifikasi, *output* dari *layer* terakhir jaringan saraf berupa distribusi probabilitas untuk setiap kelas. Sehingga, *decoding* dilakukan dengan tujuan

mengonversi distribusi probabilitas menjadi label kelas tunggal atau membuat keputusan klasifikasi berdasarkan nilai output tertinggi (Goodfellow dkk, 2016).

2.1.3.8 Evaluasi Model

Dalam upaya untuk memperoleh prediksi yang lebih akurat, dapat menggunakan matriks evaluasi untuk mengukur kinerja dan performa dari model atau sistem yang digunakan. Matriks evaluasi memberikan gambaran komprehensif tentang hasil prediksi dengan membandingkan nilai prediksi dengan nilai sebenarnya dalam bentuk *confusion matrix*. Matriks evaluasi tersebut umumnya digunakan dalam masalah klasifikasi, di mana hasil prediksi dibandingkan dengan label kelas sebenarnya. Matriks ini terdiri dari empat istilah, yaitu: *true positive* (TP) yaitu hasil prediksi kelas positif sesuai dengan data aktual yang ada, *true negative* (TN) yang mana hasil prediksi kelas negatif sesuai dengan data aktual yang ada, *false positive* (FP) menunjukkan hasil prediksi kelas negatif tidak sesuai dengan data aktual yang sebenarnya, dan berada dalam kelas positif, dan *false negative* (FN) yaitu hasil prediksi kelas positif tidak sesuai dengan data aktual yang sebenarnya dan berada dalam kelas negatif (Haykin, 2009).

Tabel 2.1 *Confusion Matrix*

<i>Confusion Matrix</i>		Kelas Prediksi			
		<i>kelas₁</i>	<i>kelas₂</i>	...	<i>kelas_N</i>
Kelas Aktual	<i>kelas₁</i>	TN	FP	...	TN
	<i>kelas₂</i>	FN	TP	...	FN

	<i>kelas_N</i>	TN	FP	...	TN

Dengan menggunakan matriks evaluasi, maka dapat menghitung dan mendapatkan berbagai metrik evaluasi penting seperti akurasi, presisi, *recall*, *F1-score*, dan lainnya. Berikut beberapa matrik evaluasi yang dapat digunakan untuk evaluasi model (Haykin, 2009):

1. Akurasi (*Accuracy*)

Akurasi merupakan rasio jumlah prediksi yang benar terhadap jumlah total data yang dievaluasi. Metrik ini memberikan gambaran tentang seberapa baik model dapat melakukan prediksi secara keseluruhan.

Rumusnya adalah:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.23)$$

2. Presisi (*Precision*)

Presisi Merupakan rasio *True Positive* (TP) terhadap total prediksi positif yang dilakukan oleh model. Metrik ini memberikan gambaran tentang seberapa banyak prediksi positif yang benar dari semua prediksi positif yang dilakukan. Rumusnya adalah:

$$Precision = \frac{TP}{TP + FP} \quad (2.24)$$

$$Precision_{weighted} = \frac{1}{N} \sum_{i=1}^N Precision(support) \quad (2.25)$$

dengan *support* merupakan nilai sampel dalam suatu kelas dan *N* merupakan jumlah nilai sampel dari seluruh kelas.

3. *Recall*

Recall juga dikenal sebagai sensitivitas atau *True Positive Rate* (TPR), adalah rasio *True Positive* (TP) terhadap total jumlah *instance* yang sebenarnya positif. Metrik ini memberikan gambaran tentang seberapa

banyak *instance* positif yang berhasil diidentifikasi oleh model.

Rumusnya adalah:

$$Recall = \frac{TP}{TP + FN} \quad (2.26)$$

$$Recall_{weighted} = \frac{1}{N} \sum_{i=1}^N Recall(support) \quad (2.27)$$

dengan *support* merupakan nilai sampel dalam suatu kelas dan N merupakan jumlah nilai sampel dari seluruh kelas.

4. F1-Score

F1-Score adalah ukuran kinerja yang umum digunakan dalam evaluasi klasifikasi model, terutama ketika terdapat ketidakseimbangan antara kelas-kelas yang diamati. Metrik ini memberikan gambaran tentang keselarasan antara presisi dan *recall*. Rumusnya adalah:

$$F1\ score = 2 \frac{Precision \times Recall}{Precision + Recall} \quad (2.28)$$

$$F1\ score_{weighted} = \frac{1}{N} \sum_{i=1}^N F1\ score(support) \quad (2.29)$$

dengan *support* merupakan nilai sampel dalam suatu kelas dan N merupakan jumlah nilai sampel dari seluruh kelas.

Hasil dari matriks evaluasi memberikan wawasan tentang berbagai aspek performa model, seperti kemampuan untuk mengenali kelas yang benar (presisi), kemampuan untuk mengidentifikasi semua instans kelas tertentu (*recall*), dan keseimbangan antara presisi dan recall (F1-score). Dengan demikian, matriks evaluasi sangat penting untuk digunakan sebagai penilaian kualitas prediksi dan melakukan analisis yang lebih mendalam tentang performa model atau sistem yang digunakan.

Selain itu, untuk melihat keakuratan model juga dapat dilihat dari seberapa besar model melakukan kesalahan (*error*). *Mean Absolute Error* (MAE) merupakan salah satu metode perhitungan untuk melihat berapa nilai kesalahan yang dilakukan oleh suatu model. MAE mengukur rata-rata absolut selisih antara prediksi model dan nilai sebenarnya.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_{target_k} - y_k| \quad (2.30)$$

Dimana N merupakan jumlah observasi, y_{target_k} merupakan data sebenarnya, dan y_k merupakan data prediksi. MAE menghitung selisih absolut antara setiap pasangan nilai sebenarnya dan prediksi, kemudian mengambil rata-rata dari seluruh selisih tersebut. Dengan demikian, nilai MAE akan memberikan gambaran tentang seberapa besar kesalahan prediksi secara keseluruhan, tanpa memperhatikan arah kesalahan.

2.1.4 Curah Hujan

Curah hujan adalah jumlah volume air yang jatuh ke permukaan bumi dalam satu waktu tertentu, biasanya dalam satuan milimeter (mm) atau liter per meter persegi (L/m²). Teori tentang curah hujan berhubungan dengan siklus hidrologi, di mana uap air dari permukaan bumi menguap dan membentuk awan, kemudian awan itu berkumpul dan memicu hujan. Beberapa faktor yang mempengaruhi curah hujan meliputi topografi, kelembaban udara, dan arus angin. Selain beberapa faktor tersebut, curah hujan juga dipengaruhi oleh faktor geografis, seperti ketinggian tempat, kontur lahan, dan arah angin. Faktor iklim seperti kelembapan udara dan suhu juga berperan dalam mempengaruhi curah hujan (Pranowo, 2019).

Salah satu teori yang menjelaskan pola curah hujan adalah teori konvektif, yang menjelaskan bahwa hujan terjadi karena udara hangat yang naik ke atmosfer dan kemudian mendingin, membentuk awan dan akhirnya hujan. Teori ini dikembangkan oleh Sir George Simpson pada tahun 1920 dan telah diuji oleh para peneliti selama beberapa dekade (Simpson, 1920). Namun, teori lain juga dianggap penting untuk memahami curah hujan, seperti teori orografik, di mana curah hujan dipengaruhi oleh topografi dan elevasi, serta teori konvergensi, di mana hujan terjadi karena aliran udara yang bertemu dan memicu pembentukan awan (Hidrologi, 2018). Secara umum, curah hujan adalah fenomena yang sangat kompleks dan dipengaruhi oleh banyak faktor yang saling terkait. Oleh karena itu, pemahaman yang komprehensif tentang teori curah hujan adalah penting untuk mengelola sumber daya air dan lingkungan secara efektif.

Menurut BMKG (2010) satuan untuk mengukur curah hujan adalah milimeter. 1 milimeter curah hujan artinya air hujan jatuh dengan ketinggian 1mm pada luasan 1 meter persegi, dan tidak mengalir, meresap atau menguap. Sebagai contoh, jika terjadi curah hujan sebesar 10 mm pada luasan 100 meter persegi, maka jumlah air yang terkumpul adalah 1000 liter atau setara dengan 1 meter kubik air. Sedangkan jika terjadi curah hujan sebesar 100 mm pada luasan 1 kilometer persegi, maka jumlah air yang terkumpul adalah 100 juta liter air atau setara dengan 100 ribu meter kubik air.

Menurut BMKG (2010) curah hujan terdiri dari empat kelas, yaitu :

1. Rendah (R) $\Rightarrow < 20$ mm/hari
2. Sedang (S) $\Rightarrow 20-50$ mm/hari
3. Lebat (L) $\Rightarrow 50-100$ mm/hari

4. Sangat Lebat (SL) => >100 mm/hari

Kelas tersebut didasarkan pada analisis data historis yang dilakukan oleh BMKG dengan mengacu pada standar internasional dalam bidang meteorologi dan klimatologi. Dengan mempertimbangkan data curah hujan yang terkumpul dan dianalisis, BMKG dapat mengklasifikasikan curah hujan dalam kategori yang relevan dan bermanfaat dalam pemantauan cuaca dan prediksi iklim.

Terdapat beberapa faktor yang mempengaruhi pembagian kategori curah hujan, antara lain (Badwi dkk, 2017)

1. Suhu (X1)

Menurut Aini (2020), suhu dapat mempengaruhi curah hujan di suatu daerah. Daerah yang memiliki suhu rendah cenderung memiliki curah hujan yang juga rendah, sedangkan daerah dengan suhu yang tinggi memiliki curah hujan yang tinggi.

2. Kelembaban (X2)

Curah hujan dipengaruhi oleh tingkat kelembaban udara. Semakin tinggi tingkat kelembaban, maka kemungkinan curah hujan juga akan semakin tinggi. Sebaliknya, apabila tingkat kelembaban rendah, maka curah hujan cenderung rendah pula.

3. Kecepatan Angin (X3)

Kecepatan angin mempengaruhi curah hujan melalui proses pengangkatan uap air, namun kecepatan angin yang terlalu tinggi dapat menghambat pembentukan awan dan menurunkan curah hujan. sehingga jika kecepatan angin rendah, maka akan memungkinkan curah hujan yang tinggi.

4. Tekanan Udara (X4)

Tekanan udara dapat mempengaruhi curah hujan melalui pergerakan massa udara. Tekanan tinggi menurunkan suhu dan membentuk awan hujan ketika turun, sementara tekanan rendah dapat membentuk awan hujan ketika massa udara naik. Hubungan ini tergantung pada suhu, kelembaban, dan kondisi atmosfer.

5. Penyinaran Matahari (X5)

Penyinaran matahari dapat mempengaruhi curah hujan melalui pemanasan permukaan bumi yang dapat memicu pembentukan awan hujan. Sehingga, jumlah sinar matahari yang diterima suatu wilayah dapat mempengaruhi curah hujan.

6. Ketinggian Wilayah (X6)

Perubahan ketinggian wilayah dapat mempengaruhi kondisi iklim dan pola curah hujan. Wilayah yang lebih tinggi cenderung memiliki suhu yang lebih rendah dan kondisi atmosfer yang berbeda, yang dapat mempengaruhi pola pembentukan awan dan proses presipitasi.

7. Luas Wilayah (X7)

Luas wilayah dapat memberikan informasi tentang ukuran skala geografis yang relevan. Seperti saat memprediksi curah hujan di suatu wilayah yang luas, perlu mempertimbangkan variasi spasial yang mungkin terjadi dalam luas wilayah tersebut. Hal ini dapat membantu dalam pengambilan keputusan terkait pengelolaan sumber daya air, irigasi, perencanaan pertanian, dan mitigasi risiko banjir.

8. Jumlah Hari Hujan (X8)

Jumlah hari hujan dapat mempengaruhi curah hujan secara keseluruhan, dengan daerah yang banyak mengalami hujan menunjukkan kelebihan uap air di atmosfer dan daerah yang jarang hujan menunjukkan kondisi atmosfer yang kurang mendukung terbentuknya awan dan hujan.

2.2 Kajian Integrasi Topik dengan Al-Quran/Hadits

Hujan adalah salah satu anugerah Allah SWT yang diberikan kepada makhluk-Nya di dunia ini. Dalam al-Quran, hujan disebutkan sebagai salah satu dari banyak tanda kebesaran dan kekuasaan Allah SWT atas ciptaan-Nya. Allah SWT berfirman dalam Surat An-Nur ayat 43, yang artinya (Kemenag, 2023b):

"Tidaklah kamu melihat bahwa Allah mengarak awan, kemudian mengumpulkan antara (bagian-bagian)nya, kemudian menjadikannya bertindih-tindih, maka kelihatanlah olehmu hujan keluar dari celah-celahnya dan Allah (juga) menurunkan (butiran-butiran) es dari langit, (yaitu) dari (gumpalan-gumpalan awan seperti) gunung-gunung, maka ditimpakan-Nya (butiran-butiran) es itu kepada siapa yang dikehendaki-Nya dan dipalingkan-Nya dari siapa yang dikehendaki-Nya. Kilauan kilat awan itu hampir-hampir menghilangkan penglihatan."

Hujan dalam kebesaran-Nya, merupakan karunia Allah yang melimpah bagi makhluk-Nya. Hal ini mengingatkan akan kebesaran Allah yang mampu mengatur datangnya hujan b. Dalam hadits-hadits Rasulullah SAW, hujan juga diangkat sebagai tanda kebaikan dan rahmat Allah. Beliau menyebutkan bahwa hujan adalah tanda dari kebaikan dan karunia-Nya, yang memberi kehidupan pada bumi yang gersang dan menghidupkan tanaman-tanaman. Hadits ini mengajarkan untuk mensyukuri nikmat hujan dan memanfaatkannya dengan bijak.

Pemahaman mendalam mengenai proses dan pola curah hujan di Jawa Tengah melalui penerapan *Artificial Neural Network* (ANN) dalam penelitian ini dapat dipandang sebagai bentuk manifestasi ilmiah dari kebesaran Allah SWT.

Seseorang dapat lebih menghargai betapa canggihnya sistem alam yang diciptakan oleh Allah SWT, dengan memahami secara lebih mendalam mekanisme curah hujan.

Demikian pula, penelitian ini membantu untuk lebih menghormati dan mengapresiasi nikmat curah hujan. Seseorang dapat memaksimalkan manfaat dari rahmat Allah ini, dengan memanfaatkan teknologi canggih seperti ANN. Ini merupakan wujud konkret dari tindakan sebagai khalifah di muka bumi untuk memelihara dan memanfaatkan sumber daya alam dengan bijak.

Sehingga dapat disimpulkan bahwa integrasi antara ilmu pengetahuan dan ajaran agama dapat menghasilkan perspektif yang lebih menyeluruh terkait pentingnya hujan dalam kehidupan manusia. Al-Quran dan hadits Rasulullah SAW memberikan petunjuk tentang betapa besar dan bermaknanya nikmat hujan dalam tatanan alam semesta. Oleh karena itu, sebagai makhluk yang bertanggung jawab, adalah tugas kita untuk memanfaatkan pengetahuan ilmiah dan kearifan agama untuk merawat serta memanfaatkan nikmat alam ini secara bijak dan berkelanjutan. Dengan demikian, kita dapat memperkuat hubungan erat antara ilmu pengetahuan, agama, dan kehidupan sehari-hari dalam menghargai karunia Allah berupa hujan.

BAB III METODE PENELITIAN

3.1 Jenis Penelitian

Pada penelitian ini, akan digunakan pendekatan kombinasi yaitu antara studi literatur dan kuantitatif. Pendekatan studi literatur dilakukan dengan mengumpulkan referensi dari artikel, jurnal, dan buku yang relevan untuk mendukung penelitian ini. Sementara itu, pendekatan kuantitatif digunakan untuk mengolah data dalam bentuk angka yang diperoleh dari *website* Badan Pusat Statistik (BPS) pada masing-masing wilayah kabupaten/kota di Jawa Tengah.

3.2 Data dan Sumber Data

Jenis data yang digunakan pada penelitian ini yaitu jenis data sekunder yang diambil dari *website* Badan Pusat Statistik (BPS) pada beberapa kabupaten/kota di Jawa Tengah <https://jateng.bps.go.id/>. Data yang digunakan yaitu data periode 2019 hingga 2022 seperti pada Lampiran 1. Variabel yang digunakan dalam penelitian ini terdiri dari 1 variabel respon dan 8 variabel prediktor seperti pada Tabel 3.1.

Tabel 3.1 Variabel Penelitian

Simbol	Variabel	Jenis Data
Y	Curah Hujan	Kategorik
X_1	Suhu	Numerik
X_2	Kelembaban	Numerik
X_3	Kecepatan Angin	Numerik
X_4	Tekanan Udara	Numerik
X_5	Penyinaran Matahari	Numerik
X_6	Ketinggian Wilayah	Numerik
X_7	Luas Wilayah	Numerik
X_8	Jumlah Hari Hujan	Numerik

Dimana variabel respon yaitu curah hujan dalam penelitian ini akan dikelompokkan kedalam 4 kelas seperti pada Tabel 3.2.

Tabel 3.2 Kelas Curah Hujan

Simbol	Kelas	Interval
<i>R</i>	Rendah	< 20
<i>S</i>	Sedang	20 – 50
<i>L</i>	Lebat	50 – 100
<i>SL</i>	Sangat Lebat	> 100

3.3 Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini merupakan teknik dokumenter. Teknik dokumentasi dilakukan dengan mencari data atau informasi dari sumber tertulis atau dokumen tertentu yang relevan dengan topik penelitian (Nilamsari, 2014).

3.4 Langkah Penelitian

Penelitian ini melibatkan serangkaian tahapan yang dilakukan secara terstruktur dan sistematis agar memudahkan penulis dalam menentukan langkah selanjutnya untuk mencapai hasil yang diinginkan. Berikut merupakan langkah penelitian yang akan dilakukan:

1. Proses membuat model ANN pada data curah hujan di Jawa Tengah:
 - a. Mendefinisikan data *input* dan *output*. Berikut ilustrasi pada Tabel 3.3 penentuan data *input* dan *output* dimana data yang digunakan merupakan data bulanan tahun 2019 hingga 2022. Data tersebut akan dibuat model *backpropagation* untuk memprediksi curah hujan pada periode selanjutnya.

Tabel 3.3 Data Ilustrasi

X1	X2	X3	X4	X5	X6	X7	X8	Y
27.9	83	3	1008.6	44.4	7010	2124.47	24	SL
27.7	81	2	1009.1	48.6	7010	2124.47	24	R
27.3	85	2	1009.5	69.9	7010	2124.47	25	L
28.3	82	3	1010.7	80.4	7010	2124.47	10	R
27.9	81	5	1010.1	83.1	7010	2124.47	13	SL
25.5	84	4	1011.2	73	7010	2124.47	3	S

Pada Tabel 3.3 menunjukkan bahwa variabel *input* yang digunakan terdapat 8 yaitu variabel X1 hingga X8 dan terdapat 1 variabel *output* yaitu Y yang berupa kategorik dengan 4 kelas yaitu R, S, L, SL. Sehingga, model yang digunakan memiliki 8 buah *input* dan 4 buah *output*.

b. Pengolahan Data

i) Data Kategorik

Pada data ketegorik perlu dilakukan *one-hot ncoding* untuk menentukan fitur biner pada masing-masing kategori.

ii) Data Numerik

Pada data numrik perlu dilakukan normalisasi data agar setiap variabel berada pada skala yang serupa. Proses normalisasi dilakukan dengan menggunakan metode *min-max normalization* dengan persamaan (2.22).

c. Membagi data menjadi data *training* dan data *testing*.

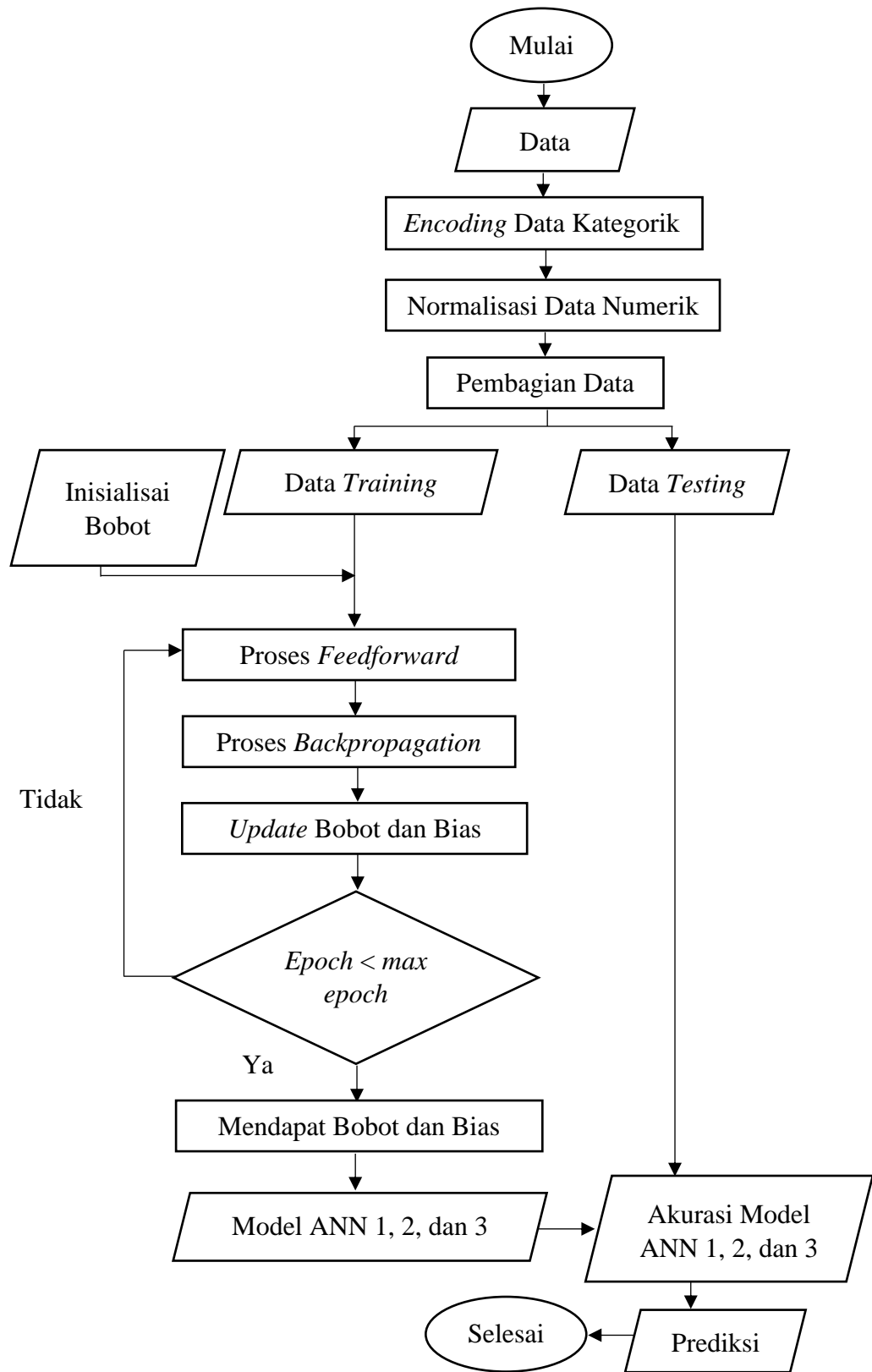
d. Membangun model ANN dengan menggunakan data *training*.

i) Memilih model yang memiliki nilai *loss* terkecil

ii) Inisialisasi bobot dan bias awal

- iii) Melakukan tiga tahapan utama *backpropagation*. Pertama yaitu melakukan propagasi maju dimana setiap unit pada *hidden layer* akan dilakukan aktivasi dengan fungsi aktivasi Relu pada persamaan (2.2), sedangkan setiap unit pada *output layer* menggunakan fungsi aktivasi *softmax* pada persamaan (2.3). Tahap kedua yaitu melakukan propagasi mundur dengan menghitung *gradient loss* pada setiap *output* dengan metode *Categorical Cross-Entropy* menggunakan persamaan (2.4). kemudian, untuk tahap *update* bobot dan bias menggunakan algoritma *optimizer* Adam dengan persamaan (2.14) hingga (2.21).
2. Proses perhitungan nilai akurasi prediksi:
 - a. Mengevaluasi model ANN untuk klasifikasi menggunakan data *training* dan data *testing*. Setelah dilakukan pelatihan pada model ANN, maka model ANN dievaluasi.
 - b. Menghitung nilai akurasi model ANN. Untuk mendapatkan nilai akurasi akan digunakan metode akurasi, presisi, *recall*, *F1 score*, dan MAE untuk data *testing* dengan persamaan yaitu (2.23) hingga (2.30).
 3. Proses melakukan prediksi untuk klasifikasi curah hujan:
 - a. Melakukan prediksi menggunakan model yang sudah dilatih.
 - b. Membuat kesimpulan.

3.5 Flowchart

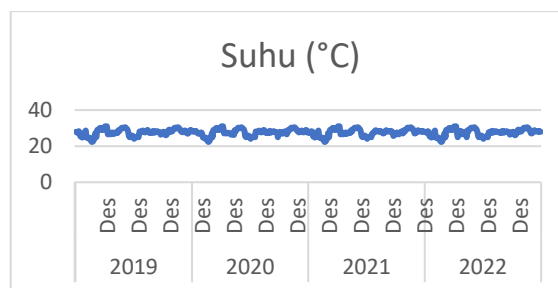


Gambar 3.1 Flowchart Tahapan Prediksi Menggunakan ANN dengan Algoritma Backpropagation

BAB IV PEMBAHASAN

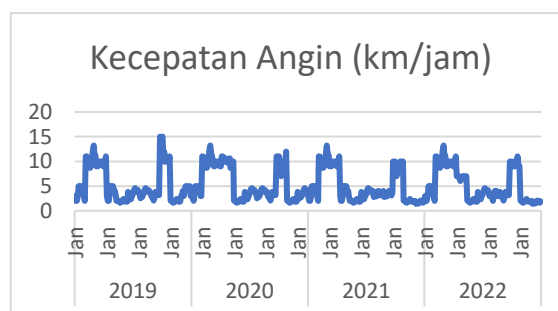
4.1 Analisis Deskriptif

Data yang digunakan adalah data dari curah hujan dalam bentuk bulanan dari 11 wilayah kabupaten/koata di Jawa Tengah. Data yang digunakan dari Januari 2019 – Desember 2022 berjumlah 528 (lihat Lampiran 1). Untuk pengolahan data dibagi jadi dua, yaitu data pelatihan (*training*) dan juga data pengujian (*testing*). Data pelatihan digunakan sebesar 80% (422 data) dan data pengujian sebesar 20% (106 data). Plot dari setiap variabel yang digunakan sebagai masukan dalam ANN ditunjukkan dalam Gambar 4.1 hingga Gambar 4.7 kecuali variabel ketinggian dan luas wilayah dikarenakan tidak terdapat perubahan pada variabel tersebut.



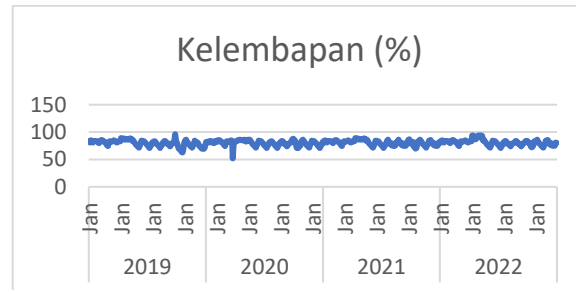
Gambar 4.1 Plot Suhu pada Tahun 2019-2022

Pada Gambar 4.1, dapat dilihat bahwa grafik pada variabel suhu menunjukkan bahwa data stasioner.



Gambar 4.2 Plot Kecepatan Angin pada Tahun 2019-2022

Pada Gambar 4.2 menunjukkan bahwa pola kecepatan angin membentuk pola siklis. Kecepatan angin akan mengalami kenaikan pada setiap awal dan akan akhir tahun.



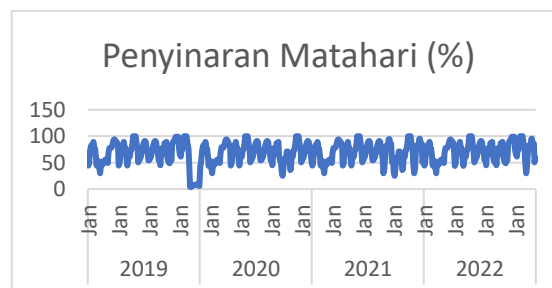
Gambar 4.3 Plot Kelembaban pada Tahun 2019-2022

Pada Gambar 4.3 menunjukkan bahwa pola kelembaban sudah cenderung stabil atau stasioner.



Gambar 4.4 Plot Tekanan Udara pada Tahun 2019-2022

Pada Gambar 4.4 menunjukkan bahwa pola tekanan udara cenderung membentuk pola musiman. Tekanan udara akan mengalami penurunan setiap akhir tahun.



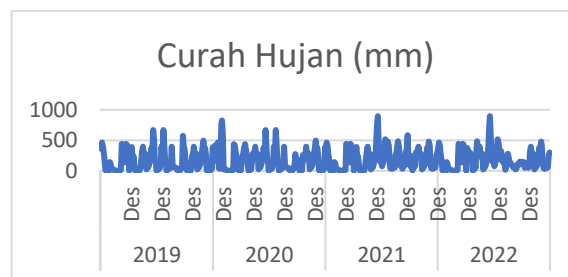
Gambar 4.5 Plot Penyinaran Matahari pada Tahun 2019-2022

Pada Gambar 4.5 menunjukkan bahwa grafik penyinaran matahari telah membentuk pola stasioner.



Gambar 4.6 Plot Hujan pada Tahun 2019-2022

Pada Gambar 4.6 juga menunjukkan bahwa grafik hari hujan telah membentuk pola stasioner.



Gambar 4.7 Plot Curah Hujan pada Tahun 2019-2022

Sama dengan plot sebelumnya, pada Gambar 4.7 juga menunjukkan bahwa grafik curah hujan telah membentuk pola stasioner.

4.2 Analisis Data

4.2.1 *Encoding* Data Kategorik

Proses *encoding* data adalah langkah penting untuk mengubah data dari bentuk bahasa manusia menjadi format yang dapat diproses oleh program komputer. Dalam tahap ini, data disusun ulang menjadi bentuk yang sesuai dengan tata aturan yang dikenali oleh komputer. Hasilnya, program dapat melakukan berbagai operasi, manipulasi, dan analisis pada data tersebut. Dalam konteks

penelitian ini, variabel *output* memiliki 4 kategori berbeda. Untuk mengintegrasikan variabel tersebut ke dalam analisis, langkah pertama adalah menggunakan *label encoder* untuk mentransformasikannya menjadi bentuk numerik. Selanjutnya, guna menghindari penafsiran yang tidak tepat berdasarkan urutan atau nilai berurutan pada variabel kategorikal, digunakan metode *one-hot encoding* yaitu dengan mengubah data menjadi vektor biner. Setiap kategori mendapatkan representasi biner dengan panjang vektor yang konsisten, mewakili keberadaan atau ketiadaan suatu kategori. Pada Lampiran 2 merupakan hasil *encoding* untuk data kategorik. Tabel 4.1 menunjukkan *encoding* masing-masing kategori.

Tabel 4.1 *Encoding* Data Kategorik

Kategori	Label Encoding	One-Hot Encoding
R	0	[1, 0, 0, 0,]
S	1	[0, 1, 0, 0,]
L	2	[0, 0, 1, 0,]
SL	3	[0, 0, 0, 1,]

4.2.2 Normalisasi Data Numerik

Setiap variabel masukan dalam penelitian ini memiliki skala yang berbeda, sehingga dilakukan normalisasi untuk menyamakan rentang nilai masing-masing variabel. Hal ini bertujuan agar nilai-nilai tersebut terstandarisasi dan dapat digunakan bersama-sama dalam analisis pembelajaran tanpa menghadapi hambatan akibat perbedaan skala. Persamaan (2.22) adalah contoh perhitungan yang mengikuti metode *Min-Max Normalization*. Hasil normalisasi dari data *training* dan data *testing* terdapat pada Lampiran 3 dan 4.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

$$x_{min} \quad : \quad 22.3^{\circ}\text{C}$$

$$x_{max} \quad : \quad 31^{\circ}\text{C}$$

$$x_{min} - x_{max} \quad : \quad 8.7^{\circ}\text{C}$$

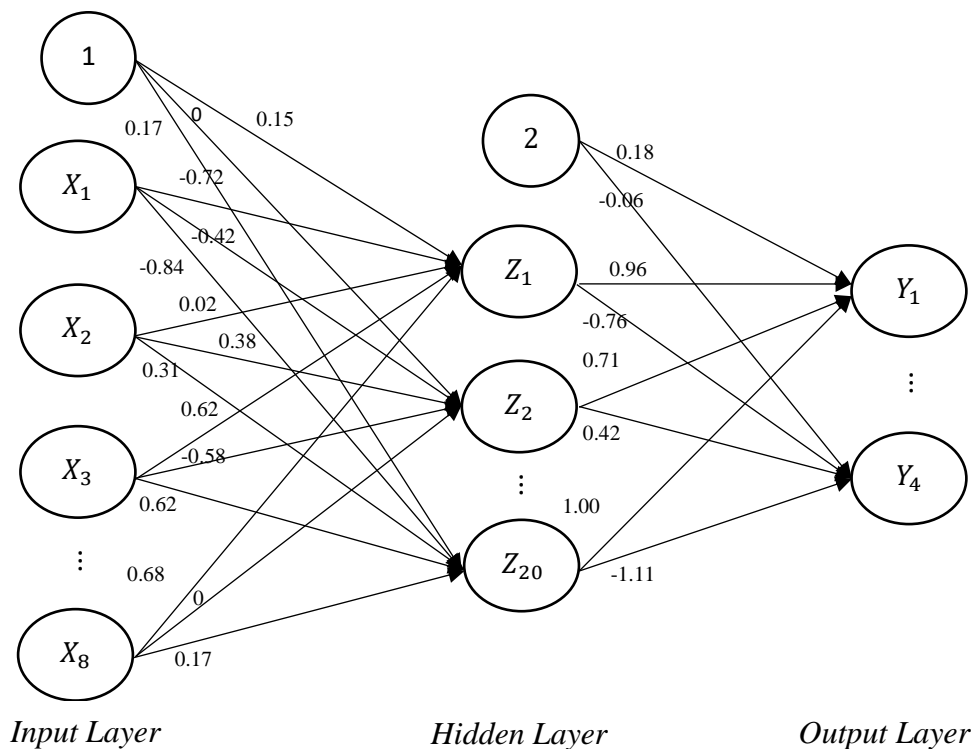
4.2.3 Model ANN

Pada penelitian untuk memprediksi curah hujan di 11 wilayah di Jawa Tengah, digunakan sebuah model ANN dengan arsitektur *multi-layer neural network* yang melibatkan 3 variasi model dengan perbedaan dalam jumlah lapisan pada hidden layer. Setiap model akan memiliki jumlah *input* sebanyak 8 neuron dan jumlah *output* sebanyak 4 neuron. Model pertama akan terdiri dari 1 lapisan pada *hidden layer*. Model kedua akan memiliki 2 lapisan pada *hidden layer*. Model ketiga akan memiliki 3 lapisan pada *hidden layer*, dengan jumlah neuron pada setiap *hidden layer* dalam rentang 4 hingga 32 neuron akan menjadi fokus ujiannya. Dalam pelatihan model ini, algoritma *backpropagation* digunakan untuk menyesuaikan bobot jaringan guna mendekati prediksi yang akurat. Fungsi aktivasi yang digunakan untuk *hidden layer* adalah ReLU, sementara untuk lapisan output menggunakan fungsi softmax untuk pengklasifikasian hasil prediksi.

4.2.3.1 Model Pertama

Setelah proses *encoding* atau normalisasi data dilakukan, langkah berikutnya adalah membagi data menjadi dua subset utama yaitu data *training* dan data *testing*. Pembagian dataset dilakukan secara acak dengan perbandingan 80% untuk data *training* dan 20% untuk data *testing*. Pada Lampiran 5 menunjukkan percobaan beberapa model untuk mendapatkan model terbaik. Dari percobaan tersebut, ditemukan bahwa model dengan kinerja paling baik memiliki nilai *loss* terendah pada tahap pelatihan sebesar 0.4729. Oleh karena itu, model

yang akan diadopsi dalam tahap pengujian pertama adalah model tipe E, yang dibangun dengan memanfaatkan 20 neuron pada *hidden layer*. Berdasarkan dari hasil pengujian menegaskan bahwa asumsi umum yang mengaitkan penurunan nilai *loss* secara *linear* dengan jumlah neuron dalam *hidden layer* tidak selalu berlaku. Dari hasil tersebut, tidak dapat dijamin bahwa jumlah neuron yang lebih sedikit atau lebih banyak dalam *hidden layer* secara konsisten akan menghasilkan nilai *loss* yang lebih rendah atau lebih tinggi. Kesimpulan tersebut menunjukkan kompleksitas hubungan antara arsitektur model dan kinerja yang dihasilkannya, yang memerlukan pendekatan eksperimental yang teliti dan tidak bersifat kaku atau mengikat. Gambar 4.8 menunjukkan model jaringan dari model E.



Gambar 4.8 Model Jaringan E dengan Bobot Akhir

Setelah ditentukan model terbaik, maka akan dilakukan perhitungan dengan langkah-langkah perhitungan pada data *training* sebagai berikut:

Langkah 1: Inisialisasi bobot dan bias awal dengan bilangan acak yang dapat dilihat pada Lampiran 6 untuk inisialisasi bobot dan bias awal dari *input* ke *hidden layer*, sedangkan pada Lampiran 7 untuk inisialisasi bobot dan bias awal dari *hidden* ke *output layer*.

Langkah 2: Menghitung nilai *hidden layer* setiap unit *hidden layer* $Z_{net_j}, j = 1, 2, 3, \dots, p$.

$$Z_{net_j} = \left(\sum_{i=1}^n X_i W_j \right) + W_{j0}$$

$$\begin{aligned} Z_{net_1} = & (0.6552 (-0.1191) + 0.5343 (-0.1184) + 0.0441 \times \\ & \times (0.1211) + 0.9377 (0.3606) + 0.7975(-0.3822) + 1 \times \\ & \times (0.1778) + 0 (-0.1537) + 0.2258 (0.4272)) + 0 \end{aligned}$$

$$Z_{net_1} = 0.1716$$

Z_{net_1} merupakan *output hidden layer* pada neuron ke-1 sebelum menggunakan fungsi aktivasi yang akan ditransfer ke *output layer*. Nilai 0.6552, 0.5343, ..., 0.2258 merupakan variabel *input* X_1, X_2, \dots, X_8 . Nilai (-0.1191), (-0.1184), ..., 0.4272 merupakan bobot awal antara *input layer* dan *hidden layer* dan nilai 0 merupakan bias awal *input layer* dan *hidden layer*. Dari persamaan tersebut menyatakan bahwa jika X_2 hingga X_8 bernilai tetap dan X_1 dinaikkan satu satuan, maka Z_{net_1} akan bertambah sebesar 0.1393 satuan. Jika X_1 hingga X_4 dan X_6 hingga X_8 bernilai tetap dan X_5 dinaikkan satu satuan, maka Z_{net_1} akan berkurang sebesar 0.2057 satuan, begitupun untuk X yang lainnya. Tabel 4.2 berikut merupakan hasil pada *hidden layer*.

