

2019

Adaptive Moment Estimation Untuk Meminimalkan Kuadrat Error pada Algoritma Backpropagation

Singarimbun, Roy Nuary

Universitas Sumatera Utara

<http://repositori.usu.ac.id/handle/123456789/20255>

Downloaded from Repotori Institusi USU, Univseritas Sumatera Utara

***ADAPTIVE MOMENT ESTIMATION UNTUK MEMINIMALKAN
KUADRAT ERROR PADA ALGORITMA BACKPROPAGATION***

TESIS

**ROY NUARY SINGARIMBUN
157038007**



**PROGRAM STUDI S2 TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2019**

***ADAPTIVE MOMENT ESTIMATION UNTUK MEMINIMALKAN
KUADRAT ERROR PADA ALGORITMA BACKPROPAGATION***

TESIS

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Magister Teknik Informatika

ROY NUARY SINGARIMBUN
157038007



PROGRAM STUDI S2 TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2019

PERSETUJUAN

Judul : *ADAPTIVE MOMENT ESTIMATION UNTUK MEMINIMALKAN KUADRAT ERROR PADA ALGORITMA BACKPROPAGATION*

Kategori : TESIS

Nama : ROY NUARY SINGARIMBUN

Nomor Induk Mahasiswa : 157038007

Program Studi : MAGISTER (S2) TEKNIK INFORMATIKA

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA.

Komisi Pembimbing :

Pembimbing 2,

Dr. Erna Budhiarti Nababan, M.I.T

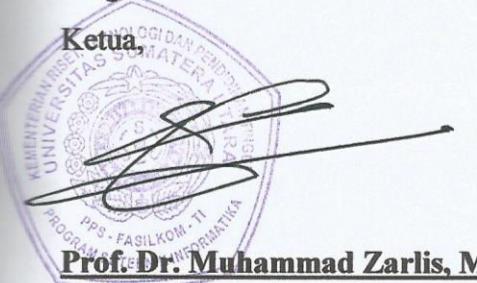
Pembimbing 1,

Prof. Dr. Opim Salim Sitompul, M.Sc

Diketahui/disetujui oleh

Program Studi S2 Teknik Informatika

Ketua,



Prof. Dr. Muhammad Zarlis, M.Sc

NIP. 19570701 198601 1 003

PERNYATAAN***ADAPTIVE MOMENT ESTIMATION UNTUK MEMINIMALKAN KUADRAT
ERROR PADA ALGORITMA BACKPROPAGATION*****TESIS**

Saya mengakui bahwa tesis ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing–masing telah disebutkan sumbernya.

Medan, 16 Agustus 2019

ROY NUARY SINGARIMBUN
157038007

**PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS**

Sebagai sivitas akademika Universitas Sumatera Utara, saya yang bertanda tangan di bawah ini:

Nama	:	Roy Nuary Singarimbun
NIM	:	157038007
Program Studi	:	Magister (S2) Teknik Informatika
Jenis Karya Ilmiah	:	Tesis

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Sumatera Utara Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas tesis saya yang berjudul:

***ADAPTIVE MOMENT ESTIMATION UNTUK MEMINIMALKAN KUADRAT
ERROR PADA ALGORITMA BACKPROPAGATION***

Berserta perangkat yang ada (*jika diperlukan*). Dengan Hak Bebas Royalti Non-Eksklusif ini, Universitas Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan tesis saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan/atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 16 Agustus 2019

ROY NUARY SINGARIMBUN

NIM. 157038007

Telah diuji pada

Tanggal : 16 Agustus 2019

PANITIA PENGUJI TESIS

Ketua : Prof. Dr. Opim Salim Sitompul, M. Sc.

Anggota : 1. Dr. Erna Budhiarti Nababan, M. IT.

2. Prof. Dr. Muhammad Zarlis, M. Sc.

3. Dr. Syahril Efendi, S. Si., M. IT.

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Roy Nuary Singarimbun
 Tempat dan Tanggal Lahir : Medan, 06 Januari 1990
 Alamat Rumah : Jalan Kopi Raya – I No. 16 LK. VI Perumnas Simalingkar Kelurahan Mangga Kecamatan Medan Tuntungan Kota Medan.
 Handphone / Telepon : -
 E-mail : roy90singarimbun@gmail.com

DATA PENDIDIKAN

SD	: SD Swasta Timbul Jaya 2	TAMAT: 2001
SMP	: SMP Negeri 31 Medan	TAMAT: 2004
SMA	: SMA Negeri 17 Medan	TAMAT: 2007
S1	: STMIK Logika	TAMAT: 2011
S2	: Teknik Informatika USU	TAMAT: 2019

UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala limpahan rahmat dan karunia-Nya yang diberikan kepada penulis sehingga dapat menyelesaikan penyusunan tesis ini sebagai syarat untuk memperoleh ijazah Magister Teknik Informatika. Penulis menyadari bahwa dalam pelaksanaannya banyak pihak yang telah membantu hingga selesaiya tesis ini.

Untuk itu penulis ingin menyampaikan ucapan terima kasih yang sebesar – besarnya kepada :

1. Kedua orang tua penulis tercinta yang telah memberikan kasih sayangnya, doa yang tidak pernah putus, serta dorongan moril maupun materil kepada penulis sehingga dapat menyelesaikan tesis ini dengan baik.
2. Bapak Prof. Dr. Runtung Sitepu, S.H., M.Hum., selaku rektor Universitas Sumatera Utara.
3. Bapak Prof. Dr. Opim Salim Sitompul, M.Sc., selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara Medan.
4. Bapak Prof. Dr. Muhammad Zarlis, M.Sc, selaku Ketua Program Studi Pascasarjana Teknik Informatika Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara Medan.
5. Bapak Dr. Syahril Efendi, S.Si., M.IT., selaku Sekretaris Program Studi Magister Teknik Informatika Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
6. Bapak dan ibu dosen pembimbing Prof. Dr. Opim Salim Sitompul, M.Sc., dan Dr. Erna Budhiarti Nababan, M.IT., yang telah banyak memberikan masukkan dan arahan kepada penulis.
7. Bapak Prof. Dr. Muhammad Zarlis, M.Sc., dan bapak Dr. Syahril Efendi, S.Si., M.IT., selaku Dosen Pembanding yang telah memberikan kritik dan saran kepada penulis.

8. Bapak dan Ibu dosen yang telah memberikan materi perkuliahan dan ilmu pengetahuan selama penulis menyelesaikan Program Studi Magister Teknik Informatika.
9. Seluruh staf atau pegawai pada Program Studi Magister Teknik Informatika Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
10. Teman – teman seperjuangan angkatan 2015 Kom – A khususnya kepada Kristina Annatasia Br. Sitepu dan Zulkarnain Lubis yang saling memberikan dukungan dan sama – sama berjuang didalam menyelesaikan tesis.
11. Kakanda Ristika Julianty Singarimbun dan Adinda Aldo Putra Singarimbun yang telah memberikan kasih sayangnya, doa yang tidak pernah putus, serta dorongan moril kepada penulis untuk dapat menyelesaikan tesis.
12. Abangnda Jhohannes H. Purba yang telah membantu dan memberikan motivasi sehingga penulis dapat menyelesaikan tesis ini.

Penulis menyadari bahwa masih banyak kekurangan di dalam tesis ini, untuk itu penulis mengharapkan saran dan kritik yang bersifat membangun dari semua pihak demi kesempurnaan penelitian selanjutnya. Akhir kata penulis mengucapkan terima kasih.

Medan, 16 Agustus 2019

Penulis,

ROY NUARY SINGARIMBUN

NIM. 157038007

ABSTRAK

Backpropagation didalam *Artificial neural network (ANN)* adalah suatu metode *gradient descent* untuk meminimalkan nilai kuadrat *error* pada *output*. Di dalam proses pembelajaran algoritma *backpropagation* masih lambat untuk meminimalkan nilai kuadrat *error* dan *mean square error (MSE)* diterapkan sebagai gantinya. Hal ini dikarena jumlah *epoch* pada pelatihan masih tinggi yang menyebabkan penyelesaian sering terjebak ke lokal optima. Sehingga dibutuhkan suatu pendekatan yang sesuai agar dapat meningkatkan proses pembelajaran untuk mendapatkan penurunan nilai *MSE* didalam pengujian. Dengan meningkatkan proses pembelajaran metode *gradient descent* pada *backpropagation* untuk meminimalkan nilai *MSE*, diharapkan dapat menurunkan nilai *MSE* disetiap *epoch* pada *output* pengujian *neural network*. *Adaptive Moment Estimation (ADAM)* yang merupakan salah satu metode mengoptimalkan banyak parameter untuk melatih *Deep Neural Network (DNN)* dari data yang cukup besar. *ADAM* dapat mempercepat pelatihan dan mendapatkan tingkat akselerasi, sehingga pengoptimalan banyak parameter dalam komputasi menjadi lebih rendah. Dari pengujian yang dilakukan dengan nilai *epoch* yang berbeda – beda, didapatkan nilai *MSE* yang minimum pada *epoch* ke 1000 adalah 0.002479194. Sedangkan dari pengujian dengan *epoch* 2000, didapatkan nilai *MSE* yang minimum adalah 0.002281315. Sehingga hasil dari penelitian ini, dapat memberikan hasil untuk meminimalkan nilai *MSE* dengan nilai *epoch* yang berbeda – beda pada *output* didalam *ANN*.

Kata kunci: *Artificial Neural Network (ANN), Gradient Descent, Backpropagation, Meminimalkan Mean Square Error (MSE), Adaptive Moment Estimation (ADAM)*.

ADAPTIVE MOMENT ESTIMATION TO MINIMIZE SQUARE ERROR IN BACKPROPAGATION ALGORITHM

ABSTRACT

Backpropagation in the Artificial Neural Networks (ANN) is a gradient descent method to minimize the value of square error on output. In learning algorithm, the backpropagation algorithm is insufficient to minimize the value of square error and the mean square error (MSE) is applied instead. This occurs due to the high number of epoch in training and often gets trapped in local optima. Therefore, an appropriate approach is needed to improve the learning gaining the decrease MSE value on testing. Through the improvement of gradient descent method learning in backpropagation to minimize the value of MSE, this is expected to reduce the value of MSE in each epoch on output neural network testing. Adaptive Moment Estimation (ADAM) is a method of parameter optimization to train Deep Neural Network (DNN) on large scale data. ADAM is able to speed up training and obtains acceleration rate, so the optimization of parameters in computing gets lower. From tests conducted with different epoch values, it was found that the minimum mean square error (MSE) value of the 1000 epoch was 0.002479194. On 2000 epoch testing, the minimum MSE value found was 0.002281315. So the results of this research, can provide results to the minimization of MSE values with different epoch values in contrast with the output in ANN.

Keywords: *Artificial Neural Network (ANN), Gradient Descent, Backpropagation, MSE Minimization, Adaptive Moment Estimation (ADAM).*

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PERSETUJUAN PUBLIKASI	iv
PANITIA PENGUJI	v
RIWAYAT HIDUP	vi
UCAPAN TERIMA KASIH	vii
ABSTRAK	ix
ABSTRACT	x
DAFTAR ISI	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xvi
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	6
1.3. Tujuan Penelitian	6
1.4. Batasan Maalah	6
1.5. Manfaat Penelitian	6
BAB 2 TINJAUAN PUSTAKA.....	7
2.1. Jaringan Saraf Tiruan (<i>Artificial Neural Networks</i>).....	7
2.1.1. Arsitektur jaringan saraf tiruan	7
A. Penggunaan jaringan <i>feed forward propagation</i>	9
B. Penggunaan jaringan <i>feed backward propagation</i>	10
C. Tahap perubahan bobot dan bias.....	10

2.1.2. Jaringan saraf <i>backpropagation</i>	10
1. Lapisan masukkan (<i>input layer</i>).....	11
2. Lapisan tersembunyi (<i>hidden layer</i>).....	11
3. Lapisan keluaran (<i>output layer</i>)	12
2.2. Pergantian <i>Missing Values</i> Menggunakan Metode Median	12
2.3. Normalisasi Data	13
2.4. <i>Gradient Descent Backpropagation</i>	13
2.5. <i>Adaptive Moment Estimation (ADAM)</i> Pada <i>Gradient Descent Backpropagation</i>	19
2.6. Penelitian Terdahulu.....	22
2.7. Perbedaan Dengan Riset Yang Lain	24
2.8. Kontribusi Riset.....	24
BAB 3 METODOLOGI PENELITIAN	25
3.1. Diagram Blok Penelitian.....	25
3.1.1 Data yang digunakan	26
3.1.2 Pra - <i>processing</i>	27
1. <i>Sorting</i> data	27
2. <i>Cleaning</i> data	28
3. Normalisasi data.....	29
3.2. Arsitektur <i>Artificial Neural Network</i>	30
3.2.1 Rancangan <i>backpropagation neural network</i>	30
3.2.2 <i>Input layer</i>	31
3.2.3 <i>Hidden layer</i>	32
3.2.4 <i>Ouput layer</i>	33
3.3. Rancangan Arsitektur <i>Adaptive Moment Estimation (ADAM)</i> Pada <i>Gradient Descent Backpropagation</i>	33
3.4. Analisis <i>Adaptive Moment Estimation (ADAM)</i> Pada <i>Gradient Descent Backpropagation</i>	35
1. <i>Input</i> data awal.....	35
2. Proses <i>feed forward propagation</i>	35
3. Proses <i>feed backward propagation</i>	36

BAB 4	HASIL DAN PEMBAHASAN	79
4.1.	Hasil Penelitian	79
4.1.1	Pengujian pertama <i>maximum epoch</i> 1000	79
4.1.2	Pengujian kedua <i>maximum epoch</i> 2000	83
4.2.	Pembahasan.....	86
BAB 5	KESIMPULAN DAN SARAN	87
5.1.	Kesimpulan	87
5.2.	Saran	87
DAFTAR PUSTAKA.....		88
LAMPIRAN.....		91

DAFTAR TABEL

	Halaman
Tabel 2.1. Penelitian terdahulu	22
Tabel 3.1. Deskripsi dari dataset WBCD	26
Tabel 3.2. <i>Breast cancer</i> dataset	27
Tabel 3.3. Dataset pada kelas kanker jinak	28
Tabel 3.4. Dataset pada kelas kanker ganas	28
Tabel 3.5. Normalisasi data pada kelas kanker jinak	29
Tabel 3.6. Normalisasi data pada kelas kanker ganas	29
Tabel 3.7. Nilai <i>input</i> setelah di normalisasi dan <i>output</i>	36
Tabel 3.8. Nilai <i>random weight</i> dan <i>bias</i> dari <i>input layer</i> ke <i>hidden layer</i> .	37
Tabel 3.9. Nilai <i>random weight</i> dan <i>bias</i> dari <i>hidden layer</i> ke <i>output layer</i>	37
Tabel 3.10.Nilai parameter pada pelatihan jaringan	38
Tabel 3.11.Nilai jaringan pada <i>neuron input layer</i> ke <i>neuron hidden layer</i>	39
Tabel 3.12.Nilai <i>derivative partial error</i> terhadap <i>weight</i> dan <i>bias</i> pada <i>output layer</i> di setiap <i>neuron</i>	42
Tabel 3.13.Nilai perhitungan <i>ADAM</i> pertama pada estimasi momen pertama terhadap <i>weight</i> dan <i>bias</i> pada <i>output layer</i> di setiap <i>neuron</i> ...	44
Tabel 3.14.Nilai perhitungan koreksi <i>weight</i> dan <i>bias ADAM</i> pertama pada estimasi momen pertama pada <i>output layer</i> di setiap <i>neuron</i> ...	45
Tabel 3.15.Nilai perhitungan <i>ADAM</i> pertama pada estimasi momen kedua terhadap <i>weight</i> dan <i>bias</i> pada <i>output layer</i> di setiap <i>neuron</i> ...	47
Tabel 3.16.Nilai perhitungan koreksi <i>weight</i> dan <i>bias ADAM</i> pertama pada estimasi momen kedua pada <i>output layer</i> di setiap <i>neuron</i>	49
Tabel 3.17. Nilai perhitungan <i>derivative partial error</i> jaringan <i>E</i> terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>	51
Tabel 3.18. Nilai perhitungan <i>ADAM</i> kedua pada estimasi momen pertama terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i> ..	57
Tabel 3.19.Nilai perhitungan koreksi <i>weight</i> dan <i>bias ADAM</i> kedua pada	

estimasi momen pertama pada <i>hidden layer</i> di setiap <i>neuron</i> ..	61
Tabel 3.20. Nilai perhitungan <i>ADAM</i> kedua pada estimasi momen kedua terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i> ..	65
Tabel 3.21. Nilai perhitungan koreksi <i>weight</i> dan <i>bias ADAM</i> kedua pada estimasi moment kedua pada <i>hidden layer</i> di setiap <i>neuron</i>	69
Tabel 3.22.Nilai perhitungan <i>update weight</i> dan <i>bias</i> pada <i>output layer</i> di setiap <i>neuron</i>	73
Tabel 3.23. Nilai perhitungan <i>update weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>	75

DAFTAR GAMBAR

	Halaman
Gambar 2.1. <i>Artificial neural network</i>	8
Gambar 2.2. <i>Multilayer artificial neural network</i>	9
Gambar 3.1. Diagram blok penelitian	25
Gambar 3.2. Rancangan <i>backpropagation neural network</i>	30
Gambar 3.3. Rancangan arsitektur <i>ADAM</i> pada <i>gradient descent backpropagation</i>	34
Gambar 4.1. Nilai MSE dalam lapisan <i>output</i> pada pengujian pertama.....	80
Gambar 4.2. Grafik pengujian pertama nilai MSE	82
Gambar 4.3. Nilai MSE dalam lapisan <i>output</i> pada pengujian kedua	83
Gambar 4.4. Grafik pengujian kedua nilai MSE.....	85

BAB 1

PENDAHULUAN

1.1. Latar Belakang

ADAM adalah *Adaptive Moment Estimation* yang merupakan salah satu metode mengoptimalkan banyak parameter untuk melatih *Deep Neural Network (DNN)* dari data yang cukup besar. Pada penelitian sebelumnya yang dilakukan Guan *et al.* (2017) dengan algoritma *Asynchronous Stochastic Gradient Descent* (ASGD) didalam struktur jaringan untuk melatih *DNN*, dimana *DNN* mempelajari representasi *hierarchical* dari data besar yang biasanya berisi banyak parameter pada *cluster*. Namun, sering meningkatnya *node/unit input* pada komputasi dan ukuran data, ASGD sering terjadi konvergensi lambat. Sehingga dibutuhkan algoritma *Delay Compensated Asynchronous ADAM* (DC – ADAM) untuk melatih DNN. Secara khusus, DC – ADAM pada pembaruan parameter dengan kenaikan momen yang merupakan pembagian momen pertama dan momen kedua. DC – ADAM konvergensi jauh lebih cepat dari pada ASGD pada *cluster* dengan *node* komputasi yang moderat (ukuran *node* tidak kecil maupun besar). Hasil percobaan, menunjukkan DC – ADAM dapat mempercepat pelatihan dan mencapai laju percepatan yang hampir *linier* dengan peningkatan *node* pada komputasi.

Wakitani *et al.* (2017) melakukan penelitian algoritma pembelajaran *online* dari *Group Method Of Data Handling Based Proportional – Integral – Derivative* (GMDH – PID) untuk sistem nonlinear. *Controller* GMDH – PID dapat mempertahankan kinerja kontrol yang baik dengan mengatur koefisien bobotnya secara tepat di jaringan GMDH. Namun koefisien bobotnya ditentukan dengan cara yang tidak tepat, sehingga pengontrol GMDH – PID tidak dapat mengadaptasi sistem jika terjadi perubahan sistem yang tidak terduga saat mengendalikan. Sehingga dalam pengaturan *online* GMDH – PID dilakukan dengan menggunakan ADAM. ADAM dapat beradaptasi dengan perubahan sistem, sehingga pengontrolan dapat diperluas dan mengoptimalkan banyak parameter dengan komputasi rendah.

Wu *et al.* (2016) pada penelitiannya Teknologi *OTN* (*Optical Transmission Network*) telah banyak digunakan dalam *Smart Grid*, yang meningkatkan kecepatan transmisi, efisien transmisi dan kapasitas transmisi jaringan. Untuk mengawasi pengoperasian jaringan secara *real time* dan meningkatkan kualitas layanan (*Quality Of Services/QOS*), diperlukan penilaian kuantitatif terhadap kualitas operasi pada teknologi OTN. Pada penelitian ini, melakukan simulasi dengan metode evaluasi *Levenberg Marquardt Backpropagation (LM-BP)* untuk menilai kualitas operasi pada teknologi OTN di *smart grid* dengan data operasi 10 tahun terakhir dari teknologi OTN di provinsi pada sebuah perusahaan tenaga listrik *smart grid*. Hasil simulasi menunjukkan bahwa jaringan saraf *BP* dioptimalkan oleh *LM* memiliki waktu pelatihan yang lebih singkat, tingkat konvergensi yang lebih cepat dan akurasi prediksi yang lebih baik dari pada jaringan saraf *BP* standar.

Artificial Neural Network (ANN) memiliki masalah utama dalam penyesuaian yang tepat dari bobot. Ada sejumlah penelitian yang dilakukan untuk perbandingan proses pembelajaran ANN berbasis evolusi dan gradient. Tetapi hasil penelitian saling bertentangan meskipun pengembangan dataset yang sama dan standar telah dilakukan. Ahmad *et al.* (2010) melakukan penelitian untuk membandingkan antara variasi *gradient descent* dan *Genetic Algorithm* (GA) berdasarkan pelatihan ANN dengan menggunakan *dataset benchmark* kanker dan diabetes. Pada GA dengan efek operasi *crossover* menunjukkan persentase kesalahan klasifikasi sedikit lebih baik dari pada *gradient descent* yang dilakukan pada *dataset* kanker, tetapi pada *dataset* diabetes *gradient descent* jauh lebih baik dari pada GA. Dari penelitian tersebut masih perlu dilakukan percobaan dengan lebih banyak *dataset* dalam pelatihan ANN.

Metode *artificial neural network* (ANN) semakin populer untuk membangun model prediksi yang menghasilkan prediksi nilai *error* yang kecil. Untuk memiliki model yang baik, ANN membutuhkan *dataset* yang besar sebagai *input*. ANN *backpropagation* adalah metode penurunan gradien (*gradient descent*) untuk meminimalkan kuadrat *error* pada *output*. Pada penelitian yang dilakukan Mahasagara *et al.* (2017) untuk pergerakan harga saham sesuai dengan persyaratan ANN ini adalah kumpulan data yang besar karena harga saham dicatat hingga setiap detik yang biasa

disebut data frekuensi tinggi. Model prediksi membantu investor dalam membangun portofolio saham dan proses pengambilan keputusan mereka, sehingga prediksi portofolio mendapatkan nilai *error* terkecil untuk kondisi resiko.

Singh *et al.* (2015) pada penelitiannya mengevaluasi dan menguji kinerja dari 3 (tiga) *gradient descent* berbasis *backpropagation artificial neural network* untuk mengklasifikasi tumor jinak dan tumor ganas dalam pencitraan *ultrasound*. Gambar – gambar ultrasonik diproses oleh *filter wavelet* untuk mengurangi *noise speckle*. Dengan menggunakan lima puluh tujuh atribut tekstur dan bentuk di ekstraksi dari gambar USG payudara yang disaring untuk mengklasifikasi tumor payudara. *Area* di bawah dari *receiving operation curve* (kurva operasi penerima), sensitivitas, spesifisitas, akurasi klasifikasi dan waktu CPU digunakan sebagai angka kelayakan pengklasifikasi. Hasilnya *adaptive gradient descent* pada *backpropagation* berdasarkan tingkat pembelajaran, variabel lebih baik dari teknik lain yang memberikan akurasi klasifikasi tertinggi 84,6%

Heravi & Hodtani (2018) dalam pelatihan jaringan saraf untuk mendapatkan nilai *mean square error* (MSE) merupakan permasalahan yang digunakan didalam proses pembelajaran. Pada penelitian menggunakan *correntropy-based conjugate gradient BP (CCG-BP)*. Algoritma CCG – BP berkonvergensi lebih cepat dari pada *backpropagation* berbasis *correntropy standard* berdasarkan nilai MSE. Hasil dari numerik untuk beberapa sampel aproksimasi fungsi, estimasi fungsi sintetik, dan prediksi deret waktu menggambarkan bahwa metode BP baru lebih kuat dari pada metode berbasis pada nilai MSE dalam arti kebisingan impulsif terutama ketika SNR (*Signal to Noise Ratio*) rendah.

Popa (2014) pada penelitian ini untuk meningkatkan pembelajaran algoritma *gradient descent* untuk jaringan saraf tiruan yang bernilai kompleks. Algoritma yang disempurnakan seperti yang paling dikenal untuk jaringan saraf bernilai ril adalah: *quickprop*, *resilient backpropagation*, *Delta-Bar-Delta* dan *Super SAB* untuk memperluas metode pembelajaran pada jaringan saraf. Varian yang kompleks dari empat algoritma yang dicontohkan pada berbagai masalah pendekatan fungsi, dan pendekatan pada aplikasi prediksi deret waktu. Dari penelitian didapat bahwa

algoritma *gradient descent* yang ditingkatkan lebih baik dari pada *gradient descent clasic/standard* dan *gradient descent momentum*. Untuk perkiraan fungsi sintetik, algoritma *Delta-Bar-Delta* yang lebih stabil dan algoritma *resilient backpropagation*, sedangkan prediksi deret waktu algoritma *quickprop* dan *Super SAB* yang tidak stabil dikarenakan metode ini cenderung terjebak kelokal minimum. Dari penelitian masih perlu dilakukan percobaan karena memiliki masalah pada fungsi sintetik, dan percobaan masih memungkinkan terjebak dalam lokal optima.

Li *et al.* (2011) melakukan penelitian untuk meningkatkan konvergensi dan kemampuan pencarian global pada jaringan *backpropagation* (BP). BP memiliki kelemahan seperti konvergensi lama, *over – fitting* dan mudah terjebak ke lokal optima. *Genetic Algorithm* (GA) dapat memperbaiki BP dengan mengevaluasi *natural selection*, *genetic crossover* dan *gen mutation* yang memiliki kelebihan seperti paralel yang sangat tinggi, stokastik dan pencarian probabilitas global yang dapat mengatasi kekurangan BP. GA merupakan algoritma pengoptimalan acak dengan kemampuan optimum global untuk dapat memberikan resolusi yang sesuai dengan masalah yang ada pada BP.