Tabel 4.2 Hasil *Hidden Layer* Sebelum Aktivasi

Z_{net_1}	0.1716	$Z_{net_{11}}$	-0.1971
Z_{net_2}	0.1738	$Z_{net_{12}}$	-0.0166
Z_{net_3}	1.0247	$Z_{net_{13}}$	-0.4683
Z_{net_4}	-0.8618	$Z_{net_{14}}$	0.4520
Z_{net_5}	0.5691	$Z_{net_{15}}$	0.6740
Z_{net_6}	-0.4307	$Z_{net_{16}}$	0.6903
Z_{net_7}	0.1181	$Z_{net_{17}}$	-0.4044
Z_{net_8}	0.1162	$Z_{net_{18}}$	0.4280
Z_{net_9}	-0.1273	$Z_{net_{19}}$	0.5271
$Z_{net_{10}}$	0.3230	$Z_{net_{20}}$	-0.1905

Tabel 4.2 merupakan contoh hasil dari neuron ke 1, untuk hasil lengkapnya dapat dilihat pada Lampiran 8. Selanjutnya, setiap nilai pada Tabel 4.2 tersebut dilakukan perhitungan dengan menggunakan fungsi aktivasi ReLu untuk mendapatkan nilai akhir *output* pada *hidden layer*.

$$Z_j = f(Z_{net_j}) = \max(0, Z_{net_j})$$

$$Z_1 = f(Z_{net_1}) = \max(0, 0.1716) = 0.1716$$

Z_1 adalah Z_{net_1} yang telah diaktivasi menggunakan fungsi aktivasi ReLu. Z_{net_1} bernilai 0.1716. Nilai *output* pada *hidden layer* (Z_1) didapatkan dengan memasukkan nilai Z_{net_1} kedalam fungsi aktivasi ReLu dengan mengambil nilai paling besar antara 0 dengan nilai Z_{net_1} sehingga didapatkan nilai sebesar 0.1716. Nilai Z_1 akan menjadi nilai *output* pada *hidden layer* ke 1 yang akan ditransfer ke *output layer*. Tabel 4.3 berikut merupakan hasil pada *hidden layer* neuron ke 1 setelah diaktivasi, untuk hasil lengkapnya dapat dilihat pada Lampiran 9.

Tabel 4.3 Hasil Hidden Layer Setelah Aktivasi

Z_1	0.1716	Z_{11}	0.0000
Z_2	0.1738	Z_{12}	0.0000
Z_3	1.0247	Z_{13}	0.0000
Z_4	0.0000	Z_{14}	0.4520
Z_5	0.5691	Z_{15}	0.6740
Z_6	0.0000	Z_{16}	0.6903
Z_7	0.1181	Z_{17}	0.0000
Z_8	0.1162	Z_{18}	0.4280
Z_9	0.0000	Z_{19}	0.5271
Z_{10}	0.3230	Z_{20}	0.0000

Langkah 3: Menghitung nilai pada *output layer* ($y_k, k = 0, 1, 2, 3$) dengan menggunakan rumus:

$$y_{net_k} = \left(\sum_{i=1}^p Z_j W_j \right) + W_{k0}$$

$$\begin{aligned} y_{net_0} &= (Z_1(0.3621) + Z_2(0.3955) + \dots + Z_{20}(0.3667)) + 0 \\ &= -0.8258 \end{aligned}$$

$$\begin{aligned} y_{net_1} &= (Z_1(0.1571) + Z_2(-0.4988) + \dots + Z_{20}(0.1229)) + 0 \\ &= 0.2241 \end{aligned}$$

$$\begin{aligned} y_{net_2} &= (Z_1(0.1042) + Z_2(0.4152) + \dots + Z_{20}(0.2462)) + 0 \\ &= 0.0267 \end{aligned}$$

$$\begin{aligned} y_{net_3} &= (Z_1(-0.0849) + Z_2(0.0687) + \dots + Z_{20}(-0.4895)) + 0 \\ &= -0.0225 \end{aligned}$$

y_{net_k} adalah *output* pada *output layer* pada neuron ke 1 yang akan menjadi nilai klasifikasi curah hujan sebelum dilakukan aktivasi. Nilai 0.3621, 0.3955, ..., 0.3667 merupakan bobot awal antara *input layer* dan *hidden layer* pada y_{net_0} dan nilai 0 merupakan bias awal *input layer* dan *hidden layer*, dan begitu seterusnya

untuk y_{net_1} hingga y_{net_3} . Setelah didapatkan *output* tersebut, kemudian menghitung sinyal *output* menggunakan fungsi aktivasi *softmax*,

$$y_k = softmax(y_{net_k}) = \frac{e^{y_{net_k}}}{\sum e^{y_{net_k}}}$$

$$y_0 = \frac{e^{(-0.8258)}}{e^{(-0.8258)} + e^{(0.2241)} + e^{(0.0267)} + e^{(0-0.0225)}} \approx 0.1185$$

$$y_1 = \frac{e^{(0.2241)}}{e^{(-0.8258)} + e^{(0.2241)} + e^{(0.0267)} + e^{(0-0.0225)}} \approx 0.3387$$

$$y_2 = \frac{e^{(0.0267)}}{e^{(-0.8258)} + e^{(0.2241)} + e^{(0.0267)} + e^{(0-0.0225)}} \approx 0.2780$$

$$y_3 = \frac{e^{(0-0.0225)}}{e^{(-0.8258)} + e^{(0.2241)} + e^{(0.0267)} + e^{(0-0.0225)}} \approx 0.2647$$

y_k merupakan nilai *output* yang telah diaktivasi menggunakan fungsi aktivasi *softmax*. Untuk hasil lengkap prediksi klasifikasi sesudah aktivasi dapat dilihat pada Lampiran 10. Hasil keluaran tersebut membentuk matriks [0.1185 0.3387 0.2780 0.2647]. Dari matriks hasil prediksi ini, langkah selanjutnya adalah melakukan *decoding* dengan menetapkan nilai tertinggi dari seluruh nilai dalam matriks sebagai representasi hasil klasifikasinya. Karena nilai tertinggi adalah 0.3387, maka matriks akan berubah menjadi [0 1 0 0]. Oleh karena itu, prediksi menunjukkan bahwa klasifikasi curah hujan adalah kelas 1 atau S (sedang).

Langkah 4: Menghitung nilai *loss* dengan rumus berikut:

$$Loss_t = - \sum_t (y_{target_k} (\log(y_k))) ; \text{ dengan } k = 0, 1, 2, 3.$$

$$Loss_0 = -(0(\log(0.1185))) \approx 0$$

$$Loss_1 = -(0(\log(0.3387))) \approx 0$$

$$Loss_2 = -(0(\log(0.2780))) \approx 0$$

$$Loss_3 = -(1(\log(0.2647))) \approx -0.5773$$

$$Loss_t = -(0 + 0 + 0 + 0.4208) = 0.5773$$

Dari hasil *loss* tersebut dapat dikatakan bahwa model telah memahami dengan baik karakteristik data pertama dan mampu mengenali kelas yang sebenarnya dengan akurat.

Langkah 5: Menghitung *gradient loss* terhadap aktivasi *output*:

$$dy_k = -\left(\frac{y}{y_k}\right)$$

$$dy_1 = -\left(\frac{[0, 0, 0, 1]}{[0.1185, 0.3387, 0.2780, 0.2647]}\right)$$

$$dy_1 = [0 \quad 0 \quad 0 \quad 3.78]$$

Menghitung *gradient aktivasi* terhadap nilai masukan *output layer (softmax)*:

$$dy_{net_k} = y_k - y$$

$$dy_{net_1} = [0.1185 \quad 0.3387 \quad 0.2780 \quad 0.2647] - [0 \quad 0 \quad 0 \quad 1]$$

$$dy_{net_1} = [0.1185 \quad 0.3387 \quad 0.2780 \quad -0.7353]$$

Menghitung *gradient bobot dan bias output layer*:

$$dW_j = Z_j(dy_{net_k})$$

$$dW_1 = 0.1716([0.1185 \quad 0.3387 \quad 0.2780 \quad -0.7353])$$

$$dW_1 = [0.0203 \quad 0.0581 \quad 0.0477 \quad -0.1262]$$

$$db_j = \frac{1}{m} \sum_{i=1}^m dy_{net_k}$$

$$db_1 = \frac{1}{4}(0.1185 + 0.3387 + 0.2780 - 0.2647)$$

$$db_1 = 0.1176$$

Menghitung *loss gradient* terhadap aktivasi *hidden layer*:

$$dZ_j = dy_{net_k}(W_j)$$

$$dZ_1 = [0.1185, 0.3387, 0.2780, -0.7353](0.3621)$$

$$dZ_1 = [0.0429 \quad 0.1226 \quad 0.1006 \quad -0.2662]$$

Menghitung *gradient* aktivasi terhadap masukan *hidden layer (ReLU)*:

$$dZ_{net_j} = dZ_j(Z_j)$$

$$dZ_{net_1} = [0.0429 \quad 0.1226 \quad 0.1006 \quad -0.2662](0.1716)$$

$$dZ_{net_1} = [0.0074 \quad 0.0210 \quad 0.0173 \quad -0.0457]$$

Menghitung *gradient* bobot dan bias *hidden layer*:

$$dW_j = X_i(dZ_{net_j})$$

$$dW_1 = (0.6552)[0.0074 \quad 0.0210 \quad 0.0173 \quad -0.0457]$$

$$dW_1 = [0.0048 \quad 0.0138 \quad 0.0113 \quad -0.0299]$$

$$db_j = \frac{1}{m} \sum_{i=1}^m dZ_{net_j}$$

$$db_1 = \frac{1}{4}(0.0048 + 0.0138 + 0.0113 - 0.0299)$$

$$db_1 = 0$$

Langkah 6: Melakukan optimasi untuk mendapatkan bobot akhir. Selama tahap pelatihan model, algoritma optimisasi akan menerapkan perubahan pada bobot $W_{1,1}$ sesuai dengan besaran gradien ini. Pada setiap iterasi, nilai bobot $W_{1,1}$ akan *update* sebagai langkah menuju peningkatan kualitas prediksi yang dihasilkan oleh neuron pertama. Dalam algoritma optimasi Adam, terdapat beberapa langkah

perhitungan dengan parameter *optimizer* Adam yang telah diinisialisasi sebagai berikut:

$$\text{Learning Rate} = \alpha = 0.001$$

$$\text{Beta 1} = \beta_1 = 0.9$$

$$\text{Beta 2} = \beta_2 = 0.999$$

$$\text{Epsilon} = \epsilon = 0.00000001 = 1 \times 10^{-7}$$

(2) Menghitung *moving average* gradien sebelumnya (m_t) pada *output*:

$$m_{t+1} = \beta_1(m_t) + (1 - \beta_1)(dW_j)$$

$$m_{t+1} = 0.9(m_t) + (0.1)[0.0203 \quad 0.0581 \quad 0.0477 \quad -0.1262]$$

Jika $m_0 = 0$ (dimulai dari 0), maka

$$m_1 = (0.1)[0.0203 \quad 0.0581 \quad 0.0477 \quad -0.1262]$$

$$m_1 = [0.00203 \quad 0.00581 \quad 0.00477 \quad -0.01262]$$

$$\hat{m}_{t+1} = \frac{m_{t+1}}{1 - \beta_1^{t+1}}$$

$$\hat{m}_1 = \frac{[0.00203 \quad 0.00581 \quad 0.00477 \quad -0.01262]}{1 - 0.9}$$

$$\hat{m}_1 = [0.0203 \quad 0.0581 \quad 0.0477 \quad -0.1262]$$

Menghitung *moving average* gradien kuadrat sebelumnya (v_t) pada *output*:

$$v_{t+1} = \beta_2(v_t) + (1 - \beta_2)(dW_j)^2$$

$$v_{t+1} = 0.999(v_t) + 0.001([0.0203 \quad 0.0581 \quad 0.0477 \quad -0.1262])^2$$

Jika $v_0 = 0$ (dimulai dari 0), maka

$$v_1 = 0.001([0.0203 \quad 0.0581 \quad 0.0477 \quad -0.1262])^2$$

$$v_1 = [0.00000041 \quad 0.0000034 \quad 0.0000023 \quad 0.000016]$$

$$\hat{v}_{t+1} = \frac{v_{t+1}}{1 - \beta_2^{t+1}}$$

$$\hat{v}_1 = \frac{[0.00000041 \quad 0.0000034 \quad 0.0000023 \quad 0.000016]}{1 - 0.999}$$

$$\hat{v}_1 = [0.00041 \quad 0.0034 \quad 0.0023 \quad 0.016]$$

(3) Pembaruan Bobot dan Bias *output*:

$$W_{1(baru)} = W_{1(lama)} - \alpha \frac{(\hat{m}_1)}{\sqrt{\hat{v}_1 + \epsilon}}$$

$$W_{1(baru)} = [0.3621 \quad 0.1571 \quad 0.1042 \quad -0.0849] - ((0.001) \times \\ \times \frac{[0.0203 \quad 0.0581 \quad 0.0477 \quad -0.1262]}{\sqrt{[0.00041 \quad 0.0034 \quad 0.0023 \quad 0.016] + (1 \times 10^{-7})}})$$

$$W_{1(baru)} = [0.9620 \quad 0.2426 \quad 0.1506 \quad -0.7571]$$

$$b_{1(baru)} = b_{1(lama)} - \alpha \frac{(\hat{m}_1)}{\sqrt{\hat{v}_1 + \epsilon}}$$

$$b_{1(baru)} = [0.0000 \quad 0.0000 \quad 0.0000 \quad 0.0000] - ((0.001) \times \\ \times \frac{[0.0203 \quad 0.0581 \quad 0.0477 \quad -0.1262]}{\sqrt{[0.00041 \quad 0.0034 \quad 0.0023 \quad 0.016] + (1 \times 10^{-7})}})$$

$$b_{1(baru)} = [0.1785 \quad 0.0786 \quad -0.1775 \quad -0.0613]$$

(4) Menghitung *moving average* gradien sebelumnya (m_t) pada *hidden*:

$$m_{t+1} = \beta_1(m_t) + (1 - \beta_1)(dW_j)$$

$$m_{t+1} = 0.9(m_t) + (0.1)[0.0048 \quad 0.0138 \quad 0.0113 \quad -0.0299]$$

Jika $m_0 = 0$ (dimulai dari 0), maka

$$m_1 = (0.1)[0.0048 \quad 0.0138 \quad 0.0113 \quad -0.0299]$$

$$m_1 = [0.00048 \quad 0.00138 \quad 0.00113 \quad -0.00299]$$

$$\hat{m}_{t+1} = \frac{m_{t+1}}{1 - \beta_1^{t+1}}$$

$$\hat{m}_1 = \frac{[0.00048 \quad 0.00138 \quad 0.00113 \quad -0.00299]}{1 - 0.9}$$

$$\hat{m}_1 = [0.0048 \quad 0.0138 \quad 0.0113 \quad -0.0299]$$

Menghitung *moving average* gradien kuadrat sebelumnya (v_t) pada *hidden*:

$$v_{t+1} = \beta_2(v_t) + (1 - \beta_2)(dW_j)^2$$

$$v_{t+1} = 0.999(v_t) + 0.001([0.0048 \quad 0.0138 \quad 0.0113 \quad -0.0299])^2$$

Jika $v_0 = 0$ (dimulai dari 0), maka

$$v_1 = 0.001([0.0048 \quad 0.0138 \quad 0.0113 \quad -0.0299])^2$$

$$v_1 = [0.00000023 \quad 0.00000019 \quad 0.00000013 \quad 0.00000089]$$

$$\hat{v}_{t+1} = \frac{v_{t+1}}{1 - \beta_2^{t+1}}$$

$$\hat{v}_1 = \frac{[0.00000023 \quad 0.00000019 \quad 0.00000013 \quad 0.00000089]}{1 - 0.999}$$

$$\hat{v}_1 = [0.000023 \quad 0.000019 \quad 0.000013 \quad 0.000089]$$

(5) Pembaruan Bobot dan Bias *hidden*:

$$W_{1(bar)} = W_{1(lama)} - \alpha \frac{(\hat{m}_1)}{\sqrt{\hat{v}_1} + \epsilon}$$

$$W_{1(bar)} = [-0.12 \quad -0.12 \quad 0.12 \quad 0.36 \quad \dots \quad 0.43] - ((0.001) \times \frac{[0.0048 \quad 0.0138 \quad 0.0113 \quad -0.0299]}{\sqrt{[0.000023 \quad 0.000019 \quad 0.000013 \quad 0.000089] + (10^{-7})}})$$

$$W_{1(bar)} = [-0.72 \quad 0.02 \quad 0.62 \quad 0.35 \quad \dots \quad 0.68]$$

$$b_{1(bar)} = b_{1(lama)} - \alpha \frac{(\hat{m}_1)}{\sqrt{\hat{v}_1} + \epsilon}$$

$$b_{1(bar)} = [0.00 \quad 0.00 \quad 0.00 \quad 0.00 \quad \dots \quad 0.00] - (0.001) \times$$

$$\times \frac{[0.0203 \quad 0.0581 \quad 0.0477 \quad -0.1262]}{\sqrt{[0.00041 \quad 0.0034 \quad 0.0023 \quad 0.016] + (1 \times 10^{-7})}}$$

$$b_{1(\text{baru})} = [0.15 \quad 0.00 \quad -0.21 \quad 0.11 \quad \dots \quad 0.17]$$

Cara yang sama terus dilakukan pada neuron-neuron selanjutnya untuk *update* bobot dan bias. Perubahan bobot dilakukan sampai terget *error* yang diinginkan tercapai. Bobot dan bias akhir dari *input* ke *hidden layer* dapat dilihat pada Lampiran 11. Bobot dan bias akhir antara *hidden* ke *output layer* terdapat pada Lampiran 12.

Penelitian yang dilakukan pada model pertama didapatkan arsitektur jaringan terbaik yaitu 8 neuron *input layer*, 20 neuron *hidden layer*, dan 4 *output layer* dengan *epoch* sebanyak 100. Bobot yang didapatkan ditentukan secara otomatis melalui *software* Python. Berikut adalah interpretasi dari perhitungan *output hidden layer* pada Z_1 dan *output layer* pada y :

$$Z_j = f\left(\left(\sum_{i=1}^n X_i V_j\right) + V_{j0}\right)$$

$$Z_j = f(-0.72x_1 + 0.02x_2 + 0.62x_3 + 0.35x_4 - 0.83x_5 + 0.18x_6 + 0.70x_7 + 0.68x_8) + 0.15$$

Z_j merupakan *output* pada *hidden layer* yang akan ditransfer ke *output layer* menggunakan fungsi aktivasi ReLu (f). Nilai -0.72, 0.02, ..., 0.68 adalah bobot antara *input* dan *hidden layer* dimana jika bobot tersebut bernilai positif artinya terdapat hubungan positif antara bobot dan Z_j yang dapat menaikkan nilai Z_j , begitupun sebaliknya jika bobot tersebut bernilai negatif artinya terdapat hubungan negatif antara bobot dan Z_j yang dapat menurunkan nilai Z_j . Sedangkan

0.15 merupakan bias antara *input* dan *hidden layer*. Berikut adalah model secara matematis ANN:

$$y_k = f \left(\left(\sum_{i=1}^p Z_j W_j \right) + W_{k0} \right)$$

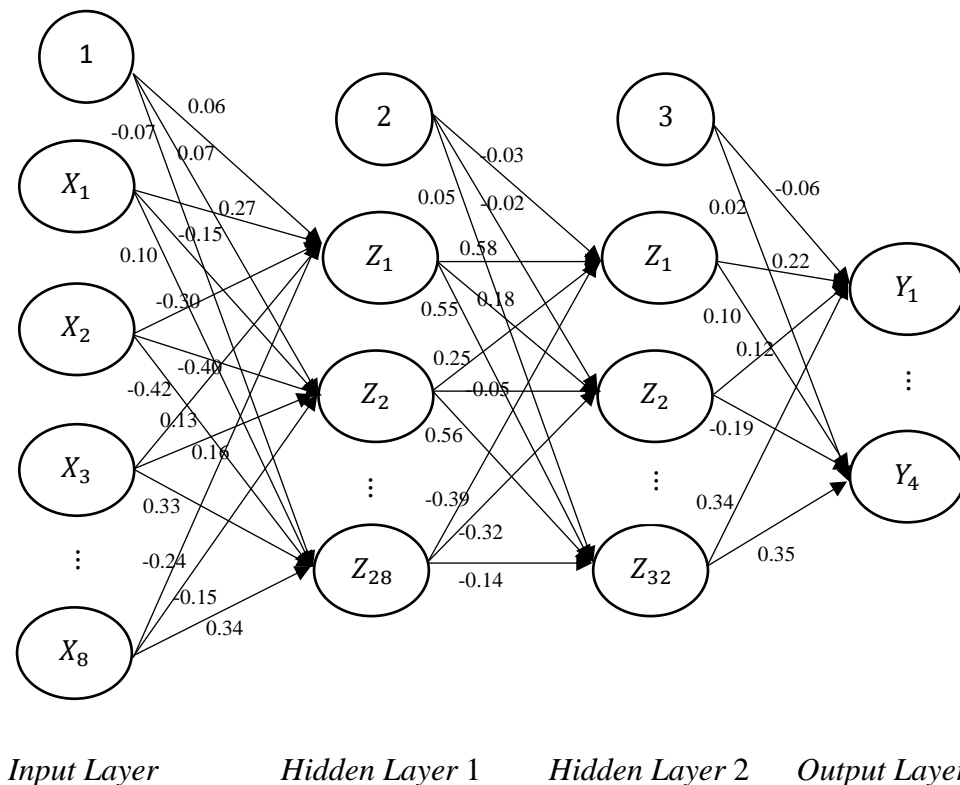
$$y_k = f(0.9620Z_1 + 0.2426Z_2 + \dots + 1.0037Z_{20}) + 0.1785$$

y_k merupakan *output* atau hasil prediksi dari curah hujan di 11 wilayah di Jawa Tengah setelah diaktivasi menggunakan fungsi aktivasi *softmax* (f). Nilai 0.9620, 0.2426, ..., 1.0037 merupakan bobot antara *hidden* ke *output layer*. Jika Z_2, \dots, Z_{20} nilainya tetap dan Z_1 (nilai *hidden layer* ke 1) dinaikkan satu satuan, maka nilai dari prediksi curah hujan sebelum diaktivasi akan bertambah sebesar 0.9620, bernilai positif artinya terdapat hubungan positif antara bobot dengan nilai prediksi curah hujan sebelum diaktivasi, begitu seterusnya hingga Z_{20} (nilai pada *hidden layer* ke 20).

4.2.3.2 Model Kedua

Pada Model Kedua, dilakukan eksplorasi melalui penambahan *layer* pada bagian *hidden layer* dengan tujuan meningkatkan kompleksitas representasi yang dihasilkan. Serangkaian variasi model dievaluasi untuk memperoleh model optimal, seperti pada Lampiran 13. Hasil percobaan mengungkapkan satu model unggul yang berhasil mencapai konvergensi dengan nilai *loss* terendah pada tahap pelatihan, yakni sebesar 0.4532. Oleh karena itu, model yang akan diadopsi dalam tahap pengujian kedua adalah model tipe AQ. Pengaturan model AQ ini menggunakan sejumlah 28 neuron pada *hidden layer* pertama dan 32 neuron pada *hidden layer* kedua. Dari hasil yang ditunjukkan pada Lampiran 13, tidak dapat

dijamin bahwa jumlah neuron atau jumlah *layer* pada *hidden layer* yang lebih sedikit atau lebih banyak dalam *hidden layer* secara konsisten akan menghasilkan nilai *loss* yang lebih rendah atau lebih tinggi. Kesimpulan tersebut menunjukkan kompleksitas hubungan antara arsitektur model dan kinerja yang dihasilkannya, yang memerlukan pendekatan eksperimental yang teliti dan tidak bersifat kaku atau mengikat. Gambar 4.9 menunjukkan model jaringan dari model AQ.



Gambar 4.9 Model Jaringan AQ dengan Bobot Akhir

Setelah ditentukan model terbaik, langkah selanjutnya adalah melakukan perhitungan dengan langkah yang sama dengan Model Pertama pada setiap *layer* nya. Dimana bobot awal *input layer* ke *hidden layer* 1 ditunjukkan pada Lampiran 14, *hidden layer* 1 ke *hidden layer* 2 pada Lampiran 15, dan *hidden layer* 2 ke *output layer* pada Lampiran 16. Bias awal pada setiap *layer* nya yaitu

0. Berikut adalah contoh hasil *output* dari *hidden layer* 1 sebelum dilakukan fungsi aktivasi ReLu:

$$Z_{net_{jk}} = \left(\sum_{i=1}^n X_i U_j \right) + U_{j0}$$

$$\begin{aligned} Z_{net_{11}} = & (0.6552 (0.12) + 0.5343 (-0.26) + 0.0441(0.13) + \\ & +0.9377(0.35) + 0.7975(0.37), +1 (0.04) + 0 (-0.01) + \\ & + 0.2258 (-0.29)) + 0 \end{aligned}$$

$$Z_{net_{11}} = 0.5419$$

$Z_{net_{11}}$ merupakan *output* dari neuron ke-1 pada *hidden layer* 1 sebelum menggunakan fungsi aktivasi yang akan ditransfer ke *hidden layer* 2. Nilai 0.6552, 0.5343, ..., 0.2258 merupakan variabel *input* X_1, X_2, \dots, X_8 . Nilai 0.12, -0.26, ..., -0.29 merupakan bobot awal antara *input layer* ke *hidden layer* 1 pada neuron ke-1 dan nilai 0 merupakan bias awal *input layer* ke *hidden layer* 1. Dari persamaan tersebut menyatakan bahwa jika X_2 hingga X_8 bernilai tetap dan X_1 dinaikkan satu satuan, maka $Z_{net_{11}}$ akan bertambah sebesar 0.12 satuan. Jika X_1 hingga X_4 dan X_6 hingga X_8 bernilai tetap dan X_5 dinaikkan satu satuan, maka $Z_{net_{11}}$ akan bertambah sebesar 0.35 satuan, begitupun untuk X yang lainnya. Tabel 4.4 berikut merupakan contoh hasil pada *hidden layer* 1 pada neuron ke 1 sebelum diaktivasi, untuk hasil lengkapnya dapat dilihat pada Lampiran 17.

Tabel 4.4 Hasil *Hidden Layer* 1 Sebelum Aktivasi

$Z_{net_{1_1}}$	0.5419	$Z_{net_{15_1}}$	0.2134
$Z_{net_{2_1}}$	-0.1590	$Z_{net_{16_1}}$	0.1744
$Z_{net_{3_1}}$	0.7809	$Z_{net_{17_1}}$	0.7294
$Z_{net_{4_1}}$	0.0538	$Z_{net_{18_1}}$	-0.4767
$Z_{net_{5_1}}$	-0.3211	$Z_{net_{19_1}}$	-0.3196
$Z_{net_{6_1}}$	-0.0149	$Z_{net_{20_1}}$	-0.5609
$Z_{net_{7_1}}$	0.5239	$Z_{net_{21_1}}$	0.1770
$Z_{net_{8_1}}$	0.0136	$Z_{net_{22_1}}$	0.0925
$Z_{net_{9_1}}$	-0.0218	$Z_{net_{23_1}}$	-0.1343
$Z_{net_{10_1}}$	-0.2961	$Z_{net_{24_1}}$	0.0046
$Z_{net_{11_1}}$	0.2385	$Z_{net_{25_1}}$	-0.1252
$Z_{net_{12_1}}$	0.2866	$Z_{net_{26_1}}$	0.0240
$Z_{net_{13_1}}$	-0.6949	$Z_{net_{27_1}}$	-0.1018
$Z_{net_{14_1}}$	-0.4482	$Z_{net_{28_1}}$	0.3718

Selanjutnya, setiap nilai pada Tabel 4.4 tersebut dilakukan perhitungan dengan menggunakan fungsi aktivasi ReLu untuk mendapatkan nilai akhir *output* pada *hidden layer* 1.

$$Z_{jk} = f(Z_{net_j}) = \max(0, Z_{net_j})$$

$$Z_{1_1} = f(Z_{net_{1_1}}) = \max(0, 0.5419) = 0.5419$$

Z_{1_1} adalah $Z_{net_{1,1_1}}$ yang telah diaktivasi menggunakan fungsi aktivasi ReLu. $Z_{net_{1_1}}$ bernilai 0.5419. Nilai *output* dari neuron 1 Z_{1_1} pada *hidden layer* 1 didapatkan dengan memasukkan nilai $Z_{net_{1_1}}$ kedalam fungsi aktivasi ReLu dengan mengambil nilai paling besar antara 0 dengan nilai $Z_{net_{1_1}}$ sehingga didapatkan nilai sebesar 0.5419. Nilai $Z_{net_{1_1}}$ akan menjadi nilai *output* pada *hidden layer* 1 neuron ke 1 yang akan ditransfer ke *hidden layer* 2. Tabel 4.5 berikut merupakan contoh hasil pada *hidden layer* 1 setelah diaktivasi, untuk hasil lengkapnya dapat dilihat pada Lampiran 18.

Tabel 4.5 Hasil *Hidden Layer 1* Setelah Aktivasi

Z_{1_1}	0.5419	Z_{15_1}	0.2134
Z_{2_1}	0.0000	Z_{16_1}	0.1744
Z_{3_1}	0.7809	Z_{17_1}	0.7294
Z_{4_1}	0.0538	Z_{18_1}	0.0000
Z_{5_1}	0.0000	Z_{19_1}	0.0000
Z_{6_1}	0.0000	Z_{20_1}	0.0000
Z_{7_1}	0.5239	Z_{21_1}	0.1770
Z_{8_1}	0.0136	Z_{22_1}	0.0925
Z_{9_1}	0.0000	Z_{23_1}	0.0000
Z_{10_1}	0.0000	Z_{24_1}	0.0046
Z_{11_1}	0.2385	Z_{25_1}	0.0000
Z_{12_1}	0.2866	Z_{26_1}	0.0240
$Z_{13,1_1}$	0.0000	$Z_{27,1_1}$	0.0000
$Z_{14,1_1}$	0.0000	$Z_{28,1_1}$	0.3718

Selanjutnya, setiap nilai pada Tabel 4.5 akan menjadi *input* pada *layer* berikutnya yaitu *hidden layer 2*. Dengan langkah yang sama seperti pada *layer* sebelumnya, maka diperoleh *output* dari *hidden layer 2* sebelum dilakukan fungsi aktivasi ReLu:

$$Z_{net_{jk}} = \left(\sum_{i=1}^n X_i V_j \right) + V_{j0}$$

$$\begin{aligned} Z_{net_{1_2}} = & (0.5419 (0.2055) + 0(-0.0667) + 0.7809(0.2299) + \\ & +0.0538(0.0116) + 0(-0.0449) + 0 (0.1730) + \\ & +0.5239(-0.1448) + \dots + 0.3718(-0.2652)) + 0 \end{aligned}$$

$$Z_{net_{1_2}} = 0.1264$$

$Z_{net_{1_2}}$ merupakan *output* dari neuron ke-1 pada *hidden layer 2* sebelum menggunakan fungsi aktivasi yang akan ditransfer ke *output*. Nilai 0.5419, 0, ..., 0.3718 merupakan variabel *input* X_1, X_2, \dots, X_{28} . Nilai 0.2055, -0.0667, ..., -0.2652 merupakan bobot awal antara *hidden layer 1* dan *hidden layer 2* pada

neuron ke-1 dan nilai 0 merupakan bias awal *hidden layer 1* dan *hidden layer 2* pada neuron ke-1. Dari persamaan tersebut menyatakan bahwa jika X_2 hingga X_{28} bernilai tetap dan X_1 dinaikkan satu satuan, maka $Z_{net_{12}}$ akan bertambah sebesar 0.2055 satuan, begitupun untuk X yang lainnya. Tabel 4.4 berikut merupakan contoh hasil pada *hidden layer 2* pada neuron ke 1 sebelum diaktivasi, untuk hasil lengkapnya dapat dilihat pada Lampiran 19.

Tabel 4.6 Hasil *Hidden Layer 2* Sebelum Aktivasi

$Z_{net_{12}}$	0.1264	$Z_{net_{172}}$	0.1692
$Z_{net_{22}}$	0.1118	$Z_{net_{182}}$	-0.3147
$Z_{net_{32}}$	0.2944	$Z_{net_{192}}$	-0.2849
$Z_{net_{42}}$	0.0648	$Z_{net_{202}}$	-0.5755
$Z_{net_{52}}$	-0.0835	$Z_{net_{212}}$	0.0166
$Z_{net_{62}}$	0.1385	$Z_{net_{222}}$	0.6169
$Z_{net_{72}}$	-0.0842	$Z_{net_{232}}$	-0.3647
$Z_{net_{82}}$	-0.0510	$Z_{net_{242}}$	-0.0574
$Z_{net_{92}}$	0.0610	$Z_{net_{252}}$	-0.4027
$Z_{net_{102}}$	-0.7110	$Z_{net_{262}}$	0.3038
$Z_{net_{112}}$	0.0066	$Z_{net_{272}}$	-0.3198
$Z_{net_{122}}$	-0.0458	$Z_{net_{282}}$	-0.2462
$Z_{net_{132}}$	-0.2218	$Z_{net_{292}}$	0.4047
$Z_{net_{142}}$	-0.1651	$Z_{net_{302}}$	0.0255
$Z_{net_{152}}$	-0.1509	$Z_{net_{312}}$	-0.2121
$Z_{net_{162}}$	-0.4978	$Z_{net_{322}}$	0.3348

Selanjutnya, setiap nilai pada Tabel 4.6 tersebut dilakukan perhitungan dengan menggunakan fungsi aktivasi ReLu untuk mendapatkan nilai akhir *output* pada *hidden layer 2*.

$$Z_{jk} = f(Z_{net_{jk}}) = \max(0, Z_{net_{jk}})$$

$$Z_{12} = f(Z_{net_{12}}) = \max(0, 0.1264) = 0.1264$$

Z_{1_2} adalah $Z_{net_{1_2}}$ yang telah diaktivasi menggunakan fungsi aktivasi ReLu. $Z_{net_{1_2}}$ bernilai 0.1264. Nilai *output* dari neuron 1 Z_{1_2} pada *hidden layer* 1 didapatkan dengan memasukkan nilai $Z_{net_{1_2}}$ kedalam fungsi aktivasi ReLu dengan mengambil nilai paling besar antara 0 dengan nilai $Z_{net_{1_2}}$ sehingga didapatkan nilai sebesar 0.1264. Nilai Z_{1_2} akan menjadi nilai *output* pada *hidden layer* 2 ke 1 yang akan ditransfer ke *output layer*. Tabel 4.7 berikut merupakan contoh hasil *hidden layer* 2 pada neuron ke 1 setelah diaktivasi untuk hasil lengkapnya dapat dilihat pada Lampiran 20.

Tabel 4.7 Hasil *Hidden Layer* 2 Setelah Aktivasi

Z_{1_2}	0.1264	Z_{17_2}	0.1692
Z_{2_2}	0.1118	Z_{18_2}	0.0000
Z_{3_2}	0.2944	Z_{19_2}	0.0000
Z_{4_2}	0.0648	Z_{20_2}	0.0000
Z_{5_2}	0.0000	Z_{21_2}	0.0166
Z_{6_2}	0.1385	Z_{22_2}	0.6169
Z_{7_2}	0.0000	Z_{23_2}	0.0000
Z_{8_2}	0.0000	Z_{24_2}	0.0000
Z_{9_2}	0.0610	Z_{25_2}	0.0000
Z_{10_2}	0.0000	Z_{26_2}	0.3038
Z_{11_2}	0.0066	Z_{27_2}	0.0000
Z_{12_2}	0.0000	Z_{28_2}	0.0000
Z_{13_2}	0.0000	Z_{29_2}	0.4047
Z_{14_2}	0.0000	Z_{30_2}	0.0255
Z_{15_2}	0.0000	Z_{31_2}	0.0000
Z_{16_2}	0.0000	Z_{32_2}	0.3348

Selanjutnya, setiap nilai pada Tabel 4.7 akan menjadi *input* pada *layer* berikutnya yaitu *output layer*. Berikut adalah perhitungan untuk *output layer*:

$$y_{net_k} = \left(\sum_{i=1}^p Z_{jk} W_j \right) + W_{j0}$$

$$y_{net_0} = \left(Z_{1_2} (0.1044) + Z_{2_2} (-0.0321) + \dots + Z_{32_2} (0.1652) \right) + 0$$

$$= 0.4021$$

$$y_{net_1} = (Z_{1_2}(-0.3590) + Z_{1_2}(-0.1074) + \dots + Z_{3_2_2}(0.1404)) + 0$$

$$= -0.0827$$

$$y_{net_2} = (Z_{1_2}(-0.2386) + Z_{1_2}(0.3383) + \dots + Z_{3_2_2}(-0.1883)) + 0$$

$$= -0.3305$$

$$y_{net_3} = (Z_{1_2}(-0.2443) + Z_{2_2}(-0.2055) + \dots + Z_{3_2_2}(0.2180)) + 0$$

$$= -0.2719$$

y_{net_k} adalah *output* pada *output layer* yang akan menjadi nilai klasifikasi curah hujan sebelum dilakukan aktivasi. Nilai -0.3590, -0.1074, ..., 0.1404 merupakan bobot awal antara *hidden layer 2* dan *output layer* pada y_{net_0} dan nilai 0 merupakan bias awal *hidden layer 2* dan *output layer*, dan begitu seterusnya untuk y_{net_1} hingga y_{net_3} . Setelah didapatkan *output* tersebut, kemudian menghitung sinyal *output* menggunakan fungsi aktivasi *softmax*,

$$y_k = \text{softmax}(y_{net_k}) = \frac{e^{y_{net_k}}}{\sum e^{y_{net_k}}}$$

dimana $k = (0, 1, 2, 3)$ jumlah kelas pada *output*.

$$y_0 = \frac{e^{(0.4021)}}{e^{(0.4021)} + e^{(-0.0827)} + e^{(-0.3305)} + e^{(-0.2719)}} \approx 0.3837$$

$$y_1 = \frac{e^{(-0.0827)}}{e^{(0.4021)} + e^{(-0.0827)} + e^{(-0.3305)} + e^{(-0.2719)}} \approx 0.2363$$

$$y_2 = \frac{e^{(-0.3305)}}{e^{(0.4021)} + e^{(-0.0827)} + e^{(-0.3305)} + e^{(-0.2719)}} \approx 0.1844$$

$$y_3 = \frac{e^{(-0.2719)}}{e^{(0.4021)} + e^{(-0.0827)} + e^{(-0.3305)} + e^{(-0.2719)}} \approx 0.1956$$

y_k merupakan nilai *output* yang telah diaktivasi menggunakan fungsi aktivasi *softmax*. Untuk hasil lengkap prediksi klasifikasi sesudah aktivasi dapat dilihat pada Lampiran 21. Hasil keluaran tersebut membentuk matriks $[0.3837 \quad 0.2363 \quad 0.1844 \quad 0.1956]$. Dari matriks hasil prediksi ini, langkah selanjutnya adalah melakukan *decoding* dengan menetapkan nilai tertinggi dari seluruh nilai dalam matriks sebagai representasi hasil klasifikasinya. Karena nilai tertinggi adalah 0.3387, maka matriks akan berubah menjadi $[1 \quad 0 \quad 0 \quad 0]$. Oleh karena itu, prediksi menunjukkan bahwa klasifikasi curah hujan adalah kelas 0 atau R (rendah). Selanjutnya, dilakukan perhitungan *loss*, *gradient loss*, dan *update* bobot. Hasil *update* bobot dan bias untuk *hidden layer* 1 dapat dilihat pada Lampiran 22, untuk *hidden layer* 2 dapat dilihat pada Lampiran 23, dan untuk *output layer* dapat dilihat pada Lampiran 24.