Achkar *et al.* (2016) pada penelitiannya menggunakan *Adaptive Modified Backpropagation Approach (AMBP)* dalam meningkatkan kinerja Algoritma *Artificial Intelligence Modern*. Algoritma *backpropagation standard* dapat memperbarui bobot dengan tingkat pembelajaran yang tepat, untuk dioptimalkan menggunakan *trial* dan *error*. *AMBP* dapat mempercepat konvergensi dengan menyesuaikan laju pembelajaran (*learning rate*) pada setiap lapisan dan pada setiap *epoch*. Kemudian kesalahan yang didapat secara signifikan berkurang dengan mempertimbangkan kesalahan linier baru beserta kesalahan nonlinear standar. Algoritma *AMBP* untuk memantau dan mengklasifikasikan gerak manusia ditunjukkan dengan membandingkan kinerja sistem, kecepatan konvergensi dan kesalahan akhir, dengan algoritma standar yang diperoleh. Hasil penelitian, *AMBP* mampu belajar lebih cepat dari pada *standard backpropagation (SBP)* dan *backpropagation with momentum (BPM)* jauh lebih baik. Selain itu *AMBP* mendapatkan MSE yang lebih rendah dalam jangka panjang dari pada *SBP* dan *BPM*.

Jaringan saraf tiruan (JST) dengan menggunakan pelatihan *levenberg – marquardt* untuk prediksi deret waktu keuangan. JST berdasarkan algoritma pelatihan *levenberg-marquardt (LM)* mengungguli *gradient descent*, *conjugate gradient* dan algoritma lainnya yang menggunakan turunan orde pertama dari indeks kinerja untuk mengoptimalkan bobot pada JST. Metode statistik dan JST merupakan metode yang digunakan untuk memprediksi. Namun JST tidak membuat struktur dasar sistem dibandingkan pada metode statistik. JST juga merupakan regresi linier yang bersifat kompleks, nonlinier dan dinamis. Algoritma *levenberg – marquardt* untuk mendekati kecepatan latihan, supaya fungsi kinerja yang memiliki bentuk jumlah kuadrat dalam pelatihan jaringan *feedforward*. Hasilnya algoritma *levenberg - marquardt* menunjukkan bahwa *error* yang relatif kurang dari 3 % (Mammadli, 2017).

Andayani *et al.* (2017) pada penelitiannya untuk mengoptimalkan proses algoritma *backpropagation* dengan menambahkan metode Nguyen – Widrow pada lapisan *input* dari proses *feed – forward* dan menyesuaikan parameter *learning rate* (laju pembelajaran) pada proses *backpropagation*. Dalam *preprocessing* biasanya data belum dinormalisasi sehingga signifikan terhadap *target output* yang perlu dikurangi pada proses didalam lapisan *input*. Dengan metode Nguyen – Widrow yang ada didalamnya untuk mengadaptasi bobot dari data hasil yang akan disesuaikan dengan bobotnya. Nilai yang berubah akan mempengaruhi proses dalam langkah *feed forward*. Jika nilai lebih tinggi dari *target output*, proses akan terus menyesuaikan tingkat pembelajaran pada langkah *backward*. Hasilnya pada fase *feed – forward* dengan metode bobot awal Nguyen – Widrow mampu memberi nilai mendekati nilai *error* yang mempengaruhi *update* bobot ke fase *backward*. Pada hasil fase *backward* parameter *adaptive learning rate* dapat megurangi jumlah iterasi (*epoch*).

Dari uraian diatas, penulis membuat penelitian dengan judul “*Adaptive Moment Estimation Untuk Meminimalkan Kuadrat Error Pada Algoritma Backpropagation*”.

1.2. Rumusan Masalah

Backpropagation didalam *Artificial neural network* adalah suatu metode penurunan gradien (*gradient descent*) untuk meminimalkan nilai kuadrat *error* pada *output*. Proses pembelajaran algoritma *backpropagation* masih lambat dalam meminimalkan nilai kuadrat *error*, hal ini dikarena jumlah *epoch* pada pelatihan masih tinggi yang menyebabkan penyelesaian sering terjebak ke lokal optima. Sehingga dibutuhkan satu pendekatan yang sesuai agar dapat meningkatkan proses pembelajaran untuk mendapatkan nilai kuadrat *error* terkecil didalam pengujian.

1.3. Tujuan Penelitian

Tujuan penelitian ini yaitu meningkatkan proses pembelajaran metode *gradient descent* pada *backpropagation* dalam klasifikasi dengan menambahkan metode *ADAM* pada *output layer* dan *ADAM* pada *hidden layer* didalam proses *backward propagation* untuk meminimalkan nilai kuadrat *error*, sehingga untuk mendapatkan nilai kuadrat *error* terkecil didalam pengujian dengan jumlah *epoch* yang sedikit pada keluaran *neural network*.

1.4. Batasan Masalah

Batasan masalah perlu dilakukan agar penelitian dapat terarah dengan baik dan tidak menyimpang dari pokok permasalahan adalah membahas proses pembelajaran metode *ADAM* pada *gradient descent backpropagation* untuk meminimalkan kuadrat *error* dengan menggunakan *mean square error* pada keluaran *neural network*. Penelitian ini menguji *maximum epoch* 1000 dan *maximum epoch* 2000.

1.5. Manfaat Penelitian

Penelitian ini diharapkan dengan menggunakan metode *ADAM* dapat memberikan manfaat pada penggunaan Jaringan Saraf Tiruan (*Artificial Neural Network*) dalam meningkatkan proses kinerja pada metode *gradient decent backpropagation* untuk meminimalkan nilai kuadrat *error* pada *output* didalam *neural network*.

BAB 2

TINJAUAN PUSTAKA

2.1. Jaringan Saraf Tiruan (*Artificial Neural Networks*)

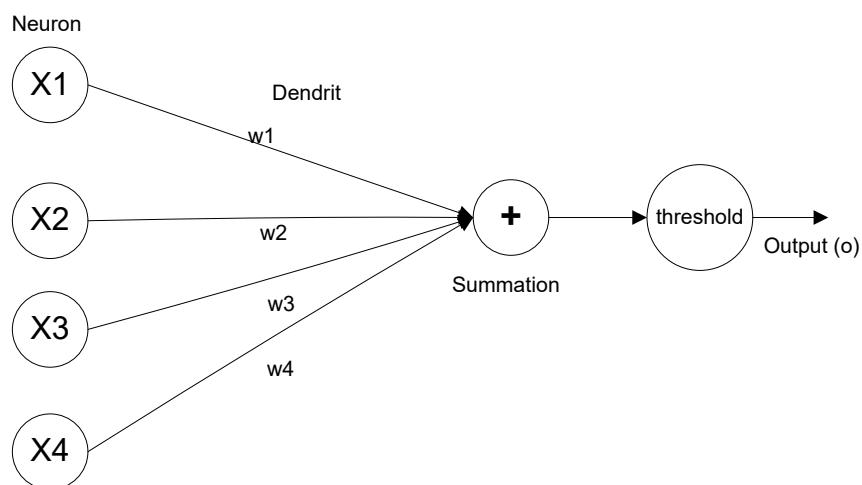
Teixeira & Fernandes (2012) pada penelitiannya Jaringan Saraf Tiruan adalah model kompetitif dibandingkan dengan metodologi lain seperti model ARIMA *time series* atau model *linear*, *Feedforward*, *Cascade Forward* dan arsitektur *Recurrent* sering dibandingkan. Metodologi Jaringan Saraf Tiruan telah membangkitkan minat yang besar dibidang ilmu ekonomi dan bisnis, karena dari penelitian yang dilakukan sejauh ini dapat dilihat untuk mewakili alternatif yang *valid* terhadap metode peramalan klasik untuk situasi yang sulit diobati dengan metode klasik. Jaringan saraf tiruan bertujuan untuk membuat jaringan saraf seperti jaringan saraf manusia yang mampu belajar terhadap lingkungan dan dapat mengelola lingkungan berdasarkan pengalaman yang didapat.

2.1.1 Arsitektur jaringan saraf tiruan

Abraham (2005) Jaringan Saraf Tiruan terdiri dari *artificial neuron* (*neuron* buatan) atau simpul/*nodes*. Arsitektur dasar terdiri dari tiga jenis lapisan *Neuron* yaitu *input layer*, *hidden layer* (lapisan tersembunyi) dan *output layer*. Di dalam jaringan ada proses *feedforward* (umpan maju) dimana aliran sinyal *input* ke *unit output*. Jaringan saraf harus di atur sedemikian rupa sehingga aplikasi dari *set input* menghasilkan *set output* yang diinginkan. Salah satu cara metode untuk mengatur kekuatan koneksi adalah dengan mengatur bobot secara eksplisit, dan cara lain untuk melatih jaringan saraf memberikan pola/aturan pengajaran dan membiarkan mengubah bobotnya sesuai dengan beberapa aturan pembelajaran.

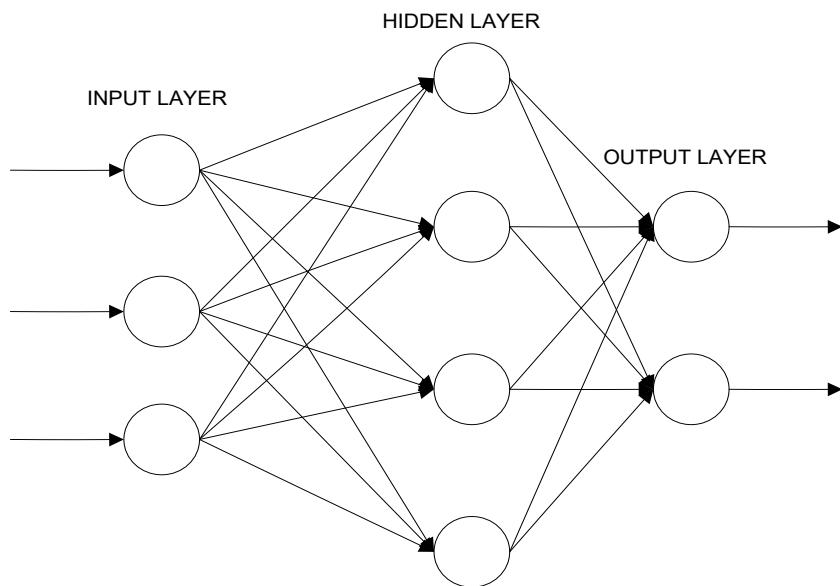
Jaringan saraf terdiri dari tiga klasifikasi yaitu *supervised learning* (pembelajaran terbimbing), *unsupervised learning* (pembelajaran tanpa pengawasan) dan *reinforcement learning* (pembelajaran penguatan). Pembelajaran terbimbing biasanya vektor *input* disiapkan pada *input* bersama dengan respon yang diinginkan yaitu satu sinyal *input* untuk setiap *node* pada lapisan *output*. *Forward pass* dilakukan dan

kesalahan (*error*) saling berhubungan untuk setiap *node* dilapisan *output* sampai ditemukan, sehingga digunakan perubahan bobot dalam jaringan sesuai dengan aturan pembelajaran yang berlaku. Contoh paling dikenal ada pada algoritma *backpropagation*, aturan *delta* dan aturan *perceptron*. Dalam pembelajaran tanpa pengawasan *unit output* dilatih untuk pola dalam *input*, dimana pada sistem seharusnya menemukan fitur secara statistik dari populasi *input*. Pada pembelajaran tanpa pengawasan sistem harus mengembangkan representasi rangsangan *input* nya sendiri. Sedangkan pada pembelajaran penguatan adalah belajar untuk memetakan situasi ke tindakan sehingga memaksimalkan sinyal yang berupa numerik. Berikut adalah gambar *artificial neuron* (*neuron* buatan) dan *multilayer artificial neural network* (lapisan banyak jaringan saraf tiruan) dapat dilihat pada gambar 2.1. dan gambar 2.2.:



Gambar 2.1 *Artificial neural network*

Gambar 2.1. *Artificial Neuron Network* pada *input neuron* tertentu mengumpulkan sinyal X_1 , X_2 , X_3 dan X_4 berupa rangsangan dari neuron lain melalui *dendrit* w_1 , w_2 , w_3 dan w_4 . Sinyal yang datang dan diterima oleh *dendrit* akan dijumlahkan (*summation*). Sinyal tersebut akan diterima oleh neuron lain jika memenuhi nilai *threshold* tertentu.



Gambar 2.2. *Multilayer artificial neural network*

Gambar 2.2. *Multilayer Artificial Neural Network* tersebut memiliki lapisan *input* tiga *unit neuron* yang terhubung dengan lapisan tersembunyi (*hidden layer*) yang memiliki empat *unit neuron*. Pada masing – masing *neuron hidden layer* terhubung ke lapisan *output* yang memiliki dua *unit neuron*.

Ada tiga tahap pada jaringan *backpropagation* yaitu tahap perambatan maju (*feed forward propagation*), tahap perambatan balik (*feed back propagation*) dan tahap perubahan bobot dan bias. Berikut adalah tiga tahap pada penggunaan jaringan sebagai berikut :

A. Penggunaan jaringan *feed forward propagation*

Feed forward propagation merupakan tahap awal pada jaringan saraf tiruan dimana setiap unit *input* menerima sinyal *input* dan meneruskan sinyal *input* tersebut kesemua *unit* pada lapisan tersembunyi (*hidden layer*). Setiap unit *hidden layer* menjumlahkan sinyal *input* dengan bobot yang sudah diinisialisasi secara acak terkecil. Setelah penjumlahan *hidden layer* dilakukan, hasil dari *hidden layer* tersebut dilakukan perhitungan dengan menerapkan fungsi aktivasi sigmoid supaya nilai *output* terletak pada interval 0 (nol) sampai 1 (satu). Setelah perhitungan *hidden layer* dengan menerapkan fungsi aktivasi *sigmoid*, sinyal *input* diteruskan ke lapisan *output* untuk

menjumlahkan bobot yang didapat pada sinyal *input*. Dan hasil penjumlahan pada lapisan *output* dilakukan kembali perhitungan fungsi aktivasi sigmoid pada lapisan *output*. Setelah nilai *output* didapat, pada proses *feed forward propagation* untuk *check error* pada setiap *epoch* bila *error* sudah mencapai proses berhenti. Proses yang belum mencapai, dilakukan proses *feed back propagation*.

B. Penggunaan jaringan *feed backward propagation*

Jaringan yang paling banyak digunakan adalah jaringan saraf *backpropagation*, yang juga dikenal sebagai *error* jaringan saraf *backpropagation* dan jaringan saraf *multilayer forward*. Jaringan saraf *backpropagation* terutama digunakan untuk tujuan berikut:

- a Pendekatan fungsi berarti bahwa aproksimasi fungsi dengan vektor masukan dan jaringan pelatihan vektor keluaran yang sesuai.
- b Pengenalan pola tertentu berarti hanya sebagian kecil vektor output dan vektor masukan spesifik terkait.
- c Mengklasifikasikan vektor input yang didefinisikan dengan metode yang sesuai.

Kompresi data berarti mengurangi dimensi vektor keluaran agar memudahkan transmisi atau penyimpanan.

C. Tahap perubahan bobot dan bias

Pada tahap perubahan bobot dan bias merupakan tahap dimana bertujuan untuk mengurangi/memiminimalkan nilai kuadrat *error* yang diterapkan pada proses jaringan saraf *backpropagation*. Salah satu cara meminimalkan nilai kuadrat *error* pada jaringan saraf *backpropagation* adalah dengan menggunakan metode *gradient descent* (penurunan gradien).

2.1.2 Jaringan saraf *backpropagation*

Zhao *et al.* (2012) pada penelitiannya algoritma *backpropagation* yang berbeda dapat mempengaruhi kecepatan, sedangkan jumlah *neuron hidden layer* (lapisan tersembunyi) juga merupakan faktor sensitif terhadap kinerja jaringan saraf. Jaringan saraf *backpropagation* digunakan untuk megklasifikasikan data berdasarkan penelitian jaringan saraf *backpropagation*, yang berfokus pada dampak simpul / *node hidden* dan

algoritma pelatihan pada kinerja jaringan. Jaringan saraf merupakan aplikasi dari model matematika pengolahan informasi yang mirip dengan koneksi sinapsi otak, ini adalah abstraksi, penyederhanaan dan simulasi otak manusia yang mencerminkan karakteristik dasar otak manusia. Jaringan saraf juga merupakan model komputasi, yang terdiri dari hubungan bobot antara sejumlah besar *neuron* dan satu sama lain. Setiap *neuron* mewakili fungsi aktivasi tertentu, hubungan antara dua *neuron* mewakili nilai bobot dari sinyal koneksi, yang setara dengan memori jaringan saraf. *Output* jaringan saraf yang terhubung ke jaringan, nilai bobot dan fungsi eksitasi. *Backpropagation* adalah metode penurunan gradien (*gradient descent*) untuk meminimalkan nilai kuadrat *error* pada keluaran (*output*).

Jaringan *Backpropagation* memiliki beberapa *neuron* yang berada dalam satu atau lebih lapisan tersembunyi (*hidden layer*). Setiap *neuron* yang berada dilapisan *input* terhubung dengan setiap neuron yang berada di *hidden layer*. Begitu juga pada *hidden layer*, setiap neuronnya terhubung dengan setiap neuron yang ada di *output layer*. Jaringan saraf tiruan *Backpropagation* terdiri dari banyak lapisan (*multi layer*), yaitu:

1 Lapisan masukkan (*input layer*)

Input layer sebanyak 1 (satu) lapis yang terdiri dari *neuron – neuron input*, mulai dari *neuron input* pertama sampai *neuron input* ke-*n*. *Input layer* merupakan penghubung yang mana lingkungan luar memberikan sebuah pola kedalam jaringan saraf. Sekali sebuah pola diberikan kedalam *input layer*, maka *output layer* akan memberikan pola yang lainnya (Heaton, 2008). Pada intinya *input layer* akan merepresentasikan kondisi yang dilatihkan kedalam jaringan. Setiap *input* akan merepresentasikan beberapa variabel bebas yang memiliki pengaruh terhadap *output layer*.

2 Lapisan tersembunyi (*hidden layer*)

Hidden layer berjumlah minimal 1 (satu) lapis yang terdiri dari *neuron – neuron tersembunyi* mulai dari *neuron tersembunyi* pertama sampai *neuron tersembunyi* ke-*p*. Menentukan jumlah neuron pada *hidden layer* merupakan bagian yang sangat penting dalam arsitektur jaringan saraf.

Menurut Heaton (2008), ada beberapa aturan yang dapat digunakan untuk menentukan banyaknya jumlah *neuron* pada *hidden layer* yaitu:

- a Jumlah *hidden neuron* harus berada diantara ukuran *input layer* dan *output layer*.
- b Jumlah *hidden neuron* harus $2/3$ dari ukuran *input layer*, ditambah ukuran *output layer*.
- c Jumlah *hidden neuron* harus kurang dari dua kali jumlah *input layer*.

Aturan – aturan tersebut hanya berupa pertimbangan dalam menentukan arsitektur jaringan saraf tiruan. Bagaimanapun, penentuan arsitektur jaringan akan kembali pada *trial and error* sesuai dengan masalah yang ditangani oleh jaringan.

3 Lapisan keluaran (*output layer*)

Output layer berjumlah satu lapis yang terdiri dari *neuron – neuron output* mulai dari *neuron output* pertama sampai *neuron output* ke-*m*. *Output layer* dari jaringan saraf adalah pola yang sebenarnya diberikan oleh lingkungan luarnya (*external environment*). Pola yang diberikan *output layer* dapat secara langsung ditelusuri kembali ke *input layer* nya. Jumlah dari *neuron output* tergantung dari tipe dan performa dari jaringan saraf itu sendiri.

2.2. Pergantian *Missing Values* Menggunakan Metode Median

Dataset yang dipakai pada penelitian ini adalah data dari *Wisconsin Breast Cancer Dataset* dari *University of California Irvine (UCI) Machine Learning Repository*. Data *breast cancer* memiliki 9 (sembilan) atribut dan 1 (satu) atribut *output* yang menghasilkan 2 (dua) kelas pada atribut *output* dalam dataset *breast cancer*. Jumlah dataset *breast cancer* yaitu 699 data yang memiliki *missing values* sebanyak 16 sampel data yang tidak memiliki nilai. Penggantian nilai-nilai yang hilang dengan menggunakan metode median sebagai berikut (Paulin & Santhakumaran 2010) :

1. Menghitung median pada variabel X_6 yaitu *Bare Nuclei* (atribut yang berisi nilai yang hilang) berjumlah 16 data dapat dihitung dengan menggunakan rumus sebagai berikut :

$$\text{MEDIAN} = \text{Size of } \frac{(N+1)}{2} \quad (2.1)$$

2. Semua nilai yang hilang dari atribut X_6 diganti dengan nilai hasil perhitungan median.

2.3. Normalisasi Data

Pada penelitian Chen *et al.* (2015) untuk meningkatkan *backpropagation neural network*, menggunakan fungsi normalisasi untuk meminimalkan kesalahan dari sistem. *Input variabel* merupakan elemen yang paling relatif untuk mengevaluasi model seperti *mean absolute percentage error (MAPE)* dan *mean square error (MSE)*. Tujuan normalisasi data adalah untuk nilai pada *input variable* bernilai kisaran [0, 1]. Berikut adalah persamaan untuk menormalisasi data pada *input variabel* sebagai berikut :

$$X'_i = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \quad (2.2)$$

Keterangan :

X_i	= nilai <i>input</i> data X_i ($X_i, i = 1, 2, 3, \dots, n$).
X_{\min}	= nilai <i>input</i> data min X_i ($\min(X_2, X_3, \dots, X_N), X_1 \times (1 - \alpha)$).
X_{\max}	= nilai <i>input</i> data max X_i ($\max(X_2, X_3, \dots, X_N), X_1 \times (1 + \alpha)$). dimana α adalah <i>markup</i> .

2.4. Gradient Descent Backpropagation

Menurut Srinivasan *et al.* (2002) pada penelitiannya algoritma *gradient descent backpropagation* dapat meminimalkan nilai *Square Error (SE)* untuk jaringan *multilayer feedforward neural*. Aturan pembelajaran untuk mengubah bobot (*weight*) dan *bias* pada lapisan *neuron output* dan juga *neuron hidden layer* (lapisan tersembunyi). Berikut adalah proses langkah – langkah *gradient descent backpropagation*:

1. Inisialisasi bobot / *weight*

Pertama kali yang dilakukan adalah penentuan bobot / *weight* secara acak (ambil nilai *random* yang paling kecil) pada setiap *neuron* yang berada pada *input layer*, *hidden layer* dan *output layer*.

2. Tahapan Perambatan maju (*feed forward propagation*) antara lain :

- Setiap *unit input* (X_i , $i = 1, 2, 3, \dots, n$) menerima sinyal X_i dan meneruskan sinyal tersebut ke semua *unit* pada *hidden layer* (lapisan tersembunyi).
- Proses perhitungan jaringan pada setiap *unit hidden layer* dengan menjumlahkan *weight* dan *bias* yang dikirim dari *neuron unit input layer*. *Input* jaringan ke *unit hidden layer* j -th menggunakan persamaan berikut :

$$net_{ij}^h = \sum_{i=0} X_i W_{ij}^h + w_{bias_j}^h \quad (2.3)$$

Keterangan :

net_{ij}^h : *Input* jaringan dari i – th *unit input layer* ke j – th *unit hidden layer*.

i : *neuron* pada *unit input layer*.

j : *neuron* pada *unit hidden layer*.

h : *hidden layer*.

X_i : nilai pada *unit input layer* dari jaringan.

W_{ij}^h : nilai bobot pada koneksi dari i – th *unit input layer* ke j – th *unit hidden layer*.

$w_{bias_j}^h$: nilai *bias* pada koneksi *unit input layer* ke j – th *unit hidden layer*.

- Setelah mendapatkan nilai perhitungan pada *unit hidden layer*, maka dilakukan perhitungan fungsi aktivasi pada *unit hidden layer* dengan menggunakan fungsi aktivasi *sigmoid* pada setiap *neuron* di *hidden layer* :

$$i_{ij} = f_{in_{net}n} = f_j^h(\mathbf{net}_{ij}^h) \quad (2.4)$$

Dimana : $f_j^h(\mathbf{net}_{ij}^h) = \frac{1}{1+e^{-(\mathbf{net}_{ij}^h)}}$

nilai e : nilai konstanta dengan nilai = 2.718281828

Keterangan :

i_{ij} : *Output* fungsi aktivasi dari jaringan i – th *unit input layer* ke j – th *unit hidden layer*.

i : *neuron* pada *unit input layer*.

j : *neuron* pada *unit hidden layer*.

$f_{in_net\ n}$: perhitungan fungsi aktivasi *neuron network* di setiap nilai *weight* dan *bias* pada perhitungan jaringan *neuron hidden layer*.

f_j^h : fungsi aktivasi dari j – th *unit hidden layer*.

net_{ij}^h : hasil nilai *input* jaringan dari i – th *unit input layer* ke j – th *unit hidden layer*.

- Perhitungan nilai jaringan pada *input layer* yang masuk ke *output layer* yang diterima dari perhitungan fungsi aktivasi pada *unit hidden layer* untuk node *output layer* k – th dengan menggunakan persamaan sebagai berikut:

$$\text{net}_{ik}^o = \sum_{j=6} (f_{in_net\ n} * w_{kj}^o) + w_{bias_k}^o \quad (2.5)$$

Keterangan :

net_{ik}^o : *Input* jaringan dari i – th *unit hidden layer* ke k – th *unit output layer*.

O : *output layer*.

i : *neuron* pada *unit hidden layer*.

k : *neuron* pada *unit output layer*.

j : *neuron* pada *unit hidden layer*.

$f_{in_net\ n}$: perhitungan fungsi aktivasi *neuron network* di setiap nilai *weight* dan *bias* pada perhitungan jaringan *neuron hidden layer*.

w_{kj}^o : nilai bobot pada koneksi fungsi aktivasi dari j – th *unit hidden layer* ke k – th *unit output layer*.

$w_{bias_k}^o$: nilai *bias* pada koneksi dari *unit hidden layer* ke k – th *unit output layer*.