Penelitian yang dilakukan pada Model Kedua didapatkan arsitektur jaringan terbaik yaitu 8 neuron *input layer*, 28 neuron *hidden layer* 1, 32 neuron *hidden layer* 2, dan 4 *output layer*. Bobot yang didapatkan ditentukan secara otomatis melalui *software* Python. Berikut adalah interpretasi dari perhitungan *output hidden layer* 1, *output hidden layer* 2, dan *output layer* pada y :

$$Z_{jk(H1)} = f \left(\left(\sum_{i=1}^n X_i U_j \right) + U_{j0} \right)$$

$$Z_{1_1} = f(0.27x_1 - 0.30x_2 + \dots - 0.24x_8) + 0.06$$

Z_{jk} merupakan *output* pada *hidden layer* yang akan ditransfer ke *output layer* menggunakan fungsi aktivasi ReLu (f). Nilai 0.27, -0.30, ..., -0.24 adalah bobot akhir antara *input* dan *hidden layer* dimana jika bobot tersebut bernilai positif

artinya terdapat hubungan positif antara bobot dan Z_{1_1} yang dapat menaikkan nilai Z_{1_1} , begitupun sebaliknya jika bobot tersebut bernilai negatif artinya terdapat hubungan negatif antara bobot dan Z_{1_1} yang dapat menurunkan nilai Z_{1_1} . Sedangkan 0.06 merupakan bias antara *input* dan *hidden layer*. Berikut bentuk persamaan pada *output hidden layer 2*:

$$Z_{jk(H2)} = f \left(\left(\sum_{i=1}^n Z_{jk(H1)} V_j \right) + V_{j0} \right)$$

$$Z_{1_2} = f(0.5760Z_{1,1_1} + 0.2454Z_{2,1_1} + \dots - 0.3916Z_{28,1_1}) - 0.0245$$

Berikut adalah model secara matematis ANN:

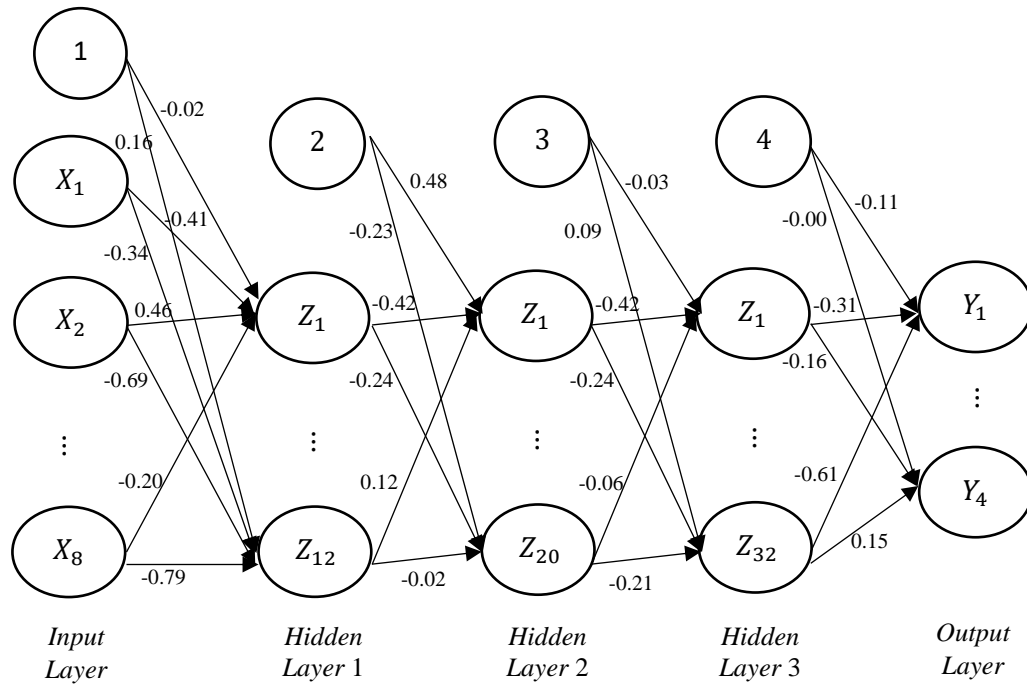
$$y_k = f \left(\left(\sum_{i=1}^p Z_{jk(H2)} W_j \right) + W_{j0} \right)$$

$$y_k = f(0.2160Z_{1,1_2} - 0.0521Z_{2,1_2} + \dots + 0.3370Z_{32,1_2}) - 0.0632$$

y_k merupakan keluaran dari masing masing kelas klasifikasi yang merupakan hasil prediksi dari curah hujan di 11 wilayah di Jawa Tengah setelah diaktivasi menggunakan fungsi aktivasi *softmax* (f). Nilai 0.2160, -0.0521, ..., 0.3370 merupakan bobot akhir antara *hidden layer 1* ke *output layer* pada kelas 1 dan $Z_{1_2}, Z_{2_2}, \dots, Z_{32_2}$ adalah nilai *output* dari *hidden layer 2*. Jika Z_{2_2}, \dots, Z_{32_2} nilainya tetap dan Z_{1_2} dinaikkan satu satuan, maka nilai dari prediksi curah hujan sebelum diaktivasi akan bertambah sebesar 0.2160, bernilai positif artinya terdapat hubungan positif antara bobot dengan nilai prediksi curah hujan sebelum diaktivasi, begitu seterusnya hingga Z_{32_2} .

4.2.3.3 Model Ketiga

Pada Model Ketiga, dilakukan eksplorasi melalui penambahan satu *layer* lagi pada bagian *hidden layer* dengan tujuan meningkatkan kompleksitas representasi yang dihasilkan. Serangkaian variasi model dievaluasi untuk memperoleh model optimal, seperti pada Lampiran 25. Hasil percobaan mengungkapkan satu model unggul yang berhasil mencapai konvergensi dengan nilai *loss* terendah pada tahap pelatihan, yakni sebesar 0.363. Oleh karena itu, model yang akan diadopsi dalam tahap pengujian kedua adalah model tipe DQ. Pengaturan model DQ ini menggunakan sejumlah 12 neuron pada *hidden layer* pertama, 20 neuron pada *hidden layer* kedua, dan 32 neuron pada *hidden layer* ketiga. Dari hasil yang ditunjukkan pada Lampiran 25, tidak dapat dijamin bahwa jumlah neuron atau jumlah *layer* pada *hidden layer* yang lebih sedikit atau lebih banyak dalam *hidden layer* secara konsisten akan menghasilkan nilai *loss* yang lebih rendah atau lebih tinggi. Kesimpulan tersebut menunjukkan kompleksitas hubungan antara arsitektur model dan kinerja yang dihasilkannya, yang memerlukan pendekatan eksperimental yang teliti dan tidak bersifat kaku atau mengikat. Gambar 4.10 menunjukkan model jaringan dari model DQ.



Gambar 4.10 Model Jaringan DQ dengan Bobot Akhir

Setelah ditentukan model terbaik, langkah selanjutnya adalah melakukan perhitungan dengan langkah yang sama dengan Model Pertama pada setiap *layer* nya. Dimana bobot awal *input layer* ke *hidden layer 1* ditunjukkan pada Lampiran 26, *hidden layer 1* ke *hidden layer 2* pada Lampiran 27, *hidden layer 2* ke *hidden layer 3* pada Lampiran 28, dan *hidden layer 3* ke *output layer* pada Lampiran 29. Bias awal pada setiap *layer* nya yaitu 0. Berikut adalah contoh hasil *output* dari *hidden layer 1* sebelum dilakukan fungsi aktivasi ReLu:

$$Z_{net_{jk}} = \left(\sum_{i=1}^n X_i T_j \right) + T_{j0}$$

$$\begin{aligned} Z_{net_{11}} = & (0.6552 (-0.37) + 0.5343 (0.48) + 0.0441 (-0.49) + 0.9377 \times \\ & \times (0.51) + 0.7975(0.32) + 1 (-0.33) + 0 (0.04) + 0.2258 \times \\ & \times (-0.19)) + 0 \end{aligned}$$

$$Z_{net_{11}} = 0.3492$$

$Z_{net_{11}}$ merupakan *output* dari neuron ke-1 pada *hidden layer* 1 sebelum menggunakan fungsi aktivasi yang akan ditransfer ke *hidden layer* 2. Nilai 0.6552, 0.5343, ..., 0.2258 merupakan variabel *input* X_1, X_2, \dots, X_8 . Nilai (-0.37), 0.48, ..., (-0.19) merupakan bobot awal antara *input layer* dan *hidden layer* 1 dan nilai 0 merupakan bias awal *input layer* dan *hidden layer* 1. Dari persamaan tersebut menyatakan bahwa jika X_2 hingga X_8 bernilai tetap dan X_1 dinaikkan satu satuan, maka $Z_{net_{1,11}}$ akan berkurang sebesar 0.37 satuan. Jika X_1 hingga X_4 dan X_6 hingga X_8 bernilai tetap dan X_5 dinaikkan satu satuan, maka $Z_{net_{11}}$ akan bertambah sebesar 0.32 satuan, begitupun untuk X yang lainnya. Tabel 4.8 berikut merupakan contoh hasil *hidden layer* 1 pada neuron ke 1 sebelum diaktivasi, untuk hasil lengkapnya dapat dilihat pada Lampiran 30.

Tabel 4.8 Hasil *Hidden Layer* 1 Sebelum Aktivasi

$Z_{net_{11}}$	0.3492	$Z_{net_{71}}$	0.2809
$Z_{net_{21}}$	0.5528	$Z_{net_{81}}$	-1.1991
$Z_{net_{31}}$	0.8156	$Z_{net_{91}}$	0.6939
$Z_{net_{41}}$	-0.0928	$Z_{net_{101}}$	-0.1248
$Z_{net_{51}}$	-0.2173	$Z_{net_{111}}$	-1.0530
$Z_{net_{61}}$	-0.4408	$Z_{net_{121}}$	0.1715

Selanjutnya, setiap nilai pada Tabel 4.8 tersebut dilakukan perhitungan dengan menggunakan fungsi aktivasi ReLu untuk mendapatkan nilai akhir *output* pada *hidden layer* 1.

$$Z_{jk} = f(Z_{net_{jk}}) = \max(0, Z_{net_{jk}})$$

$$Z_{11} = f(Z_{net_{11}}) = \max(0, 0.3492) = 0.3492$$

Z_{1_1} adalah $Z_{net_{1_1}}$ yang telah diaktivasi menggunakan fungsi aktivasi ReLu. $Z_{net_{1_1}}$ bernilai 0.3492. Nilai *output* dari neuron 1 (Z_{1_1}) pada *hidden layer* 1 didapatkan dengan memasukkan nilai $Z_{net_{1_1}}$ kedalam fungsi aktivasi ReLu dengan mengambil nilai paling besar antara 0 dengan nilai $Z_{net_{1_1}}$ sehingga didapatkan nilai sebesar 0.3492. Nilai $Z_{net_{1_1}}$ akan menjadi nilai *output* pada *hidden layer* 1 ke 1 yang akan ditransfer ke *hidden layer* 2. Tabel 4.9 berikut merupakan contoh hasil *hidden layer* 1 pada neuron ke 1 setelah diaktivasi, untuk hasil lengkapnya dapat dilihat pada Lampiran 31.

Tabel 4.9 Hasil *Hidden Layer* 1 Setelah Aktivasi

Z_{1_1}	0.3492	Z_{7_1}	0.2809
Z_{2_1}	0.5527	Z_{8_1}	0.0000
Z_{3_1}	0.8156	Z_{9_1}	0.6939
Z_{4_1}	0.0000	Z_{10_1}	0.0000
Z_{5_1}	0.0000	Z_{11_1}	0.0000
Z_{6_1}	0.0000	Z_{12_1}	0.1715

Selanjutnya, setiap nilai pada Tabel 4.9 akan menjadi *input* pada *layer* berikutnya yaitu *hidden layer* 2. Dengan langkah yang sama seperti pada *layer* sebelumnya, maka diperoleh *output* dari *hidden layer* 2 sebelum dilakukan fungsi aktivasi ReLu:

$$Z_{net_{jk}} = \left(\sum_{i=1}^n X_i U_j \right) + U_{j0}$$

$$\begin{aligned} Z_{net_{1_2}} = & (0.3492(-0.4264) + 0.5527(0.4026) + 0.8156(-0.1359) + \\ & + 0(-0.1123) + 0(-0.2371), + 0(-0.2014) + \\ & + 0.2809(-0.1756) + 0(0.3112) + 0.6939(-0.4186) + \\ & + 0(0.3095) + 0(0.1987) + 0.1715(-0.4033)) + 0 \end{aligned}$$

$$Z_{net_{12}} = -0.4462$$

$Z_{net_{12}}$ merupakan *output* dari neuron ke-1 pada *hidden layer 2* sebelum menggunakan fungsi aktivasi yang akan ditransfer ke *output*. Nilai 0.3492, 0.5527, ..., 0.1715 merupakan variabel *input* $Z_{1_1}, Z_{2_1}, \dots, Z_{12_1}$. Nilai -0.4264, 0.4026, ..., -0.4033 merupakan bobot awal antara *hidden layer 1* dan *hidden layer 2* dan nilai 0 merupakan bias awal *hidden layer 1* dan *hidden layer 2*. Dari persamaan tersebut menyatakan bahwa jika Z_{2_1} hingga Z_{12_1} bernilai tetap dan Z_{1_1} dinaikkan satu satuan, maka $Z_{net_{12}}$ akan bertambah sebesar -0.4264 satuan, begitupun untuk X yang lainnya. Tabel 4.4 berikut contoh hasil *hidden layer 2* pada neuron ke 1 sebelum diaktivasi, untuk hasil lengkapnya dapat dilihat pada Lampiran 32.

Tabel 4.10 Hasil *Hidden Layer 2* Sebelum Aktivasi

$Z_{net_{12}}$	-0.4462	$Z_{net_{112}}$	0.5139
$Z_{net_{22}}$	-0.1806	$Z_{net_{122}}$	-0.2080
$Z_{net_{32}}$	0.4683	$Z_{net_{132}}$	-0.1843
$Z_{net_{42}}$	-0.1566	$Z_{net_{142}}$	0.0493
$Z_{net_{52}}$	-0.4923	$Z_{net_{152}}$	0.6912
$Z_{net_{62}}$	-0.2313	$Z_{net_{162}}$	-0.2893
$Z_{net_{72}}$	-0.6504	$Z_{net_{172}}$	-0.1671
$Z_{net_{82}}$	0.3189	$Z_{net_{182}}$	-0.6179
$Z_{net_{92}}$	0.1977	$Z_{net_{192}}$	0.1872
$Z_{net_{102}}$	0.0530	$Z_{net_{202}}$	0.1098

Selanjutnya, setiap nilai pada Tabel 4.10 tersebut dilakukan perhitungan dengan menggunakan fungsi aktivasi ReLu untuk mendapatkan nilai akhir *output* pada *hidden layer 2*.

$$Z_{jk} = f(Z_{net_{jk}}) = \max(0, Z_{net_{jk}})$$

$$Z_{1_2} = f(Z_{net_{1_2}}) = \max(0, -0.4462) = 0$$

Z_{1_2} adalah $Z_{net_{1_2}}$ yang telah diaktivasi menggunakan fungsi aktivasi ReLu. Nilai *output* dari neuron 1 (Z_{1_2}) pada *hidden layer* 2 didapatkan dengan memasukkan nilai $Z_{net_{1_2}}$ kedalam fungsi aktivasi ReLu dengan mengambil nilai paling besar antara 0 dengan nilai $Z_{net_{1_2}}$ sehingga didapatkan nilai sebesar 0. Nilai Z_{1_2} menjadi nilai *output* pada *hidden layer* 2 ke 1 yang ditransfer ke *output layer*. Tabel 4.7 berikut contoh hasil *hidden layer* 2 pada neuron ke 1 setelah diaktivasi, untuk hasil lengkapnya dapat dilihat pada Lampiran 33.

Tabel 4.11 Hasil *Hidden Layer* 2 Setelah Aktivasi

Z_{1_2}	0.0000	Z_{11_2}	0.5139
Z_{2_2}	0.0000	Z_{12_2}	0.0000
Z_{3_2}	0.4683	Z_{13_2}	0.0000
Z_{4_2}	0.0000	Z_{14_2}	0.0493
Z_{5_2}	0.0000	Z_{15_2}	0.6912
Z_{6_2}	0.0000	Z_{16_2}	0.0000
Z_{7_2}	0.0000	Z_{17_2}	0.0000
Z_{8_2}	0.3189	Z_{18_2}	0.0000
Z_{9_2}	0.1977	Z_{19_2}	0.1872
Z_{10_2}	0.0530	Z_{20_2}	0.1098

Selanjutnya, setiap nilai pada Tabel 4.11 menjadi *input* pada *hidden layer* 3. Dengan langkah yang sama seperti pada *layer* sebelumnya, maka diperoleh *output* dari *hidden layer* 3 sebelum dilakukan fungsi aktivasi ReLu:

$$Z_{net_{jk}} = \left(\sum_{i=1}^n X_i V_j \right) + V_{j0}$$

$$Z_{net_{1_3}} = (0(-0.2882) + 0(0.2594) + 0.4683(-0.1265) + 0(0.2852) + \\ + 0(-0.0323) + \dots + 0.1098(-0.0387)) + 0$$

$$Z_{net_{1_3}} = 0.0401$$

$Z_{net_{13}}$ merupakan *output* dari neuron ke-1 pada *hidden layer 3* sebelum menggunakan fungsi aktivasi yang akan ditransfer ke *output*. Nilai 0, 0, ..., 0.1098 merupakan variabel *input* $Z_{1_2}, Z_{2_2}, \dots, Z_{12_2}$. Nilai -0.2882, 0.2594, ..., -0.0387 merupakan bobot awal antara *hidden layer 2* dan *hidden layer 3* pada neuron ke-1 dan nilai 0 merupakan bias awal *hidden layer 2* dan *hidden layer 3* pada neuron ke-1. Dari persamaan tersebut menyatakan bahwa jika Z_{2_2} hingga Z_{12_2} bernilai tetap dan Z_{1_2} dinaikkan satu satuan, maka $Z_{net_{13}}$ akan bertambah sebesar -0.2882 satuan, begitupun untuk X yang lainnya. Tabel 4.4 berikut contoh hasil *hidden layer 3* pada neuron ke 1 sebelum diaktivasi, untuk hasil lengkapnya dapat dilihat pada Lampiran 34.

Tabel 4.12 Hasil *Hidden Layer 3* Sebelum Aktivasi

$Z_{net_{13}}$	0.0401	$Z_{net_{173}}$	0.2387
$Z_{net_{23}}$	0.1903	$Z_{net_{183}}$	-0.1522
$Z_{net_{33}}$	0.0867	$Z_{net_{193}}$	0.2906
$Z_{net_{43}}$	-0.1172	$Z_{net_{203}}$	-0.2248
$Z_{net_{53}}$	-0.0950	$Z_{net_{213}}$	-0.1462
$Z_{net_{63}}$	-0.3458	$Z_{net_{223}}$	0.4946
$Z_{net_{73}}$	0.0964	$Z_{net_{233}}$	-0.1903
$Z_{net_{83}}$	-0.1677	$Z_{net_{243}}$	0.1104
$Z_{net_{93}}$	-0.0401	$Z_{net_{253}}$	-0.1693
$Z_{net_{103}}$	-0.0330	$Z_{net_{263}}$	0.2036
$Z_{net_{113}}$	-0.2741	$Z_{net_{273}}$	-0.1376
$Z_{net_{123}}$	-0.3792	$Z_{net_{283}}$	0.0189
$Z_{net_{133}}$	0.1630	$Z_{net_{293}}$	0.1200
$Z_{net_{143}}$	0.0372	$Z_{net_{303}}$	-0.3104
$Z_{net_{153}}$	-0.2252	$Z_{net_{313}}$	0.3007
$Z_{net_{163}}$	-0.2551	$Z_{net_{323}}$	0.2822

Selanjutnya, setiap nilai pada Tabel 4.12 tersebut dilakukan perhitungan dengan menggunakan fungsi aktivasi ReLu untuk mendapatkan nilai akhir *output* pada *hidden layer* 3.

$$Z_{jk} = f(Z_{net_{jk}}) = \max(0, Z_{net_{jk}})$$

$$Z_{1_3} = f(Z_{net_{1_3}}) = \max(0, 0.0401) = 0.0401$$

Z_{1_3} adalah $Z_{net_{1_3}}$ yang telah diaktivasi menggunakan fungsi aktivasi ReLu. Nilai *output* dari neuron 1 (Z_{1_3}) pada *hidden layer* 3 didapatkan dengan memasukkan nilai $Z_{net_{1_3}}$ kedalam fungsi aktivasi ReLu dengan mengambil nilai paling besar antara 0 dengan nilai $Z_{net_{1_3}}$ sehingga didapatkan nilai sebesar 0.0401. Nilai Z_{1_3} menjadi nilai *output* pada *hidden layer* 3 ke 1 yang ditransfer ke *output layer*. Tabel 4.13 berikut contoh hasil *hidden layer* 3 pada neuron ke 1 setelah diaktivasi, untuk hasil lengkapnya dapat dilihat pada Lampiran 35.

Tabel 4.13 Hasil *Hidden Layer* 3 Setelah Aktivasi

Z_{1_3}	0.0401	Z_{17_3}	0.2387
Z_{2_3}	0.1903	Z_{18_3}	0.0000
Z_{3_3}	0.0867	Z_{19_3}	0.2906
Z_{4_3}	0.0000	Z_{20_3}	0.0000
Z_{5_3}	0.0000	Z_{21_3}	0.0000
Z_{6_3}	0.0000	Z_{22_3}	0.4946
Z_{7_3}	0.0964	Z_{23_3}	0.0000
Z_{8_3}	0.0000	Z_{24_3}	0.1104
Z_{9_3}	0.0000	Z_{25_3}	0.0000
Z_{10_3}	0.0000	Z_{26_3}	0.2036
Z_{11_3}	0.0000	Z_{27_3}	0.0000
Z_{12_3}	0.0000	Z_{28_3}	0.0189
Z_{13_3}	0.1630	Z_{29_3}	0.1200
Z_{14_3}	0.0372	Z_{30_3}	0.0000
Z_{15_3}	0.0000	Z_{31_3}	0.3007
Z_{16_3}	0.0000	Z_{32_3}	0.2822

Selanjutnya, setiap nilai pada Tabel 4.13 akan menjadi *input* pada *layer* berikutnya yaitu *output layer*. Berikut adalah perhitungan untuk *output layer*:

$$y_{net_k} = \left(\sum_{i=1}^p Z_i W_j \right) + W_{j0}$$

$$\begin{aligned} y_{net_0} &= \left(Z_{1_3}(-0.3405) + Z_{2_3}(0.3616) + \dots + Z_{32_3}(-0.2182) \right) + 0 \\ &= -0.1898 \end{aligned}$$

$$\begin{aligned} y_{net_1} &= \left(Z_{1_3}(0.2168) + Z_{2_3}(-0.3902) + \dots + Z_{32_3}(-0.0904) \right) + 0 \\ &= -0.1909 \end{aligned}$$

$$\begin{aligned} y_{net_2} &= \left(Z_{1_3}(-0.1866) + Z_{2_3}(0.0793) + \dots + Z_{32_3}(-0.1898) \right) + 0 \\ &= -0.2303 \end{aligned}$$

$$\begin{aligned} y_{net_3} &= \left(Z_{1_3}(-0.1866) + Z_{2_3}(0.3434) + \dots + Z_{32_3}(0.0474) \right) + 0 \\ &= -0.0234 \end{aligned}$$

y_{net_k} adalah *output* pada *output layer* yang akan menjadi nilai klasifikasi curah hujan sebelum dilakukan aktivasi. Nilai -0.3405, 0.3616, ..., -0.2182 merupakan bobot awal antara *hidden layer* 3 dan *output layer* pada y_{net_0} dan nilai 0 merupakan bias awal *hidden layer* 3 dan *output layer*, dan begitu seterusnya untuk y_{net_1} hingga y_{net_3} . Setelah didapatkan *output* tersebut, kemudian menghitung sinyal *output* menggunakan fungsi aktivasi *softmax*,

$$y_k = softmax(y_{net_k}) = \frac{e^{y_{net_k}}}{\sum e^{y_{net_k}}}$$

dimana $k = (0, 1, 2, 3)$ jumlah kelas pada *output*.

$$y_0 = \frac{e^{(-0.1898)}}{e^{(-0.1898)} + e^{(-0.1909)} + e^{(-0.2303)} + e^{(-0.0234)}} \approx 0.2415$$

$$y_1 = \frac{e^{(-0.1909)}}{e^{(-0.1898)} + e^{(-0.1909)} + e^{(-0.2303)} + e^{(-0.0234)}} \approx 0.2413$$

$$y_2 = \frac{e^{(-0.2303)}}{e^{(-0.1898)} + e^{(-0.1909)} + e^{(-0.2303)} + e^{(-0.0234)}} \approx 0.2319$$

$$y_3 = \frac{e^{(-0.0234)}}{e^{(-0.1898)} + e^{(-0.1909)} + e^{(-0.2303)} + e^{(-0.0234)}} \approx 0.2853$$

y_k merupakan *output* yang telah diaktivasi menggunakan fungsi aktivasi *softmax*.

Untuk hasil lengkap prediksi klasifikasi sesudah aktivasi dapat dilihat pada

Lampiran 36. Hasil keluaran tersebut membentuk matriks

[0.2415 0.2413 0.2319 0.2853]. Dari matriks hasil prediksi ini, langkah

selanjutnya adalah melakukan *decoding* dengan menetapkan nilai tertinggi dari seluruh

nilai dalam matriks sebagai representasi hasil klasifikasinya. Karena nilai tertinggi adalah

0.2853, maka matriks akan berubah menjadi [0 0 0 1]. Oleh karena itu, prediksi

menunjukkan bahwa klasifikasi curah hujan adalah kelas 3 atau SL (sangat lebat).

Selanjutnya, dilakukan perhitungan *loss*, *gradient loss*, dan *update* bobot. Hasil

update bobot dan bias untuk *hidden layer* 1 dapat dilihat pada Lampiran 37, untuk

hidden layer 2 dapat dilihat pada Lampiran 38, *hidden layer* 3 dapat dilihat pada

Lampiran 39, dan untuk *output layer* dapat dilihat pada Lampiran 40.

Penelitian yang dilakukan pada Model Ketiga didapatkan arsitektur jaringan

terbaik yaitu 8 neuron *input layer*, 12 neuron *hidden layer* 1, 20 neuron *hidden*

layer 2, 32 neuron *hidden layer* 3, dan 4 *output layer*. Bobot yang didapatkan

ditentukan secara otomatis melalui *software* Python. Berikut adalah interpretasi

dari perhitungan *output hidden layer* 1, *output hidden layer* 2, *output hidden layer*

3, dan *output layer*:

$$Z_{jk(H1)} = f \left(\left(\sum_{i=1}^n X_i T_j \right) + T_{j0} \right)$$

$$Z_{1_1} = f(-0.41x_1 + 0.46x_2 + \dots - 0.20x_8) - 0.02$$

Z_{1_1} merupakan *output* ke pada *hidden layer* yang akan ditransfer ke *output layer* menggunakan fungsi aktivasi ReLu (f). Nilai -0.41, 0.46, ..., -0.20 adalah bobot antara *input* dan *hidden layer* 1 dimana jika bobot tersebut bernilai positif artinya terdapat hubungan positif antara bobot dan Z_{1_1} yang dapat menaikkan nilai Z_{1_1} , begitupun sebaliknya jika bobot tersebut bernilai negatif artinya terdapat hubungan negatif antara bobot dan Z_{1_1} yang dapat menurunkan nilai Z_{1_1} . Sedangkan -0.02 merupakan bias antara *input* dan *hidden layer* 1. Berikut bentuk persamaan pada *output hidden layer* 2:

$$Z_{jk(H2)} = f \left(\left(\sum_{i=1}^n Z_{jk(H1)} U_j \right) + U_{j0} \right)$$

$$Z_{1_2} = f(-0.4173Z_{1_1} + 0.5423Z_{2_1} + \dots + 0.1243Z_{12_1}) + 0.4756$$

Z_{1_2} merupakan *output* ke pada *hidden layer* yang akan ditransfer ke *output layer* menggunakan fungsi aktivasi ReLu (f). Nilai -0.4173, 0.5423, ..., 0.1243 adalah bobot antara *hidden layer* 1 dan *hidden layer* 2 dimana jika bobot tersebut bernilai positif artinya terdapat hubungan positif antara bobot dan Z_{1_2} yang dapat menaikkan nilai Z_{1_2} , begitupun sebaliknya jika bobot tersebut bernilai negatif artinya terdapat hubungan negatif antara bobot dan Z_{1_2} yang dapat menurunkan

nilai Z_{1_2} . Sedangkan 0.4756 merupakan bias antara *hidden layer* 1 dan *hidden layer* 2. Berikut bentuk persamaan pada *output hidden layer* 3:

$$Z_{j_k(H3)} = f \left(\left(\sum_{i=1}^n Z_{j_k(H2)} V_j \right) + V_{j0} \right)$$

$$Z_{1_3} = f(-0.3115Z_{1_2} + 0.2258Z_{2_2} + \dots - 0.0596Z_{20_2}) - 0.0293$$

Z_{1_3} merupakan *output* ke pada *hidden layer* yang akan ditransfer ke *output layer* menggunakan fungsi aktivasi ReLu (f). Nilai -0.3115, 0.2258, ..., 0.0596 adalah bobot antara *hidden layer* 2 dan *hidden layer* 3 dimana jika bobot tersebut bernilai positif artinya terdapat hubungan positif antara bobot dan Z_{1_3} yang dapat menaikkan nilai Z_{1_3} , begitupun sebaliknya jika bobot tersebut bernilai negatif artinya terdapat hubungan negatif antara bobot dan Z_{1_3} yang dapat menurunkan nilai Z_{1_3} . Sedangkan -0.0293 merupakan bias antara *hidden layer* 2 dan *hidden layer* 3

Berikut adalah model secara matematis ANN:

$$y_k = f \left(\left(\sum_{i=1}^p Z_{j_k(H3)} W_j \right) + W_{j0} \right)$$

$$y_k = f(-0.3092Z_{1_3} + 0.2727Z_{2_3} + \dots - 0.6112Z_{32_3}) - 0.1055$$

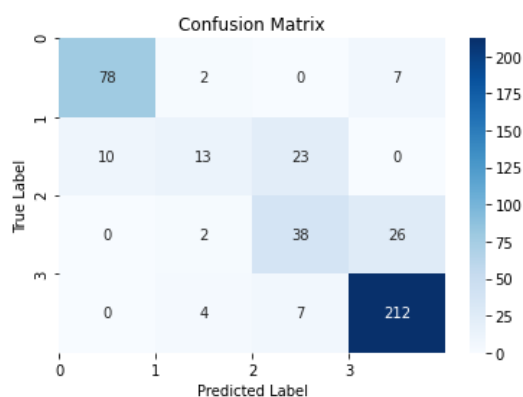
y_k merupakan keluaran dari masing masing kelas klasifikasi yang merupakan hasil prediksi dari curah hujan di 11 wilayah di Jawa Tengah setelah diaktivasi menggunakan fungsi aktivasi *softmax* (f). Nilai -0.3092, 0.2727, ..., -0.6112 merupakan bobot antara *hidden layer* 3 ke *output layer* pada kelas 1 dan Z_{1_3} ,

Z_{2_3}, \dots, Z_{23_3} adalah nilai *output* dari *hidden layer* 3. Jika Z_{2_3}, \dots, Z_{23_3} nilainya tetap dan Z_{1_3} dinaikkan satu satuan, maka nilai dari prediksi curah hujan setelah diaktivasi akan berkurang sebesar 0.3092, bernilai positif artinya terdapat hubungan positif antara bobot dengan nilai prediksi curah hujan sebelum diaktivasi, begitu seterusnya hingga Z_{23_3} .

4.2.4 Evaluasi Model ANN

Untuk menilai sejauh mana kinerja sebuah model dalam menganalisis pola dalam data dan membuat prediksi yang akurat, diperlukan evaluasi model yang sesuai. Beberapa metode evaluasi yang digunakan yaitu akurasi, presisi, *recall*, dan *F1-Score*. Melalui evaluasi ini, pemahaman lebih mendalam dapat diperoleh tentang sejauh mana model tersebut efektif dalam menjalankan tugas yang diberikan dan seberapa baiknya dalam mengenali pola-pola dalam data.

4.2.4.1 Model Pertama



Gambar 4.11 *Confusion Matrix* Model E Menggunakan Data Training

Dari Gambar 4.11 kemudian dilakukan perhitungan pada metrik evaluasi dengan rumus akurasi, presisi, *recall*, dan *F1-Score* pada persamaan (2.20) hingga (2.26). Sehingga didapatkan nilai sebagai berikut:

$$Accuracy = \frac{341 + 1185}{341 + 1185 + 81 + 81} = 0.90$$

$$Precision_{weighted} = \frac{0.89(87) + 0.62(46) + 0.56(66) + 0.87(223)}{422}$$

$$= 0.79$$

$$Recall_{weighted} = \frac{0.90(87) + 0.28(46) + 0.58(66) + 0.95(223)}{422}$$

$$= 0.81$$

$$F1 - Score_{weighted} = \frac{0.89(87) + 0.39(46) + 0.57(66) + 0.79(223)}{422}$$

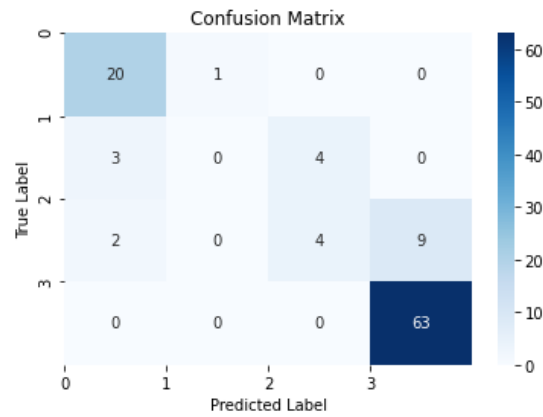
$$= 0.79$$

Setelah didapatkan nilai dari setiap evaluasi, kemudian dilakukan perulangan untuk melihat konsistensi dari model. Tabel 4.14 berikut merupakan hasil perulangan Model E.

Tabel 4.14 Hasil Perulangan Model E Menggunakan Data *Training*

Perulangan ke-	Akurasi	Presisi	Recall	F1-Score
1	90%	79%	81%	79%
2	91%	81%	82%	79%
3	90%	79%	81%	79%
4	91%	81%	82%	79%
5	90%	78%	80%	78%
Rata-rata	90%	79%	81%	79%

Dari hasil perulangan Model E pada data *training* dapat dilihat bahwa antara perulangan 1, 2, 3, dan 4 terjadi model yang siklik atau berulang. Karena perulangan ke-1 sama dengan perulangan ke-3 dan perulangan ke 2 sama dengan perulangan ke-4, maka bisa dipilih perulangan ke-2 yang bisa dilakukan untuk Model E.



Gambar 4.12 Confusion Matrix Model E Menggunakan Data Testing

$$Accuracy = \frac{87 + 299}{87 + 299 + 19 + 19} = 0.91$$

$$Precision_{weighted} = \frac{0.80(21) + 0.00(7) + 0.50(15) + 0.88(63)}{106}$$

$$= 0.75$$

$$Recall_{weighted} = \frac{0.95(21) + 0.00(7) + 0.27(15) + 1(63)}{106}$$

$$= 0.82$$

$$F1 - Score_{weighted} = \frac{0.87(21) + 0.00(7) + 0.35(15) + 0.93(63)}{106}$$

$$= 0.78$$

$$MAE = \frac{1}{106} (|3 - 3| + |3 - 3| + \dots + |2 - 0|)$$

$$= 0.198$$

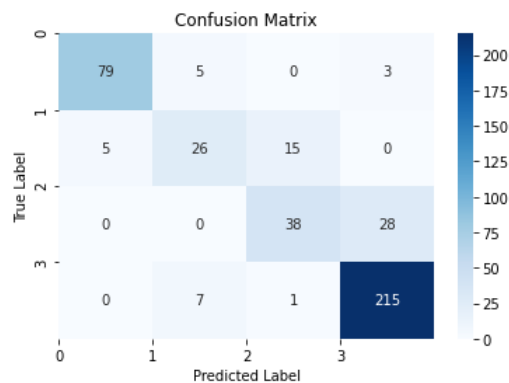
Setelah didapatkan nilai dari setiap evaluasi, maka akan dilakukan perulangan untuk melihat konsistensi dari model. Tabel 4.15 berikut merupakan hasil perulangan Model E.

Tabel 4.15 Hasil Perulangan Model E Menggunakan Data *Testing*

Perulangan ke-	Accuracy	Precision	Recall	F1-Score	MAE
1	91%	75%	82%	78%	0.198
2	91%	78%	81%	78%	0.198
3	91%	75%	82%	78%	0.189
4	91%	78%	81%	78%	0.226
5	90%	75%	80%	76%	0.235
Rata-rata	91%	76%	81%	78%	0.209

Dari Tabel 4.15, menunjukkan hasil perulangan Model E pada data *testing* yang dapat dilihat bahwa nilai konvergen pada perulangan perulangan 1, 2, 3, dan 4. Karena perulangan 1 sama dengan perulangan 3 dan perulangan 2 sama dengan perulangan 4, maka dapat digunakan perulangan ke-2 untuk Model E dimana dari perulangan tersebut didapatkan nilai MAE yaitu 0.198.