- Setelah mendapatkan perhitungan nilai *weight* dan *bias* di *neuron network* yang masuk ke *neuron output layer*, maka dilakukan perhitungan fungsi aktivasi pada *neuron output layer* dengan menggunakan fungsi aktivasi *sigmoid* pada setiap *neuron* di *neuron output layer* :

$$O_{ik} = f_{out_net\ n} = f_k^o(\text{net}_{ik}^o) \quad (2.6)$$

Dimana : $f_k^o(\text{net}_{ik}^o) = \frac{1}{1+e^{-(\text{net}_{ik}^o)}}$

Keterangan :

O_{ik} : *Output* dari jaringan i – th *unit hidden layer* ke k – th *unit output layer*.

i : *neuron* pada *unit hidden layer*.

k : *neuron* pada *unit output layer*.

$f_{out_net\ n}$: perhitungan fungsi aktivasi *neuron unit output* pada nilai *weight* dan *bias* pada perhitungan jaringan *neuron output layer*.

f_k^o : fungsi aktivasi dari ke k – th *unit output layer*.

net_{ik}^o : *Input* jaringan dari i – th *unit hidden layer* ke k – th *unit output layer*.

- Pada satu proses *feed forward propagation* membentuk *output* dari jaringan saraf pada *output layer*, setelah proses tersebut dilakukan pengidentifikasi *gradient error* pada *output layer*. Berikut adalah persamaan untuk menghitung *error* pada setiap *output layer* sebagai berikut :

$$\text{error}_o = \text{Target} - O_{ik} \quad (2.7)$$

Keterangan :

error_o : perhitungan nilai *error* dari *output layer*.

Target : *output aktual / target aktual*.

O_{ik} : Nilai *output* dari i – th *unit hidden layer* ke k – th *unit output layer*.

i : *neuron* pada *unit hidden layer*.

k : *neuron* pada *unit output layer*.

- Menghitung *mean square error* pada penurunan gradient (*gradient descent*) untuk semua *output layer* adalah sebagai berikut :

$$\text{Mean Square Error} = \frac{1}{2} * \sum_{k=1} (\text{Target} - O_{ik})^2 \quad (2.8)$$

Keterangan :

O_{ik} : Nilai *output* dari jaringan i – th *unit hidden layer* ke k – th *unit output layer*.

i : *neuron* pada *unit hidden layer*.

k : *neuron* pada *unit output layer*.

Target : *output aktual / target aktual*.

3. Tahapan *gradient descent* pada perambatan balik (*gradient descent feed back propagation*) antara lain :

- Setelah mendapatkan nilai *error*, perhitungan *derivative partial error* jaringan E terhadap *weight* dan *bias* pada *output layer* menggunakan persamaan:

$$\frac{\partial E}{\partial W_{ik}^o} = -(Target - O_{ik}) * f_k^{o'}(net_{ik}^o) * f_{in_netn}$$

Dimana $f_k^{o'}$ adalah turunan dari f_k^o (fungsi aktivasi) dari *neuron output layer*

$$f_k^o(net_{ik}^o) = \frac{1}{1+e^{-(net_{ik}^o)}}$$

Maka turunannya :

$$f_k^{o'}(net_{ik}^o) = f_k^o(net_{ik}^o) * [1 - f_k^o(net_{ik}^o)]$$

Sehingga persamaan :

$$\frac{\partial E}{\partial W_{ik}^o} = - \sum_{k=6} (Target - O_{ik}) * f_k^o(net_{ik}^o) * [1 - f_k^o(net_{ik}^o)] * f_{in_netn} \quad (2.9)$$

Keterangan :

∂E : *derivative partial* pada *error*.

∂W_{ik}^o : *derivative partial* pada *bobot* dan *bias output layer* dari i – th *unit hidden layer* ke k – th *unit output layer*.

O_{ik} : Nilai *output* dari i – th *unit hidden layer* ke k – th *unit output layer*.

Target : *output aktual / target aktual*.

f_k^o : fungsi aktivasi dari ke k – th *unit output layer*.

f_{in_netn} : perhitungan fungsi aktivasi *neuron network* di setiap nilai *weight* dan *bias* pada perhitungan jaringan *neuron hidden layer*.

net_{ik}^o : *Input* jaringan dari i – th *unit hidden layer* ke k – th *unit output layer*.

k : *neuron* pada *unit output layer*.

Derivative partial didefinisikan sebagai laju perubahan nilai *gradient descent* untuk meminimalkan nilai *error* dan sampai pada set bobot yang benar.

- Perhitungan *derivative partial error* jaringan E terhadap *weight* dan *bias* pada *hidden layer* sebagai berikut:

$$\frac{\partial E}{\partial W_{ij}^h} = - \sum_{i=0} (\text{Target} - O_{ik}) * f_k^{o'}(net_{ik}^o) * w_{kj}^0 * f_j^{h'}(net_{ij}^h) * X_i$$

Pada $f_k^{o'}$ adalah turunan dari f_k^o (fungsi aktivasi) dari *output layer*, dimana

$$f_k^o(net_{ik}^o) = \frac{1}{1+e^{-(net_{ik}^o)}}$$

Maka turunannya :

$$f_k^{o'}(net_{ik}^o) = f_k^o(net_{ik}^o) * [1 - f_k^o(net_{ik}^o)]$$

Sedangkan pada $f_j^{h'}$ adalah turunan dari f_j^h (fungsi aktivasi) dari *hidden layer*, dimana

$$f_j^h(net_{ij}^h) = \frac{1}{1+e^{-(net_{ij}^h)}}$$

Maka turunannya :

$$f_j^{h'}(net_{ij}^h) = f_j^h(net_{ij}^h) * [1 - f_j^h(net_{ij}^h)]$$

Sehingga persamaan :

$$\begin{aligned} \frac{\partial E}{\partial W_{ij}^h} &= - \sum_{i=0} (\text{Target} - O_{ik}) * f_k^o(net_{ik}^o) * [1 - f_k^o(net_{ik}^o)] \\ &\quad * w_{ik}^0 \\ &\quad * f_j^h(net_{ij}^h) * [1 - f_j^h(net_{ij}^h)] * X_i \end{aligned} \tag{2.10}$$

Keterangan :

∂E : *derivative partial* pada *error*.

∂W_{ij}^h : *derivative partial* pada *weight* dari jaringan dari i – th *unit input layer* ke j – th *unit hidden layer*.

f_k^o : fungsi aktivasi dari ke k – th *unit output layer*.

net_{ik}^o : Hasil *input* dari jaringan i – th *unit hidden layer* ke k – th *unit output layer*.

w_{kj}^0 : nilai bobot pada koneksi fungsi aktivasi dari j – th *unit hidden layer* ke k – th *unit output layer*.

- f_j^h : fungsi aktivasi dari j – th *unit hidden layer*.
- net_{ij}^h : hasil nilai *input* jaringan dari i – th *unit input layer* ke j – th *unit hidden layer*.
- X_i : nilai *unit input layer*.
- i : *neuron* pada *unit input layer*.

2.5. ADAM Pada Gradient Descent Backpropagation

ADAM untuk mempercepat proses di sepanjang dimensi dimana *gradient* secara konsisten menunjukkan kearah yang sama. Gao *et al.* (2017) pada penelitiannya menilai hipotesis menggabungkan *Recurrent Neural Networks (RNNs)* dengan variabel *input* dapat memberikan metode yang lebih baik untuk memprediksi. Pada pelatihan model, strategi optimasi seperti *ADAM* digunakan untuk perkiraan dalam memprediksi akurasi pada model pelatihan. *Gradient descent* adalah salah satu algoritma terpenting untuk mengoptimalkan jaringan saraf, dan *ADAM* dapat menghitung tingkat pembelajaran adaptif (*adaptive learning rate*) untuk setiap bobot jaringan saraf. *ADAM* adalah memperkirakan momen *gradient* pertama dan momen *gradient* kedua untuk melakukan pembaruan bobot.

Pada penelitian Indolia *et al.* (2018) menggunakan *ADAM* untuk pengoptimasian *gradient descent* merupakan algoritma pembelajaran untuk meminimalkan fungsi objektif (sering disebut *loss function* E (x)) pada berbagai parameter seperti bobot dan bias. *Error backpropagation* adalah mekanisme yang digunakan untuk memodifikasi parameter sebelum inisialisasi jaringan untuk mendapatkan parameter yang dioptimalkan dan dapat menghasilkan keluaran (*output*) yang mendekati keluaran target (*output target*). Pada *neural network* digunakan *error backpropagation*, pada proses *feedforward* proses menghitung *output* satu persatu dan menghitung komponen *error* yang didapat pada lapisan terakhir. *Gradient* yang dihitung adalah *backpropagation* untuk mendapatkan jaringan yang dapat dioptimalkan. Berikut adalah proses langkah – langkah *ADAM* sebagai berikut :

- Inisialisasi $m_{\text{weight}(t)}$, $m_{\text{bias}(t)}$, $v_{\text{weight}(t)}$ dan $v_{\text{bias}(t)} = 0$

Jika *epoch* pertama adalah $t = 1$, maka $t = 1 - 1 = 0$ (*time step / epoch* awal pada *input*)

- Perhitungan *gradient* pada *epoch / time step* $g_t = \frac{\partial E}{\partial W_{ik}^0}$
- Perhitungan estimasi moment pertama (m_t) *weight* dan *bias* dilakukan setelah menerima perhitungan *derivative square error* (kuadrat *error*) pada *output layer* dengan persamaan sebagai berikut:

$$m_{weight(t)} = \beta_1 * m_{weight(t-1)} + (1 - \beta_1) * g_t \quad (2.11)$$

$$m_{bias(t)} = \beta_1 * m_{bias(t-1)} + (1 - \beta_1) * g_t \quad (2.12)$$

Keterangan :

$m_{weight(t)}$: perhitungan estimasi moment pertama pada *weight*.

$m_{bias(t)}$: perhitungan estimasi moment pertama pada *bias*.

β_1 : *exponential / hyperparameter* untuk estimasi momen pertama
($\beta_1 = 0.9$).

g_t : *gradient derivative partial* pada *error* pada iterasi / *time step*.

$m_{weight(t-1)}$: estimasi moment pertama pada *weight* vektor pada *time step*
 $t-1$.

$m_{bias(t-1)}$: estimasi moment pertama pada *bias* vektor pada *time step* $t-1$.

- Perhitungan koreksi *weight* dan *bias* estimasi momen pertama \hat{m}_t dengan persamaan sebagai berikut :

$$\hat{m}_{weight(t)} = \frac{m_{weight(t)}}{1 - \beta_1^t} \quad (2.13)$$

$$\hat{m}_{bias(t)} = \frac{m_{bias(t)}}{1 - \beta_1^t} \quad (2.14)$$

Keterangan :

$\hat{m}_{weight(t)}$: perhitungan koreksi *weight* estimasi momen pertama.

$\hat{m}_{bias(t)}$: perhitungan koreksi *bias* estimasi momen pertama.

β_1 : *exponential / hyperparameter* untuk estimasi momen pertama
($\beta_1 = 0.9$).

$m_{weight(t)}$: hasil perhitungan dari estimasi moment pertama *weight*.

$m_{bias(t)}$: hasil perhitungan dari estimasi moment pertama *bias*.

- Perhitungan *gradient* pada *epoch / time step* $g_t = \frac{\partial E}{\partial W_{ij}^h}$
- Perhitungan estimasi moment kedua (v_t) *weight* dan *bias* dilakukan setelah menerima Perhitungan *derivative output* ke *hidden layer* dengan persamaan sebagai berikut:

$$v_{weight(t)} = \beta_2 v_{weight(t-1)} + (1 - \beta_2) * (g_t)^2 \quad (2.15)$$

$$v_{bias(t)} = \beta_2 v_{bias(t-1)} + (1 - \beta_2) * (g_t)^2 \quad (2.16)$$

Keterangan :

$v_{weight(t)}$: perhitungan estimasi moment kedua pada *weight* yang pada waktu t.

$v_{bias(t)}$: perhitungan estimasi moment kedua pada *bias* yang pada waktu t.

β_2 : *exponential / hyperparameter* untuk estimasi momen kedua
($\beta_2 = 0.999$).

g_t : *gradient derivative output* ke *hidden layer* pada *epoch / time step*.

$v_{weight(t-1)}$: estimasi moment kedua vektor dengan parameter *weight* pada *time step* t-1.

$v_{bias(t-1)}$: estimasi moment kedua vektor dengan parameter *bias* pada *time step* t-1.

- Perhitungan koreksi *weight* dan *bias* estimasi momen kedua \tilde{V}_t dengan persamaan sebagai berikut :

$$\tilde{V}_{weight(t)} = \frac{v_{weight(t)}}{1 - \beta_2^t} \quad (2.17)$$

$$\tilde{V}_{bias(t)} = \frac{v_{bias(t)}}{1 - \beta_2^t} \quad (2.18)$$

Keterangan :

$\tilde{V}_{weight(t)}$: perhitungan koreksi *weight* estimasi momen kedua.

$\tilde{V}_{bias(t)}$: perhitungan koreksi *bias* estimasi momen kedua.

β_2 : *exponential/hyperparameter* untuk estimasi momen kedua
($\beta_2 = 0.999$).

$v_{weight(t)}$: hasil perhitungan dari estimasi moment kedua *weight*.

$v_{bias(t)}$: hasil perhitungan dari estimasi moment kedua *bias*.

➤ Perbaharuan parameter *weight* dan *bias* dengan persamaan sebagai berikut :

$$w_{weight(t)} = w_{weight(t-1)} - \alpha * \frac{\hat{m}_{weight(t)}}{\sqrt{\tilde{V}_{weight(t)} + \varepsilon}} \quad (2.19)$$

$$w_{bias(t)} = w_{bias(t-1)} - \alpha * \frac{\hat{m}_{bias(t)}}{\sqrt{\tilde{V}_{bias(t)} + \varepsilon}} \quad (2.20)$$

Keterangan :

- $w_{weight(t)}$: perbaruan parameter *weight*.
- $w_{bias(t)}$: perbaruan parameter *bias*.
- $\tilde{V}_{weight(t)}$: hasil perhitungan koreksi *weight* estimasi momen kedua
- $\tilde{V}_{bias(t)}$: hasil perhitungan koreksi *bias* estimasi momen kedua
- $w_{weight(t-1)}$: parameter *weight* awal.
- $w_{bias(t-1)}$: parameter *bias* awal.
- $\hat{m}_{weight(t)}$: perhitungan koreksi *weight* estimasi momen pertama.
- $\hat{m}_{bias(t)}$: perhitungan koreksi *bias* estimasi momen pertama.
- ε : nilai *epsilon* ($\varepsilon = 10^{-8}$).
- α : *learning rate*.

2.6 Penelitian Terdahulu

Dalam melakukan penelitian, penulis menggunakan beberapa penelitian yang relevan dengan permasalahan yang diteliti yang digunakan sebagai acuan agar penelitian ini berjalan dengan lancar.

Tabel 2.1. Penelitian terdahulu

No	Nama Peneliti dan Tahun	Metode yang digunakan	Keterangan / Hasil Penelitian
1.	Endah <i>et al.</i> (2017)	<i>Gradient Descent</i> , <i>Momentum</i> dan <i>Adaptive Learning Rate</i>	Melakukan penelitian tentang meningkatkan laju konvergensi dengan menggunakan <i>adaptive learning rate</i> dan optimasi momentum. Identifikasi dilakukan terhadap <i>True Positive Rate (TPR)</i> dan <i>False Positif Rate (FPR)</i> yang bisa berpengaruh pada laju konvergen. Data yang digunakan dalam penelitian tersebut adalah data rekam medis

			diabetes, dan data juga bisa mempengaruhi konvergensi pada proses pelatihan.
--	--	--	--

Tabel 2.1. Penelitian terdahulu (Lanjutan)

No	Nama Peneliti dan Tahun	Metode yang digunakan	Keterangan / Hasil Penelitian
2.	Wijnhoven & With (2010)	<i>Stochastic Gradient Descent</i>	Pada penelitian ini menggunakan algoritma <i>Stochastic Gradient Descent</i> (<i>SGD</i>) memakai hanya satu sampel pelatihan baru disetiap iterasi dan memproses sampel dengan cara seperti aliran. <i>SGD</i> juga memiliki fitur yang memungkinkan adaptasi pada fungsi klasifikasi, dan dengan memasukkan <i>SGD</i> dapat mempercepat proses optimasi.
3.	Chen & Wang (2012)	<i>Weighted Stochastic Gradient Descent</i>	Pada penelitian ini menggunakan <i>weighted Stochastic Gradient Descent</i> untuk pembelajaran kamus (<i>dictionary learning</i>) dengan matriks pembobotan. Matriks pembobotan dimasukkan kedalam fungsi <i>cost</i> , yang mewakili <i>mean square error</i> dan mendapatkan rumus pembaruan kamus.
4.	Lenka & Mohapatra (2015)	<i>Gradient Descent with Momentum</i>	Penelitian ini untuk memprediksi kadar air tanah dalam pertanian presisi dengan klasifikasi pola jaringan saraf tiruan menggunakan <i>gradient descent with momentum</i> . Pada penelitian menggunakan <i>gradient descent with momentum</i> dengan menganalisis menggunakan MSE (<i>Mean Square Error</i>), RMSE (<i>Root Mean Square Error</i>) dan <i>R squared Error</i> .
5.	Rahmat <i>et al.</i> (2017)	<i>Multilayer perceptron</i> dengan Algoritma pelatihan <i>backpropagation Levenberg-Marquardt</i>	Pada penelitian ini membantu dokter gigi forensik untuk mengidentifikasi gigi molar dan premolar dan teknik pemrosesan gambar juga diterapkan untuk mengatasi kontras rendah dari gambar radiografi panoramik gigi
6.	Seniman <i>et al.</i> (2016)	<i>Backpropagation</i> dan <i>Direction Feature Extraction</i> (<i>DFE</i>)	Pada penelitian ini menggunakan <i>Backpropagation</i> (<i>BP</i>) dan <i>Direction Feature Extraction</i> (<i>DFE</i>) untuk pengenalan karakter catur cina. <i>DFE</i> adalah metode

			ekstraksi fitur untuk mendapatkan fitur, hasilnya BP dan kinerja <i>DFE</i> mampu mengenali karakter catur Cina dengan akurasi 98%.
--	--	--	---

2.7 Perbedaan Dengan Riset Yang Lain

Berdasarkan riset yang telah dilakukan, peneliti membuat beberapa perbedaan dalam penelitian ini, yaitu Metode yang digunakan adalah *ADAM* untuk mengoptimasi metode *Gradient Descent Backpropagation* agar meminimalkan nilai kuadrat *error* dan dapat mempercepat proses pengoptimasian pada *neural network*.

2.8 Kontribusi Riset

Dalam penelitian ini, metode *ADAM* digunakan untuk memperkirakan momen *gradient* pertama dan momen *gradient* kedua untuk melakukan pembaruan bobot (*update weight*) pada metode *Backpropagation Neural Network*. Diharapkan dari penelitian ini akan didapatkan metode yang lebih efektif dalam meminimalkan nilai kuadrat *error / Square Error (SE)*.

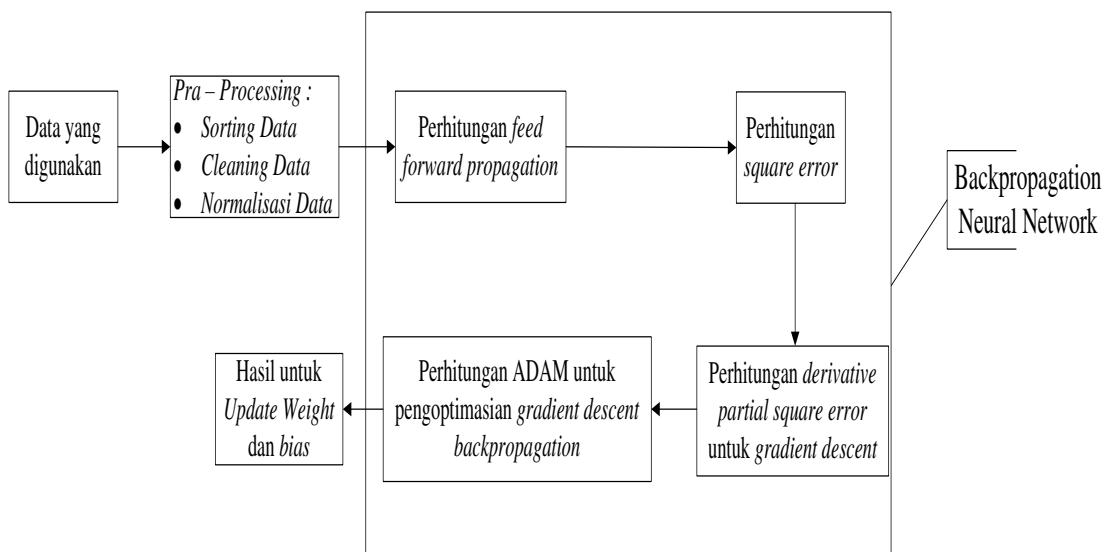
BAB 3

METODOLOGI PENELITIAN

Pada *backpropagation* merupakan jaringan saraf tiruan (*artificial neural network*) yang biasa digunakan untuk klasifikasi. *Backpropagation* adalah metode *Gradient descent* / penurunan gradient untuk meminimalkan nilai *square error* (kuadrat *error*) pada *output*. Pengoptimasian *gradient descent* pada *backpropagation* dengan menggunakan *ADAM* diharapkan dapat meningkatkan pembelajaran adaptif (*adaptive learning rate*) untuk mendapatkan perbaharuan disetiap bobot pada jaringan saraf (*neural network*). Berdasarkan uraian diatas, maka tujuan dari penelitian ini adalah untuk meningkatkan proses pembelajaran *backpropagation* dalam klasifikasi dengan mengoptimasi *gradient descent* menggunakan *ADAM* untuk dapat meminimalkan nilai kuadrat *error* dengan menggunakan *mean square error (MSE)*.

3.1. Diagram Blok Penelitian

Diagram blok penelitian bertujuan supaya penelitian yang dilakukan tidak keluar dari jalur yang ditentukan. Diagram blok penelitian yang dilakukan dapat dilihat pada gambar 3.1:



Gambar 3.1. Diagram blok penelitian

3.1.1. Data yang digunakan

Data yang digunakan pada penelitian ini adalah data tentang Wisconsin Breast Cancer Dataset dari University of California Irvine (UCI) Machine Learning Repository. Dataset berjumlah 699 data dengan distribusi kelas jinak sebanyak 458 (65,5%) dan distribusi kelas ganas sebanyak 241 (34,5%). Data ini dikumpulkan di University of Wisconsin Hospital dan data ini hanya didasarkan pada tes *Fine Needle Aspiration* (FNA) yaitu cara menyuntikan dengan jarum halus untuk mengambil cairan pada benjolan dan dapat memeriksanya. Data memiliki 9 (Sembilan) karakteristik/atribut di nilai secara visual dengan variabel kelas yang sesuai dan didefinisikan untuk setiap record pada dataset. Semua nilai pada sembilan atribut di indeks dari 1 – 10 rentang interval, sedangkan nilai rentang *output/class* pada sel *breast cancer* adalah 2 (dua) untuk kategori jinak dan 4 (empat) untuk kategori ganas. Table 3.1 deskripsi dari dataset WBCD:

Tabel 3.1. Deskripsi dari dataset WBCD

No.	Atribut	Rentang Interval (<i>range</i>)
1.	Clump Thickness	1 – 10
2.	Uniformity of Cell Size	1 – 10
3.	Uniformity of Cell Shape	1 – 10
4.	Marginal Adhesion	1 – 10
5.	Single Epithelial Cell Size	1 – 10
6.	Bare Nuclei	1 – 10
7.	Bland Chromatin	1 – 10
8.	Normal Neucleoli	1 – 10
9.	Mitoses	1 – 10
10.	Class variable	Sel jinak (2) dan Sel ganas (4)

Tabel 3.1. merupakan deskripsi dari dataset yang dikumpulkan di University of Wisconsin Hospital dan pada data ini hanya didasarkan pada tes *Fine Needle Aspiration* (FNA). Data tersebut nantinya akan digunakan untuk pengujian jaringan saraf tiruan yang dihasilkan.

Tabel 3.2 yang menjadi *attribut* atau *neuron input* pada jaringan saraf tiruan *backpropagation* adalah X1 (*Clump Thickness*), X2 (*Uniformity of Cell Size*), X3 (*Uniformity of Cell Shape*), X4 (*Marginal Adhesion*), X5 (*Single Epithelial Cell Size*), X6 (*Bare Nuclei*), X7 (*Bland Chromatin*), X8 (*Normal Neucleoli*), X9 (*Mitoses*) dan

X10 (*Output / Class*). *Neuron / node input* tersebut digunakan sebagai variabel pada jaringan saraf tiruan *backpropagation*. *Breast cancer* dataset dapat dilihat pada Table 3.2:

Tabel 3.2. *Breast cancer* dataset

No.	id number	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10 (Output)
1.	1000025	5	1	1	1	2	1	3	1	1	2
2.	1002945	5	4	4	5	7	10	3	2	1	2
3.	1015425	3	1	1	1	2	2	3	1	1	2
4.	1016277	6	8	8	1	3	4	3	7	1	2
5.	1017023	4	1	1	3	2	1	3	1	1	2
6.	1017122	8	10	10	8	7	10	9	7	1	4
7.	1018099	1	1	1	1	2	10	3	1	1	2
8.	1018561	2	1	2	1	2	1	3	1	1	2
9.	1033078	2	1	1	1	2	1	1	1	5	2
...											
697.	888820	5	10	10	3	7	3	8	10	2	4
698.	897471	4	8	6	4	3	4	10	6	1	4
699.	897471	4	8	8	5	4	5	10	4	1	4

Tabel 3.2 merupakan *breast cancer dataset* yang digunakan untuk menentukan klasifikasi kelas. Data *breast cancer* memiliki 699 *id number*, 9 (sembilan) atribut dan 1 (satu) atribut *output* yang menghasilkan 2 (dua) kelas pada atribut *output* dalam dataset *breast cancer*. Untuk tabel 3.2. pada *breast cancer dataset* dapat dilihat selengkapnya pada lampiran 1.