4.2.4.2 Model Kedua

**Gambar 4.13** *Confusion Matrix* Model AQ Menggunakan Data *Training*

Dari Gambar 4.13 akan dilakukan perhitungan pada metrik evaluasi dengan rumus akurasi, presisi, *recall*, dan F1-Score pada persamaan (2.20) hingga (2.26). Sehingga didapatkan nilai sebagai berikut:

$$Accuracy = \frac{358 + 1202}{358 + 1202 + 64 + 64} = 0.92$$

$$Precision_{weighted} = \frac{0.94(87) + 0.68(46) + 0.70(66) + 0.87(223)}{422}$$

$$= 0.84$$

$$Recall_{weighted} = \frac{0.91(87) + 0.57(46) + 0.58(66) + 0.96(223)}{422}$$

$$= 0.85$$

$$F1 - Score_{weighted} = \frac{0.92(87) + 0.62(46) + 0.63(66) + 0.92(223)}{422}$$

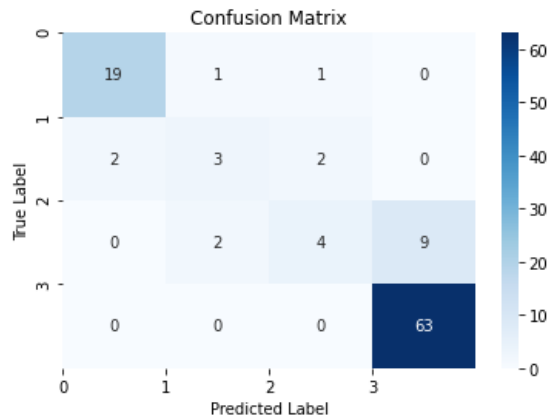
$$= 0.84$$

Setelah didapatkan nilai dari setiap evaluasi, kemudian dilakukan perulangan untuk melihat konsistensi dari model. Tabel 4.16 berikut merupakan hasil perulangan Model AQ.

Tabel 4.16 Hasil Perulangan Model AQ Menggunakan Data *Training*

Perulangan ke-	Accuracy	Precision	Recall	F1-Score
1	92%	84%	85%	84%
2	91%	81%	82%	81%
3	92%	83%	84%	83%
4	92%	84%	85%	84%
5	92%	78%	80%	79%
Rata-rata	92%	82%	83%	82%

Dari hasil perulangan Model AQ pada data *training* dapat dilihat bahwa antara perulangan 1, 2, 3, dan 4 terjadi model yang siklik atau berulang dengan nilai akurasi yang konvergen pada 92%. Perulangan tersebut memiliki nilai yang berulang setiap 4 perulangan. Karena perulangan ke-1 sama dengan perulangan ke-4, maka bisa dipilih perulangan ke-1 yang bisa dilakukan untuk Model AQ.



Gambar 4.14 Confusion Matrix Model AQ Menggunakan Data Testing

$$Accuracy = \frac{87 + 301}{87 + 301 + 17 + 17} = 0.92$$

$$Precision_{weighted} = \frac{0.90(21) + 0.50(7) + 0.57(15) + 0.88(63)}{106}$$

$$= 0.81$$

$$Recall_{weighted} = \frac{0.90(21) + 0.43(7) + 0.27(15) + 1(63)}{106}$$

$$= 0.84$$

$$F1 - Score_{weighted} = \frac{0.90(21) + 0.46(7) + 0.36(15) + 0.93(63)}{106}$$

$$= 0.82$$

$$MAE = \frac{1}{106} (|3 - 3| + |3 - 3| + \dots + |2 - 0|)$$

$$= 0.189$$

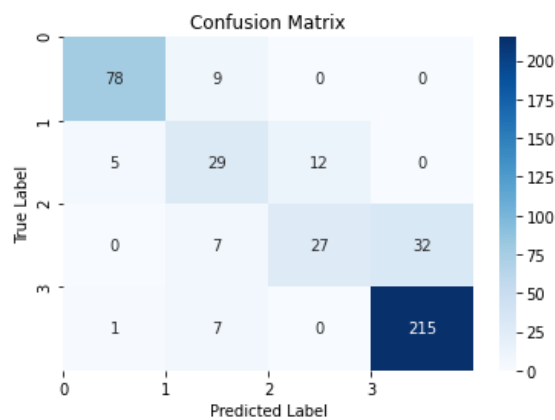
Setelah didapatkan nilai dari setiap evaluasi, kemudian dilakukan perulangan untuk melihat konsistensi dari model. Tabel 4.17 berikut merupakan hasil perulangan Model AQ.

Tabel 4.17 Hasil Perulangan Model AQ Menggunakan Data *Testing*

Perulangan ke-	Accuracy	Precision	Recall	F1-Score	MAE
1	92%	81%	84%	82%	0.189
2	91%	80%	82%	80%	0.198
3	91%	80%	82%	80%	0.198
4	92%	81%	84%	82%	0.208
5	93%	84%	85%	83%	0.189
Rata-rata	92%	81%	83%	81%	0.197

Dari Tabel 4.17, menunjukkan hasil perulangan Model AQ pada data *testing* yang dapat dilihat bahwa nilai konvergen pada perulangan perulangan 2 dan 3. Karena kedua perulangan tersebut memiliki nilai yang sama, maka dapat digunakan perulangan ke-2 dimana dari perulangan tersebut didapatkan nilai MAE yaitu 0.198.

4.2.4.3 Model Ketiga

**Gambar 4.15** *Confusion Matrix* Model DQ Menggunakan Data *Training*

Dari Gambar 4.15 akan dilakukan perhitungan pada metrik evaluasi dengan rumus akurasi, presisi, *recall*, dan *F1-Score* pada persamaan (2.20) hingga (2.26). Sehingga didapatkan nilai sebagai berikut:

$$Accuracy = \frac{349 + 1193}{349 + 1193 + 73 + 73} = 0.91$$

$$Precision_{weighted} = \frac{0.93(87) + 0.56(46) + 0.69(66) + 0.87(223)}{422}$$

$$= 0.82$$

$$Recall_{weighted} = \frac{0.90(87) + 0.63(46) + 0.41(66) + 0.96(223)}{422}$$

$$= 0.83$$

$$F1 - Score_{weighted} = \frac{0.91(87) + 0.59(46) + 0.51(66) + 0.91(223)}{422}$$

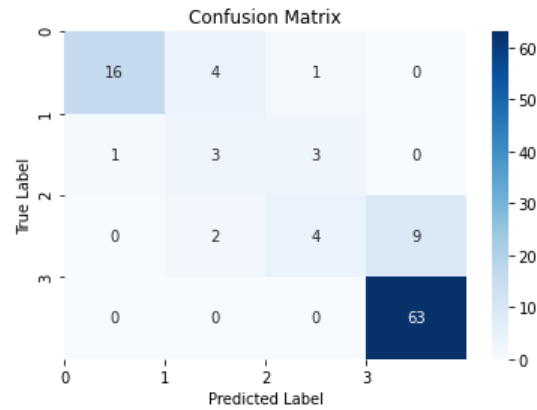
$$= 0.82$$

Setelah didapatkan nilai dari setiap evaluasi, maka akan dilakukan perulangan untuk melihat konsistensi dari model. Tabel 4.18 berikut merupakan hasil perulangan Model DQ.

Tabel 4.18 Hasil Perulangan Model DQ Menggunakan Data *Training*

Perulangan ke-	Accuracy	Precision	Recall	F1-Score
1	91%	82%	83%	82%
2	92%	83%	84%	83%
3	91%	82%	83%	82%
4	92%	83%	84%	83%
5	91%	78%	80%	78%
Rata-rata	91%	82%	83%	82%

Dari hasil perulangan Model DQ pada data *training* dapat dilihat bahwa antara perulangan 1, 2, 3, dan 4 terjadi model yang siklik atau berulang. Karena perulangan ke-1 sama dengan perulangan ke-3 dan perulangan ke 2 sama dengan perulangan ke-4, maka bisa dipilih perulangan ke-2 yang bisa dilakukan untuk Model DQ.



Gambar 4.16 *Confusion Matrix* Model DQ Menggunakan Data *Testing*

$$Accuracy = \frac{87 + 301}{87 + 301 + 17 + 17} = 0.92$$

$$Precision_{weighted} = \frac{0.94(21) + 0.33(7) + 0.50(15) + 0.88(63)}{106}$$

$$= 0.80$$

$$Recall_{weighted} = \frac{0.76(21) + 0.43(7) + 0.27(15) + 1(63)}{106}$$

$$= 0.81$$

$$F1 - Score_{weighted} = \frac{0.84(21) + 0.38(7) + 0.35(15) + 0.93(63)}{106}$$

$$= 0.80$$

$$MAE = \frac{1}{106} (|3 - 3| + |3 - 3| + \dots + |2 - 0|)$$

$$= 0.189$$

Setelah didapatkan nilai dari setiap evaluasi, maka akan dilakukan perulangan untuk melihat konsistensi dari model. Tabel 4.19 berikut merupakan hasil perulangan Model DQ.

Tabel 4.19 Hasil Perulangan Model DQ Menggunakan Data *Testing*

Perulangan ke-	Accuracy	Precision	Recall	F1-Score	MAE
1	91%	80%	81%	80%	0.189
2	91%	78%	80%	78%	0.208
3	91%	78%	81%	78%	0.189
4	91%	78%	83%	79%	0.198
5	91%	80%	82%	80%	0.226
Rata-rata	91%	79%	81%	79%	0.202

Dari Tabel 4.19, menunjukkan hasil perulangan Model DQ pada data *testing* yang dapat dilihat bahwa nilai akurasi dan presisi konvergen secara berturut-turut dengan nilai 91% dan 78% pada perulangan 2, 3, dan 4. Dari perulangan ketiga perulangan tersebut tersebut dapat ditetapkan perulangan ke-4 yang digunakan oleh Model DQ dimana didapatkan nilai MAE yaitu 0.198.

4.2.5 Prediksi Hasil Klasifikasi Curah Hujan di Jawa Tengah

4.2.5.1 Model Pertama

Setelah berhasil menemukan model terbaik pada Model Pertama, yaitu Model E dengan arsitektur jaringan 8-20-4 yang menggunakan satu *layer* pada *hidden layer*, langkah selanjutnya adalah melakukan prediksi menggunakan model tersebut. Berdasarkan data yang ada pada tahun 2019-2022 dari 35 kabupaten/kota di Jawa Tengah hanya 11 kabupaten/kota yang didapatkan data yang lengkap berdasarkan dari sumber Badan Pusat Statistik (BPS) (<https://jateng.bps.go.id/>). Berikut merupakan hasil prediksi curah hujan di Jawa Tengah dengan menggunakan Model E.

Tabel 4.20 Hasil Prediksi Curah Hujan di Jawa Tengah Menggunakan Model E

Bulan	Wilayah	Prediksi	Wilayah	Prediksi	Wilayah	Prediksi	Wilayah	Prediksi
Jan	Cilacap	SL	Wono-giri	L	Tegal	S	Sema-rang Kota	S
Feb		SL		SL		S		S
Mar		R		R		R		S
Apr		R		R		R		S
Mei		R		R		R		S
Jun		S		SL		S		S
Jul		S		S		R		R
Agt		SL		SL		R		R
Sept		R		S		R		R
Okt		R		R		R		R
Nov		SL		S		S		S
Des		R		SL		SL		SL
Jan	Banjar-negara	S	Sema-rang	S	Brebes	SL	Tegal Kota	R
Feb		S		L		SL		R
Mar		SL		R		R		S
Apr		S		SL		R		SL
Mei		R		L		L		L
Jun		R		S		SL		SL
Jul		R		SL		R		R
Agt		R		R		SL		R
Sept		L		R		S		S
Okt		SL		R		SL		S
Nov		R		SL		R		L
Des		SL		L		R		R
Jan	Boyolali	S	Pema-lang	S	Surakarta Kota	S		
Feb		L		S		SL		
Mar		R		R		SL		
Apr		R		SL		S		
Mei		SL		L		L		
Jun		S		SL		L		
Jul		SL		R		SL		
Agt		L		R		SL		
Sept		L		R		L		
Okt		L		SL		R		
Nov		R		R		R		
Des		SL		R		SL		

Berdasarkan Tabel 4.17 menunjukkan bahwa terdapat daerah-daerah yang secara periodik dalam satu tahun dominan mengalami curah hujan ringan yaitu di

wilayah Cilacap, Banjarnegara, Wonogiri, Semarang, Pemalang, Tegal, Brebes, Semarang Kota, dan Tegal Kota. Untuk wilayah yang dominan mengalami curah hujan lebat yaitu di wilayah Boyolali. Sedangkan untuk wilayah yang dominan mengalami curah hujan sangat lebat yaitu di wilayah Wonogiri dan Surakarta Kota.

4.2.5.2 Model Kedua

Setelah berhasil menemukan model terbaik pada Model Kedua, yaitu Model AQ dengan arsitektur jaringan 8-28-32-4 yang menggunakan dua *layer* pada *hidden layer*, langkah selanjutnya adalah melakukan prediksi menggunakan model tersebut. Berdasarkan data yang ada pada tahun 2019-2022 dari 35 kabupaten/kota di Jawa Tengah hanya 11 kabupaten/kota yang didapatkan data yang lengkap berdasarkan dari sumber Badan Pusat Statistik (BPS) (<https://jateng.bps.go.id/>). Berikut merupakan hasil prediksi curah hujan di Jawa Tengah dengan menggunakan Model AQ.

Tabel 4.21 Hasil Prediksi Curah Hujan di Jawa Tengah Menggunakan Model AQ

Bulan	Wilayah	Prediksi	Wilayah	Prediksi	Wilayah	Prediksi	Wilayah	Prediksi
Jan	Cilacap	SL	Wono- giri	L	Tegal	SL	Sema- rang Kota	SL
Feb		SL		SL		SL		
Mar		R		R		R		
Apr		R		R		R		
Mei		R		R		SL		
Jun		L		SL		S		
Jul		SL		S		SL		
Agt		S		SL		R		
Sept		L		L		R		
Okt		R		SL		SL		
Nov		SL		SL		S		
Des		R		SL		SL		
Jan	Banjar- negara	S	Sema- rang	SL	Brebes	SL	Tegal Kota	SL
Feb		S		L		SL		
Mar		SL		SL		R		
Apr		S		L		R		
Mei		SL		R		L		

Jun		R		S		SL		S
Jul		R		L		R		R
Agt		SL		SL		L		SL
Sept		L		SL		S		SL
Okt		SL		R		SL		R
Nov		L		SL		R		R
Des		SL		L		R		R
Jan	Boyolali	S	Pema- lang	SL	Surakarta Kota	S		
Feb		L		L		L		
Mar		L		SL		S		
Apr		R		R		S		
Mei		S		L		R		
Jun		R		R		R		
Jul		R		R		S		
Agt		S		S		SL		
Sept		S		S		L		
Okt		R		SL		R		
Nov		SL		SL		R		
Des		SL		R		SL		

Berdasarkan Tabel 4.18 menunjukkan bahwa terdapat daerah-daerah yang secara periodik dalam satu tahun dominan mengalami curah hujan ringan yaitu di wilayah Cilacap, Boyolali, Pemalang, Brebes, Surakarta Kota, dan Tegal Kota. Untuk wilayah yang dominan mengalami curah hujan sedang yaitu di wilayah Boyolali dan Surakarta Kota. Sedangkan untuk wilayah yang dominan mengalami curah hujan sangat lebat yaitu di wilayah Banjarnegara, Wonogiri, Semarang, Pemalang, Tegal, dan Semarang Kota.

4.2.5.3 Model Ketiga

Setelah berhasil menemukan model terbaik pada Model Ketiga, yaitu Model DQ dengan arsitektur jaringan 8-12-20-32-4 yang menggunakan tiga *layer* pada *hidden layer*, langkah selanjutnya adalah melakukan prediksi menggunakan model tersebut. Berdasarkan data yang ada pada tahun 2019-2022 dari 35 kabupaten/kota di Jawa Tengah hanya 11 kabupaten/kota yang didapatkan data yang lengkap

berdasarkan dari sumber Badan Pusat Statistik (BPS) (<https://jateng.bps.go.id/>).

Berikut hasil prediksi curah hujan di Jawa Tengah menggunakan Model DQ.

Tabel 4.22 Hasil Prediksi Curah Hujan di Jawa Tengah Menggunakan Model DQ

Bulan	Wilayah	Prediksi	Wilayah	Prediksi	Wilayah	Prediksi	Wilayah	Prediksi
Jan	Cilacap	SL	Wono-giri	L	Tegal	SL	Sema-rang Kota	S
Feb		SL		SL		SL		SL
Mar		R		R		R		R
Apr		R		R		R		R
Mei		R		R		R		SL
Jun		SL		SL		SL		SL
Jul		SL		S		SL		SL
Agt		SL		SL		R		SL
Sept		SL		L		R		R
Okt		R		SL		SL		SL
Nov		SL		SL		S		SL
Des		R		SL		SL		SL
Jan	Banjar-negara	SL	Sema-rang	SL	Brebes	SL	Tegal Kota	R
Feb		SL		L		SL		R
Mar		SL		SL		R		SL
Apr		SL		SL		R		SL
Mei		SL		SL		L		SL
Jun		R		SL		SL		SL
Jul		R		SL		R		SL
Agt		SL		SL		SL		SL
Sept		L		SL		SL		SL
Okt		SL		R		SL		SL
Nov		SL		SL		R		L
Des		SL		L		R		R
Jan	Boyolali	SL	Pema-lang	SL	Surakarta Kota	SL		
Feb		SL		SL		SL		
Mar		SL		SL		SL		
Apr		R		SL		SL		
Mei		SL		L		SL		
Jun		SL		SL		SL		
Jul		SL		R		SL		
Agt		L		SL		SL		
Sept		SL		SL		L		
Okt		SL		SL		R		
Nov		SL		SL		R		
Des		SL		R		SL		

Berdasarkan Tabel 4.19 menunjukkan bahwa kesebelas wilayah kabupaten/kota tersebut yang secara periodik dalam satu tahun dominan mengalami curah hujan sangat lebat.

4.3 Integrasi Nilai-Nilai Agama dengan Hasil Penelitian

ANN adalah salah satu representasi menggunakan program komputer yang mampu menyelesaikan sejumlah perhitungan pada otak manusia selama proses pembelajaran. Seperti otak manusia, ANN terdiri dari beberapa neuron dan penghubung antar neuron tersebut. Seperti model dalam penelitian ini yang menghasilkan arsitektur yang terdiri dari beberapa jumlah neuron pada setiap *layer* (penghubung antar neuron) untuk memproses penklasifikasian curah hujan. Namun meskipun ANN bekerja seperti cara kerja manusia bukan berarti akal manusia dapat digantikan oleh otak buatan. Firman Allah dalam surat Al-Baqarah ayat 269 yang artinya (Kemenag, 2023c):

“Dia (Allah) menganugerahkan hikmah kepada siapa yang Dia kehendaki. Siapa yang dianugerahi hikmah, sungguh dia telah dianugerahi kebaikan banyak. Tidak ada yang dapat mengambil pelajaran (darinya), kecuali ululalbab.”

Maksud dari ululalbab dari arti surat Al-Baqarah ayat 269 merupakan orang yang berakal. Pemberian hikmah dan akal merupakan karunia dari Allah, dan penggunaan akal dalam pengembangan teknologi dapat dianggap sebagai bentuk pengambilan pelajaran dari karunia tersebut.

Hasil dari penelitian menggunakan metode ANN yaitu klasifikasi curah hujan pada beberapa wilayah di Jawa Tengah. Dari hasil tersebut diharapkan dapat dilakukan antisipasi terjadinya bencana yang disebabkan tinggi rendahnya curah hujan. Menurut Al-Quran turunnya air hujan secara bertahap dan teratur menurut

kadarnya dan dapat terjadi kekeringan juga atas Kuasa-Nya. Sesuai dengan firman Allah dalam Surat Al-Mu'minun ayat 18 yang artinya (Kemenag, 2023a):

“Dan Kami turunkan air dari langit menurut kadar yang ditentukan, kemudian Kami pancarkan air itu dalam bumi, dan sesungguhnya Kami benar-benar kuasa mengambilnya kembali (menjadi hilang).”

Hujan bisa menjadi berkah, namun apabila turun secara terus menerus atau tidak turun dalam jangka waktu yang lama akan menjadi bencana. Hujan yang turun dengan volume yang terlalu banyak akan menyebabkan banjir hingga tanah longsor. Sedangkan jika tidak turun hujan dalam jangka waktu yang panjang akan mengakibatkan kekeringan.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan rumusan masalah dan pembahasan pada bab sebelumnya, maka diperoleh kesimpulan sebagai berikut:

1. Model jaringan ANN yang dapat dan baik digunakan dalam melakukan klasifikasi curah hujan di Jawa Tengah merupakan Model E, Model AQ, dan Model DQ. Model E memiliki arsitektur jaringan 8-20-4, yaitu 8 neuron *input layer*, 20 neuron *hidden layer*, dan 4 neuron pada *output layer*. Model AQ memiliki arsitektur jaringan 8-28-32-4, yaitu 8 neuron *input layer*, 28 neuron *hidden layer 1*, 32 neuron *hidden layer 2*, dan 4 neuron pada *output layer*. Model DQ memiliki arsitektur jaringan 8-12-20-32-4, yaitu 8 neuron *input layer*, 12 neuron *hidden layer 1*, 20 neuron *hidden layer 2*, 32 neuron *hidden layer 3*, dan 4 neuron pada *output layer*. Bentuk Model ANN secara matematis dapat dilihat pada Lampiran 41.
2. Tingkat akurasi dari Model E didapatkan pada perulangan ke-2 sebesar 91%, presisi 78%, *recall* 81%, dan *F1-score* 78%. Model AQ didapatkan pada perulangan ke-2 nilai akurasi sebesar 91%, presisi 80%, *recall* 82%, dan *F1-score* 80%. Model DQ didapatkan pada perulangan ke-4 nilai akurasi sebesar 91%, presisi 78%, *recall* 83%, dan *F1-score* 79%. MAE pada ketiga model tersebut menunjukkan nilai yang sama yaitu 0.198. Hal tersebut menunjukkan bahwa ketiga model tersebut memiliki nilai kesalahan yang sama dalam melakukan klasifikasi curah hujan yaitu sebesar 0.198.

3. Hasil prediksi curah hujan pada 11 wilayah kabupaten/kota di Jawa Tengah didapatkan bahwa daerah yang dominan mengalami curah hujan ringan yaitu di wilayah Cilacap, Boyolali, Pemalang, Brebes, Surakarta Kota, dan Tegal Kota. Daerah yang dominan mengalami curah hujan sedang yaitu di wilayah Boyolali dan Surakarta Kota. Sedangkan untuk wilayah yang dominan mengalami curah hujan sangat lebat yaitu di wilayah Banjarnegara, Wonogiri, Semarang, Pemalang, Tegal, dan Semarang Kota. Namun tidak ada wilayah yang dominan mengalami curah hujan lebat.

5.2 Saran

Saran yang dapat penulis sampaikan untuk penelitian selanjutnya adalah menambahkan faktor lain atau menggunakan faktor lain selain 8 faktor yang telah digunakan dalam penelitian ini.

DAFTAR PUSTAKA

- Abiodun, D. A. (2019). Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition. *International Journal of Computer Applications*, 181(38), 1–9.
- Aini, N. (2020). *9 Faktor yang Mempengaruhi Curah Hujan di Indonesia Beserta Penjelasannya*. Haloedukasi.Com.
- Al-Sheikh, A. bin M. bin, & Ishaq, A. Bin. (2003). Tafsir Ibnu Katsir Jilid 5. *Pustaka Imam Syafi'i*, 1, 458–615.
- Back, B., Sere, K., & Vanharanta, H. (1998). Managing complexity in large data bases using self-organizing maps. *Accounting, Management and Information Technologies*, 8(4), 191–210. [https://doi.org/10.1016/S0959-8022\(98\)00009-5](https://doi.org/10.1016/S0959-8022(98)00009-5)
- Badwi, N., Maru, R., & Abbas, I. (2017). Analisis Spasial Tingkat Kekeringan Sebagai Upaya Mitigasi Kegagalan Panen di Kabupaten Takalar Sulawesi Selatan. *Jurnal Scientific Phinisi*, 2(3), 116–122.
- Bilgili, M. Y. (2010). Prediction of film thickness for two-phase flow in horizontal and slightly inclined pipes using artificial neural networks. *Nuclear Engineering and Design*, 240(12), 3653–3663.
- Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 1, Issue 4). Springer.
- BMKG. (2010). Kondisi Cuaca Ekstrem dan Iklim Tahun 2010-2011. *Jakarta: BMKG*.
- Brownlee, J. (2017). How to one hot encode sequence data in python. *Machine Learning Mastery*, 12.
- Enders, W. (2004). Applied Econometric Time Series. “2th ed.” In *New York (US): University of Alabama*.
- Erdil, A., & Arcaklioglu, E. (2013). The prediction of meteorological variables using artificial neural network. *Neural Computing and Applications*, 22, 1677–1683.
- Ghosh, S. K., & Ghoshdastidar, P. S. (2017). Artificial Neural Network Based Rainfall Prediction in Purulia District of West Bengal, India. *International Journal of Scientific and Research Publications*, 7(5), 420–426.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
- Hadihardaja, I. K., & Sutikno, S. (2010). Pemodelan Curah Hujan-Limpasan Menggunakan Artificial Neural Network (ANN) dengan Metode Backpropagation. *Jurnal Teknik Sipil*, 12(4), 250. <https://doi.org/10.5614/jts.2005.12.4.3>
- Hadisuwito, A. S., & Hassan, F. H. (2019). Using backpropagation to predict drought factor in Keetch-Byram Drought Index. *Baghdad Science Journal*, 16(2 Special Issue), 477–484.
- Handayanto, R. T., & Herlawati. (2020). *DATA MINING DAN MACHINE LEARNING MENGGUNAKAN MATLAB DAN PYTHON*.
- Hanke, J. E., & Wichern, D. W. (2005). *Business forecasting*. Pearson Educación.
- Hassoun, M. H. (1995). *Fundamentals of artificial neural networks*. MIT Press. https://www.google.co.id/books/edition/Fundamentals_of_Artificial_Neural_Networ/Otk32Y3QkxQC?hl=id&gbpv=1&dq=artificial+neural+network&pg=PA1&printsec=frontcover

- Haykin, S. (2010). *Neural networks: a comprehensive foundation*. 1999. *Mc Millan, New Jersey*, 1–24.
- Haykin, S. O. (2009). *Neural Networks and Learning Machines 3rd Ed*. NY: *NYL Pearson Prentice Hall* (p. 24).
- Hidrologi, T. (2018). Analisis Curah Hujan Ekstrem di Indonesia Berdasarkan Teori Konveksi. *Jurnal Pengelolaan Sumberdaya Alam Dan Lingkungan*, 8(2), 190–200. <https://doi.org/https://doi.org/10.29244/jpsl.8.2.190-200>
- Hudnurkar, S., & Rayavarapu, N. (2022). Binary classification of rainfall time-series using machine learning algorithms. *International Journal of Electrical and Computer Engineering*, 12(2), 1945–1954.
- Jumarwanto, A., Hartanto, R., & Prastiyanto, D. (2009). Aplikasi jaringan saraf tiruan backpropagation untuk memprediksi penyakit THT di Rumah Sakit Mardi Rahayu Kudus. *Jurnal Teknik Elektro*, 1(1), 11.
- Kemenag. (2023a). *kemenag al mu'minin ayat 18*. <https://kemenag.go.id/>
- Kemenag. (2023b). *kemenag an nur ayat 43*. <https://kemenag.go.id/>
- Kemenag. (2023c). *kemenag al baqarah ayat 269*. <https://kemenag.go.id/>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
- Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., & Hinton, G. (2020). Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6), 335–346.
- Logar, J., Turk, G., Marsden, P., & Ambrožič, T. (2017). Prediction of rainfall induced landslide movements by artificial neural networks. *Natural Hazards and Earth System Sciences Discussions*, 1–18.
- Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). *Introduction to time series analysis and forecasting*. John Wiley & Sons.
- Montgomery, D. C., Johnson, L. A., & Gardiner, J. S. (2012). Forecasting and Time Series Analysis (7th Ed.). In *McGraw-Hill Education*.
- Mustafidah, H., & Suwarsito, S. (2016). Testing Design of Neural Network Parameters in Optimization Training Algorithm. *International Conference of Result and Community Services, 6th August 2016*, 139–146.
- Nabila, R. C., Arnita, A., Fitria, A., & Suryani, N. (2023). APPLICATION OF THE BACKPROPAGATION METHOD TO PREDICT RAINFALL IN NORTH SUMATRA PROVINCE. *BAREKENG: Jurnal Ilmu Matematika Dan Terapan*, 17(1), 0449–0456. <https://doi.org/10.30598/barekengvol17iss1pp0449-0456>
- Nguyen, H. N., Nguyen, T.-A., Ly, H.-B., Tran, V. Q., Nguyen, L. K., Nguyen, M. V., & Ngo, C. T. (2021). Prediction of daily and monthly rainfall using a backpropagation neural network. *Journal of Applied Science and Engineering*, 24(3), 367–379.
- Nilamsari, N. (2014). Memahami studi dokumen dalam penelitian kualitatif. *WACANA: Jurnal Ilmiah Ilmu Komunikasi*, 13(2), 177–181.
- Paras, S. M., Kumar, A., & Chandra, M. (2009). A feature based neural network model for weather forecasting. *International Journal of Computational Intelligence*, 4(3), 209–216.
- Pranowo, W. S. (2019). Faktor-Faktor yang Mempengaruhi Curah Hujan. *Jurnal Kebijakan Dan Pengembangan Pendidikan*, 7(1), 61–72. <https://doi.org/https://doi.org/10.29210/128002>

- Puspitasari, D. (2006). Implementasi Jaringan Syaraf Tiruan Backpropagation untuk Pengenalan Huruf dengan Matlab 7.0. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, 1(1), 41–47.
- Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning Third edition*, vol. 53. Packt Publishing Ltd.
- Remya, S. S., & Sunil Kumar, B. (2019). Rainfall Prediction Using Artificial Neural Networks: A Comprehensive Review. *Journal of Water Resource and Hydraulic Engineering*, 8(6), 139–153.
- Roza, S., Aidha, Z. R., Yuliza, M., & Yondri, S. (2018). Artificial Neural Network Application for Aroma Monitoring on The Coffe Beans Blending Process. *JOIV: International Journal on Informatics Visualization*, 2(3), 147–152.
- Sagioglu, S., Kaya, Y., & Kaynar, O. (2003). An expert system based on artificial neural networks and genetic algorithms to predict the blast-induced ground vibrations. *Soil Dynamics and Earthquake Engineering*, 23(4), 279–289.
- Santoso, M. A. F. (2014). Air Dan Pemeliharaannya Dalam Perspektif Islam. *Tarjih: Jurnal Tarjih Dan Pengembangan Pemikiran Islam*, 12(1), 97–114. <https://www.jurnal.tarjih.or.id/index.php/tarjih/article/view/84>
- Shihab, M. Q. (2002). Tafsir Al-Misbāh: pesan, kesan dan keserasian al-Qur'an. In *Lentera Hati*. Lentera Hati.
- Shofura, S., Suryani, S., Salma, L., & Harini, S. (2021). The Effect of Number of Factors and Data on Monthly Weather Classification Performance Using Artificial Neural Networks. *International Journal on Information and Communication Technology (IJoICT)*, 7(2), 23–35.
- Simpson, G. C. (1920). Some aspects of the rainfall of the earth. *Quarterly Journal of the Royal Meteorological Society*, 46(193), 77–93. <https://doi.org/https://doi.org/10.1002/qj.49704619305>
- Sunarsa, S. (2018). Isyarat Sains Tentang Air Dalam Al-Qur'an. *Jurnal NARATAS*, 1(2), 9–18. www.journal.stai-musaddadiyah.ac.id
- Syukri, S., & Samsuddin, S. (2019). Pengujian Algoritma Artificial Neural Network (ANN) Untuk Prediksi Kecepatan Angin. *Jurnal Nasional Komputasi Dan Teknologi Informasi (JNKTI)*, 2(1), 43–47.
- Vamsidhar, A., & Veeravalli, B. (2010). Artificial Neural Networks: An Overview. *International Journal of Computer Science and Information Security*, 8(1), 81–22.
- Wanto, A., Zarlis, M., & Hartama, D. (2017). Analysis of Artificial Neural Network Backpropagation Using Conjugate Gradient Fletcher Reeves in the Predicting Process. *Journal of Physics: Conference Series*, 930(1), 012018.
- Wibowo, S. W., Ernawati, T., & Jumadi, O. (2019). Pengembangan Model Jaringan Saraf Tiruan Untuk Klasifikasi Curah Hujan di Wilayah Pulau Jawa, Bali dan Lombok. *Jurnal Sains Dan Teknologi Lingkungan*, 11(1), 50–57.
- Zhang, G., Xiao, Q., Zhang, Y., & Yao, Y. (2020). The Impact of Wind Speed on Rainfall in the Eastern Himalayas Based on High-Resolution Observational Data. *Atmosphere*, 11(4), 375.
- Zhang, X., & LeCun, Y. (2015). Text understanding from scratch. *ArXiv Preprint ArXiv:1502.01710*.
- Zhang, X., Xu, J., Yang, J., Chen, L., Zhou, H., Liu, X., Li, H., Lin, T., & Ying, Y. (2020). Understanding the learning mechanism of convolutional neural networks in spectral analysis. *Analytica Chimica Acta*, 1119, 41–51.

LAMPIRAN

Lampiran 1. Data Bulanan Curah Hujan dan Faktor-Faktor yang Mempengaruhi di Kabupaten/Kota Jawa Tengah.

Kab/ Kota	Bu- lan	X1	X2	X3	X4	X5	X6	X7	X8	Y
2019										
Cila- cap	Jan	27.9	83	3	1008.6	44.4	7010	2124.47	24	SL
	Feb	27.7	81	2	1009.1	48.6	7010	2124.47	24	SL
	Mar	27.3	85	2	1009.5	69.9	7010	2124.47	25	SL
	Apr	28.3	82	3	1010.7	80.4	7010	2124.47	10	SL
	Mei	27.9	81	5	1010.1	83.1	7010	2124.47	13	SL
	Jun	25.5	84	4	1011.2	73	7010	2124.47	3	S
	Jul	25.1	84	5	1011.5	89	7010	2124.47	2	S
	Agt	24.7	84	5	1012.4	77	7010	2124.47	3	S
	Sept	25.2	83	4	1012	77.2	7010	2124.47	1	S
	Okt	25.2	83	4	1011.5	60.8	7010	2124.47	4	S
	Nov	27.7	82	3	1009.4	44.3	7010	2124.47	11	SL
	Des	28.6	80	2	1010.6	54	7010	2124.47	21	SL
Banjar- negara	Jan	24.7	82.3	11	944	46.7	77160	1335.3	24	L
	Feb	24.6	85.2	11	945.3	41.5	77160	1335.3	22	L
	Mar	24.3	85.7	10.4	944.3	30.1	77160	1335.3	23	L
	Apr	24.8	85.1	9.9	943.9	48.3	77160	1335.3	21	S
	Mei	24.5	83.8	8.8	945	51.6	77160	1335.3	18	S
	Jun	22.9	81.5	8.7	945.2	46.2	77160	1335.3	3	S
	Jul	22.3	81.7	9.1	945.9	50.2	77160	1335.3	4	S
	Agt	22.4	79.5	9.9	946.5	51.2	77160	1335.3	2	S
	Sept	23.4	76.3	12.2	947.4	55.9	77160	1335.3	2	S
	Okt	24.9	74.7	13.2	945.5	54.5	77160	1335.3	2	S
	Nov	27.3	80.2	11.5	945.2	56.1	77160	1335.3	12	S
	Des	24.9	83.2	11.2	944.2	49.6	77160	1335.3	25	S
Boyo- lali	Jan	29	82	9	1010	68	77160	1335.3	21	SL
	Feb	29	82	9	1010	78	77160	1335.3	20	SL
	Mar	29	83	9	1010	78	77160	1335.3	18	SL
	Apr	30	84	10	1010	78	77160	1335.3	15	SL
	Mei	30	85	10	1010	80	77160	1335.3	7	SL
	Jun	29	84	10	1011	88	77160	1335.3	21	SL
	Jul	30	82	10	1011	93	77160	1335.3	20	SL
	Agt	29	81	10	1011	94	77160	1335.3	18	SL
	Sept	30	82	9	1011	92	77160	1335.3	15	SL
	Okt	31	83	9	1011	90	77160	1335.3	0	S

	Nov	31	83	10	1012	88	77160	1335.3	18	SL
	Des	31	83	11	1012	80	77160	1335.3	19	SL
Wono- giri	Jan	26.5	89	3	1008.6	44.4	77160	1335.3	19	SL
	Feb	26.66	87.6	2	1009.1	48.6	77160	1335.3	15	SL
	Mar	27.65	87.5	2	1009.5	69.9	77160	1335.3	18	SL
	Apr	26.55	88.2	3	1010.7	80.4	77160	1335.3	10	S
	Mei	27.46	86.4	5	1010.1	83.1	77160	1335.3	2	S
	Jun	27.29	86.5	4	1011.2	73	77160	1335.3	0	S
	Jul	27.52	87	5	1011.5	89	77160	1335.3	0	S
	Agt	26.74	86.4	5	1012.4	77	77160	1335.3	0	S
	Sept	27.48	87.3	4	1012	77	77160	1335.3	0	S
	Okt	27.34	86.8	4	1011.5	60.8	77160	1335.3	0	S
	Nov	27.51	88.7	3	1009.4	44.3	77160	1335.3	3	S
	Des	27.24	88	2	1010.6	54	77160	1335.3	10	SL
Sema- rang	Jan	28.76	84.2	2.1	1008.8	65.7	77160	1335.3	14	SL
	Feb	28.21	85.9	2.1	1009.6	61.7	77160	1335.3	21	SL
	Mar	29.06	83.7	1.8	1009	72.7	77160	1335.3	18	SL
	Apr	29.65	80.9	1.6	1009.2	74.8	77160	1335.3	15	SL
	Mei	30.17	78.5	2	1008.7	86.3	77160	1335.3	11	SL
	Jun	30.08	76.8	2	1009	100	77160	1335.3	3	L
	Jul	29.85	74.8	1.9	1008.5	100	77160	1335.3	6	S
	Agt	30.37	73.4	2.4	1009	100	77160	1335.3	4	S
	Sept	30.46	71.8	2.1	1008.8	100	77160	1335.3	8	S
	Okt	29.85	75.9	1.9	1008.3	85.5	77160	1335.3	14	SL
	Nov	29.5	79.2	1.8	1008.5	77	77160	1335.3	20	SL
	Des	28.25	84.1	1.9	1007.3	50.5	77160	1335.3	25	SL
Pema- lang	Jan	25.5	82	3.8	1010.4	55	77160	1335.3	20	SL
	Feb	25	83	3.5	1011.1	58	77160	1335.3	22	SL
	Mar	25.4	83	2.5	1010.5	61	77160	1335.3	19	SL
	Apr	25.8	81	2.3	1010.8	72	77160	1335.3	15	SL
	Mei	25.8	79	2.8	1010.2	76	77160	1335.3	10	SL
	Jun	25.1	77	3.2	1010.9	85	77160	1335.3	1	S
	Jul	24.1	75	3.8	1010.3	90	77160	1335.3	4	S
	Agt	24.5	73	4.4	1011	91	77160	1335.3	3	L
	Sept	24.9	71	4.6	1011.2	89	77160	1335.3	5	L
	Okt	25.3	75	4.1	1009.9	71	77160	1335.3	11	L
	Nov	25.5	76	3.9	1009.9	72	77160	1335.3	11	SL
	Des	25	81	4.2	1009.2	54	77160	1335.3	20	SL
Tegal	Jan	28.1	82	3.8	1009.2	55	77160	1335.3	20	SL
	Feb	27.6	83	3.5	1010.1	58	77160	1335.3	22	SL
	Mar	27.9	83	2.5	1010	61	77160	1335.3	19	SL
	Apr	28.4	81	3.3	1010.2	72	77160	1335.3	15	SL
	Mei	28.7	79	2.8	1010.1	76	77160	1335.3	10	SL