3.1.2. *Pra – processing*

Pada blok diagram penelitian, dataset dilakukan proses *pra – processing* yang digunakan sebagai masukkan dalam proses pengujian. Tahap *pra – processing* memiliki tiga proses:

1. *Sorting data*

Data untuk pengujian yang akan digunakan adalah data dengan nilai *range* [1-10] yang ada pada tabel 3.1. Data yang telah di *input* akan melalui proses *sorting data*. Pada tahap *sorting data* digunakan untuk mengurutkan data sesuai kelas yaitu jinak dan ganas. Tabel 3.3 dan tabel 3.4 merupakan data yang sesuai pada kelas yaitu jinak dan ganas setelah di *sortir*:

Tabel 3.3. Dataset pada kelas kanker jinak

No.	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10 (Output)
1.	5	1	1	1	2	1	3	1	1	2
2.	5	4	4	5	7	10	3	2	1	2
3.	3	1	1	1	2	2	3	1	1	2
4.	6	8	8	1	3	4	3	7	1	2
5.	4	1	1	3	2	1	3	1	1	2
...										
457.	3	1	1	1	3	2	1	1	1	2
458.	2	1	1	1	2	1	1	1	1	2

Tabel 3.4. Dataset pada kelas kanker ganas

No.	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10 (Output)
459.	8	10	10	8	7	10	9	7	1	4
460.	5	3	3	3	2	3	4	4	1	4
461.	8	7	5	10	7	9	5	5	4	4
462.	7	4	6	4	6	1	4	3	1	4
463.	10	7	7	6	4	10	4	1	2	4
...										
698.	4	8	6	4	3	4	10	6	1	4
699.	4	8	8	5	4	5	10	4	1	4

Tabel 3.3. merupakan data pada kelas kanker jinak setelah melalui tahap *sorting data* yang memiliki jumlah data sebanyak 458 di kelas jinak. Tabel 3.4. merupakan data pada kelas kanker ganas setelah melalui tahap *sorting data* yang memiliki jumlah data sebanyak 241 di kelas ganas. Untuk tabel 3.3. pada *dataset* kelas kanker jinak dan tabel 3.4 pada *dataset* kelas kanker ganas dapat dilihat selengkapnya pada lampiran 2.

2. Cleaning data

Pada data di tabel 3.3 dan tabel 3.4 yang sudah dilakukan proses pada tahap *sorting data*, memiliki 16 data berisi nilai yang hilang pada variabel X₆ (*bare nuclei*) dari keseluruhan 699 data. Pengisian nilai yang hilang dilakukan dengan menggunakan metode median. Metode median digunakan untuk mengisi seluruh nilai yang hilang dari dataset 699 pada atribut *bare nuclei*. Teknik untuk mengisi nilai yang hilang pada dataset dengan metode median pada persamaan 2.1:

$$\text{MEDIAN} = \text{Size of } \frac{(N+1)}{2}$$

$$\text{MEDIAN} = \text{Size of } \frac{(683+1)}{2}$$

$$\text{MEDIAN} = \text{Size of } 342$$

$$\text{Size of } 342 = 1$$

Dari teknik perhitungan metode median diatas, N merupakan total keseluruhan data pada atribut *bare nuclei* setelah dikurangin 16 data yang bernilai hilang, sehingga memiliki jumlah 683 data. Pada hasil perhitungan metode median, *Size of* merupakan urutan/ukuran data dari perhitungan metode median. Data yang sudah di urutkan, maka pada data urutan ke - 342 adalah bernilai 1 (satu). Untuk itu semua nilai 16 data nilai yang hilang dapat diisi dengan nilai 1 (satu).

3. Normalisasi data

Setelah data di proses pada tahap *sorting data* dan *cleaning data*, dilakukan tahap normalisasi data yang digunakan untuk mengubah nilai *range* dari 1–10 supaya nilai *output* terletak pada interval 0–1 sesuai kebutuhan yang dipakai pada penelitian ini yang memakai fungsi aktivasi *sigmoid*. Tabel 3.5 dan tabel 3.6 merupakan data sesuai kelas yaitu jinak dan ganas setelah di normalisasi dengan menggunakan persamaan 2.2:

Tabel 3.5. Normalisasi data pada kelas kanker jinak

No.	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
1.	0.444	0	0	0	0.111	0	0.222	0	0	0
2.	0.444	0.333	0.333	0.444	0.666	1	0.222	0.111	0	0
3.	0.222	0	0	0	0.111	0.111	0.222	0	0	0
4.	0.555	0.777	0.777	0	0.222	0.333	0.222	0.666	0	0
5.	0.333	0	0	0.222	0.111	0	0.222	0	0	0
...										
457.	0.222	0	0	0	0.222	0.111	0	0	0	0
458.	0.111	0	0	0	0.111	0	0	0	0	0

Tabel 3.6. Normalisasi data pada kelas kanker ganas

No.	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
459.	0.777	1	1	0.777	0.666	1	0.888	0.666	0	1
460.	0.444	0.222	0.222	0.222	0.111	0.222	0.333	0.333	0	1
461.	0.777	0.666	0.444	1	0.666	0.888	0.444	0.444	0.333	1
462.	0.666	0.333	0.555	0.333	0.555	0	0.333	0.222	0	1
463.	1	0.666	0.666	0.556	0.333	1	0.333	0	0.111	1
...										

Tabel 3.6. Normalisasi data pada kelas kanker ganas (Lanjutan).

No.	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
698.	0.333	0.777	0.555	0.333	0.222	0.333	1	0.555	0	1
699.	0.333	0.777	0.777	0.444	0.333	0.444	1	0.333	0	1

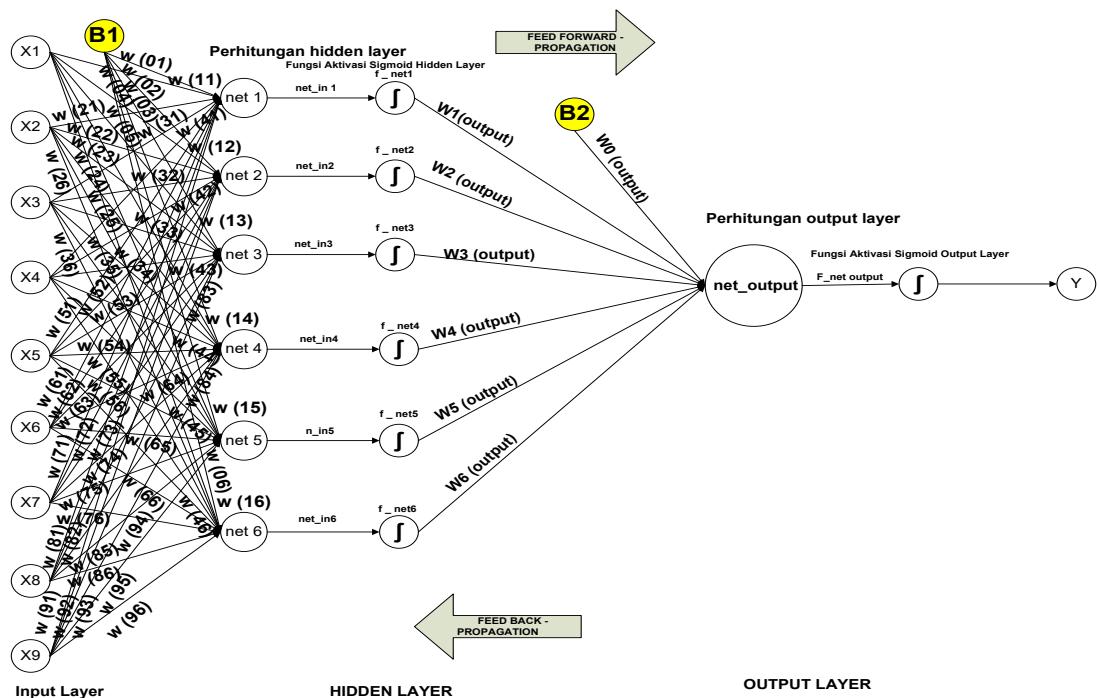
Tabel 3.5. merupakan hasil dari normalisasi data pada kelas kanker jinak setelah melalui tahap *sorting data* dan *cleaning data*. Tabel 3.6. merupakan hasil dari normalisasi data pada kelas kanker ganas setelah melalui tahap *sorting data* dan *cleaning data*. Untuk tabel 3.5. pada *dataset* kelas kanker jinak dan tabel 3.6 pada *dataset* kelas kanker ganas dapat dilihat selengkapnya pada lampiran 3.

3.2. Arsitektur Artificial Neural Network

Rancangan arsitektur *artificial neural network* ini bertujuan untuk menentukan jumlah *neuron* pada masing – masing *layer*.

3.2.1 Rancangan backpropagation neural network

Berdasarkan data yang telah disebutkan maka rancangan *backpropagation neural network* dijelaskan melalui gambar 3.2:



Gambar 3.2: Rancangan backpropagation neural network

Keterangan :

$X_1 - X_9$: Input Neuron
$w_{11} \text{ s.d } w_{96}$: Bobot awal <i>input</i> ke <i>hidden layer</i>
B_1	: Bias jaringan ke <i>hidden layer</i>
w_{01}	: Bobot bias jaringan ke <i>hidden layer</i>
$net_1 \text{ s.d } net_6$: Jaringan <i>neuron</i> pada <i>hidden layer</i>
$net_in1 \text{ s.d } net_in6$: Perhitungan jaringan pada <i>hidden layer</i>
$f_{net1} \text{ s.d } f_{net6}$: Fungsi aktivasi <i>sigmoid</i> jaringan pada <i>hidden layer</i>
$w_{1(\text{output})} \text{ s.d } w_{6(\text{output})}$: Bobot jaringan ke <i>output layer</i>
B_2	: Bias jaringan ke <i>output layer</i>
w_0	: Bobot bias jaringan ke <i>output layer</i>
net_output	: Perhitungan jaringan pada <i>output layer</i>
$f_{net\ output}$: Fungsi aktivasi <i>sigmoid</i> pada <i>output layer</i>
Y	: Output Neuron

Gambar 3.2. merupakan rancangan *backpropagation neural network*. *Input* pada rancangan ini merupakan arsitektur yang disesuaikan dengan *feature* dataset dari *UCI Machine Learning* yaitu 9 (sembilan) *neuron* pada *input layer*. Pada *hidden layer* berdasarkan perhitungan yang sudah dihitung dan ditentukan, maka *neuron* pada *hidden layer* berjumlah 6 (enam) *neuron*. Dan untuk *output layer* sesuai dengan data yang disebut pada *output layer* yaitu sebanyak 1 (satu) buah *neuron*.

3.2.2 *Input layer*

Input layer merupakan *layer* untuk memasukkan nilai yang akan dipelajari oleh *artificial neural network*. Data tersebut memiliki *node* pada *input layer* disediakan sebanyak 9 (sembilan) *node*. Pada dataset *breast cancer* semua nilai pada sembilan atribut untuk keseluruhan data diindeks dari nilai rentang 1 – 10. Setelah dataset di normalisasi, semua nilai pada sembilan atribut untuk keseluruhan data bernilai rentang 0 – 1.

3.2.3 Hidden layer

Hidden layer merupakan *layer* yang melakukan proses nilai *input* dan menghasilkan nilai *function* yang diteruskan ke *layer output*. *Node* atau *neuron* yang disediakan pada *hidden layer* dipenelitian ini ditentukan berdasarkan (Heaton, 2008) sebagai berikut:

- Jumlah *hidden neuron* harus berada diantara ukuran *input layer* dan *output layer*.
- Jumlah *hidden neuron* $\frac{2}{3}$ dari ukuran *input layer*, ditambah ukuran *output layer*.

Neuron hidden layer yang digunakan pada penelitian ini sesuai pada gambar 3.2., adalah menggunakan peraturan pada huruf b yaitu jumlah *hidden neuron* harus $\frac{2}{3}$ dari ukuran *input layer*, ditambah ukuran *output layer*. Jumlah *neuron* pada *hidden layer* yang digunakan adalah sebagai berikut :

$$\text{Jumlah } \textit{neuron} \text{ pada } \textit{hidden layer} : \frac{2}{3} * (\text{jumlah } \textit{neuron} \text{ } \textit{input} + \text{jumlah } \textit{neuron} \text{ } \textit{output})$$

$$\text{Jumlah } \textit{neuron} \text{ pada } \textit{hidden layer} : \frac{2}{3} * (9 + 1)$$

$$\text{Jumlah } \textit{neuron} \text{ pada } \textit{hidden layer} : \frac{2}{3} * (10)$$

Jumlah *neuron* pada *hidden layer* : 6 *neuron* pada *hidden layer*

Pada sinyal *input layer* akan meneruskan sinyal tersebut ke semua *unit* pada lapisan tersembunyi (*hidden layer*). Setiap *unit* *hidden layer* menjumlahkan sinyal *input* dengan bobot dan *bias* yang sudah diinisialisasi secara acak nilai yang terkecil. Setelah penjumlahan *hidden layer* dilakukan, hasil dari *hidden layer* tersebut dilakukan perhitungan dengan menerapkan fungsi aktivasi *sigmoid* supaya nilai *output* terletak pada interval 0–1. Setelah perhitungan *hidden layer* dengan menerapkan fungsi aktivasi *sigmoid*, sinyal *input* diteruskan ke lapisan *output* untuk menjumlahkan bobot dan *bias* pada lapisan *output*. Dan hasil penjumlahan pada lapisan *output* dilakukan kembali perhitungan fungsi aktivasi *sigmoid* pada lapisan *output*. Setelah nilai *output* didapat, pada proses *feed forward propagation* dilakukan *check error* pada setiap *epoch* bila *error* sudah dicapai proses akan berhenti. Proses yang belum mencapai *error*, akan dilakukan proses *feed backward propagation*.