	Jun	28.1	77	3.2	1010.8	85	77160	1335.3	1	S
	Jul	27.6	75	3.8	1011.3	90	77160	1335.3	4	S
	Agt	27.8	73	4.4	1010.8	91	77160	1335.3	3	L
	Sept	28.4	71	4.6	1010.9	89	77160	1335.3	5	L
	Okt	29	75	4.1	1009.8	71	77160	1335.3	11	L
	Nov	28.7	76	3.9	1007.8	72	77160	1335.3	11	SL
	Des	27.5	81	4.2	1008.9	54	77160	1335.3	20	SL
Bre- bes	Jan	27.7	82	3.8	1009.4	62	77160	1335.3	16	S
	Feb	27.2	84	3.8	1008.6	45	77160	1335.3	18.5	S
	Mar	27.3	83	3.2	1008.2	70	77160	1335.3	19	S
	Apr	28.3	79	2.7	1008.7	78	77160	1335.3	16.5	S
	Mei	28.4	79	2.7	1008.8	75	77160	1335.3	9.5	S
	Jun	27.5	79	2.1	1009.3	67	77160	1335.3	1.5	S
	Jul	27.7	76	2.7	1009.1	87	77160	1335.3	3	S
	Agt	28.1	74	3.8	1009.6	86	77160	1335.3	0	S
	Sept	28.3	76	3.2	1010.2	89	77160	1335.3	0	S
	Okt	27.9	79	3.8	1009.5	64	77160	1335.3	3	S
	Nov	28	80	3.2	1008.9	54	77160	1335.3	6.5	S
	Des	27.7	81	3.2	1008.1	50	77160	1335.3	18	S
Kota Sura- karta	Jan	26.4	86	15	1011.7	49	77160	1335.3	24	SL
	Feb	26.8	96	12	1014.5	55	77160	1335.3	21	SL
	Mar	26.7	84	15	1012.1	54	77160	1335.3	23	SL
	Apr	27.9	80	15	1011.5	85	77160	1335.3	17	SL
	Mei	27.9	74	10	1012.6	89	77160	1335.3	3	L
	Jun	26.5	70	12	1012.9	92	77160	1335.3	0	S
	Jul	26	70	10	1014.1	94	77160	1335.3	0	S
	Agt	26.4	67	10	1014.5	99	77160	1335.3	0	S
	Sept	27.4	64	10	1015	99	77160	1335.3	0	S
	Okt	29	63	10	1013.1	99	77160	1335.3	0	S
	Nov	29.2	67	10	1012.6	99	77160	1335.3	6	S
	Des	27.9	79	11	1011.7	75	77160	1335.3	19	SL
Kota Sema- rang	Jan	28.8	84.2	2.1	1008.8	65.7	77160	1335.3	14	SL
	Feb	28.2	86	2.1	1009.6	61.7	77160	1335.3	21	SL
	Mar	29.1	83.7	1.8	1009	72.7	77160	1335.3	18	SL
	Apr	29.7	80.9	1.6	1009.2	74.8	77160	1335.3	15	SL
	Mei	30.2	78.5	2	1008.7	86.3	77160	1335.3	11	SL
	Jun	30.1	76.8	2	1009	100	77160	1335.3	3	L
	Jul	29.8	74.8	1.9	1008.5	100	77160	1335.3	6	S
	Agt	30.4	73.4	2.4	1009	100	77160	1335.3	4	S
	Sept	30.5	71.8	2.1	1009	100	77160	1335.3	8	S
	Okt	29.9	75.9	1.9	1008.2	85.5	77160	1335.3	14	SL
	Nov	29.5	79.2	1.8	1008.5	77	77160	1335.3	20	SL
	Des	28.3	84.1	1.9	1007.3	50.5	77160	1335.3	25	SL

Kota Tegal	Jan	27.7	82	3	1010.1	4.2	77160	1335.3	15	SL
	Feb	27.8	81	3	1011.5	6.4	77160	1335.3	16	SL
	Mar	27.7	81	4	1010.5	4.3	77160	1335.3	16	SL
	Apr	28.6	79	3	1009.7	6.7	77160	1335.3	14	SL
	Mei	28.7	75	4	1010.5	7.4	77160	1335.3	3	S
	Jun	27.5	73	5	1010.9	7.8	77160	1335.3	0	S
	Jul	27	72	5	1011.8	8.2	77160	1335.3	3	S
	Agt	26.9	70	5	1012	8.5	77160	1335.3	0	S
	Sept	27.4	70	5	1012.4	8.8	77160	1335.3	0	S
	Okt	28.7	69	5	1010.6	9.1	77160	1335.3	1	S
	Nov	29	71	5	1010.1	7.4	77160	1335.3	3	L
	Des	28.6	78	3	1009.9	5.8	77160	1335.3	13	SL
:										
2022										
Cila-cap	Jan	27.9	83	3	1008.6	44.4	7010	2124.47	24	SL
	Feb	27.7	81	2	1009.1	48.6	7010	2124.47	24	SL
	Mar	27.3	85	2	1009.5	69.9	7010	2124.47	25	SL
	Apr	28.3	82	3	1010.7	80.4	7010	2124.47	10	SL
	Mei	27.9	81	5	1010.1	83.1	7010	2124.47	13	SL
	Jun	25.5	84	4	1011.2	73	7010	2124.47	3	S
	Jul	25.1	84	5	1011.5	89	7010	2124.47	2	S
	Agt	24.7	84	5	1012.4	77	7010	2124.47	3	S
	Sept	25.2	83	4	1012	77.2	7010	2124.47	1	S
	Okt	25.2	83	4	1011.5	60.8	7010	2124.47	4	S
	Nov	27.7	82	3	1009.4	44.3	7010	2124.47	11	SL
	Des	28.6	80	2	1010.6	54	7010	2124.47	21	SL
Banjar-negara	Jan	24.7	82.3	11	944	46.7	77160	1335.3	24	L
	Feb	24.6	85.2	11	945.3	41.5	77160	1335.3	22	L
	Mar	24.3	85.7	10.4	944.3	30.1	77160	1335.3	23	L
	Apr	24.8	85.1	9.9	943.9	48.3	77160	1335.3	21	S
	Mei	24.5	83.8	8.8	945	51.6	77160	1335.3	18	S
	Jun	22.9	81.5	8.7	945.2	46.2	77160	1335.3	3	S
	Jul	22.3	81.7	9.1	945.9	50.2	77160	1335.3	4	S
	Agt	22.4	79.5	9.9	946.5	51.2	77160	1335.3	2	S
	Sept	23.4	76.3	12.2	947.4	55.9	77160	1335.3	2	S
	Okt	24.9	74.7	13.2	945.5	54.5	77160	1335.3	2	S
	Nov	27.3	80.2	11.5	945.2	56.1	77160	1335.3	12	S
	Des	24.9	83.2	11.2	944.2	49.6	77160	1335.3	25	S
Boyo-lali	Jan	29	82	9	1010	68	77160	1335.3	21	SL
	Feb	29	82	9	1010	78	77160	1335.3	20	SL
	Mar	29	83	9	1010	78	77160	1335.3	18	SL
	Apr	30	84	10	1010	78	77160	1335.3	15	SL
	Mei	30	85	10	1010	80	77160	1335.3	7	SL

	Jun	29	84	10	1011	88	77160	1335.3	21	SL
	Jul	30	82	10	1011	93	77160	1335.3	20	SL
	Agt	29	81	10	1011	94	77160	1335.3	18	SL
	Sept	30	82	9	1011	92	77160	1335.3	15	SL
	Okt	31	83	9	1011	90	77160	1335.3	0	S
	Nov	31	83	10	1012	88	77160	1335.3	18	SL
	Des	31	83	11	1012	80	77160	1335.3	19	SL
Wono-giri	Jan	26.9	93.5	7	1008.6	44	77160	1335.3	20	SL
	Feb	25	93.85	7.1	1009.1	49	77160	1335.3	18	SL
	Mar	26.5	92	6.9	1009.5	70	77160	1335.3	23	SL
	Apr	26.75	91	6.7	1010.7	80	77160	1335.3	13	SL
	Mei	28.5	87	6	1010.1	83	77160	1335.3	12	SL
	Jun	28	89.5	6.5	1011.2	73	77160	1335.3	0	S
	Jul	26.5	93	7	1011.5	89	77160	1335.3	2	L
	Agt	28	93.5	7	1012.4	77	77160	1335.3	4	SL
	Sept	27	90.5	6.7	1012	77	77160	1335.3	5	L
	Okt	28	93.5	7	1011.5	61	77160	1335.3	16	SL
	Nov	27	93.5	7	1009.4	44	77160	1335.3	19	SL
	Des	25	93	7	1010.6	54	77160	1335.3	21	SL
Sema-rang	Jan	28.76	84.2	2.1	1008.8	65.7	77160	1335.3	14	SL
	Feb	28.21	85.9	2.1	1009.6	61.7	77160	1335.3	21	SL
	Mar	29.06	83.7	1.8	1009	72.7	77160	1335.3	18	SL
	Apr	29.65	80.9	1.6	1009.2	74.8	77160	1335.3	15	SL
	Mei	30.17	78.5	2	1008.7	86.3	77160	1335.3	11	SL
	Jun	30.08	76.8	2	1009	100	77160	1335.3	3	L
	Jul	29.85	74.8	1.9	1008.5	100	77160	1335.3	6	S
	Agt	30.37	73.4	2.4	1009	100	77160	1335.3	4	S
	Sept	30.46	71.8	2.1	1008.8	100	77160	1335.3	8	S
	Okt	29.85	75.9	1.9	1008.3	85.5	77160	1335.3	14	SL
	Nov	29.5	79.2	1.8	1008.5	77	77160	1335.3	20	SL
	Des	28.25	84.1	1.9	1007.3	50.5	77160	1335.3	25	SL
Pema-lang	Jan	25.5	82	3.8	1010.4	55	77160	1335.3	24	SL
	Feb	25	83	3.5	1011.1	58	77160	1335.3	25	SL
	Mar	25.4	83	2.5	1010.5	61	77160	1335.3	23	SL
	Apr	25.8	81	2.3	1010.8	72	77160	1335.3	15	SL
	Mei	25.8	79	2.8	1010.2	76	77160	1335.3	10	SL
	Jun	25.1	77	3.2	1010.9	85	77160	1335.3	12	SL
	Jul	24.1	75	3.8	1010.3	90	77160	1335.3	5	S
	Agt	24.5	73	4.4	1011	91	77160	1335.3	7	SL
	Sept	24.9	71	4.6	1011.2	89	77160	1335.3	11	SL
	Okt	25.3	75	4.1	1009.9	71	77160	1335.3	11	SL
	Nov	25.5	76	3.9	1009.9	72	77160	1335.3	22	SL
	Des	25	81	4.2	1009.2	54	77160	1335.3	21	SL

Tegal	Jan	27.7	82	4	1009.4	62	77160	1335.3	17	SL
	Feb	27.2	84	3	1008.6	45	77160	1335.3	15	SL
	Mar	27.3	83	3	1008.2	70	77160	1335.3	19	SL
	Apr	28.3	79	3	1008.7	78	77160	1335.3	10	SL
	Mei	28.4	79	3	1008.8	75	77160	1335.3	8	SL
	Jun	27.5	79	2	1009.3	67	77160	1335.3	8	SL
	Jul	27.7	76	3	1009.1	87	77160	1335.3	7	SL
	Agt	28.1	74	4	1009.6	86	77160	1335.3	1	S
	Sept	28.3	76	3	1010.2	89	77160	1335.3	6	S
	Okt	27.9	79	4	1009.5	64	77160	1335.3	8	SL
	Nov	28	80	3	1008.9	54	77160	1335.3	13	SL
	Des	27.7	81	3	1008.1	50	77160	1335.3	13	SL
Bre-bes	Jan	27.7	82	3.8	1009.4	62	77160	1335.3	28	SL
	Feb	27.2	84	3.8	1008.6	45	77160	1335.3	28	SL
	Mar	27.3	83	3.2	1008.2	70	77160	1335.3	31	SL
	Apr	28.3	79	2.7	1008.7	78	77160	1335.3	30	S
	Mei	28.4	79	2.7	1008.8	75	77160	1335.3	29	S
	Jun	27.5	79	2.1	1009.3	67	77160	1335.3	30	S
	Jul	27.7	76	2.7	1009.1	87	77160	1335.3	14	S
	Agt	28.1	74	3.8	1009.6	86	77160	1335.3	30	L
	Sept	28.3	76	3.2	1010.2	89	77160	1335.3	24	L
	Okt	27.9	79	3.8	1009.5	64	77160	1335.3	31	S
	Nov	28	80	3.2	1008.9	54	77160	1335.3	31	SL
	Des	27.7	81	3.2	1008.1	50	77160	1335.3	31	S
Kota Sura-karta	Jan	26.4	84	10	1010.9	49	77160	1335.3	24	SL
	Feb	26.8	84	10	1009.9	55	77160	1335.3	20	SL
	Mar	26.7	84	9	1009.6	54	77160	1335.3	25	SL
	Apr	27.9	81	10	1010.5	85	77160	1335.3	16	SL
	Mei	27.9	81	9	1010.5	89	77160	1335.3	16	SL
	Jun	26.5	80	9	1010.9	92	77160	1335.3	17	SL
	Jul	26	74	9	1011	94	77160	1335.3	5	L
	Agt	26.4	72	9	1011	99	77160	1335.3	8	S
	Sept	27.4	73	10	1012	99	77160	1335.3	13	S
	Okt	29	78	11	1011	99	77160	1335.3	16	S
	Nov	29.2	83	9	1011	99	77160	1335.3	23	L
	Des	27.9	81	9	1010	75	77160	1335.3	22	S
Kota Sema-rang	Jan	28.8	84.2	2.1	1008.8	65.7	77160	1335.3	14	SL
	Feb	28.2	86	2.1	1009.6	61.7	77160	1335.3	21	SL
	Mar	29.1	83.7	1.8	1009	72.7	77160	1335.3	18	SL
	Apr	29.7	80.9	1.6	1009.2	74.8	77160	1335.3	15	SL
	Mei	30.2	78.5	2	1008.7	86.3	77160	1335.3	11	SL
	Jun	30.1	76.8	2	1009	100	77160	1335.3	3	L
	Jul	29.8	74.8	1.9	1008.5	100	77160	1335.3	6	S

	Agt	30.4	73.4	2.4	1009	100	77160	1335.3	4	S
	Sept	30.5	71.8	2.1	1009	100	77160	1335.3	8	S
	Okt	29.9	75.9	1.9	1008.2	85.5	77160	1335.3	14	SL
	Nov	29.5	79.2	1.8	1008.5	77	77160	1335.3	20	SL
	Des	28.3	84.1	1.9	1007.3	50.5	77160	1335.3	25	SL
Kota Tegal	Jan	27.1	84.23	1.9	1008.4	30.1	77160	1335.3	21	SL
	Feb	27	85.71	2	1009.2	38.2	77160	1335.3	22	SL
	Mar	27.7	82	1.4	1009	66.8	77160	1335.3	17	SL
	Apr	28.4	78.61	1.5	1009.8	82.4	77160	1335.3	8	SL
	Mei	28.9	76.75	1.6	1009	79.8	77160	1335.3	4	L
	Jun	28.1	79.96	1.5	1010.5	77.4	77160	1335.3	11	S
	Jul	28.1	75.26	1.7	1010.1	94.5	77160	1335.3	4	L
	Agt	28.2	74.96	2.1	1010.7	95.3	77160	1335.3	5	L
	Sept	28	75.51	2	1010.1	80.6	77160	1335.3	7	S
	Okt	28.5	74.83	2	1009.6	86.1	77160	1335.3	6	S
	Nov	28	80.79	1.6	1008.3	50.6	77160	1335.3	15	SL
	Des	28	80.18	1.9	1008.3	59	77160	1335.3	12	SL

Lampiran 2. Hasil *Encoding* Data Kategorik (y_{target_k})

Data Training			Data Testing		
Y	Label Encoding	One-Hot Encoding	Y	Label Encoding	One-Hot Encoding
SL	3	[0, 0, 0, 1]	SL	3	[0, 0, 0, 1]
SL	3	[0, 0, 0, 1]	SL	3	[0, 0, 0, 1]
SL	3	[0, 0, 0, 1]	SL	3	[0, 0, 0, 1]
SL	3	[0, 0, 0, 1]	SL	3	[0, 0, 0, 1]
SL	3	[0, 0, 0, 1]	SL	3	[0, 0, 0, 1]
S	1	[0, 1, 0, 0]	SL	3	[0, 0, 0, 1]
S	1	[0, 1, 0, 0]	SL	3	[0, 0, 0, 1]
S	1	[0, 1, 0, 0]	SL	3	[0, 0, 0, 1]
S	1	[0, 1, 0, 0]	SL	3	[0, 0, 0, 1]
S	1	[0, 1, 0, 0]	SL	3	[0, 0, 0, 1]
SL	3	[0, 0, 0, 1]	S	1	[0, 1, 0, 0]
SL	3	[0, 0, 0, 1]	SL	3	[0, 0, 0, 1]
L	2	[0, 0, 0, 1]	SL	3	[0, 0, 0, 1]
L	2	[0, 0, 1, 0]	SL	3	[0, 0, 0, 1]
L	2	[0, 0, 1, 0]	SL	3	[0, 0, 0, 1]
S	1	[0, 1, 0, 0]	SL	3	[0, 0, 0, 1]
S	1	[0, 1, 0, 0]	SL	3	[0, 0, 0, 1]
S	1	[0, 1, 0, 0]	S	1	[0, 1, 0, 0]
⋮	⋮	⋮	⋮	⋮	⋮
S	1	[0, 1, 0, 0]	SL	3	[0, 0, 0, 1]

Lampiran 3. Hasil Normalisasi Data *Training*.

No	X1	X2	X3	X4	X5	X6	X7	X8
1	0.6552	0.5343	0.0441	0.9377	0.7975	1	0	0.2258
2	0.2989	0.7091	0.7206	0.0042	0.4739	1	0	0.8065
3	0.6782	0.4318	0.6324	0.9646	0.666	1	0	0.3871
4	0.3448	0.5227	0.1985	0.9348	0.6973	1	0	0.3548
5	0.5747	0.4773	0.4853	0.9759	0.6973	1	0	0.2258
6	0.5989	0.8341	0.1176	0.9278	0.4186	1	0	0.0968
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
422	0.4253	0.4091	0.6324	0.9943	0.9374	1	0	0

Lampiran 4. Hasil Normalisasi Data *Testing*.

No	X1	X2	X3	X4	X5	X6	X7	X8
1	0.7701	0.7045	0.5588	0.9363	0.7704	1	0	0.5806
2	0.6552	0.6405	0.0368	0.9122	0.572	1	0	0.3871
3	0.3218	0.7273	0.2647	0.9575	0.8852	0	1	0.0645
4	0.2989	0.5159	0.8676	0.0227	0.5251	1	0	0.0645
5	0.6322	0.4773	0.2206	0.9476	0.9061	1	0	0.0968
6	0.7011	0.6136	0.0956	0.9193	0.739	1	0	0.3065
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
106	0.7701	0.5227	0.1985	0.9348	0.6973	1	0	0.0323

Lampiran 5. Percobaan Beberapa Model Pertama.

No	Model	Arsitektur			Loss
		Input	Hidden	Output	
1	A	8	4	4	0.616
2	B	8	8	4	0.5776
3	C	8	12	4	0.5785
4	D	8	16	4	0.6126
5	E	8	20	4	0.4729
6	F	8	24	4	0.5124
7	G	8	28	4	0.5365
8	H	8	32	4	0.5001

Lampiran 6. Bobot dan Bias Awal (*Input ke Hidden Layer*) Model Pertama

<i>Input to Hidden</i>	1	2	3	4	5	6	7	8	Bias
1	-0.12	-0.12	0.12	0.36	-0.38	0.18	-0.15	0.43	0.00
2	-0.30	0.26	-0.37	0.02	0.15	0.14	-0.21	-0.17	0.00
3	0.29	0.31	-0.17	0.36	0.23	0.06	-0.06	0.45	0.00
4	-0.19	0.14	-0.42	-0.34	-0.38	-0.08	0.38	-0.43	0.00

5	0.01	-0.24	0.09	0.45	0.44	-0.14	0.11	0.22	0.00
6	0.42	0.02	-0.27	-0.20	-0.45	-0.13	-0.31	-0.07	0.00
7	0.30	0.26	0.46	-0.20	0.34	-0.25	-0.10	-0.31	0.00
8	-0.03	0.01	0.24	-0.05	0.30	-0.04	0.42	-0.13	0.00
9	0.14	-0.44	-0.45	0.26	-0.19	-0.05	-0.36	-0.03	0.00
10	0.22	0.07	-0.33	0.26	-0.40	0.31	-0.07	-0.35	0.00
11	0.14	0.32	-0.43	-0.22	-0.41	0.10	0.11	-0.07	0.00
12	0.22	-0.25	0.22	0.28	-0.15	-0.12	0.03	-0.20	0.00
13	-0.21	-0.43	-0.31	0.08	-0.35	0.15	0.38	-0.11	0.00
14	-0.37	0.10	0.17	0.31	0.07	0.22	-0.15	0.32	0.00
15	0.26	-0.17	0.39	0.15	-0.02	0.39	0.01	0.24	0.00
16	0.19	0.42	-0.31	-0.38	0.42	0.30	-0.35	0.36	0.00
17	0.01	-0.08	0.09	-0.03	-0.39	0.00	-0.11	-0.16	0.00
18	0.34	0.33	-0.07	-0.13	0.28	-0.10	-0.39	0.12	0.00
19	0.00	0.46	-0.34	0.11	0.36	-0.17	0.23	0.34	0.00
20	-0.43	0.23	0.11	-0.08	0.42	-0.29	0.09	-0.04	0.00

Lampiran 7. Bobot dan Bias Awal (*Hidden ke Output Layer*) Model Pertama

<i>Hidden to output</i>	1	2	3	4
1	0.3621	0.1571	0.1042	-0.0849
2	0.3955	-0.4988	0.4152	0.0687
3	-0.3229	-0.0943	0.1721	0.1289
4	-0.0804	-0.0812	-0.2123	0.2754
5	-0.2226	0.4676	0.0150	-0.1673
6	-0.1448	0.4337	-0.3087	-0.4628
7	-0.2410	-0.3181	-0.2155	0.0904
8	-0.3487	-0.2794	0.4544	0.3518
9	-0.1743	-0.4453	0.0480	-0.0632
10	-0.1481	-0.3153	0.4921	-0.0307
11	0.3547	-0.3724	0.1092	0.4696
12	0.3605	0.0707	0.1754	0.1745
13	-0.3982	-0.4842	-0.3019	0.1276
14	-0.2546	-0.1893	-0.1467	0.4797
15	-0.1271	0.3568	-0.2056	-0.3683
16	-0.1841	0.4769	-0.2899	0.0538
17	-0.0141	-0.2434	-0.0038	0.1713
18	-0.2680	0.0866	-0.0728	0.1108
19	0.1144	-0.4456	0.0032	-0.2875
20	0.3667	0.1229	0.2462	-0.4895
Bias	0.0000	0.0000	0.0000	0.0000

Lampiran 8. *Output (Input ke Hidden Layer) Model Pertama Sebelum Aktivasi*

<i>Input to Hidden</i>	1	2	3	4	5	6	7	8
1	0.1716	0.3104	0.3811	0.3211	0.2935	0.2379	0.1780	0.4626
2	0.1738	-0.0965	-0.1257	0.1684	0.0048	0.2044	0.1186	0.0876
3	1.0247	0.7129	0.9549	0.9401	0.9020	0.9419	0.8611	1.1680
4	-0.8618	-0.8674	-1.1561	-0.8850	-1.0137	-0.6366	-0.9196	-0.9537
5	0.5691	0.1504	0.6386	0.5673	0.5965	0.3023	0.5171	0.5045
6	-0.4307	-0.4647	-0.5369	-0.5643	-0.5445	-0.2847	-0.6104	-0.4112
7	0.1181	0.2620	0.2666	0.0198	0.2394	0.1257	0.3709	-0.0328
8	0.1162	0.1687	0.1924	0.1169	0.1912	0.0428	0.2519	0.0144
9	-0.1273	-0.7612	-0.3182	-0.2223	-0.2845	-0.2265	-0.4466	-0.2527
10	0.3230	-0.2803	0.1305	0.1968	0.2056	0.5010	0.2003	0.2159
11	-0.1971	-0.1884	-0.4411	-0.2790	-0.3820	0.0279	-0.3486	-0.1214
12	-0.0166	-0.3161	0.1385	-0.0587	0.1034	-0.0061	0.0419	-0.1175
13	-0.4683	-0.7056	-0.5833	-0.4257	-0.5271	-0.4622	-0.6296	-0.5272
14	0.4520	0.5921	0.5835	0.6267	0.5554	0.4432	0.5701	0.6508
15	0.6740	0.8107	0.9693	0.6863	0.8407	0.6112	0.7437	0.7702
16	0.6903	0.9138	0.4614	0.5845	0.4578	0.5828	0.5040	0.7724
17	-0.4044	-0.3061	-0.3191	-0.3738	-0.3231	-0.2555	-0.3364	-0.3788
18	0.4280	0.4170	0.3405	0.2953	0.3185	0.3800	0.3575	0.4277
19	0.5271	0.3558	0.2924	0.4767	0.3205	0.4594	0.3754	0.6277
20	-0.1905	-0.0038	-0.2239	-0.0898	-0.1668	-0.2408	-0.0213	-0.2134

Lampiran 9. *Output (Input ke Hidden Layer) Model E Setelah Aktivasi*

<i>Input to Hidden</i>	1	2	3	4	5	6	7	8
1	0.1716	0.3104	0.3811	0.3211	0.2935	0.2379	0.1780	0.4626
2	0.1738	0.0000	0.0000	0.1684	0.0048	0.2044	0.1186	0.0876
3	1.0247	0.7129	0.9549	0.9401	0.9020	0.9419	0.8611	1.1680
4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5	0.5691	0.1504	0.6386	0.5673	0.5965	0.3023	0.5171	0.5045
6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
7	0.1181	0.2620	0.2666	0.0198	0.2394	0.1257	0.3709	0.0000
8	0.1162	0.1687	0.1924	0.1169	0.1912	0.0428	0.2519	0.0144
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.3230	0.0000	0.1305	0.1968	0.2056	0.5010	0.2003	0.2159
11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0279	0.0000	0.0000
12	0.0000	0.0000	0.1385	0.0000	0.1034	0.0000	0.0419	0.0000
13	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
14	0.4520	0.5921	0.5835	0.6267	0.5554	0.4432	0.5701	0.6508
15	0.6740	0.8107	0.9693	0.6863	0.8407	0.6112	0.7437	0.7702
16	0.6903	0.9138	0.4614	0.5845	0.4578	0.5828	0.5040	0.7724

17	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
18	0.4280	0.4170	0.3405	0.2953	0.3185	0.3800	0.3575	0.4277
19	0.5271	0.3558	0.2924	0.4767	0.3205	0.4594	0.3754	0.6277
20	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Lampiran 10. Hasil Prediksi Model E (y_k)

Data Training				
No.	Prediksi	Decoding	Kelas	Label
1	[0.1185, 0.3387, 0.2780, 0.2647]	[0, 1, 0, 0]	1	S
2	[0.3270, 0.5401, 0.0814, 0.0516]	[0, 0, 1, 0]	2	L
3	[0.0122, 0.1797, 0.4764, 0.3317]	[0, 0, 1, 0]	2	L
4	[0.0029, 0.0963, 0.2370, 0.6637]	[0, 0, 0, 1]	3	SL
5	[0.0283, 0.2278, 0.5479, 0.1960]	[0, 0, 1, 0]	2	L
6	[0.1880, 0.3076, 0.2644, 0.2400]	[0, 1, 0, 0]	1	S
7	[0.7552, 0.2063, 0.0357, 0.0027]	[1, 0, 0, 0]	0	R
8	[0.0001, 0.0123, 0.0712, 0.9164]	[0, 0, 0, 1]	3	SL
9	[0.8881, 0.0977, 0.0134, 0.0008]	[1, 0, 0, 0]	0	R
10	[0.0002, 0.0207, 0.0952, 0.8840]	[0, 0, 0, 1]	3	SL
...
422	[0.9336, 0.0535, 0.0128, 0.0001]	[1, 0, 0, 0]	0	R
Data Testing				
No.	Prediksi	Decoding	Kelas	Label
1	[0.0002, 0.0155, 0.1026, 0.8817]	[0, 0, 0, 1]	3	SL
2	[0.0019, 0.0613, 0.2105, 0.7263]	[0, 0, 0, 1]	3	SL
3	[0.9788, 0.0187, 0.0007, 0.0018]	[1, 0, 0, 0]	0	R
4	[0.9987, 0.0013, 0.0000, 0.0000]	[1, 0, 0, 0]	0	R
5	[0.3868, 0.2603, 0.2825, 0.0704]	[1, 0, 0, 0]	0	R
6	[0.0097, 0.5293, 0.1240, 0.3370]	[0, 1, 0, 0]	1	S
7	[0.0019, 0.7263, 0.0613, 0.2105]	[0, 1, 0, 0]	1	S
8	[0.0001, 0.0114, 0.0651, 0.9234]	[0, 0, 0, 1]	3	SL
9	[0.6859, 0.0039, 0.0812, 0.2290]	[1, 0, 0, 0]	0	R
10	[0.8448, 0.0840, 0.0634, 0.0077]	[1, 0, 0, 0]	0	R
...
106	[0.4108, 0.2428, 0.2474, 0.0989]	[1, 0, 0, 0]	0	R

Lampiran 11. Bobot dan Bias Akhir (*Input ke Hidden Layer*) Model Pertama

Input to Hidden	1	2	3	4	5	6	7	8	Bias
1	-0.72	0.02	0.62	0.35	-0.83	0.18	0.70	0.68	0.15
2	-0.42	0.38	-0.58	0.18	0.07	0.14	-0.10	-0.12	0.00
3	0.58	0.12	-0.43	0.56	0.49	0.06	-0.91	0.16	-0.21
4	-0.18	0.27	-0.07	-0.23	-0.54	-0.08	0.60	-0.27	0.11
5	0.23	-0.34	0.40	0.46	0.73	-0.14	-0.24	0.28	0.01

6	0.29	0.12	-0.21	-0.36	-0.53	-0.13	-0.27	-0.04	-0.10
7	0.57	0.34	0.05	0.02	0.54	-0.25	-0.82	-0.44	-0.01
8	0.32	-0.07	0.11	0.20	0.57	-0.04	-0.35	-0.29	-0.10
9	0.59	-0.28	-0.67	0.63	0.23	-0.05	-0.91	0.09	0.13
10	0.24	0.01	-0.06	0.26	-0.37	0.31	-0.17	-0.42	-0.05
11	0.26	0.60	-0.72	0.01	-0.43	0.10	0.38	0.01	0.17
12	0.47	-0.10	-0.51	0.34	-0.12	-0.12	0.12	-0.41	0.11
13	-0.21	-0.43	-0.31	0.08	-0.35	0.15	0.38	-0.11	0.00
14	-0.12	0.21	-0.32	0.44	0.21	0.22	-0.79	0.20	0.01
15	0.09	-0.29	0.87	0.23	-0.06	0.39	-0.03	0.29	-0.07
16	0.48	0.50	-0.61	-0.44	0.59	0.30	-0.53	0.28	0.07
17	0.27	-0.20	0.09	0.28	-0.40	0.00	-0.27	-0.35	0.23
18	0.72	0.35	-0.42	0.05	0.63	-0.10	-1.03	-0.02	0.00
19	-0.37	0.59	-0.20	0.10	0.00	-0.17	0.90	0.53	0.14
20	-0.84	0.31	0.62	-0.20	0.07	-0.29	1.02	0.21	0.17

Lampiran 12. Bobot dan Bias Akhir (*Hidden ke Output Layer*) Model Pertama

<i>Hidden to Output</i>	1	2	3	4
1	0.9620	0.2426	0.1506	-0.7571
2	0.7115	-0.8464	-0.0487	0.4233
3	-0.6485	-0.1814	0.2244	0.2884
4	0.8991	-0.9742	-0.9123	-0.5735
5	-0.6592	0.4485	0.3588	-0.2443
6	-0.1619	0.3312	-0.4279	-0.3553
7	-0.9316	-0.3607	-0.1310	0.3316
8	-0.9151	-0.2989	0.6084	0.4449
9	-1.1085	-0.8760	0.1630	0.3937
10	-0.2282	-0.5203	0.0931	0.3148
11	0.8947	-0.3271	-0.6673	0.6271
12	-0.1325	-0.3079	-0.0015	0.6254
13	-0.3982	-0.4842	-0.3019	0.1276
14	-0.4797	-0.3337	0.0061	0.5906
15	-0.1560	0.4105	0.2221	-0.8574
16	-0.6018	0.6144	-0.3981	0.2345
17	-0.0068	-0.5966	-0.7383	0.7638
18	-0.9063	0.0536	-0.0248	0.3631
19	0.4569	-0.3750	-0.2901	-0.4084
20	1.0037	0.3430	-0.0248	-1.1063
Bias	0.1785	0.0786	-0.1775	-0.0613

Lampiran 13. Percobaan Beberapa Model Kedua.

Model	Arsitektur				Loss
	Input Layer	Hidden Layer 1	Hidden Layer 2	Output Layer	
I	8	4	4	4	0.5874
J	8	4	8	4	0.5794
K	8	4	12	4	0.5379
L	8	4	16	4	0.5361
M	8	4	20	4	0.7197
N	8	4	24	4	0.4999
O	8	4	28	4	0.5035
P	8	4	32	4	0.4568
Q	8	8	8	4	0.4875
R	8	8	12	4	0.5344
S	8	8	16	4	0.5151
T	8	8	20	4	0.5302
U	8	8	24	4	0.4997
V	8	8	28	4	0.4937
W	8	8	32	4	0.5218
X	8	12	12	4	0.5122
Y	8	12	16	4	0.4856
Z	8	12	20	4	0.4713
AA	8	12	24	4	0.4822
AB	8	12	28	4	0.5264
AC	8	12	32	4	0.4913
AD	8	16	16	4	0.4821
AE	8	16	20	4	0.4887
AF	8	16	24	4	0.4744
AG	8	16	28	4	0.4748
AH	8	16	32	4	0.5124
AI	8	20	20	4	0.482
AJ	8	20	24	4	0.4623
AK	8	20	28	4	0.4576
AL	8	20	32	4	0.5289
AM	8	24	24	4	0.4703
AN	8	24	28	4	0.5003
AO	8	24	32	4	0.4629
AP	8	28	28	4	0.4716
AQ	8	28	32	4	0.4532
AR	8	32	32	4	0.4737

Lampiran 14. Bobot dan Bias Awal (*Input ke Hidden Layer 1*) Model AQ.

<i>Input to H1</i>	1	2	3	4	5	6	7	8	Bias
1	0.12	-0.26	0.13	0.35	0.37	0.04	-0.01	-0.29	0.00
2	-0.15	-0.23	0.16	0.23	-0.05	-0.08	-0.26	-0.20	0.00
3	0.04	0.31	0.09	-0.09	0.32	0.36	0.11	0.19	0.00
4	-0.24	-0.30	-0.16	0.36	-0.10	0.09	-0.33	0.14	0.00
5	0.37	-0.13	-0.15	-0.06	-0.33	-0.22	-0.27	0.27	0.00
6	0.05	0.39	-0.16	0.18	-0.25	-0.30	0.30	0.36	0.00
7	0.38	0.02	-0.01	0.33	-0.21	0.11	0.29	0.05	0.00
8	0.35	-0.02	0.36	-0.16	0.17	-0.15	-0.26	-0.26	0.00
9	0.22	-0.21	-0.36	0.06	-0.03	-0.03	0.07	-0.16	0.00
10	-0.25	-0.10	-0.39	0.23	0.24	-0.38	0.37	-0.38	0.00
11	0.13	0.12	-0.18	0.28	0.19	-0.24	-0.23	-0.34	0.00
12	0.23	0.08	-0.09	-0.38	0.19	0.38	0.17	-0.34	0.00
13	-0.34	0.09	-0.09	-0.04	-0.29	-0.20	0.10	-0.18	0.00
14	-0.29	-0.04	-0.15	-0.04	-0.05	-0.11	0.28	-0.19	0.00
15	0.32	-0.07	0.38	-0.36	-0.03	0.29	0.09	0.40	0.00
16	0.28	-0.24	-0.33	-0.06	-0.16	0.40	0.38	-0.36	0.00
17	-0.01	0.39	-0.27	0.22	0.11	0.32	-0.07	-0.32	0.00
18	0.07	0.15	0.02	-0.29	-0.05	-0.28	-0.34	-0.06	0.00
19	0.30	-0.37	0.04	-0.01	-0.28	0.01	-0.02	-0.41	0.00
20	-0.23	-0.29	-0.06	-0.04	0.12	-0.36	-0.40	0.38	0.00
21	0.08	-0.36	0.16	0.23	-0.18	0.32	-0.28	-0.37	0.00
22	0.14	-0.37	0.19	-0.06	0.37	-0.13	0.11	0.35	0.00
23	0.00	-0.02	0.01	0.08	0.07	-0.29	-0.03	0.18	0.00
24	0.00	-0.17	-0.37	0.03	-0.31	0.36	0.30	-0.12	0.00
25	0.05	-0.08	0.40	0.07	0.03	-0.25	0.08	0.11	0.00
26	-0.05	0.00	0.13	0.08	-0.18	0.06	-0.41	0.24	0.00
27	0.32	-0.09	0.25	-0.30	0.10	-0.05	-0.21	-0.10	0.00
28	0.16	-0.28	0.33	0.30	-0.24	0.24	-0.37	0.28	0.00

Lampiran 15. Bobot dan Bias Awal (*Hidden Layer 1 ke Hidden Layer 2*) Model AQ.

<i>H1 to H2</i>	1	2	3	4	5	6	7
1	0.2055	-0.0667	0.2299	0.0116	-0.0449	0.1730	-0.1448
2	0.1774	-0.0546	0.1038	0.0219	-0.1796	0.1342	-0.2823
3	-0.1421	-0.2518	0.2283	0.0121	0.2508	-0.0526	0.0466
4	-0.0384	-0.1467	0.1789	-0.0620	-0.1999	0.1372	-0.0770
5	0.1943	-0.2230	-0.2730	0.0760	-0.2890	0.2721	0.1035
6	-0.0901	0.1666	0.1383	-0.1203	0.0931	0.2527	0.0248

7	0.0287	-0.0103	-0.1208	-0.0085	0.2736	-0.0529	0.1196
8	0.1745	0.2628	-0.3027	0.0809	-0.0111	-0.1309	0.1682
9	0.1898	0.0769	0.2864	-0.1795	0.1832	0.2164	-0.0142
10	-0.3112	0.2314	-0.1573	-0.2577	-0.2234	-0.0615	-0.2741
11	0.2283	-0.1469	-0.2956	-0.0161	-0.1619	0.1631	-0.1135
12	-0.2180	-0.2205	-0.0031	0.0656	0.1099	0.0843	0.2111
13	0.1258	-0.2773	0.0172	0.0031	0.3034	-0.0657	-0.2124
14	0.0911	0.2151	-0.0293	-0.0451	-0.0978	0.2106	0.1057
15	0.1506	-0.1810	-0.1724	0.0644	0.2130	0.0038	-0.0968
16	-0.2143	-0.0413	-0.2438	0.1433	-0.0156	-0.1211	-0.0956
17	-0.2771	-0.2589	0.1655	0.2013	-0.0135	0.0253	0.2060
18	-0.1456	0.0077	-0.0148	0.0258	0.1046	0.2844	-0.1943
19	0.0905	0.2344	0.0001	-0.1782	-0.1471	0.2622	-0.1386
20	-0.1482	-0.1574	-0.2176	0.0476	-0.2499	-0.3086	-0.1233
21	-0.1743	-0.1431	-0.2727	-0.0947	0.2635	0.1353	-0.1320
22	0.2604	-0.1273	0.0697	-0.0438	0.2926	-0.0057	0.3048
23	-0.0989	-0.0032	0.0354	0.0766	-0.2452	0.0875	-0.1636
24	-0.0031	0.2599	0.0587	-0.0817	0.0820	0.3078	-0.0071
25	-0.0827	-0.2431	-0.1373	0.0931	0.1122	0.1304	-0.1804
26	0.0756	0.1481	0.2848	0.0514	0.2066	-0.1154	0.0018
27	-0.1537	-0.3032	-0.1406	-0.1328	-0.1263	0.2728	-0.2088
28	-0.2585	0.1110	-0.0977	0.2406	-0.1518	-0.3039	0.1129
29	-0.2483	0.2708	0.2020	0.1249	0.1707	-0.1485	0.0969
30	-0.0694	-0.1563	0.0315	0.0188	-0.0753	0.0035	-0.0314
31	-0.0375	-0.1558	-0.1681	0.2886	-0.0705	0.2576	0.2171
32	0.2339	0.2467	0.1985	0.1395	0.2629	0.1746	-0.0878
H 1 to H 2	8	9	10	11	12	13	14
1	0.2062	0.0189	-0.1416	0.0828	-0.1445	0.2088	0.2062
2	-0.0140	-0.2964	0.2281	-0.1812	0.1580	0.0209	-0.0140
3	-0.0099	0.0707	0.1530	0.2874	-0.0471	-0.3105	-0.0099
4	0.2227	0.2615	0.1326	-0.1538	-0.0270	0.1047	0.2227
5	-0.0556	-0.2547	-0.1962	0.0348	-0.0869	0.2139	-0.0556
6	0.2471	0.2587	0.1063	0.2842	-0.1854	0.2946	0.2471
7	-0.2447	-0.3085	0.1627	-0.2250	-0.1249	0.0904	-0.2447
8	-0.0954	-0.0413	-0.2159	-0.1339	-0.0048	-0.2958	-0.0954
9	0.0038	0.2830	-0.2240	-0.0315	-0.0291	0.0350	0.0038
10	0.1188	0.2838	0.2700	0.2440	-0.1791	0.3061	0.1188
11	0.2284	-0.1438	-0.0493	0.2746	-0.1539	0.2310	0.2284
12	0.1086	0.0583	-0.0293	0.2079	-0.0457	-0.1041	0.1086
13	-0.2146	0.2407	-0.2845	-0.2559	-0.0595	0.1392	-0.2146
14	-0.1119	0.1950	0.2404	-0.0238	-0.2724	-0.2412	-0.1119
15	-0.2026	0.2205	-0.0924	-0.0868	-0.2344	-0.0717	-0.2026

16	-0.2877	-0.1198	0.0505	-0.1971	0.0982	0.2858	-0.2877
17	-0.2738	0.2405	-0.2141	-0.1667	-0.1781	-0.0098	-0.2738
18	0.0163	0.2516	0.2217	0.2368	0.1659	-0.1165	0.0163
19	0.2904	0.0905	0.1790	-0.3086	0.1798	-0.3144	0.2904
20	-0.2966	-0.2240	-0.0081	0.0602	0.2322	0.2693	-0.2966
21	0.2135	0.2746	-0.1636	-0.0485	0.3103	0.2958	0.2135
22	0.1475	-0.0684	0.1664	0.2438	-0.0508	0.2220	0.1475
23	0.0501	-0.1308	0.1820	-0.0603	0.2169	0.1679	0.0501
24	-0.0001	-0.1945	-0.1333	-0.0139	-0.0463	0.1385	-0.0001
25	-0.1661	-0.2915	0.2673	-0.2207	0.2367	0.2732	-0.1661
26	-0.1167	-0.0532	-0.0148	-0.1886	0.2535	0.2266	-0.1167
27	-0.1328	-0.1747	0.0218	0.1164	-0.1663	0.1484	-0.1328
28	0.0406	-0.2231	-0.0853	0.0491	0.1681	0.1707	0.0406
29	0.2446	-0.2133	0.0518	0.3066	0.2862	-0.2405	0.2446
30	-0.2537	-0.2794	-0.3080	-0.2953	0.2553	0.1824	-0.2537
31	0.2429	-0.1194	0.0738	-0.1486	0.2012	0.2936	0.2429
32	0.2546	0.0633	0.2920	0.3037	0.2943	-0.1177	0.2546

<i>H 1</i> to <i>H 2</i>	15	16	17	18	19	20	21
1	0.3067	0.2906	-0.1063	0.2502	-0.0403	0.0225	0.0317
2	0.2320	-0.1483	0.2560	0.0424	0.1795	-0.0413	-0.2366
3	-0.1905	-0.0735	0.2854	-0.1876	-0.3151	0.2527	0.1281
4	-0.3035	-0.0165	0.1240	0.1111	-0.1651	-0.2660	0.0338
5	0.3107	0.1978	-0.1974	0.1117	-0.1813	-0.0236	-0.1395
6	0.0687	0.2030	-0.0026	0.0116	-0.2687	0.2512	0.1508
7	0.3111	-0.2032	0.1861	0.3110	0.1376	-0.2456	-0.1606
8	0.2023	0.1310	-0.0704	0.0452	0.2679	0.2533	-0.0271
9	0.2156	-0.0870	-0.2378	0.3105	-0.1350	-0.1884	-0.0143
10	-0.2341	-0.1740	-0.1300	0.0300	-0.1185	-0.0130	-0.0264
11	-0.2647	0.1024	0.0852	-0.0677	0.2040	-0.1912	0.2443
12	0.0220	-0.1130	-0.0053	-0.2231	0.3006	-0.2206	-0.0270
13	0.1902	0.1368	-0.1740	-0.0045	0.1468	0.2886	0.1533
14	0.0780	-0.0594	-0.1393	0.2495	0.0578	0.1299	-0.2153
15	0.0571	-0.0495	0.0443	0.1864	0.0029	-0.0048	0.2649
16	0.1702	-0.2554	-0.2246	0.3030	-0.1391	-0.2369	0.1436
17	0.0422	-0.2428	0.2506	0.0145	-0.3013	-0.0430	0.1651
18	-0.2676	-0.0997	-0.2067	0.1026	-0.0426	0.3131	0.1023
19	-0.2197	-0.3081	-0.1883	0.1394	-0.2786	-0.1561	0.0782
20	-0.2129	0.2953	-0.2513	-0.2060	0.0595	-0.2782	-0.2951
21	0.2426	0.1826	0.2223	0.2380	0.1562	0.1048	-0.1614
22	-0.0046	0.0755	0.2967	-0.2888	-0.0746	-0.1418	-0.1600

23	-0.0998	-0.1705	-0.2253	-0.0529	0.1313	0.2559	0.0487
24	0.2067	0.1351	-0.1008	-0.1492	-0.0932	0.2756	-0.1138
25	0.0545	0.0294	-0.2616	0.1927	-0.0810	0.2056	0.2399
26	0.2106	0.0134	-0.1072	0.1965	0.1893	0.2213	0.1327
27	0.0360	-0.1305	0.2137	-0.1863	0.0568	-0.0329	-0.2338
28	0.2410	0.1809	-0.2182	0.0834	0.1330	-0.1386	-0.0070
29	-0.3021	0.1276	0.3044	-0.0969	-0.0628	0.2923	0.0545
30	0.2853	0.2708	0.0420	0.1360	0.2953	0.2486	-0.1927
31	0.3122	-0.0536	-0.2443	0.1174	0.2421	0.0116	-0.3021
32	-0.2713	0.2353	-0.0239	-0.2413	0.0986	-0.1337	0.0420

<i>H 1</i> <i>to</i> <i>H 2</i>	22	23	24	25	26	27	28	Bias
1	-0.1625	0.2845	0.1150	0.0212	-0.0621	-0.1326	-0.2652	0.00
2	0.2415	-0.3017	-0.2378	-0.2117	-0.0465	-0.1868	-0.2937	0.00
3	0.0789	-0.0340	0.3037	-0.0576	-0.1345	-0.2305	-0.1875	0.00
4	-0.2938	-0.1999	-0.0696	0.2982	0.0689	-0.2384	0.0754	0.00
5	-0.0757	0.0413	0.1656	0.1843	0.2292	-0.1059	0.1401	0.00
6	0.1494	0.2991	0.0743	-0.0631	0.1197	-0.2472	-0.0997	0.00
7	-0.1725	-0.3026	-0.1663	-0.2058	-0.0842	-0.1639	-0.2538	0.00
8	-0.3053	-0.2394	-0.0545	-0.0131	0.2278	0.0480	0.1239	0.00
9	0.0856	0.2107	0.2610	-0.1886	-0.1939	-0.3141	-0.2472	0.00
10	-0.2969	0.1961	0.2355	0.1039	-0.2787	0.2892	-0.1550	0.00
11	0.2515	-0.0646	0.0797	0.2718	-0.1361	-0.2422	0.1678	0.00
12	-0.2520	0.1935	0.2208	-0.1266	-0.0418	-0.2469	-0.0817	0.00
13	0.2657	0.2038	0.1369	-0.0330	-0.2148	0.3087	-0.2582	0.00
14	-0.0447	0.0742	0.3101	0.2694	-0.2251	0.0397	-0.0477	0.00
15	-0.0224	0.2072	0.0000	0.0315	-0.1068	-0.0588	-0.1024	0.00
16	-0.2671	0.1591	-0.0371	0.1863	0.0077	-0.3047	0.1216	0.00
17	0.2113	0.0364	-0.2769	0.2930	0.1713	-0.2444	-0.0947	0.00
18	-0.2760	-0.0954	0.1291	0.0294	-0.0810	0.0621	0.0159	0.00
19	0.2329	0.2154	0.1518	0.3039	0.0131	-0.1223	-0.0865	0.00
20	-0.0869	0.0990	-0.0982	0.2072	0.2741	0.3075	-0.2925	0.00
21	0.1961	-0.0543	-0.2997	0.2337	-0.3057	0.0790	0.2458	0.00
22	0.2454	-0.0537	-0.2534	0.0394	0.0350	-0.0233	-0.0112	0.00
23	-0.1643	0.1077	0.0971	-0.0120	-0.2972	-0.0272	-0.2072	0.00
24	0.1979	0.0857	0.1320	-0.0746	0.1229	-0.3051	-0.1956	0.00
25	0.1997	0.3123	-0.2747	-0.2926	0.2116	0.1920	-0.1743	0.00
26	0.0801	0.1044	0.0790	0.0057	-0.0530	-0.2426	0.0312	0.00
27	0.2949	0.0752	-0.0244	0.0619	-0.1694	0.1924	-0.2985	0.00
28	0.2176	0.0121	-0.3143	0.2608	0.1782	-0.2633	-0.2902	0.00
29	0.0555	-0.0997	-0.2567	-0.1896	0.1205	-0.1981	-0.0826	0.00

30	-0.0325	0.2854	0.0452	0.1428	-0.2145	0.0814	-0.1131	0.00
31	0.2081	-0.0415	-0.1857	0.2842	0.0955	0.0513	-0.1657	0.00
32	0.2955	0.2995	0.0227	-0.0764	-0.3118	-0.0929	-0.1662	0.00

Lampiran 16. Bobot dan Bias Awal (*Hidden Layer 2 ke Output Layer*) Model AQ

<i>H 2 to Output</i>	1	2	3	4
1	0.1044	-0.3590	-0.2386	-0.2443
2	-0.0321	-0.1074	0.3383	-0.2055
3	-0.3681	-0.3415	-0.3925	-0.2536
4	-0.0062	-0.2582	-0.0388	0.2614
5	0.2079	-0.0898	-0.3354	-0.0121
6	0.3675	0.3525	0.4048	0.2086
7	0.1392	-0.1937	0.0904	0.1007
8	0.1476	-0.2700	0.2971	-0.2464
9	0.1925	0.3193	-0.3777	0.1155
10	-0.3694	-0.0708	0.1688	-0.1406
11	-0.2031	-0.2687	0.3502	0.0086
12	-0.2350	0.0899	-0.0701	0.4070
13	-0.0547	0.2764	0.3773	-0.2969
14	0.1454	0.1490	0.0772	-0.1300
15	0.2168	0.1136	0.3572	-0.0054
16	-0.1248	0.3659	-0.0367	-0.4038
17	-0.0029	-0.0523	-0.1737	-0.2963
18	0.3619	0.0758	-0.1960	0.2192
19	0.0915	0.0578	0.2825	0.1355
20	-0.2851	0.1638	-0.2942	-0.4053
21	0.0727	-0.0152	-0.2851	0.1124
22	0.2570	0.2471	-0.1808	-0.3792
23	-0.0928	-0.2995	0.0619	0.1647
24	0.2318	-0.2123	0.3838	-0.2612
25	0.1276	-0.0830	-0.0730	-0.0400
26	0.2882	-0.3777	0.3267	0.2074
27	0.1160	-0.1106	0.3952	-0.0115
28	0.2813	-0.0035	0.2205	0.3120
29	0.3450	-0.1103	-0.3457	-0.1029
30	-0.0723	-0.2183	-0.2377	-0.3344
31	-0.2613	-0.0847	0.0687	0.0894
32	0.1652	0.1404	-0.1883	0.2180

Lampiran 17. Output (*Input ke Hidden Layer 1*) Model AQ Sebelum Aktivasi

<i>Input to H 1</i>	1	2	3	4	5	6	7	8
1	0.5419	-0.0676	0.5639	0.4531	0.5817	0.3595	0.5670	0.2588

2	-0.1590	-0.3528	-0.0649	-0.1063	-0.0503	-0.1648	-0.0595	-0.2480
3	0.7809	0.9654	0.7885	0.7698	0.7636	0.7311	0.8304	0.8476
4	0.0538	-0.2484	0.0274	0.1340	0.0413	-0.0199	-0.0541	0.0813
5	-0.3211	-0.2576	-0.3028	-0.3887	-0.3789	-0.3008	-0.5222	-0.1924
6	-0.0149	0.0496	-0.0544	0.0123	-0.0815	0.1360	-0.0830	0.2632
7	0.5239	0.1756	0.5677	0.4300	0.5195	0.5778	0.4446	0.5600
8	0.0136	0.0740	0.1691	-0.0904	0.1240	-0.0151	0.1568	-0.1428
9	-0.0218	-0.5138	-0.2276	-0.1602	-0.1817	-0.0936	-0.2543	-0.1945
10	-0.2961	-1.0003	-0.6097	-0.3510	-0.4589	-0.3851	-0.4070	-0.5803
11	0.2385	-0.4305	0.0536	0.1070	0.1363	0.2242	0.1966	0.0138
12	0.2866	0.2505	0.1378	0.1366	0.1881	0.2661	0.2577	0.0691
13	-0.6949	-0.5946	-0.7567	-0.5981	-0.6841	-0.5202	-0.6151	-0.6574
14	-0.4482	-0.5063	-0.5641	-0.3987	-0.4854	-0.4137	-0.4408	-0.5157
15	0.2134	0.9255	0.5128	0.2289	0.3503	0.1675	0.2652	0.4127
16	0.1744	-0.2890	-0.0268	0.0077	0.0337	0.1705	-0.0221	-0.0609
17	0.7294	0.1955	0.4686	0.6349	0.5852	0.8256	0.7156	0.5953
18	-0.4767	-0.2135	-0.4907	-0.4981	-0.4883	-0.4037	-0.4409	-0.4592
19	-0.3196	-0.5965	-0.2805	-0.4272	-0.2778	-0.2852	-0.3500	-0.5249
20	-0.5286	-0.3173	-0.4945	-0.4260	-0.5327	-0.6993	-0.6300	-0.4286
21	0.1770	-0.1827	0.2782	0.1499	0.2875	0.1918	0.2401	-0.0060
22	0.0925	0.2444	0.2523	0.0911	0.1468	-0.1970	0.0202	0.0928
23	-0.1343	-0.1206	-0.1038	-0.1167	-0.1330	-0.1907	-0.1665	-0.0747
24	0.0046	-0.2707	-0.1715	-0.0351	-0.1151	0.0600	-0.1751	-0.0469
25	-0.1252	0.1032	0.1345	-0.0684	0.0501	-0.1501	0.0261	-0.0552
26	0.0240	0.2494	0.1622	0.1066	0.1046	0.0722	0.0631	0.2000
27	-0.1018	0.1311	0.0263	-0.1842	-0.0328	-0.1487	-0.0437	-0.1841
28	0.3718	0.4413	0.6806	0.4336	0.5543	0.3566	0.4145	0.5245

Lampiran 18. Output (Input ke Hidden Layer 1) Model AQ Sesudah Aktivasi

<i>Input to H 1</i>	1	2	3	4	5	6	7	8
1	0.5419	0.0000	0.5639	0.4531	0.5817	0.3595	0.5670	0.2588
2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.7809	0.9654	0.7885	0.7698	0.7636	0.7311	0.8304	0.8476
4	0.0538	0.0000	0.0274	0.1340	0.0413	0.0000	0.0000	0.0813
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.0000	0.0496	0.0000	0.0123	0.0000	0.1360	0.0000	0.2632
7	0.5239	0.1756	0.5677	0.4300	0.5195	0.5778	0.4446	0.5600
8	0.0136	0.0740	0.1691	0.0000	0.1240	0.0000	0.1568	0.0000
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
11	0.2385	0.0000	0.0536	0.1070	0.1363	0.2242	0.1966	0.0138
12	0.2866	0.2505	0.1378	0.1366	0.1881	0.2661	0.2577	0.0691

13	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
14	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
15	0.2134	0.9255	0.5128	0.2289	0.3503	0.1675	0.2652	0.4127
16	0.1744	0.0000	0.0000	0.0077	0.0337	0.1705	0.0000	0.0000
17	0.7294	0.1955	0.4686	0.6349	0.5852	0.8256	0.7156	0.5953
18	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
19	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
20	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
21	0.1770	0.0000	0.2782	0.1499	0.2875	0.1918	0.2401	0.0000
22	0.0925	0.2444	0.2523	0.0911	0.1468	0.0000	0.0202	0.0928
23	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
24	0.0046	0.0000	0.0000	0.0000	0.0000	0.0600	0.0000	0.0000
25	0.0000	0.1032	0.1345	0.0000	0.0501	0.0000	0.0261	0.0000
26	0.0240	0.2494	0.1622	0.1066	0.1046	0.0722	0.0631	0.2000
27	0.0000	0.1311	0.0263	0.0000	0.0000	0.0000	0.0000	0.0000
28	0.3718	0.4413	0.6806	0.4336	0.5543	0.3566	0.4145	0.5245

Lampiran 19. *Output (Hidden Layer 1 ke Hidden Layer 2) Model AQ Sebelum Aktivasi*

<i>H 1 to H 2</i>	1	2	3	4	5	6	7	8
1	0.1264	0.2598	0.1187	0.0739	0.1179	0.0890	0.1507	0.1013
2	0.1118	0.2321	0.0251	0.1144	0.0552	0.0534	0.1268	0.1285
3	0.2944	-0.0402	0.0623	0.2245	0.1708	0.3449	0.2709	0.1440
4	0.0648	-0.1028	0.0338	0.0879	0.0695	0.1324	0.1441	0.1003
5	-0.0835	0.0968	0.0991	-0.0675	0.0026	-0.0758	-0.1113	0.0721
6	0.1385	0.1688	0.1507	0.0898	0.1444	0.1822	0.1418	0.1680
7	-0.0842	-0.0412	-0.1419	-0.0425	-0.1185	-0.0660	-0.0879	0.0189
8	-0.0510	-0.0622	0.0404	-0.0386	0.0034	-0.0602	-0.0642	-0.0235
9	0.0610	0.2336	0.0982	0.0471	0.0801	0.0198	0.0917	0.1108
10	-0.7110	-0.6430	-0.8460	-0.6790	-0.7649	-0.6291	-0.6551	-0.7521
11	0.0066	-0.4493	0.0208	-0.0526	0.0444	-0.0117	0.0062	-0.1946
12	-0.0458	-0.0993	-0.1287	-0.0506	-0.0836	0.0432	-0.0355	0.0177
13	-0.2218	0.0218	-0.1754	-0.2105	-0.2113	-0.3055	-0.2615	-0.2753
14	-0.1651	-0.0859	-0.0988	-0.1311	-0.1458	-0.1448	-0.1766	-0.0344
15	-0.1509	-0.2772	-0.1417	-0.1160	-0.1219	-0.1649	-0.1484	-0.1997
16	-0.4978	-0.1717	-0.3535	-0.3707	-0.3994	-0.4763	-0.4558	-0.3547
17	0.1692	0.2711	0.2516	0.2636	0.2022	0.2253	0.1639	0.3631
18	-0.3147	-0.3491	-0.4370	-0.3108	-0.3577	-0.2448	-0.2870	-0.3343
19	-0.2849	-0.1475	-0.1146	-0.2230	-0.1711	-0.2837	-0.1757	-0.2153
20	-0.5755	-0.4772	-0.7884	-0.6027	-0.7317	-0.5901	-0.7018	-0.6964
21	0.0166	0.1964	0.0698	-0.0538	0.0157	0.0237	-0.0027	0.0170
22	0.6169	0.2372	0.5615	0.5011	0.5672	0.5708	0.5752	0.4906

29	0.4047	0.0360	0.1244	0.2868	0.2490	0.4359	0.3628	0.1988
30	0.0255	0.2566	0.0000	0.0000	0.0000	0.0207	0.0000	0.0447
31	0.0000	0.2358	0.0000	0.0000	0.0000	0.0000	0.0000	0.0074
32	0.3348	0.0000	0.1018	0.1670	0.2019	0.2512	0.2703	0.0131

Lampiran 21. Hasil Prediksi Model AQ (y_k)

Data Training				
No.	Prediksi	Decoding	Kelas	Label
1	[0.3837, 0.2363, 0.1844, 0.1956]	[1, 0, 0, 0]	0	R
2	[0.0016, 0.3998, 0.0007, 0.5979]	[0, 0, 0, 1]	3	SL
3	[0.0001, 0.0613, 0.2445, 0.6942]	[0, 0, 0, 1]	3	SL
4	[0.0000, 0.0428, 0.2173, 0.7398]	[0, 0, 0, 1]	3	SL
5	[0.0019, 0.2140, 0.5560, 0.2280]	[0, 0, 1, 0]	2	L
6	[0.2278, 0.4983, 0.2141, 0.0599]	[0, 1, 0, 0]	1	S
7	[0.2200, 0.7193, 0.0548, 0.0060]	[0, 1, 0, 0]	1	S
8	[0.0000, 0.0139, 0.0793, 0.9068]	[0, 0, 0, 1]	3	SL
9	[0.9842, 0.0136, 0.0020, 0.0001]	[1, 0, 0, 0]	0	R
10	[0.0142, 0.5171, 0.4105, 0.1324]	[0, 1, 0, 0]	3	SL
⋮	⋮	⋮	⋮	⋮
422	[0.2580, 0.1435, 0.0524, 0.7842]	[0, 0, 0, 1]	3	SL
Data Testing				
No.	Prediksi	Decoding	Kelas	Label
1	[0.0003, 0.0207, 0.1034, 0.8756]	[0, 0, 0, 1]	3	SL
2	[0.0008, 0.0363, 0.1635, 0.7994]	[0, 0, 0, 1]	3	SL
3	[0.9970, 0.0026, 0.0003, 0.0000]	[1, 0, 0, 0]	0	R
4	[1.0000, 0.0000, 0.0000, 0.0000]	[1, 0, 0, 0]	0	R
5	[0.4091, 0.5559, 0.2965, 0.0384]	[0, 1, 0, 0]	1	S
6	[0.0033, 0.9900, 0.6245, 0.5821]	[0, 1, 0, 0]	1	S
7	[0.0008, 0.0363, 0.1635, 0.7994]	[0, 0, 0, 1]	3	SL
8	[0.9053, 0.0394, 0.0128, 0.0475]	[1, 0, 0, 0]	0	R
9	[0.5018, 0.0598, 0.0209, 0.0175]	[1, 0, 0, 0]	0	R
10	[0.0606, 0.1551, 0.0760, 0.4182]	[0, 0, 0, 1]	3	SL
⋮	⋮	⋮	⋮	⋮
106	[0.6677, 0.2167, 0.1054, 0.0102]	[1, 0, 0, 0]	0	R

Lampiran 22. Bobot dan Bias Akhir (*Input ke Hidden Layer 1*) Model AQ.

Input to H1	1	2	3	4	5	6	7	8	Bias
1	0.27	-0.30	0.13	0.54	0.58	-0.05	0.03	-0.24	0.06
2	-0.15	-0.40	0.16	0.39	-0.05	-0.09	-0.26	-0.15	0.07
3	-0.04	0.38	0.09	0.25	0.32	0.33	0.06	0.11	-0.03
4	-0.11	-0.39	-0.16	0.23	-0.33	0.24	-0.66	0.31	0.09

5	0.49	-0.18	-0.15	0.37	-0.08	-0.29	0.08	0.29	0.09
6	0.07	0.55	-0.16	0.53	0.08	-0.24	0.68	0.38	0.14
7	0.47	0.10	-0.01	0.17	-0.44	0.44	0.08	0.29	0.21
8	0.35	-0.02	0.36	-0.16	0.17	-0.15	-0.26	-0.26	0.00
9	0.22	-0.21	-0.36	0.06	-0.03	-0.03	0.07	-0.16	0.00
10	-0.25	-0.10	-0.39	0.23	0.24	-0.38	0.37	-0.38	0.00
11	0.13	0.12	-0.18	0.28	0.19	-0.24	-0.23	-0.34	0.00
12	0.37	0.23	-0.09	0.13	0.10	0.39	-0.12	-0.39	0.07
13	-0.34	0.09	-0.09	-0.04	-0.29	-0.20	0.10	-0.18	0.00
14	-0.29	-0.04	-0.15	-0.04	-0.05	-0.11	0.28	-0.19	0.00
15	0.32	-0.22	0.38	-0.66	-0.14	0.42	-0.10	0.51	0.03
16	0.25	-0.41	-0.33	-0.17	-0.49	0.56	0.15	-0.25	0.10
17	0.05	0.73	-0.27	0.65	-0.19	0.31	-0.89	-0.39	0.05
18	0.07	0.15	0.02	-0.29	-0.05	-0.28	-0.34	-0.06	0.00
19	0.30	-0.37	0.04	-0.01	-0.28	0.01	-0.02	-0.41	0.00
20	-0.24	-0.36	-0.06	0.17	0.37	-0.56	-0.17	0.32	0.02
21	0.34	-0.41	0.16	0.66	-0.06	0.32	-0.11	-0.26	0.14
22	0.20	-0.43	0.19	0.43	0.63	-0.30	0.50	0.29	-0.01
23	0.13	0.07	0.01	0.55	0.38	-0.40	0.35	0.17	0.12
24	0.02	-0.27	-0.37	-0.46	-0.30	0.60	0.36	0.07	0.06
25	0.20	0.01	0.40	0.58	0.36	-0.40	0.52	0.13	0.20
26	0.07	-0.02	0.13	-0.05	-0.33	0.16	-0.73	0.36	0.03
27	0.30	-0.11	0.25	-0.30	0.07	-0.09	-0.24	-0.13	-0.03
28	0.10	-0.42	0.33	-0.25	-0.43	0.33	-0.57	0.34	-0.07

Lampiran 23. Bobot dan Bias Akhir (*Hidden Layer 1 ke Hidden Layer 2*) Model

AQ.

<i>H1 to H2</i>	1	2	3	4	5	6	7
1	0.5760	0.2454	0.1773	-0.0877	0.0014	0.1774	-0.1772
2	0.1774	-0.0546	0.0809	-0.0041	-0.2027	0.1111	-0.3059
3	0.2762	0.1272	0.2186	0.0444	0.2211	-0.0915	0.0272
4	-0.0384	-0.1467	0.1583	-0.0841	-0.2115	0.1227	-0.0916
5	0.5006	0.0272	-0.4107	0.0734	-0.2176	0.2025	0.0271
6	0.0488	0.3663	0.2095	-0.2026	0.0117	0.3357	0.0734
7	-0.0258	-0.1683	-0.0542	0.2592	0.3885	-0.0591	0.2131
8	0.5358	0.5820	-0.3847	0.0281	0.0998	-0.0490	0.1402
9	0.1907	0.0403	0.2102	-0.2127	0.2275	0.1964	-0.0468
10	-0.3112	0.2314	-0.1573	-0.2577	-0.2234	-0.0615	-0.2741
11	0.8330	0.4190	-0.2977	0.0512	-0.1526	0.1695	-0.1398
12	-0.4265	-0.4692	0.0929	0.3429	0.2299	0.1130	0.3458
13	0.1486	-0.1939	-0.0124	-0.1998	0.3357	-0.0144	-0.2272

14	0.2502	0.4005	0.0443	-0.1276	-0.1282	0.3191	0.1738
15	0.3538	0.0564	-0.1365	-0.0219	0.2924	0.1596	-0.0410
16	-0.2143	-0.0413	-0.2640	0.1224	-0.0369	-0.1361	-0.1162
17	-0.0589	-0.0111	0.1723	0.0285	-0.0282	0.1088	0.2113
18	-0.1456	0.0077	-0.0148	0.0258	0.1046	0.2844	-0.1943
19	0.7277	0.7137	0.0534	0.1387	-0.0836	0.3236	-0.1036
20	-0.1482	-0.1574	-0.2176	0.0476	-0.2499	-0.3086	-0.1233
21	0.0155	-0.1252	-0.2868	0.0181	0.3868	0.1421	-0.1058
22	0.4446	0.0904	0.0555	-0.2330	0.2592	0.0454	0.2908
23	-0.0989	-0.0032	0.0354	0.0766	-0.2452	0.0875	-0.1636
24	0.3174	0.5248	0.0840	-0.1308	0.0210	0.3347	0.0050
25	0.2664	0.0652	-0.1488	0.1006	0.2359	0.2349	-0.1349
26	0.3522	0.3375	0.2771	0.2026	0.2727	-0.1530	0.0072
27	0.4439	0.2681	-0.1182	0.2530	-0.0915	0.2775	-0.2287
28	-0.2717	-0.0424	-0.1060	0.4633	-0.0258	-0.3909	0.1430
29	0.0790	0.5625	0.1676	0.0786	0.2825	-0.0616	0.1260
30	0.2049	0.0958	0.0367	-0.1565	-0.0747	0.0891	-0.0240
31	-0.2997	-0.4243	0.0039	0.6499	0.0548	0.3286	0.4290
32	0.5545	0.5559	0.1901	0.1974	0.3968	0.2278	-0.0567

<i>H 1</i> <i>to</i> <i>H 2</i>	8	9	10	11	12	13	14
1	0.2062	0.0189	-0.1416	0.0828	0.1851	0.2088	-0.0086
2	-0.0140	-0.2964	0.2281	-0.1812	0.1566	0.0209	0.0738
3	-0.0099	0.0707	0.1530	0.2874	0.1214	-0.3105	0.1042
4	0.2227	0.2615	0.1326	-0.1538	0.0206	0.1047	0.1036
5	-0.0556	-0.2547	-0.1962	0.0348	-0.1541	0.2139	-0.1709
6	0.2471	0.2587	0.1063	0.2842	0.4641	0.2946	-0.1409
7	-0.2447	-0.3085	0.1627	-0.2250	-0.3913	0.0904	-0.0785
8	-0.0954	-0.0413	-0.2159	-0.1339	0.1739	-0.2958	-0.1573
9	0.0038	0.2830	-0.2240	-0.0315	-0.0494	0.0350	-0.1863
10	0.1188	0.2838	0.2700	0.2440	-0.1791	0.3061	-0.2388
11	0.2284	-0.1438	-0.0493	0.2746	-0.0132	0.2310	0.1525
12	0.1086	0.0583	-0.0293	0.2079	-0.4919	-0.1041	0.0626
13	-0.2146	0.2407	-0.2845	-0.2559	-0.4736	0.1392	-0.1637
14	-0.1119	0.1950	0.2404	-0.0238	0.3682	-0.2412	0.2174
15	-0.2026	0.2205	-0.0924	-0.0868	0.2953	-0.0717	-0.1654
16	-0.2877	-0.1198	0.0505	-0.1971	0.0824	0.2858	-0.0979
17	-0.2738	0.2405	-0.2141	-0.1667	0.5014	-0.0098	-0.2943
18	0.0163	0.2516	0.2217	0.2368	0.1659	-0.1165	0.3096
19	0.2904	0.0905	0.1790	-0.3086	0.5533	-0.3144	-0.2297
20	-0.2966	-0.2240	-0.0081	0.0602	0.2322	0.2693	0.1872

21	0.2135	0.2746	-0.1636	-0.0485	0.1263	0.2958	0.0114
22	0.1475	-0.0684	0.1664	0.2438	0.5248	0.2220	0.1872
23	0.0501	-0.1308	0.1820	-0.0603	0.2169	0.1679	-0.1857
24	-0.0001	-0.1945	-0.1333	-0.0139	0.3391	0.1385	-0.1935
25	-0.1661	-0.2915	0.2673	-0.2207	0.4413	0.2732	0.2780
26	-0.1167	-0.0532	-0.0148	-0.1886	0.0823	0.2266	0.1064
27	-0.1328	-0.1747	0.0218	0.1164	0.1541	0.1484	-0.3005
28	0.0406	-0.2231	-0.0853	0.0491	-0.2729	0.1707	0.0819
29	0.2446	-0.2133	0.0518	0.3066	0.7238	-0.2405	0.2038
30	-0.2537	-0.2794	-0.3080	-0.2953	0.8617	0.1824	-0.1875
31	0.2429	-0.1194	0.0738	-0.1486	-0.1795	0.2936	0.3112
32	0.2546	0.0633	0.2920	0.3037	0.2446	-0.1177	-0.1284

<i>H 1</i> to <i>H 2</i>	15	16	17	18	19	20	21
1	0.1829	0.1733	0.1014	0.2502	-0.0403	0.1554	0.3239
2	0.2088	-0.1483	0.2753	0.0424	0.1795	-0.0597	-0.2366
3	-0.3337	0.0565	0.4816	-0.1876	-0.3151	0.0455	0.3736
4	-0.3298	-0.0518	0.1220	0.1111	-0.1651	-0.2660	0.0158
5	0.2194	0.4741	-0.5851	0.1117	-0.1813	0.1328	0.1958
6	0.0146	-0.1299	0.7889	0.0116	-0.2687	0.4127	0.2038
7	0.4735	0.1754	-0.1908	0.3110	0.1376	-0.3816	0.0683
8	0.0288	-0.1157	0.0364	0.0452	0.2679	0.4917	0.2130
9	0.1872	0.2033	-0.5189	0.3105	-0.1350	-0.0205	0.1905
10	-0.2341	-0.1740	-0.1300	0.0300	-0.1185	-0.0130	-0.0264
11	-0.3235	0.0219	0.2418	-0.0677	0.2040	-0.0367	0.3882
12	0.2406	0.3843	-0.5681	-0.2231	0.3006	-0.4056	-0.0352
13	0.1790	0.1319	-0.2968	-0.0045	0.1468	0.4196	0.1602
14	0.0241	-0.3130	0.5460	0.2495	0.0578	0.3560	-0.1128
15	-0.0448	-0.3790	0.4582	0.1864	0.0029	0.2867	0.3539
16	0.1496	-0.2673	-0.2303	0.3030	-0.1391	-0.2433	0.1357
17	-0.0876	-0.6234	0.8809	0.0145	-0.3013	0.1844	0.2900
18	-0.2676	-0.0997	-0.2067	0.1026	-0.0426	0.3131	0.1023
19	-0.3273	-0.5445	0.2243	0.1394	-0.2786	-0.0685	0.5494
20	-0.2129	0.2953	-0.2513	-0.2060	0.0595	-0.2782	-0.2951
21	0.2952	0.5414	-0.2409	0.2380	0.1562	0.1633	0.1394
22	-0.1297	-0.3136	0.8520	-0.2888	-0.0746	0.0684	-0.0762
23	-0.0998	-0.1705	-0.2253	-0.0529	0.1313	0.2559	0.0487
24	0.1440	-0.1773	0.4586	-0.1492	-0.0932	0.3573	0.0286
25	-0.0529	-0.3332	0.0080	0.1927	-0.0810	0.4435	0.4953
26	0.2341	0.2283	-0.3253	0.1965	0.1893	0.1665	0.4383
27	-0.0948	-0.4862	0.5518	-0.1863	0.0568	-0.0526	0.5264

28	0.3531	0.5985	-0.8247	0.0834	0.1330	-0.3198	0.2705
29	-0.4421	-0.2476	0.5930	-0.0969	-0.0628	0.5370	0.3788
30	0.1361	-0.2223	0.6234	0.1360	0.2953	0.4729	-0.0983
31	0.6196	0.4810	-0.6437	0.1174	0.2421	-0.2213	-0.3767
32	-0.2631	0.4928	-0.2585	-0.2413	0.0986	0.0941	0.3905

<i>H 1</i> <i>to</i> <i>H 2</i>	22	23	24	25	26	27	28	Bias
1	-0.1118	0.4660	-0.1771	0.2163	-0.1431	-0.1213	-0.3916	-0.0245
2	0.2186	-0.3236	-0.2466	-0.2321	-0.0696	-0.1868	-0.3174	-0.0234
3	0.0361	0.0251	0.1827	-0.0142	-0.1433	-0.2350	-0.2583	-0.0013
4	-0.3133	-0.1957	-0.0867	0.3020	0.0460	-0.2384	0.0384	-0.0279
5	0.2085	0.2878	0.0413	0.4553	0.1143	-0.0955	0.0772	-0.0437
6	-0.1132	0.4529	-0.0207	0.2502	0.1107	-0.2351	-0.1806	0.0415
7	-0.1076	-0.4933	0.0505	-0.5090	0.1149	-0.1512	-0.0351	0.0646
8	-0.1650	0.1376	-0.3414	0.4108	0.1417	0.0480	-0.0036	-0.0260
9	0.3826	0.4582	0.2488	0.1384	-0.2913	-0.3053	-0.2925	-0.0008
10	-0.2969	0.1961	0.2355	0.1039	-0.2787	0.2892	-0.1550	0.0000
11	0.2634	-0.0114	-0.0092	0.3094	-0.0785	-0.2422	0.1273	-0.0224
12	-0.1013	-0.0050	0.5912	-0.4172	0.1474	-0.2698	0.1551	0.0998
13	0.4576	0.4122	0.1482	0.2614	-0.4031	0.3029	-0.3885	0.0160
14	-0.1609	0.3240	0.2449	0.6342	-0.2647	0.0548	-0.1373	0.0858
15	0.1053	0.5640	-0.2563	0.4444	-0.1951	-0.0366	-0.2386	0.1204
16	-0.2862	0.1591	-0.0590	0.1863	-0.0118	-0.3047	0.1011	-0.0202
17	0.1596	0.3485	-0.5370	0.7018	0.0726	-0.2322	-0.2694	0.0297
18	-0.2760	-0.0954	0.1291	0.0294	-0.0810	0.0621	0.0159	0.0000
19	0.2161	0.3274	-0.1256	0.3792	0.1983	-0.1275	-0.1330	0.0529
20	-0.0869	0.0990	-0.0982	0.2072	0.2741	0.3075	-0.2925	0.0000
21	0.5409	0.1244	-0.2330	0.4278	-0.2757	0.0867	0.3200	0.0456
22	0.2019	0.2223	-0.4857	0.4237	-0.0811	-0.0102	-0.1818	0.0114
23	-0.1643	0.1077	0.0971	-0.0120	-0.2972	-0.0272	-0.2072	0.0000
24	-0.0041	0.1124	0.0186	-0.0137	0.1310	-0.2916	-0.2548	-0.0046
25	0.3497	0.6412	-0.5408	0.0473	0.1418	0.2043	-0.2743	0.0531
26	0.0900	0.0217	0.1099	-0.1618	0.0328	-0.2330	0.1197	-0.0005
27	0.2527	0.1474	-0.3614	0.1028	0.0660	0.1962	-0.4035	-0.0112
28	0.4398	-0.1364	-0.1504	0.0523	0.3348	-0.2541	-0.0997	0.0296
29	0.2066	0.2491	-0.5628	0.1527	0.0291	-0.1682	-0.1981	0.0565
30	-0.0415	0.6052	-0.2022	0.5540	-0.3432	0.0940	-0.3087	0.0471
31	0.2549	-0.3116	0.3099	-0.0600	0.3914	0.0381	0.1605	0.1468
32	0.6876	0.6869	-0.0174	0.3963	-0.3205	-0.0833	-0.1440	0.0511

Lampiran 24. Bobot dan Bias Akhir (*Hidden Layer 2 ke Output Layer*) Model AQ

<i>H 2 to Output</i>	1	2	3	4
1	0.2160	0.1890	0.1292	0.0986
2	-0.0521	-0.1254	0.3327	-0.1857
3	0.1931	-0.6193	-0.8634	-0.2833
4	0.0529	-0.3349	-0.0588	0.1644
5	-0.0344	-0.1539	-0.3997	0.2167
6	0.4502	0.4525	0.4644	-0.0195
7	-0.3687	-0.1591	0.2680	0.2385
8	0.6036	-0.8909	0.1425	-0.1758
9	0.0892	0.3132	-0.5177	0.2771
10	-0.3694	-0.0708	0.1688	-0.1406
11	0.3586	-0.8031	-0.2071	-0.3072
12	-0.7060	0.1259	0.0666	0.5508
13	0.0704	0.5017	-0.0317	-0.7606
14	0.2906	0.3610	0.0482	-0.5713
15	0.6121	0.2153	-0.1145	-0.5111
16	-0.1412	0.3496	-0.0531	-0.3871
17	0.2202	-0.0118	-0.0903	-0.6669
18	0.3619	0.0758	-0.1960	0.2192
19	0.6842	-0.1883	0.0000	-0.4053
20	-0.2851	0.1638	-0.2942	-0.4053
21	-0.0170	-0.0703	-0.4275	0.3161
22	0.5101	0.2920	-0.1437	-0.7764
23	-0.0928	-0.2995	0.0619	0.1647
24	0.2087	-0.1055	0.4683	-0.3969
25	0.4407	-0.1579	-0.4849	-0.0875
26	-0.0093	-0.3686	0.3938	0.3204
27	0.8751	-0.6929	-0.0107	-0.2754
28	-0.3184	-0.1467	0.1197	0.7494
29	0.9680	-0.2283	-0.7440	-0.6451
30	0.1820	-0.0530	-0.4063	-0.8831
31	-0.6360	-0.0358	0.1547	0.2036
32	0.3370	0.0987	-0.5892	0.3519
Bias	-0.0632	0.0126	0.0162	0.0202

Lampiran 25. Percobaan Beberapa Model Ketiga.

Model	Arsitektur					Loss
	Input	Hidden Layer 1	Hidden Layer 2	Hidden Layer 3	Output	
AS	8	4	4	4	4	0.5645
AT	8	4	4	8	4	0.6087
AU	8	4	4	12	4	0.5228

AV	8	4	4	16	4	0.5332
AW	8	4	4	20	4	0.5483
AX	8	4	4	24	4	0.4614
AY	8	4	4	28	4	0.524
AZ	8	4	4	32	4	0.5046
BA	8	4	8	8	4	0.5727
BB	8	4	8	12	4	0.4965
BC	8	4	8	16	4	0.5347
BD	8	4	8	20	4	0.4941
BE	8	4	8	24	4	0.5501
BF	8	4	8	28	4	0.5137
BG	8	4	8	32	4	0.4591
BH	8	4	12	12	4	0.5063
BI	8	4	12	16	4	0.4831
BJ	8	4	12	20	4	0.4497
BK	8	4	12	24	4	0.5628
BL	8	4	12	28	4	0.5336
BM	8	4	12	32	4	0.4931
BN	8	4	16	16	4	0.6073
BO	8	4	16	20	4	0.4693
BP	8	4	16	24	4	0.4444
BQ	8	4	16	28	4	0.5132
BR	8	4	16	32	4	0.5081
BS	8	4	20	20	4	0.4973
BT	8	4	20	24	4	0.4756
BU	8	4	20	28	4	0.4678
BV	8	4	20	32	4	0.4723
BW	8	4	24	24	4	0.6183
BX	8	4	24	28	4	0.4562
BY	8	4	24	32	4	0.4751
BZ	8	4	28	28	4	0.5256
CA	8	4	28	32	4	0.5328
CB	8	8	8	8	4	0.4812
CC	8	8	8	12	4	0.4762
CD	8	8	8	16	4	0.4216
CE	8	8	8	20	4	0.4857
CF	8	8	8	24	4	0.4516
CG	8	8	8	28	4	0.5039
CH	8	8	8	32	4	0.4942
CI	8	8	12	12	4	0.482
CJ	8	8	12	16	4	0.4551

CK	8	8	12	20	4	0.4654
CL	8	8	12	24	4	0.4915
CM	8	8	12	28	4	0.4758
CN	8	8	12	32	4	0.4666
CO	8	8	16	16	4	0.5029
CP	8	8	16	20	4	0.5166
CQ	8	8	16	24	4	0.4661
CR	8	8	16	28	4	0.4698
CS	8	8	16	32	4	0.4814
CT	8	8	20	20	4	0.5501
CU	8	8	20	24	4	0.4782
CV	8	8	20	28	4	0.5053
CW	8	8	20	32	4	0.4114
CX	8	8	24	24	4	0.5354
CY	8	8	24	28	4	0.5944
CZ	8	8	24	32	4	0.4721
DA	8	8	28	28	4	0.4717
DB	8	8	28	32	4	0.4241
DC	8	12	12	12	4	0.4697
DD	8	12	12	16	4	0.4621
DE	8	12	12	20	4	0.5042
DF	8	12	12	24	4	0.4456
DG	8	12	12	28	4	0.4863
DH	8	12	12	32	4	0.5024
DI	8	12	16	16	4	0.4395
DJ	8	12	16	20	4	0.4301
DK	8	12	16	24	4	0.4777
DL	8	12	16	28	4	0.4511
DM	8	12	16	32	4	0.4559
DN	8	12	20	20	4	0.4319
DO	8	12	20	24	4	0.4733
DP	8	12	20	28	4	0.4934
DQ	8	12	20	32	4	0.363
DR	8	12	24	24	4	0.4321
DS	8	12	24	28	4	0.4309
DT	8	12	24	32	4	0.4463
DU	8	12	28	28	4	0.5004
DV	8	12	28	32	4	0.5031
DW	8	16	16	16	4	0.4633
DX	8	16	16	20	4	0.4153
DY	8	16	16	24	4	0.4306

DZ	8	16	16	28	4	0.4568
EA	8	16	16	32	4	0.4922
EB	8	16	20	20	4	0.4965
EC	8	16	20	24	4	0.4659
ED	8	16	20	28	4	0.4478
EE	8	16	20	32	4	0.4829
EF	8	16	24	24	4	0.4581
EG	8	16	24	28	4	0.4408
EH	8	16	24	32	4	0.4562
EI	8	16	28	28	4	0.4855
EJ	8	16	28	32	4	0.4473
EK	8	20	20	20	4	0.4837
EL	8	20	20	24	4	0.4643
EM	8	20	20	28	4	0.4532
EN	8	20	20	32	4	0.4747
EO	8	20	24	24	4	0.4363
EP	8	20	24	28	4	0.4886
EQ	8	20	24	32	4	0.4132
ER	8	20	28	28	4	0.4887
ES	8	20	28	32	4	0.4018
ET	8	24	24	24	4	0.4606
EU	8	24	24	28	4	0.4015
EV	8	24	24	32	4	0.4297
EW	8	24	28	32	4	0.4823
EX	8	28	28	28	4	0.4147
EY	8	28	28	32	4	0.3894
EZ	8	32	32	32	4	0.4778

Lampiran 26. Bobot dan Bias Awal (*Input ke Hidden Layer 1*) Model DQ.

<i>Input to H 1</i>	1	2	3	4	5	6	7	8	Bias
1	-0.37	0.48	-0.49	0.51	0.32	-0.33	0.04	-0.19	0.00
2	0.51	0.51	0.15	-0.36	0.45	-0.02	-0.51	-0.24	0.00
3	0.39	-0.17	0.37	0.02	0.42	0.36	0.40	-0.38	0.00
4	0.52	0.01	-0.09	-0.51	0.37	-0.30	0.33	0.18	0.00
5	0.53	0.02	-0.49	-0.18	-0.30	-0.19	-0.07	0.19	0.00
6	0.44	-0.34	0.24	-0.48	-0.46	0.28	0.10	-0.10	0.00
7	0.53	-0.13	-0.38	-0.09	-0.12	0.22	-0.28	-0.11	0.00
8	0.02	-0.42	0.31	-0.29	-0.31	-0.43	0.27	-0.24	0.00
9	-0.54	0.16	0.00	0.16	0.33	0.49	0.04	0.22	0.00
10	0.34	0.21	-0.04	0.13	-0.20	-0.48	-0.30	0.27	0.00
11	-0.38	-0.18	-0.47	-0.51	0.27	-0.46	-0.20	0.16	0.00

12	-0.50	-0.35	0.47	0.05	0.33	0.47	-0.47	-0.51	0.00
----	-------	-------	------	------	------	------	-------	-------	------

Lampiran 27. Bobot dan Bias Awal (*Hidden Layer 1 ke Hidden Layer 2*) Model DQ.

<i>H 1 to H 2</i>	1	2	3	4	5	6
1	-0.4264	0.4026	-0.1359	-0.1123	-0.2371	-0.2014
2	-0.2019	0.3125	-0.2488	-0.1335	0.1359	-0.3552
3	0.2700	0.0637	0.3841	0.0637	0.4099	0.1833
4	-0.4062	0.3172	-0.2839	-0.3270	0.1392	0.1478
5	-0.0796	-0.1609	-0.2417	-0.3713	-0.0196	-0.3753
6	-0.3882	0.3451	0.0029	0.3797	0.0171	-0.0532
7	-0.2668	-0.2007	-0.3243	0.0767	-0.0450	0.1290
8	-0.2561	0.2348	0.3598	-0.1741	0.0927	0.4048
9	-0.0898	0.3063	-0.1478	-0.1758	0.0321	-0.2514
10	-0.2816	-0.3468	0.1007	-0.2234	0.2288	0.4006
11	0.3060	0.3368	-0.0860	0.1911	0.1017	0.2665
12	0.3472	-0.1217	-0.3781	0.1065	0.1262	0.0779
13	-0.0240	-0.0913	-0.1970	0.0755	0.1031	0.1316
14	-0.3227	0.2926	-0.2091	-0.4307	-0.0917	0.2017
15	0.2830	0.2129	0.2532	0.0136	-0.0715	-0.0204
16	0.1626	-0.2177	-0.3565	0.3797	-0.2267	0.1460
17	0.1464	-0.3526	-0.0950	-0.0610	-0.3661	0.2677
18	-0.1733	-0.2758	-0.3244	0.0063	-0.2581	-0.3377
19	0.1144	0.3688	-0.3223	-0.0994	-0.0154	-0.0172
20	-0.2320	-0.1554	-0.1300	-0.2561	0.0701	0.0736

<i>H 1 to H 2</i>	7	8	9	10	11	12	Bias
1	-0.1756	0.3112	-0.4186	0.3095	0.1987	-0.4033	0.0000
2	-0.2121	-0.1289	0.0572	0.1304	0.3458	-0.3499	0.0000
3	-0.1654	0.2786	0.0387	0.0194	0.1372	0.2631	0.0000
4	-0.2404	0.1338	0.0677	-0.0113	0.3397	0.3615	0.0000
5	-0.0091	-0.4292	-0.2521	0.3300	-0.0710	-0.0058	0.0000
6	-0.1065	-0.1753	-0.3081	-0.4064	-0.0946	-0.2628	0.0000
7	0.2006	-0.0508	-0.3773	0.0442	-0.0804	0.1381	0.0000
8	-0.2044	-0.3127	0.1287	-0.1561	0.2339	-0.2727	0.0000
9	0.0739	-0.3504	0.1232	-0.3760	0.1603	0.4318	0.0000
10	-0.0281	0.1484	0.3686	-0.0251	-0.1246	0.0757	0.0000
11	-0.0787	0.2077	0.4239	-0.3918	-0.2788	0.1107	0.0000
12	0.4129	0.2894	-0.0348	-0.0774	0.1385	-0.2652	0.0000
13	0.0348	0.0541	-0.0170	0.0075	-0.0083	0.2172	0.0000
14	-0.0813	0.3542	0.2877	-0.0189	0.1556	-0.0350	0.0000

15	0.4162	0.0055	0.2010	-0.4117	-0.2968	0.0688	0.0000
16	0.2764	-0.2181	0.0837	0.0260	-0.0266	-0.4127	0.0000
17	-0.1410	0.4158	0.2036	-0.3393	-0.0819	-0.2774	0.0000
18	-0.4110	0.1511	0.0448	-0.2342	-0.1836	-0.3269	0.0000
19	-0.0670	0.2131	0.3201	-0.3823	-0.1435	0.0176	0.0000
20	0.1332	-0.2060	0.4326	-0.3994	-0.0511	0.2634	0.0000

Lampiran 28. Bobot dan Bias Awal (*Hidden Layer 2 ke Hidden Layer 3*) Model

DQ.

<i>H 2 to H 3</i>	1	2	3	4	5	6	7
1	-0.2882	0.2594	-0.1265	0.2852	-0.0323	0.0588	0.0771
2	0.0298	-0.2801	-0.0065	0.0885	-0.0019	-0.0665	0.0987
3	-0.2871	-0.2403	0.0025	-0.0796	0.3311	-0.2846	0.2066
4	-0.0527	0.2434	0.1438	0.3091	0.3155	0.0501	-0.1146
5	-0.1786	-0.0143	-0.0701	-0.0878	0.3246	0.1530	-0.0812
6	0.2783	-0.2174	-0.2581	-0.1645	-0.2622	-0.2085	0.1948
7	-0.3296	0.2512	0.1932	0.1843	-0.0861	0.1210	0.0831
8	0.0251	0.2228	0.1247	-0.3259	-0.3096	-0.0777	-0.1172
9	-0.0954	-0.0588	0.0066	0.2286	-0.0920	-0.0469	0.2893
10	0.1929	-0.1300	0.1514	0.1247	-0.3185	-0.0807	0.3327
11	0.2722	0.0387	-0.2371	-0.2655	-0.2828	0.2396	-0.0051
12	0.0881	0.0556	-0.3158	-0.0834	-0.2101	-0.1259	-0.0819
13	-0.1318	-0.2959	0.2853	-0.0040	-0.1197	-0.0649	-0.1448
14	0.1872	0.1794	-0.1374	0.1627	-0.0918	0.3286	-0.2197
15	0.0839	-0.0094	-0.0334	-0.3339	-0.2431	-0.0151	-0.1741
16	0.0082	-0.0725	-0.3359	0.1602	0.1533	-0.0496	-0.2842
17	0.1691	0.1263	0.0050	0.1983	0.0457	-0.0883	0.0258
18	0.1900	-0.0777	0.0686	-0.1939	-0.3080	0.2803	-0.2495
19	0.1048	0.2114	0.2093	0.0813	-0.1807	-0.2138	-0.0624
20	-0.2777	-0.1677	0.2875	0.1919	0.2845	-0.1126	-0.1133
21	0.2179	-0.0047	0.0449	0.2485	0.2480	-0.2659	-0.1226
22	0.2107	-0.2756	0.3269	0.3124	-0.0297	0.0786	0.0217
23	0.3380	-0.2114	-0.0259	0.1698	0.1756	0.1519	0.0575
24	0.0258	-0.1766	-0.0749	-0.2379	0.0155	-0.2055	-0.2366
25	-0.2619	0.1739	0.1374	0.1947	0.2360	0.3048	-0.2482
26	-0.0874	-0.2930	0.3139	-0.2920	-0.1420	-0.3377	0.2858
27	-0.1023	0.1616	0.1090	0.0270	0.1290	-0.2225	-0.0950
28	0.1247	0.2200	0.3324	0.0142	-0.2803	-0.2261	-0.2991
29	-0.0674	-0.2560	0.0776	-0.1650	0.2901	0.0686	-0.0794
30	0.1844	0.1619	-0.3392	-0.2794	-0.3040	0.1627	-0.1175
31	0.2378	-0.2649	-0.0343	-0.1389	-0.2396	0.1615	0.2955

32	0.2931	-0.1153	0.1653	-0.1668	0.0801	0.0557	0.2837
-----------	--------	---------	--------	---------	--------	--------	--------

<i>H 2 to H 3</i>	8	9	10	11	12	13	14
1	0.0256	-0.1713	-0.2516	0.3048	0.2395	-0.2950	0.0286
2	-0.1290	-0.2325	0.1357	0.2620	-0.3137	0.2264	0.2109
3	-0.1565	0.2274	-0.2572	-0.3385	0.2488	-0.1948	0.0346
4	0.1506	0.1142	-0.1990	-0.1292	-0.2426	0.2688	0.2178
5	-0.0053	0.0877	0.2927	0.1226	-0.2980	0.1790	0.1003
6	-0.0889	-0.0557	-0.0160	-0.2366	0.0218	0.0185	0.1320
7	-0.2601	-0.0029	0.1333	-0.0422	0.2481	0.1111	0.0503
8	0.1364	-0.3264	0.2426	-0.3056	0.2215	0.2748	0.1837
9	-0.2290	-0.0107	-0.0266	-0.3320	-0.0459	0.2314	0.0142
10	-0.2779	0.2207	0.2107	-0.2677	-0.2941	-0.2754	0.0761
11	-0.2429	-0.1797	0.1218	-0.2044	0.0093	0.2649	-0.2208
12	-0.3320	-0.1135	-0.2207	-0.2665	-0.0911	0.2665	-0.0871
13	-0.0677	0.2621	0.2734	-0.0987	-0.2652	-0.0064	0.2842
14	-0.1591	-0.2999	-0.0135	0.1099	-0.1599	-0.2154	0.1035
15	-0.0144	0.3294	0.1526	-0.2983	0.2850	0.0707	0.0258
16	0.1453	-0.0614	0.2398	-0.0184	0.3244	0.2508	-0.3211
17	-0.0446	-0.2518	-0.2659	0.2555	-0.2877	-0.1676	-0.0431
18	0.0560	-0.2011	-0.1249	-0.2974	-0.2474	-0.1666	-0.1883
19	0.2493	-0.1373	0.1080	0.0400	-0.1097	-0.3329	-0.1211
20	-0.1424	0.1859	-0.2765	-0.3116	0.0333	0.0197	-0.1927
21	0.1031	-0.0476	0.1950	-0.1102	0.0633	-0.1826	-0.2852
22	-0.0426	0.2722	-0.0625	0.2888	0.1876	-0.2946	0.0442
23	-0.0120	-0.1863	0.0053	-0.0445	0.0955	0.0287	0.1035
24	0.1139	-0.0405	-0.0257	0.2121	-0.1609	-0.1788	0.2598
25	0.2181	0.2562	-0.2695	-0.1796	-0.2922	0.2819	0.0321
26	0.2067	0.1431	0.0726	-0.0312	0.1445	0.2139	0.0087
27	-0.0845	-0.1579	0.2041	-0.1831	0.2464	-0.3203	0.0576
28	-0.2393	0.0175	0.2624	0.0124	-0.2788	0.2879	0.1603
29	-0.1593	0.1697	-0.1402	-0.2184	0.2925	-0.1138	0.1664
30	0.2972	0.1913	0.1023	-0.1560	-0.1109	-0.3130	0.1576
31	0.1072	0.2401	-0.2814	0.1200	0.1474	-0.3249	0.1987
32	0.0855	-0.0246	-0.1955	0.1228	0.2037	-0.2907	0.2795

<i>H 2 to H 3</i>	15	16	17	18	19	20	Bias
1	-0.0379	0.3321	-0.0183	-0.0314	0.0575	-0.0387	0.0000
2	0.2340	-0.1178	-0.0493	-0.0196	-0.0297	-0.2543	0.0000

3	0.3313	-0.2837	0.0422	0.1619	0.3130	-0.1025	0.0000
4	-0.3051	-0.0927	-0.2844	0.1777	0.1708	-0.0912	0.0000
5	-0.1508	-0.3007	0.2428	-0.3200	-0.2458	-0.1001	0.0000
6	-0.0871	-0.1952	-0.2423	0.3396	-0.0948	0.0757	0.0000
7	0.1931	0.2682	0.0279	0.2773	-0.1379	-0.0553	0.0000
8	-0.1514	0.1347	0.1375	-0.2560	0.1698	0.0265	0.0000
9	0.2799	0.3358	0.1492	0.0493	0.0257	0.0456	0.0000
10	0.0588	0.0561	-0.3292	-0.1552	0.0127	0.1891	0.0000
11	0.1266	-0.2735	0.1623	0.0769	-0.1381	-0.0204	0.0000
12	0.1561	0.0536	0.2369	0.3162	-0.3187	0.0150	0.0000
13	0.0902	-0.1134	0.2671	0.2340	-0.2342	0.0262	0.0000
14	0.3184	0.1270	-0.2103	-0.1198	-0.2291	-0.2410	0.0000
15	-0.2081	0.2923	0.2294	0.1606	0.2033	-0.1860	0.0000
16	-0.1865	-0.3260	0.1335	0.0506	-0.1313	0.3097	0.0000
17	0.2582	-0.2397	0.2832	0.2700	-0.0077	0.0753	0.0000
18	0.0506	-0.1214	-0.2513	0.1803	0.0181	-0.2911	0.0000
19	0.0661	-0.0994	-0.2254	-0.0698	0.2760	0.2053	0.0000
20	-0.2327	0.1698	0.1384	0.1864	-0.0265	-0.0062	0.0000
21	-0.1538	-0.3174	0.0784	0.2662	-0.1789	0.0863	0.0000
22	0.1608	-0.3343	-0.1394	0.2485	0.0318	0.3362	0.0000
23	-0.1617	-0.1269	-0.0672	-0.1579	-0.0091	-0.0592	0.0000
24	-0.0516	-0.1585	0.2864	-0.0308	0.1826	-0.0163	0.0000
25	-0.2640	-0.1035	-0.1443	0.3070	-0.2442	-0.1879	0.0000
26	0.0746	0.0898	0.0402	0.2494	-0.2747	-0.2368	0.0000
27	0.0339	-0.1257	-0.1701	0.0800	-0.2708	-0.2072	0.0000
28	-0.2167	-0.1798	-0.2655	-0.1412	0.1599	0.2527	0.0000
29	0.3027	-0.3188	0.2816	0.0359	0.1268	-0.1875	0.0000
30	-0.2871	0.0434	-0.0438	0.3387	0.0879	-0.3205	0.0000
31	0.1997	-0.0110	0.0428	-0.3313	0.2616	-0.0772	0.0000
32	0.2639	0.1973	-0.2726	-0.1414	-0.2285	-0.2160	0.0000

Lampiran 29. Bobot dan Bias Awal (*Hidden Layer 3 ke Output Layer*) Model DQ.

<i>H 3 to output</i>	1	2	3	4
1	-0.3405	0.2168	-0.1866	-0.1354
2	0.3616	-0.3902	0.0793	0.3434
3	0.3536	-0.3338	0.1681	-0.0587
4	0.3989	-0.2170	-0.0793	-0.3439
5	-0.3983	0.3386	-0.2672	-0.3595
6	0.0191	0.0528	-0.1991	-0.1545
7	-0.0612	0.3796	-0.3953	0.0563
8	-0.0139	0.3410	-0.2046	0.0630
9	-0.2366	0.1513	-0.3904	-0.1400

10	0.4023	-0.2811	0.1698	-0.3918
11	0.2697	0.1794	-0.3500	0.1776
12	-0.1735	0.2833	0.1825	-0.1414
13	-0.0644	-0.1074	-0.0657	-0.1421
14	0.3382	-0.0155	0.1045	-0.2559
15	-0.1572	0.1646	0.1371	0.0273
16	-0.0460	0.2541	0.0273	0.0365
17	-0.3058	0.3317	-0.3104	0.1483
18	-0.3585	0.2681	-0.3034	-0.0511
19	-0.1728	-0.3492	-0.3011	0.0440
20	-0.3002	0.0035	-0.1928	0.1450
21	-0.2774	-0.2117	-0.0958	0.0266
22	-0.3794	-0.0533	0.2047	-0.3964
23	0.2381	-0.0144	-0.1960	0.1101
24	0.3812	-0.0296	-0.2697	-0.3130
25	0.3272	0.1036	-0.0802	-0.3260
26	0.0321	0.1086	0.0747	0.0562
27	-0.1354	0.3040	0.3091	0.0693
28	0.2583	-0.0088	-0.1463	-0.3032
29	-0.3035	-0.3106	0.1070	0.1513
30	0.2059	0.1590	-0.2120	0.0869
31	0.2778	-0.0734	-0.2965	0.3132
32	-0.2182	-0.0904	-0.1898	0.0474
Bias	0.0000	0.0000	0.0000	0.0000

Lampiran 30. Output (Input ke Hidden Layer 1) Model DQ Sebelum Aktivasi

<i>Input to H 1</i>	1	2	3	4	5	6	7	8
1	0.3492	-0.4561	-0.0564	0.3245	0.1225	0.3780	0.2856	0.2359
2	0.5527	0.6183	0.4951	0.3367	0.4912	0.5512	0.6426	0.3974
3	0.8156	0.5232	0.9467	0.6646	0.9180	0.6577	0.9377	0.5056
4	-0.0928	0.1222	-0.1670	-0.2810	-0.2312	-0.2843	-0.2795	-0.1603
5	-0.2173	-0.3577	-0.4296	-0.4020	-0.4541	-0.1894	-0.5787	-0.1592
6	-0.4408	0.0434	-0.2245	-0.5030	-0.3248	-0.3580	-0.4391	-0.4321
7	0.2809	-0.1315	0.0751	0.0548	0.0836	0.2438	-0.0135	0.1370
8	-1.1992	-0.8430	-0.9800	-1.1531	-1.0209	-1.1517	-1.0771	-1.2626
9	0.6939	0.7849	0.6639	0.8551	0.7028	0.6163	0.7814	0.7887
10	-0.1248	-0.1404	-0.0919	-0.1847	-0.1589	-0.0437	-0.2105	0.0480
11	-1.0530	-0.7869	-1.3432	-1.0119	-1.2659	-1.2373	-1.2770	-1.0962
12	0.1715	0.1586	0.3514	0.3038	0.4098	0.0681	0.5075	-0.1057

Lampiran 31. *Output (Input ke Hidden Layer 1) Model DQ Sesudah Aktivasi*

<i>Input to H 1</i>	1	2	3	4	5	6	7	8
1	0.3492	0.0000	0.0000	0.3245	0.1225	0.3780	0.2856	0.2359
2	0.5527	0.6183	0.4951	0.3367	0.4912	0.5512	0.6426	0.3974
3	0.8156	0.5232	0.9467	0.6646	0.9180	0.6577	0.9377	0.5056
4	0.0000	0.1222	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.0000	0.0434	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
7	0.2809	0.0000	0.0751	0.0548	0.0836	0.2438	0.0000	0.1370
8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
9	0.6939	0.7849	0.6639	0.8551	0.7028	0.6163	0.7814	0.7887
10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0480
11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
12	0.1715	0.1586	0.3514	0.3038	0.4098	0.0681	0.5075	0.0000

Lampiran 32. *Output (Hidden Layer 1 ke Hidden Layer 2) Model DQ Sebelum Aktivasi*

<i>H 1 to H 2</i>	1	2	3	4	5	6	7	8
1	-0.4462	-0.2372	-0.3622	-0.5832	-0.4534	-0.3569	-0.5223	-0.3487
2	-0.1806	0.0207	-0.1817	-0.1947	-0.2205	-0.1080	-0.2230	-0.0269
3	0.4683	0.3282	0.5009	0.4683	0.5382	0.3912	0.6420	0.2920
4	-0.1566	0.1245	0.0422	-0.0592	0.0211	-0.1577	0.0580	-0.0934
5	-0.4923	-0.4864	-0.4786	-0.4584	-0.4910	-0.4357	-0.5526	-0.3891
6	-0.2313	-0.0246	-0.1313	-0.3570	-0.2085	-0.1884	-0.2605	-0.2301
7	-0.6504	-0.5530	-0.5933	-0.6393	-0.6208	-0.5990	-0.7340	-0.5746
8	0.3189	0.3874	0.4311	0.2510	0.3758	0.2801	0.3772	0.2807
9	0.1977	0.2448	0.2508	0.2163	0.2735	0.1610	0.3480	0.1150
10	0.0530	0.1297	0.1929	0.1955	0.1754	-0.0059	0.1176	0.1324
11	0.5139	0.5485	0.3998	0.5473	0.4607	0.4944	0.6106	0.4673
12	-0.2080	-0.3260	-0.5035	-0.2673	-0.4630	-0.1233	-0.4953	-0.1322
13	-0.1843	-0.1235	-0.1640	-0.1161	-0.1487	-0.1762	-0.1533	-0.1498
14	0.0493	0.2479	0.1195	0.0858	0.0933	0.0569	0.1069	0.1493
15	0.6912	0.4335	0.5340	0.5473	0.5759	0.6209	0.6470	0.4752
16	-0.2893	-0.2681	-0.5140	-0.2961	-0.5015	-0.2021	-0.5717	-0.1232
17	-0.1671	-0.1477	-0.2374	-0.0522	-0.2249	-0.1293	-0.2556	-0.0286
18	-0.6179	-0.3708	-0.5596	-0.4482	-0.5913	-0.5257	-0.6618	-0.3467
19	0.1872	0.3005	0.0911	0.2225	0.1258	0.2167	0.2265	0.2355
20	0.1098	0.1891	0.1897	0.2432	0.1990	0.0581	0.1836	0.1580

Lampiran 33. *Output (Hidden Layer 1 ke Hidden Layer 2) Model DQ Sesudah Aktivasi*

<i>H 1 to H 2</i>	1	2	3	4	5	6	7	8
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.0207	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.4683	0.3282	0.5009	0.4683	0.5382	0.3912	0.6420	0.2920
4	0.0000	0.1245	0.0422	0.0000	0.0211	0.0000	0.0580	0.0000
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
8	0.3189	0.3874	0.4311	0.2510	0.3758	0.2801	0.3772	0.2807
9	0.1977	0.2448	0.2508	0.2163	0.2735	0.1610	0.3480	0.1150
10	0.0530	0.1297	0.1929	0.1955	0.1754	0.0000	0.1176	0.1324
11	0.5139	0.5485	0.3998	0.5473	0.4607	0.4944	0.6106	0.4673
12	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
13	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
14	0.0493	0.2479	0.1195	0.0858	0.0933	0.0569	0.1069	0.1493
15	0.6912	0.4335	0.5340	0.5473	0.5759	0.6209	0.6470	0.4752
16	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
17	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
18	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
19	0.1872	0.3005	0.0911	0.2225	0.1258	0.2167	0.2265	0.2355
20	0.1098	0.1891	0.1897	0.2432	0.1990	0.0581	0.1836	0.1580

Lampiran 34. *Output (Hidden Layer 2 ke Hidden Layer 3) Model DQ Sebelum Aktivasi*

<i>H 2 to H 3</i>	1	2	3	4	5	6	7	8
1	0.0401	0.1025	-0.0289	0.0129	-0.0226	0.0691	0.0263	0.0534
2	0.1903	0.1542	0.1166	0.1619	0.1309	0.1894	0.1678	0.1710
3	0.0867	-0.0111	-0.0074	0.0045	0.0150	0.0959	0.0503	0.0108
4	-0.1172	0.0362	-0.0500	-0.1122	-0.0747	-0.0924	-0.0477	-0.0760
5	-0.0950	-0.0429	-0.0236	-0.0438	-0.0381	-0.1012	-0.0732	-0.0462
6	-0.3458	-0.3088	-0.3112	-0.3268	-0.3375	-0.3145	-0.4243	-0.2514
7	0.0964	0.0285	0.0866	0.0934	0.1008	0.0710	0.1144	0.0390
8	-0.1677	-0.1223	-0.0881	-0.1207	-0.1194	-0.1618	-0.1940	-0.0739
9	-0.0401	-0.1063	-0.0642	-0.0724	-0.0634	-0.0446	-0.0817	-0.0739
10	-0.0330	-0.0267	0.0278	0.0310	0.0338	-0.0609	0.0215	-0.0335
11	-0.2741	-0.3897	-0.3132	-0.2843	-0.3156	-0.2559	-0.4090	-0.2461
12	-0.3792	-0.4909	-0.4356	-0.4340	-0.4443	-0.3427	-0.5387	-0.3637
13	0.1630	0.1504	0.2583	0.2007	0.2564	0.1090	0.2610	0.1188
14	0.0372	-0.0484	-0.0486	-0.0381	-0.0478	0.0476	-0.0567	0.0050

21	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
22	0.4946	0.4947	0.4881	0.5278	0.5307	0.4314	0.6533	0.3851
23	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
24	0.1104	0.1831	0.0882	0.1265	0.0939	0.1223	0.1262	0.1558
25	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
26	0.2036	0.0752	0.2424	0.1499	0.2401	0.1619	0.2454	0.0959
27	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
28	0.0189	0.1095	0.0899	0.1496	0.1063	0.0000	0.1280	0.0710
29	0.1200	0.0166	0.0490	0.0487	0.0682	0.1191	0.0973	0.0458
30	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
31	0.3007	0.2955	0.2167	0.2393	0.2417	0.3021	0.3297	0.2405
32	0.2822	0.1744	0.2304	0.1876	0.2395	0.2631	0.2813	0.1801

Lampiran 36. Hasil Prediksi Model DQ (y_k)

Data Training				
No.	Prediksi	Decoding	Kelas	Label
1	[0.2415, 0.2413, 0.2319, 0.2853]	[0, 0, 0, 1]	3	SL
2	[0.4291, 0.5694, 0.0015, 0.0000]	[0, 1, 0, 0]	1	S
3	[0.0058, 0.0743, 0.3217, 0.5982]	[0, 0, 0, 1]	3	SL
4	[0.0080, 0.0729, 0.2707, 0.6484]	[0, 0, 0, 1]	3	SL
5	[0.1753, 0.3242, 0.4248, 0.0756]	[0, 0, 1, 0]	2	L
6	[0.2831, 0.3488, 0.2839, 0.0843]	[0, 1, 0, 0]	1	S
7	[0.7536, 0.1755, 0.0679, 0.0030]	[1, 0, 0, 0]	0	R
8	[0.0000, 0.0028, 0.0435, 0.9537]	[0, 0, 0, 1]	3	SL
9	[0.8334, 0.1306, 0.0354, 0.0006]	[1, 0, 0, 0]	0	R
10	[0.0000, 0.0017, 0.0272, 0.9711]	[0, 0, 0, 1]	3	SL
⋮	⋮	⋮	⋮	⋮
422	[0.9566, 0.0366, 0.0068, 0.0000]	[1, 0, 0, 0]	0	R
Data Testing				
No.	Prediksi	Decoding	Kelas	Label
1	[0.0000, 0.0029, 0.0626, 0.9345]	[0, 0, 0, 1]	3	SL
2	[0.0000, 0.0068, 0.1103, 0.8829]	[0, 0, 0, 1]	3	SL
3	[0.9910, 0.0083, 0.0003, 0.0004]	[1, 0, 0, 0]	0	R
4	[0.9987, 0.0012, 0.0000, 0.0000]	[1, 0, 0, 0]	0	R
5	[0.7264, 0.4965, 0.2676, 0.0095]	[1, 0, 0, 0]	0	R
6	[0.0000, 0.0162, 0.1978, 0.7860]	[0, 0, 0, 1]	3	SL
7	[0.0000, 0.0068, 0.1103, 0.8829]	[0, 0, 0, 1]	3	SL
8	[0.0000, 0.0017, 0.0384, 0.9599]	[0, 0, 0, 1]	3	SL
9	[0.0000, 0.0361, 0.2234, 0.7405]	[0, 0, 0, 1]	3	SL
10	[0.9590, 0.0373, 0.0037, 0.0000]	[1, 0, 0, 0]	0	R
⋮	⋮	⋮	⋮	⋮
106	[0.7878, 0.1776, 0.0339, 0.0008]	[1, 0, 0, 0]	0	R

Lampiran 37. Bobot dan Bias Akhir (*Input ke Hidden Layer 1*) Model DQ.

<i>Input to H 1</i>	1	2	3	4	5	6	7	8	Bias
1	-0.41	0.46	-0.51	0.52	0.28	-0.33	0.04	-0.20	-0.02
2	0.29	0.65	0.27	-0.22	0.80	0.02	-0.49	-0.01	0.14
3	0.71	-0.13	0.41	-0.05	0.14	0.31	0.61	-0.48	0.00
4	0.81	0.02	-0.08	-0.52	0.09	-0.30	0.52	0.13	-0.01
5	0.53	0.02	-0.49	-0.18	-0.30	-0.19	-0.07	0.19	0.00
6	0.75	-0.33	0.32	-0.67	-0.89	0.28	0.23	-0.08	0.04
7	0.53	-0.13	-0.38	-0.09	-0.12	0.22	-0.28	-0.11	0.00
8	0.31	-0.40	0.38	-0.55	-0.65	-0.43	0.42	-0.32	0.03
9	-0.54	0.25	0.18	0.25	0.72	0.54	0.04	0.63	0.17
10	-0.16	0.23	-0.04	0.40	0.13	-0.44	-0.45	0.57	0.04
11	-0.38	-0.18	-0.47	-0.51	0.27	-0.46	-0.20	0.16	0.00
12	-0.34	-0.69	0.62	-0.20	0.29	0.47	-0.82	-0.79	0.16

Lampiran 38. Bobot dan Bias Akhir (*Hidden Layer 1 ke Hidden Layer 2*) Model DQ.

<i>H 1 to H 2</i>	1	2	3	4	5	6
1	-0.4173	0.5423	-0.0010	0.1059	-0.2371	-0.0393
2	-0.1886	0.3748	-0.3276	-0.4548	0.1359	-0.7696
3	0.2947	0.2512	0.5868	0.4155	0.4099	0.3395
4	-0.3784	0.4510	-0.3154	-0.5858	0.1392	-0.1225
5	-0.0796	-0.1669	-0.2413	-0.3713	-0.0196	-0.3753
6	-0.3718	0.4182	-0.0685	0.1368	0.0171	-0.5362
7	-0.2668	-0.2007	-0.3243	0.0767	-0.0450	0.1290
8	-0.2796	0.2531	0.4686	0.0946	0.0927	0.5995
9	-0.0647	0.4658	-0.1297	-0.3095	0.0321	-0.4904
10	-0.2714	-0.3747	0.0784	-0.2992	0.2288	0.5373
11	0.2794	0.2911	0.0309	0.5956	0.1017	0.5743
12	0.3472	-0.1217	-0.3781	0.1065	0.1262	0.0779
13	0.0135	0.0093	-0.2260	-0.2926	0.1031	-0.0196
14	-0.3498	0.3717	-0.2322	-0.6269	-0.0917	0.1560
15	0.2539	0.2613	0.3979	0.3769	-0.0715	0.2191
16	0.1626	-0.2264	-0.3462	0.3797	-0.2267	0.1460
17	0.1550	-0.2269	0.0766	0.2734	-0.3661	0.5376
18	-0.1733	-0.2758	-0.3244	0.0063	-0.2581	-0.3377
19	0.0886	0.3817	-0.4595	-0.4117	-0.0154	-0.6761
20	-0.2394	-0.2152	-0.1485	-0.2551	0.0701	-0.0005

<i>H 1 to H 2</i>	7	8	9	10	11	12	Bias
1	0.4756	-0.2752	0.2985	0.1987	-0.5232	0.1243	0.4756
2	-0.4313	0.2590	0.5680	0.3458	-0.1426	0.1178	-0.4313
3	0.4501	0.2577	0.0080	0.1372	-0.1872	0.1760	0.4501
4	-0.0476	0.2785	0.2546	0.3397	0.7170	0.1456	-0.0476
5	-0.4292	-0.2644	0.3178	-0.0710	-0.0058	-0.0124	-0.4292
6	-0.4895	-0.1730	-0.2314	-0.0946	-0.1069	0.0312	-0.4895
7	-0.0508	-0.3773	0.0442	-0.0804	0.1381	0.0000	-0.0508
8	-0.1683	0.1630	-0.2039	0.2339	-0.6716	0.0499	-0.1683
9	-0.5015	0.3039	-0.1303	0.1603	0.7586	0.1380	-0.5015
10	0.2535	0.3364	0.0155	-0.1246	0.5152	0.0183	0.2535
11	0.4482	0.2809	-0.7149	-0.2788	-0.1883	-0.0789	0.4482
12	0.2894	-0.0348	-0.0774	0.1385	-0.2652	0.0000	0.2894
13	-0.0036	0.1467	0.2771	-0.0083	0.7317	0.1484	-0.0036
14	0.3505	0.4408	0.2066	0.1556	0.3779	0.1160	0.3505
15	0.2068	0.1973	-0.6810	-0.2968	-0.1980	0.0312	0.2068
16	-0.2181	0.0744	0.0170	-0.0266	-0.4127	-0.0091	-0.2181
17	0.6168	0.2899	-0.2790	-0.0819	-0.7155	0.1098	0.6168
18	0.1511	0.0448	-0.2342	-0.1836	-0.3269	0.0000	0.1511
19	-0.1461	0.3934	-0.2345	-0.1435	0.2685	-0.0107	-0.1461
20	-0.2314	0.4076	-0.4262	-0.0511	0.2014	-0.0203	-0.2314

Lampiran 39. Bobot dan Bias Akhir (*Hidden Layer 2 ke Hidden Layer 3*) Model DQ.

<i>H 2 to H 3</i>	1	2	3	4	5	6	7
1	-0.3115	0.2258	-0.1530	0.2522	-0.0382	0.0396	0.0771
2	-0.0046	-0.6457	0.0319	-0.0684	0.0030	-0.1151	0.0987
3	-0.2993	-0.2507	0.0004	-0.0963	0.3311	-0.2857	0.2066
4	-0.0399	0.5815	0.0983	0.6737	0.3259	0.2784	-0.1146
5	-0.2994	-0.0205	-0.1202	-0.1387	0.3212	0.1530	-0.0812
6	0.2783	-0.2174	-0.2581	-0.1645	-0.2622	-0.2085	0.1948
7	-0.3433	0.4233	0.1653	0.4034	-0.0781	-0.0171	0.0831
8	0.0700	0.2064	0.1787	-0.3253	-0.3044	-0.4697	-0.1172
9	-0.1315	-0.0857	-0.0347	0.1710	-0.0899	-0.0469	0.2893
10	0.2054	0.2155	0.1248	0.4700	-0.3085	0.2431	0.3327
11	0.2722	0.0387	-0.2371	-0.2655	-0.2828	0.2396	-0.0051
12	0.0881	0.0556	-0.3158	-0.0834	-0.2101	-0.1259	-0.0819
13	-0.0062	-0.5753	0.3780	-0.2700	-0.1100	-0.1769	-0.1448
14	0.2683	0.5748	-0.1490	0.5233	-0.0812	0.6923	-0.2197
15	0.0689	-0.0285	-0.0512	-0.3576	-0.2524	-0.0151	-0.1741
16	0.0082	-0.0725	-0.3359	0.1602	0.1533	-0.0496	-0.2842

17	0.2686	-0.0033	0.1492	0.0250	0.0457	-0.3032	0.0258
18	0.1799	-0.0904	0.0658	-0.2066	-0.3080	0.2719	-0.2495
19	0.1111	-0.2017	0.2587	-0.1947	-0.1885	-0.5746	-0.0624
20	-0.2777	-0.1677	0.2875	0.1919	0.2845	-0.1126	-0.1133
21	0.3827	-0.2754	0.2206	0.0000	0.2397	-0.3694	-0.1226
22	0.3393	-0.2018	0.4348	0.2570	-0.0297	0.3431	0.0217
23	0.2988	-0.0525	-0.0270	0.4057	0.1851	0.0751	0.0575
24	0.0611	-0.2537	-0.1135	-0.3177	0.0272	-0.4513	-0.2366
25	-0.2488	0.6141	0.1017	0.5945	0.2432	0.6237	-0.2482
26	0.1128	-0.5361	0.5262	-0.5548	-0.1420	-0.4544	0.2858
27	-0.1073	0.1436	0.0850	0.0098	0.1191	-0.2225	-0.0950
28	0.0828	0.6042	0.2291	0.4154	-0.2694	-0.0622	-0.2991
29	0.0542	-0.5514	0.2425	-0.4218	0.2901	-0.0024	-0.0794
30	0.2362	0.5627	-0.3409	0.1189	-0.2926	0.2687	-0.1175
31	0.1745	-0.5997	-0.0589	-0.3293	-0.2396	-0.2861	0.2955
32	0.3883	-0.4092	0.3093	-0.4604	0.0696	-0.2035	0.2837

<i>H 2 to H 3</i>	8	9	10	11	12	13	14
1	0.0017	-0.1946	-0.2772	0.2805	0.2395	-0.3161	-0.0053
2	-0.1383	-0.3855	0.2599	0.3603	-0.3137	0.2837	0.2470
3	-0.1593	0.2228	-0.2587	-0.3411	0.2488	-0.1948	0.0283
4	0.0866	0.4598	-0.1499	-0.1948	-0.2426	0.7298	0.3555
5	-0.0482	0.0492	0.2477	0.0632	-0.2980	0.0034	0.0417
6	-0.0889	-0.0557	-0.0160	-0.2366	0.0218	0.0185	0.1320
7	-0.3158	-0.0252	0.4707	-0.1915	0.2481	0.5718	0.1552
8	0.1818	-0.4582	0.4293	-0.3061	0.2215	0.4707	0.2971
9	-0.2717	-0.0527	-0.0531	-0.3821	-0.0459	0.1474	-0.0269
10	-0.3132	0.5837	0.1244	-0.2907	-0.2941	0.0716	0.1926
11	-0.2429	-0.1797	0.1218	-0.2044	0.0093	0.2649	-0.2208
12	-0.3320	-0.1135	-0.2207	-0.2665	-0.0911	0.2665	-0.0871
13	0.0320	0.0712	0.1917	0.0554	-0.2652	-0.3340	0.3103
14	-0.1621	0.0704	-0.0667	0.0969	-0.1599	0.1776	0.2500
15	-0.0250	0.3110	0.1426	-0.3079	0.2850	0.0439	0.0106
16	0.1453	-0.0614	0.2398	-0.0184	0.3244	0.2508	-0.3211
17	0.1149	-0.4512	-0.1894	0.3516	-0.2877	-0.4096	0.0299
18	0.0505	-0.2123	-0.1249	-0.3029	-0.2474	-0.1666	-0.1964
19	0.2790	-0.4493	0.2060	0.1217	-0.1097	-0.4652	-0.0553
20	-0.1424	0.1859	-0.2765	-0.3116	0.0333	0.0197	-0.1927
21	0.2854	-0.2200	0.2250	0.0971	0.0633	-0.5272	-0.1500
22	0.1144	0.4495	-0.2946	0.4241	0.1876	-0.8942	0.0167
23	-0.0777	-0.1309	0.4032	-0.0791	0.0955	0.6131	0.2042

24	0.0813	-0.0755	-0.0911	0.1898	-0.1609	-0.2898	0.2313
25	0.1748	0.6212	-0.2106	-0.2714	-0.2922	0.8052	0.1899
26	0.4534	-0.0695	0.0386	0.2017	0.1445	-0.2026	0.1032
27	-0.1095	-0.1579	0.1808	-0.2071	0.2464	-0.3381	0.0360
28	-0.3694	0.3400	0.3340	-0.1593	-0.2788	0.7834	0.2878
29	0.0053	-0.0122	-0.1490	-0.0035	0.2925	-0.5196	0.2579
30	0.2843	0.4483	0.5708	-0.2780	-0.1109	0.4240	0.3750
31	0.0517	-0.0965	-0.0965	0.0930	0.1474	-0.0501	0.2365
32	0.2355	-0.3146	-0.1309	0.2783	0.2037	-0.6061	0.3696

<i>H 2</i> <i>to</i> <i>H 3</i>	15	16	17	18	19	20	Bias
1	-0.0610	0.3196	-0.0258	-0.0314	0.0260	-0.0596	-0.0293
2	0.2986	-0.1059	0.0210	-0.0196	-0.1446	-0.2309	-0.0352
3	0.3293	-0.2837	0.0422	0.1619	0.3086	-0.1025	-0.0032
4	-0.3474	-0.0778	-0.6675	0.1777	0.6082	-0.0597	0.0724
5	-0.1986	-0.3112	0.1677	-0.3200	-0.2734	-0.1195	-0.0264
6	-0.0871	-0.1952	-0.2423	0.3396	-0.0948	0.0757	0.0000
7	0.0692	0.2723	0.0346	0.2773	-0.0440	-0.0615	0.0773
8	-0.1457	0.1344	0.2534	-0.2560	-0.1237	0.0038	0.1063
9	0.2407	0.3366	0.1154	0.0493	-0.0127	0.0451	-0.0104
10	0.0555	0.0708	-0.6782	-0.1552	0.4878	0.2205	0.0710
11	0.1266	-0.2735	0.1623	0.0769	-0.1381	-0.0204	0.0000
12	0.1561	0.0536	0.2369	0.3162	-0.3187	0.0150	0.0000
13	0.2036	-0.0991	0.4940	0.2340	-0.6039	0.0248	-0.0156
14	0.3185	0.1412	-0.4561	-0.1198	0.2579	-0.2210	0.0849
15	-0.2166	0.2830	0.2001	0.1606	0.1777	-0.1946	-0.0106
16	-0.1865	-0.3260	0.1335	0.0506	-0.1313	0.3097	0.0000
17	0.3651	-0.2491	0.5484	0.2700	-0.3278	0.0722	0.1256
18	0.0463	-0.1214	-0.2513	0.1803	0.0081	-0.2911	-0.0061
19	0.1168	-0.1038	-0.0962	-0.0698	0.0372	0.2199	-0.0014
20	-0.2327	0.1698	0.1384	0.1864	-0.0265	-0.0062	0.0000
21	0.0296	-0.3321	0.3326	0.2662	-0.4578	0.0911	0.1023
22	0.3140	-0.3499	-0.0389	0.2485	-0.0550	0.3129	0.0615
23	-0.2223	-0.1095	-0.0458	-0.1579	0.2446	-0.0223	0.0366
24	-0.0919	-0.1431	0.2988	-0.0308	0.1007	0.0007	-0.0481
25	-0.3165	-0.0929	-0.4880	0.3070	0.2315	-0.1577	0.1194
26	0.2959	0.0928	0.4124	0.2494	-0.6449	-0.2582	0.1430
27	0.0096	-0.1379	-0.1929	0.0800	-0.2916	-0.2221	-0.0234
28	-0.3607	-0.1644	-0.5891	-0.1412	0.5891	0.2670	0.0531
29	0.4907	-0.3237	0.5376	0.0359	-0.1288	-0.1799	0.0712
30	-0.3612	0.0592	-0.3695	0.3387	0.4881	-0.2883	0.1683

31	0.1441	0.0000	0.1108	-0.3313	0.0178	-0.0538	-0.0270
32	0.3973	0.1812	-0.0115	-0.1414	-0.6005	-0.2125	0.0911

Lampiran 40. Bobot dan Bias Akhir (*Hidden Layer 3 ke Output Layer*) Model DQ.

<i>H 3 to output</i>	1	2	3	4
1	-0.3092	0.2072	-0.1990	-0.1612
2	0.2727	-0.5823	0.0605	0.5016
3	0.3182	-0.3328	0.1921	-0.0850
4	0.7424	-0.0751	-0.1864	-0.9135
5	-0.3856	0.3209	-0.2920	-0.3398
6	0.0191	0.0528	-0.1991	-0.1545
7	0.3679	0.3356	-0.9786	0.0247
8	-0.1264	0.4689	-0.5207	0.2055
9	-0.2178	0.1385	-0.4100	-0.1551
10	0.7459	-0.1628	0.1189	-0.9986
11	0.2697	0.1794	-0.3500	0.1776
12	-0.1735	0.2833	0.1825	-0.1414
13	-0.5238	-0.1767	-0.0327	-0.0039
14	0.5447	0.0567	0.2039	-0.7374
15	-0.1478	0.1738	0.1227	0.0126
16	-0.0460	0.2541	0.0273	0.0365
17	-0.7364	0.3467	-0.1948	0.2362
18	-0.3600	0.2452	-0.3252	-0.0385
19	-0.4375	-0.3591	-0.2258	0.1218
20	-0.3002	0.0035	-0.1928	0.1450
21	-0.8921	-0.2217	-0.0569	0.1739
22	-0.6348	-0.0312	0.3661	-0.4132
23	0.4741	-0.0563	-0.5551	0.1462
24	0.3318	0.0834	-0.4405	-0.2101
25	0.5938	0.2752	-0.0332	-0.9054
26	-0.4586	0.0897	0.1498	0.1771
27	-0.1268	0.2892	0.2892	0.0838
28	0.6935	0.0941	-0.4902	-0.7057
29	-0.7840	-0.3354	0.2260	0.2645
30	0.6455	0.3869	-0.6828	-0.5065
31	0.0888	-0.1259	-0.2212	0.4159
32	-0.6112	-0.0913	-0.0828	0.1465
Bias	-0.1055	0.0637	0.0129	-0.0021

Lampiran 41. Model ANN Secara Matematis

Model AQ (contoh salah satu model)

$$\begin{aligned}
 Z_{1,1_1} &= f(0.27x_1 - 0.30x_2 + 0.13x_3 + 0.54x_4 + 0.58x_5 - 0.05x_6 + \\
 &\quad 0.03x_7 - 0.24x_8) + 0.06 \\
 Z_{2,1_1} &= f(-0.15x_1 - 0.40x_2 + 0.16x_3 + 0.39x_4 - 0.05x_5 - 0.09x_6 - \\
 &\quad 0.26x_7 - 0.15x_8) + 0.07 \\
 Z_{3,1_1} &= f(-0.04x_1 + 0.38x_2 + 0.09x_3 + 0.25x_4 + 0.32x_5 + 0.33x_6 + \\
 &\quad 0.06x_7 + 0.11x_8) - 0.03 \\
 Z_{4,1_1} &= f(-0.11x_1 - 0.39x_2 - 0.16x_3 + 0.23x_4 - 0.33x_5 + 0.24x_6 - \\
 &\quad 0.66x_7 + 0.31x_8) + 0.09 \\
 Z_{5,1_1} &= f(0.49x_1 - 0.18x_2 - 0.15x_3 + 0.37x_4 - 0.08x_5 - 0.29x_6 + \\
 &\quad 0.08x_7 + 0.29x_8) + 0.09 \\
 Z_{6,1_1} &= f(0.07x_1 + 0.55x_2 - 0.16x_3 + 0.53x_4 + 0.08x_5 - 0.24x_6 + \\
 &\quad 0.68x_7 + 0.38x_8) + 0.14 \\
 Z_{7,1_1} &= f(0.47x_1 + 0.10x_2 - 0.01x_3 + 0.17x_4 - 0.44x_5 + 0.44x_6 + \\
 &\quad 0.08x_7 + 0.29x_8) + 0.21 \\
 Z_{8,1_1} &= f(0.35x_1 - 0.02x_2 + 0.36x_3 - 0.16x_4 + 0.17x_5 - 0.15x_6 - \\
 &\quad 0.26x_7 - 0.26x_8) + 0.00 \\
 Z_{9,1_1} &= f(0.22x_1 - 0.21x_2 - 0.36x_3 + 0.06x_4 - 0.03x_5 - 0.03x_6 + \\
 &\quad 0.07x_7 - 0.16x_8) + 0.00 \\
 Z_{10,1_1} &= f(-0.25x_1 - 0.10x_2 - 0.39x_3 + 0.23x_4 + 0.24x_5 - 0.38x_6 + \\
 &\quad 0.37x_7 - 0.38x_8) + 0.00 \\
 Z_{11,1_1} &= f(0.13x_1 + 0.12x_2 - 0.18x_3 + 0.28x_4 + 0.19x_5 - 0.24x_6 - \\
 &\quad 0.23x_7 - 0.34x_8) + 0.00 \\
 Z_{12,1_1} &= f(0.37x_1 + 0.23x_2 - 0.09x_3 + 0.13x_4 + 0.10x_5 + 0.39x_6 - \\
 &\quad 0.12x_7 - 0.39x_8) + 0.07 \\
 Z_{13,1_1} &= f(-0.34x_1 + 0.09x_2 - 0.09x_3 - 0.04x_4 - 0.29x_5 - 0.20x_6 + \\
 &\quad 0.10x_7 - 0.18x_8) + 0.00 \\
 Z_{14,1_1} &= f(-0.29x_1 - 0.04x_2 - 0.15x_3 - 0.04x_4 - 0.05x_5 - 0.11x_6 + \\
 &\quad 0.28x_7 - 0.19x_8) + 0.00
 \end{aligned}$$

$$\begin{aligned}
Z_{15,1_1} &= f(0.32x_1 - 0.22x_2 + 0.38x_3 - 0.66x_4 - 0.14x_5 + 0.42x_6 - \\
&\quad 0.10x_7 + 0.51x_8) + 0.03 \\
Z_{16,1_1} &= f(0.25x_1 - 0.41x_2 - 0.33x_3 - 0.17x_4 - 0.49x_5 + 0.56x_6 + \\
&\quad 0.15x_7 - 0.25x_8) + 0.10 \\
Z_{17,1_1} &= f(0.05x_1 + 0.73x_2 - 0.27x_3 + 0.65x_4 - 0.19x_5 + 0.31x_6 - \\
&\quad 0.89x_7 - 0.39x_8) + 0.05 \\
Z_{18,1_1} &= f(0.07x_1 + 0.15x_2 + 0.02x_3 - 0.29x_4 - 0.05x_5 - 0.28x_6 - \\
&\quad 0.34x_7 - 0.06x_8) + 0.00 \\
Z_{19,1_1} &= f(0.30x_1 - 0.37x_2 + 0.04x_3 - 0.01x_4 - 0.28x_5 + 0.01x_6 - \\
&\quad 0.02x_7 - 0.41x_8) + 0.00 \\
Z_{20,1_1} &= f(-0.24x_1 - 0.36x_2 - 0.06x_3 + 0.17x_4 + 0.37x_5 - 0.56x_6 - \\
&\quad 0.17x_7 + 0.32x_8) + 0.02 \\
Z_{21,1_1} &= f(0.34x_1 - 0.41x_2 + 0.16x_3 + 0.66x_4 - 0.06x_5 + 0.32x_6 - \\
&\quad 0.11x_7 - 0.26x_8) + 0.14 \\
Z_{22,1_1} &= f(0.20x_1 - 0.43x_2 + 0.19x_3 + 0.43x_4 + 0.63x_5 - 0.30x_6 + \\
&\quad 0.50x_7 + 0.29x_8) - 0.01 \\
Z_{23,1_1} &= f(0.13x_1 + 0.07x_2 + 0.01x_3 + 0.55x_4 + 0.38x_5 - 0.40x_6 + \\
&\quad 0.35x_7 + 0.17x_8) + 0.12 \\
Z_{24,1_1} &= f(0.02x_1 - 0.27x_2 - 0.37x_3 - 0.46x_4 - 0.30x_5 + 0.60x_6 + \\
&\quad 0.36x_7 + 0.07x_8) + 0.06 \\
Z_{25,1_1} &= f(0.20x_1 + 0.01x_2 + 0.40x_3 + 0.58x_4 + 0.36x_5 - 0.40x_6 + \\
&\quad 0.52x_7 + 0.13x_8) + 0.20 \\
Z_{26,1_1} &= f(0.07x_1 - 0.02x_2 + 0.13x_3 - 0.05x_4 - 0.33x_5 + 0.16x_6 - \\
&\quad 0.73x_7 + 0.36x_8) + 0.03 \\
Z_{27,1_1} &= f(0.30x_1 - 0.11x_2 + 0.25x_3 - 0.30x_4 + 0.07x_5 - 0.09x_6 - \\
&\quad 0.24x_7 - 0.13x_8) - 0.03 \\
Z_{28,1_1} &= f(0.10x_1 - 0.42x_2 + 0.33x_3 - 0.25x_4 - 0.43x_5 + 0.33x_6 - \\
&\quad 0.57x_7 + 0.34x_8) - 0.07 \\
Z_{1,1_2} &= f(0.5760Z_{1,1_1} + 0.2454Z_{2,1_1} + \dots - 0.3916Z_{28,1_1}) - 0.0245
\end{aligned}$$

$$\begin{aligned}
Z_{2,1_2} &= f(0.1774Z_{1,1_1} - 0.0546Z_{2,1_1} + \dots - 0.3174Z_{28,1_1}) - 0.0234 \\
Z_{3,1_2} &= f(0.2762Z_{1,1_1} + 0.1272Z_{2,1_1} + \dots - 0.2583Z_{28,1_1}) - 0.0013 \\
Z_{4,1_2} &= f(-0.0384Z_{1,1_1} - 0.1467Z_{2,1_1} + \dots + 0.0384Z_{28,1_1}) - 0.0279 \\
Z_{5,1_2} &= f(0.5006Z_{1,1_1} + 0.0272Z_{2,1_1} + \dots + 0.0772Z_{28,1_1}) - 0.0437 \\
Z_{6,1_2} &= f(0.0488Z_{1,1_1} + 0.3663Z_{2,1_1} + \dots - 0.1806Z_{28,1_1}) + 0.0415 \\
Z_{7,1_2} &= f(-0.0258Z_{1,1_1} - 0.1683Z_{2,1_1} + \dots - 0.0351Z_{28,1_1}) + 0.0646 \\
Z_{8,1_2} &= f(0.5358Z_{1,1_1} + 0.5820Z_{2,1_1} + \dots - 0.0036Z_{28,1_1}) - 0.0260 \\
Z_{9,1_2} &= f(0.1907Z_{1,1_1} + 0.0403Z_{2,1_1} + \dots - 0.2925Z_{28,1_1}) - 0.0008 \\
Z_{10,1_2} &= f(-0.3112Z_{1,1_1} + 0.2314Z_{2,1_1} + \dots - 0.1550Z_{28,1_1}) + 0.000 \\
Z_{11,1_2} &= f(0.8330Z_{1,1_1} + 0.4190Z_{2,1_1} + \dots + 0.1273Z_{28,1_1}) - 0.0224 \\
Z_{12,1_2} &= f(-0.4265Z_{1,1_1} - 0.4692Z_{2,1_1} + \dots + 0.1551Z_{28,1_1}) + 0.0998 \\
Z_{13,1_2} &= f(0.1486Z_{1,1_1} - 0.1939Z_{2,1_1} + \dots - 0.3885Z_{28,1_1}) + 0.0160 \\
Z_{14,1_2} &= f(0.2502Z_{1,1_1} + 0.4005Z_{2,1_1} + \dots - 0.1373Z_{28,1_1}) + 0.0858 \\
Z_{15,1_2} &= f(0.3538Z_{1,1_1} + 0.0564Z_{2,1_1} + \dots - 0.2386Z_{28,1_1}) + 0.1204 \\
Z_{16,1_2} &= f(-0.2143Z_{1,1_1} - 0.0413Z_{2,1_1} + \dots + 0.1011Z_{28,1_1}) - 0.0202 \\
Z_{17,1_2} &= f(-0.0589Z_{1,1_1} - 0.0111Z_{2,1_1} + \dots - 0.2694Z_{28,1_1}) + 0.0297 \\
Z_{18,1_2} &= f(-0.1456Z_{1,1_1} + 0.0077Z_{2,1_1} + \dots + 0.0159Z_{28,1_1}) + 0.0000 \\
Z_{19,1_2} &= f(0.7277Z_{1,1_1} + 0.7137Z_{2,1_1} + \dots - 0.1330Z_{28,1_1}) + 0.0529 \\
Z_{20,1_2} &= f(-0.1482Z_{1,1_1} - 0.1754Z_{2,1_1} + \dots - 0.2925Z_{28,1_1}) + 0.0000 \\
Z_{21,1_2} &= f(0.0155Z_{1,1_1} - 0.1252Z_{2,1_1} + \dots + 0.3200Z_{28,1_1}) + 0.0456 \\
Z_{22,1_2} &= f(0.4446Z_{1,1_1} + 0.0904Z_{2,1_1} + \dots - 0.1818Z_{28,1_1}) + 0.0114 \\
Z_{23,1_2} &= f(-0.0989Z_{1,1_1} - 0.0032Z_{2,1_1} + \dots - 0.2072Z_{28,1_1}) + 0.0000 \\
Z_{24,1_2} &= f(0.3174Z_{1,1_1} + 0.5248Z_{2,1_1} + \dots - 0.2548Z_{28,1_1}) - 0.0046 \\
Z_{25,1_2} &= f(0.26664Z_{1,1_1} + 0.0652Z_{2,1_1} + \dots - 0.0997Z_{28,1_1}) + 0.0296 \\
Z_{26,1_2} &= f(0.3522Z_{1,1_1} + 0.3375Z_{2,1_1} + \dots + 0.1197Z_{28,1_1}) - 0.0005 \\
Z_{27,1_2} &= f(0.4439Z_{1,1_1} + 0.2681Z_{2,1_1} + \dots - 0.4035Z_{28,1_1}) - 0.0112 \\
Z_{28,1_2} &= f(-0.2717Z_{1,1_1} - 0.0424Z_{2,1_1} + \dots - 0.0997Z_{28,1_1}) + 0.0296 \\
Z_{29,1_2} &= f(0.0790Z_{1,1_1} + 0.5625Z_{2,1_1} + \dots - 0.1981Z_{28,1_1}) + 0.0565
\end{aligned}$$

$$\begin{aligned}
Z_{30,1_2} &= f(0.2049Z_{1,1_1} + 0.0958Z_{2,1_1} + \dots - 0.3087Z_{28,1_1}) + 0.0471 \\
Z_{31,1_2} &= f(-0.2997Z_{1,1_1} - 0.4243Z_{2,1_1} + \dots + 0.1605Z_{28,1_1}) + 0.1468 \\
Z_{32,1_2} &= f(0.5545Z_{1,1_1} - 0.5559Z_{2,1_1} + \dots - 0.1440Z_{28,1_1}) + 0.0511 \\
\\
y_{net_1} &= f(0.2160Z_{1,1_2} - 0.0521Z_{2,1_2} + \dots + 0.3370Z_{32,1_2}) - 0.0632 \\
y_{net_2} &= f(0.1890Z_{1,1_2} - 0.1254Z_{2,1_2} + \dots + 0.0987Z_{32,1_2}) + 0.0126 \\
y_{net_3} &= f(0.1292Z_{1,1_2} + 0.3327Z_{2,1_2} + \dots - 0.5892Z_{32,1_2}) + 0.0162 \\
y_{net_4} &= f(0.0986Z_{1,1_2} - 0.1857Z_{2,1_2} + \dots + 0.3519Z_{32,1_2}) + 0.0202
\end{aligned}$$

Lampiran 42. Source Code

a) Menyiapkan Variabel *Input* dan *Output*

```

# Memisahkan fitur (X) dan target (y)
x = data.iloc[:, 0:-1] # Mengambil semua baris, kecuali kolom terakhir
y = data.iloc[:, -1] # Mengambil semua baris pada kolom terakhir

```

b) Membagi Data *Training* dan *Testing*

```

X_train, X_test, y_train, y_test = train_test_split(x, y_encoded, test_size=0.2,
random_state=42)

```

c) *Encoding* Data

```

# Konversi kelas menjadi angka dengan LabelEncoder
label_encoder = LabelEncoder()
custom_mapping = {"R": 0, "S": 1, "L": 2, "SL": 3}
label_encoder.fit(list(custom_mapping.values()))
y_encoded = [custom_mapping[label] for label in y]
# Konversi kelas menjadi format one-hot encoding
num_classes = len(label_encoder.classes_)
y_train_onehot = to_categorical(y_train, num_classes=4)
y_test_onehot = to_categorical(y_test, num_classes=4)
# Konversi kelas menjadi format one-hot encoding
num_classes = len(label_encoder.classes_)
y_train_onehot = to_categorical(y_train, num_classes=4)
y_test_onehot = to_categorical(y_test, num_classes=4)

```

d) *Normalisasi* Data

```

# Normalisasi menggunakan Min-Max Scaling
scaler = MinMaxScaler()
X_train_normalized = scaler.fit_transform(X_train)

```

```
X_test_normalized = scaler.transform(X_test)
```

e) Membangun Model ANN

```
# Membangun model jaringan saraf tiruan
model = Sequential()
model.add(Dense(8, input_dim= X_train_normalized.shape[1], activation='relu'))
model.add(Dense(28, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(4, activation='softmax'))
```

f) Mengompilasi Model

```
model.compile(loss='categorical_crossentropy',optimizer='adam',
metrics=['accuracy'])
```

g) Melatih Model

```
model.fit(X_train_normalized, y_train_onehot, epochs=100, batch_size=32,
verbose=1)
```

h) Evaluasi Model

```
loss, accuracy = model.evaluate(X_train_normalized, y_train_onehot)
print(f'Loss: {loss}, Accuracy: {accuracy}')
```

```
# Inisialisasi list untuk menyimpan metrik evaluasi berbobot per kelas
weighted_precision_scores = []
weighted_recall_scores = []
weighted_f1_scores = []
```

```
# Inisialisasi variabel untuk menyimpan total support (jumlah sampel) untuk semua kelas
total_support = 0
```

```
# Inisialisasi list untuk menyimpan nilai TP, TN, FP, dan FN per kelas
class_tp_values = []
class_tn_values = []
class_fp_values = []
class_fn_values = []
```

```
# Hitung multilabel confusion matrix
mcm = multilabel_confusion_matrix(y_train, predicted_classes_train)

classes = ['0', '1', '2', '3']
```

```

# Iterasi untuk setiap kelas
for i in range(len(classes)):
    # Ambil confusion matrix untuk kelas tersebut
    cm = mcm[i]

    # Ambil nilai TP, TN, FP, dan FN dari confusion matrix
    tn, fp, fn, tp = cm.ravel()

    # Tambahkan nilai TP, TN, FP, dan FN ke dalam list per kelas
    class_tp_values.append(tp)
    class_tn_values.append(tn)
    class_fp_values.append(fp)
    class_fn_values.append(fn)

    # Hitung metrik evaluasi per kelas
    precision = tp / (tp + fp) if (tp + fp) != 0 else 0.0
    recall = tp / (tp + fn)
    f1 = 2 * (precision * recall) / (precision + recall) if (precision + recall) != 0 else
        0.0

    # Ambil nilai support (jumlah sampel) untuk kelas tersebut
    support = tp + fn

    # Tambahkan metrik evaluasi yang berbobot ke dalam list
    weighted_precision_scores.append(precision * support)
    weighted_recall_scores.append(recall * support)
    weighted_f1_scores.append(f1 * support)

    # Tambahkan support ke total support
    total_support += support

    # Tampilkan metrik evaluasi untuk kelas tersebut
    print(f'Metrics for Class {classes[i]}:')
    print(f'Precision: {precision:.2f}')
    print(f'Recall: {recall:.2f}')
    print(f'F1-score: {f1:.2f}')
    print(f'Support: {support}')
    print(f'True Positives (TP): {tp}')
    print(f'True Negatives (TN): {tn}')
    print(f'False Positives (FP): {fp}')
    print(f'False Negatives (FN): {fn}')
    print()

```

```

# Hitung rata-rata metrik evaluasi yang berbobot (weighted)
weighted_precision = sum(weighted_precision_scores) / total_support
weighted_recall = sum(weighted_recall_scores) / total_support
weighted_f1 = sum(weighted_f1_scores) / total_support

print("Weighted Metrics:")
print(f"Weighted Precision: {weighted_precision:.2f}")
print(f"Weighted Recall: {weighted_recall:.2f}")
print(f"Weighted F1-score: {weighted_f1:.2f}")

# Tampilkan total nilai TP, TN, FP, dan FN untuk semua kelas
print("Total Values:")
print(f"Total True Positives (TP): {sum(class_tp_values)}")
print(f"Total True Negatives (TN): {sum(class_tn_values)}")
print(f"Total False Positives (FP): {sum(class_fp_values)}")
print(f"Total False Negatives (FN): {sum(class_fn_values)}")

```

i) Prediksi

```

predictions_train = model.predict(X_train_normalized)
np.set_printoptions(precision=3)

#Mengkonversi nilai probabilitas dari prediksi ke kelas asli
predicted_classes_train = np.argmax(predictions_train, axis=1)

#merubah hasil dalam bentuk abjad
class_labels = {
    0: "R",
    1: "S",
    2: "L",
    3: "SL"
}
predicted_labels = [class_labels[prediction] for prediction in predicted_classes_train]

```

RIWAYAT HIDUP



Nur Kholida Maulidia, lahir di Banyuwangi pada 4 Juni 2000. Penulis merupakan anak tunggal dari pasangan Bapak Ahmad Nur Halim dan Ibu Siti Zubaidah. Penulis telah menempuh pendidikan mulai dari TK Khadijah 57 dan lulus pada tahun 2007, dilanjutkan menempuh pendidikan sekolah dasar di MI Roudlotuth Tholibin dan lulus pada tahun 2013. Kemudian penulis melanjutkan pendidikan sekolah menengah pertama di SMPN 1 Muncar dan lulus pada tahun 2016. Selanjutnya menempuh pendidikan sekolah menengah atas di SMAN 1 Giri dan lulus pada tahun 2019. Pada tahun yang sama, penulis melanjutkan pendidikan di Universitas Islam Negeri Maulana Malik Ibrahim Malang pada Program Studi Matematika, Fakultas Sains dan Teknologi.

Selama menempuh pendidikan di perguruan tinggi, penulis aktif mengikuti beberapa kegiatan baik di dalam lingkup kampus maupun di luar kampus. Penulis pernah menjadi asisten praktikum Statistika Elementer. Penulis juga pernah mengikuti program Kampus Merdeka di Universitas Jember. Selain itu, penulis juga mengikuti berbagai kegiatan di luar kampus seperti pelatihan dan seminar.



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

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

BUKTI KONSULTASI SKRIPSI

Nama : Nur Kholida Maulidia
NIM : 19610055
Fakultas/Program Studi : Sains dan Teknologi/Matematika
Judul Skripsi : Penerapan *Artificial Neural Network* pada Proses Klasifikasi Curah Hujan di Jawa Tengah
Pembimbing I : Prof. Dr. Hj. Sri Harini, M.Si.
Pembimbing II : Juhari, M.Si.

No	Tanggal	Hal	Tanda Tangan
1.	7 Oktober 2022	Konsultasi Judul dan Topik	1.
2.	23 Desember 2022	Konsultasi BAB I, BAB II, dan BAB III	2.
3.	10 Januari 2023	Konsultasi Revisi BAB I, BAB II, dan BAB III	3.
4.	20 Januari 2023	Konsultasi Kajian Agama BAB I	4.
5.	7 Februari 2023	Konsultasi Revisi Kajian Agama BAB I	5.
6.	21 Februari 2023	Konsultasi Kajian Agama BAB II	6.
7.	9 Maret 2023	Konsultasi Revisi Kajian Agama BAB II	7.
8.	12 April 2023	ACC Seminar Proposal	8.
9.	4 Mei 2023	Konsultasi Revisi Seminar Proposal	9.
10.	16 Mei 2023	Konsultasi BAB IV	10.
11.	29 Mei 2023	Konsultasi Revisi BAB IV dan Konsultasi BAB V	11.
12.	5 Juni 2023	Konsultasi Revisi BAB IV dan BAB V	12.
13.	6 Juni 2023	Konsultasi Kajian Agama BAB IV	13.
14.	7 Juni 2023	Konsultasi Revisi Kajian Agama BAB IV	14.



**KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933**

15.	8 Juni 2023	ACC Seminar Hasil	15. <i>[Signature]</i>
16.	25 September 2023	Konsultasi Revisi Seminar Hasil	16. <i>[Signature]</i>
17.	6 November 2023	ACC Sidang Skripsi	17. <i>[Signature]</i>
18.	20 November 2023	ACC Keseluruhan	18. <i>[Signature]</i>

Malang, 20 November 2023

Mengetahui,
Ketua Program Studi Matematika



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