3.2.4 *Output layer*

Output layer merupakan *layer* yang memproses nilai dari *hidden layer* dan menghasilkan nilai *error*. *Neuron* pada *layer output* sesuai dengan *dataset* yaitu 1 (satu) buah *node*. Pada masing – masing *node* tersebut memiliki kelas jinak dan ganas pada *breast cancer dataset*, karena pada *output layer* sesuai kebutuhan penelitian memakai fungsi aktivasi *sigmoid* yang memakai nilai [0, 1], maka nilai kelas jinak pada data bernilai 2 (dua) akan menjadi 0 (nol). Sedangkan nilai kelas ganas pada data bernilai 4 (empat) akan menjadi 1 (satu) sesuai kebutuhan penelitian memakai fungsi aktivasi *sigmoid*.

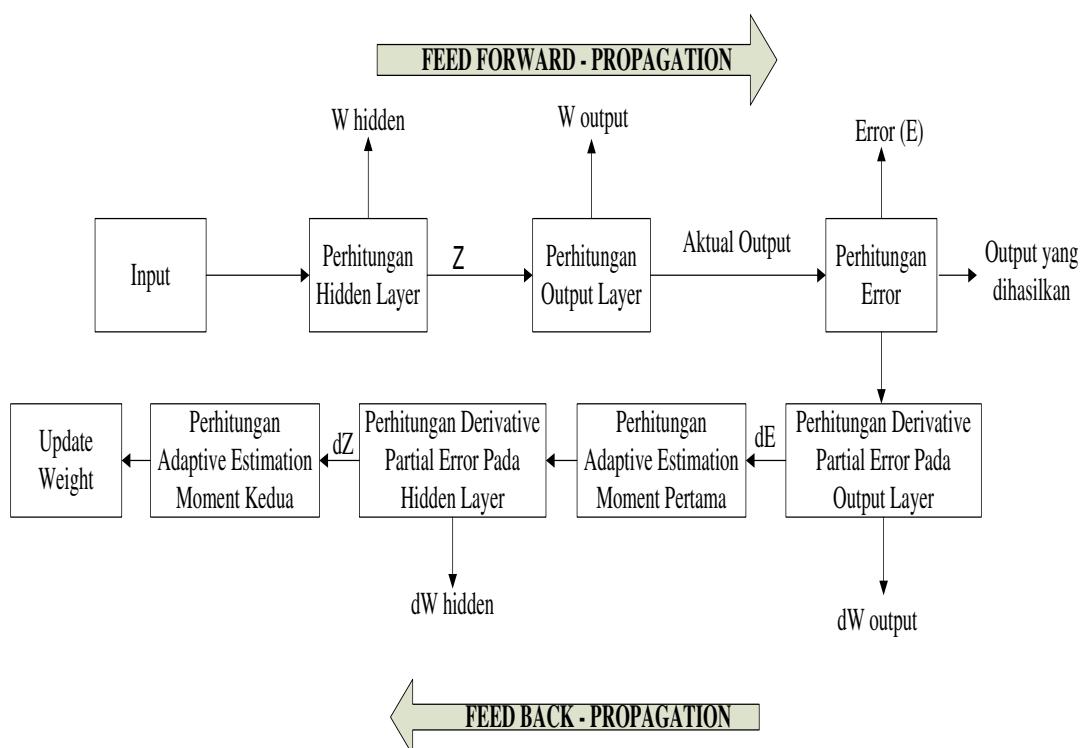
3.3. Rancangan Arsitektur *ADAM* Pada *Gradient Descent Backpropagation*

Pengoptimasian *gradient descent backpropagation* difokuskan pada waktu proses *feedforward propagation* saat pengidentifikasi/pengecekan *error*. *Gradient descent* diterapkan pada *backpropagation* untuk jaringan *multilayer feedforward propagation*. Aturan pembelajaran untuk memperbarui bobot dan *bias* (*update weight and bias*) pada *output layer* serta *neuron* pada *hidden layer*. Pada penelitian ini untuk pengoptimasian *gradient descent backpropagation* dengan menggunakan *ADAM*. *ADAM* digunakan untuk meningkatkan pembelajaran adaptif disetiap bobot dan *bias* pada jaringan saraf dan untuk memperkirakan momen pertama dan momen kedua pada *gradient* dalam melakukan pembaruan bobot dan *bias* (*update weight and bias*) sehingga dapat meminimalkan nilai kuadrat *error* (*square error*).

Penelitian ini melakukan proses menggunakan data yang memiliki nilai *output* berkelas jinak dan berkelas ganas. Pada proses yang dilakukan, diambil 1 (satu) data untuk dipelajari pada proses *feedforward propagation neural network*. Pengidentifikasian *error* sebagai *target error* yang ditentukan akan dilakukan proses *backpropagation* untuk mendapatkan perbaruan bobot dan *bias* (*update weight* dan *bias*) pada *epoch* pertama. Pada *epoch* kedua digunakan data kedua dan menggunakan bobot dan *bias* yang sudah diperbarui pada *epoch* pertama. Pada *epoch* kedua dilakukan proses yang sama dengan proses pada *epoch* pertama dan sampai proses *epoch* selanjutnya sehingga mendapatkan pengidentifikasi *error* yang diinginkan.

Setelah tercapai pada pengidentifikasi *error* yang diinginkan, proses berhenti pada *epoch* terakhir. Pada penelitian ini difokuskan pada nilai kuadrat *error* yang didapat setelah proses *epoch* dan *target error* ditentukan.

Rancangan arsitektur *ADAM* pada *gradient descent backpropagation* yang digunakan dari data berfokus pada proses *feedforward*, untuk melihat penurunan/peminalisasian kuadrat *error*. Gambar arsitektur *ADAM* pada *gradient descent backpropagation* akan ditampilkan pada Gambar 3.3 :



Gambar 3.3. Rancangan arsitektur *ADAM* pada *gradient descent backpropagation*

Gambar 3.3. merupakan rancangan arsitektur *ADAM* pada *gradient descent backpropagation*. Pada proses *feed forward propagation*, data yang *input* akan dilakukan perhitungan pada *hidden layer* dan perhitungan pada *output layer*. Pada hasil perhitungan *output layer* akan dijumlahkan dengan nilai aktual *output* (nilai target) pada data. Proses perhitungan *error* untuk pengidentifikasi *error* pada *feed forward propagation*. Setelah itu akan dilakukan proses perhitungan kuadrat *error* untuk mendapatkan minimalisasi nilai kuadrat *error* pada *output* yang dihasilkan *feed*

forward propagation. Pada *backpropagation* dibutuhkan proses perhitungan turunan partial (*derivative partial*) dari *error* pada bobot dan *bias* di *output layer* dan *hidden layer*. Setelah mendapatkan hasil perhitungan *derivative partial* pada bobot dan *bias*, dilakukan perhitungan *ADAM* pertama pada estimasi momen pertama dan *ADAM* pertama pada estimasi momen kedua di *output layer*. Hasil perhitungan *ADAM* pertama pada estimasi momen pertama dan *ADAM* pertama pada estimasi momen kedua akan dilakukan koreksi bobot dan *bias*.. Proses yang sama dilakukan perhitungan *ADAM* kedua pada estimasi momen pertama dan *ADAM* kedua pada estimasi momen kedua pada *hidden layer*. Masing – masing hasil *ADAM* kedua pada estimasi momen pertama dan *ADAM* kedua pada estimasi momen kedua akan dikoreksi bobot dan *bias* pada *hidden layer*. Sehingga hasil koreksi bobot dan *bias* pada *output layyer* dan *hidden layer* dapat dilakukan perhitungan untuk mendapatkan perbaruan bobot dan *bias* di *output layer* dan *hidden layer*.

3.4. Analisis *ADAM* Pada *Gradient Descent Backpropagation*

Pada proses perhitungan *ADAM* pada *gradient descent backpropagation* ini terlebih dahulu dijabarkan proses kerja *backpropagation neural network* secara garis besar terlebih dahulu. Proses kerja *backpropagation neural network* dapat dijabarkan:

1. *Input* data awal

Pada bagian ini memasukkan data berupa *input* data dan inisialisasi bobot dan *bias* dari nilai *random* yang cukup kecil. Data yang di *input* memiliki nilai rentang 0 – 1.

2. Proses *feedforward propagation*

Pada proses *feedforward* pada masing – masing *input* data yang dibentuk untuk menjumlahkan *neuron* – *neuron* yang telah ditetapkan pada masing – masing *input*. Setelah nilai *function* pada masing – masing *neuron* dihitung, maka dilanjutkan ke *output layer*. Pada *output layer* setiap *neuron* pada *output layer* menerima jumlah dari nilai *function neuron* pada *hidden layer* menuju ke *neuron output layer*. Setelah itu nilai *function output* dikurangi dengan nilai *output* pada dataset untuk mendapatkan nilai *error*.

3. Proses *feed backward propagation*

Pada proses *backward* dari nilai *error* yang dihasilkan dari *output layer* akan digunakan sebagai penentuan parameter seperti *learning rate* yang digunakan untuk melakukan *update* bobot dan *bias* dari *output layer* menuju ke *hidden layer*.

Pada proses kerja *backpropagation neural network* diatas setelah mencapai proses *feed backward* maka kembali ke proses *feed forward* yang dilakukan secara berulang–ulang berdasarkan pada nilai *epoch* yang telah ditetapkan ataupun berdasarkan nilai *error* yang telah ditetapkan sampai proses pembelajaran tersebut selesai.

Pada proses perhitungan ini menggunakan data dengan menggunakan 9 (sembilan) *input* variabel dan 1 (satu) data untuk memudahkan perhitungan yang dilakukan pada iterasi pertama. Arsitektur yang ditetapkan adalah 9 – 6 – 1 yaitu 9 (sembilan) *neuron* pada *input layer*, 6 (enam) *neuron* pada *hidden layer* dan 1 (satu) *neuron* pada *output layer*. Nilai *input* dan *output* setelah di lakukan proses *pra – processing* dapat ditampilkan pada Tabel 3.7. beserta dengan target / *output* nya:

Tabel 3.7. Nilai *input* setelah di normalisasi dan *output*

<i>Epoch</i>	X1	X2	X3	X4	X5	X6	X7	X8	X9	<i>Output</i>
1	0.444	0	0	0	0.111	0	0.222	0	0	0
2	0.444	0.333	0.333	0.444	0.666	1	0.222	0.111	0	0
3	0.222	0	0	0	0.111	0.111	0.222	0	0	0
...										
697	0.444	1	1	0.222	0.666	0.222	0.777	1	0.111	1
698	0.333	0.777	0.555	0.333	0.222	0.333	1	0.555	0	1
699	0.333	0.777	0.777	0.444	0.333	0.444	1	0.333	0	1

Tabel 3.7. merupakan nilai *input* dan *output* pada dataset setelah dilakukan tahap *pra processing* yaitu tahap *cleaning data*, *sorting data* dan normalisasi data. Untuk nilai masing – masing *weight* dan *bias* pada *neuron* dari *input layer* ke *hidden layer* akan diinisialisasi secara nilai *random* (acak) yang paling kecil. Berikut ini adalah Tabel 3.8 untuk *weight* dan *bias* pada *neuron* dari *input layer* ke *hidden layer* pada data *epoch* pertama sebagai berikut:

Tabel 3.8. Nilai *random weight* dan *bias* dari *input layer* ke *hidden layer*

<i>Weight</i>	X1	X2	X3	X4	X5	X6	X7	X8	X9	<i>Bias</i>
W_1^{hidden}	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111
W_2^{hidden}	0.222	0.222	0.222	0.222	0.222	0.222	0.222	0.222	0.222	0.222
W_3^{hidden}	0.333	0.333	0.333	0.333	0.333	0.333	0.333	0.333	0.333	0.333
W_4^{hidden}	0.444	0.444	0.444	0.444	0.444	0.444	0.444	0.444	0.444	0.444
W_5^{hidden}	0.555	0.555	0.555	0.555	0.555	0.555	0.555	0.555	0.555	0.555
W_6^{hidden}	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666

Tabel 3.8. merupakan nilai *random* yang cukup kecil untuk *weight* dan *bias* dari *input layer* ke *hidden layer* yang diinisialisasi secara *random* ke sembilan *neuron* dalam masing – masing *input layer* yang memiliki *weight* dan *bias* ke masing–masing enam *neuron* yang menuju ke *hidden layer*.

Untuk nilai *weight* dan *bias* pada *neuron* dari *hidden layer* ke *output layer* diinisialisasi juga secara *random*. Berikut ini adalah Tabel 3.9 untuk *weight* dan *bias* pada *neuron* dari *hidden layer* ke *output layer*:

Tabel 3.9. Nilai *random weight* dan *bias* dari *hidden layer* ke *output layer*

Neuron <i>weight</i> pada <i>hidden layer</i> ke <i>output layer</i>	Nilai <i>weight</i> pada <i>neuron</i> dari <i>hidden layer</i> ke <i>output layer</i>
W_1^{Output}	0.111
W_2^{Output}	0.222
W_3^{Output}	0.333
W_4^{Output}	0.444
W_5^{Output}	0.555
W_6^{Output}	0.666
W_{bias}^{Output}	0.777

Untuk nilai parameter yang digunakan sebagai kebutuhan pelatihan jaringan ditentukan pula pada Tabel 3.10:

Tabel 3.10. Nilai parameter pada pelatihan jaringan

Target error	Learning rate (α)	ϵ	Nilai <i>exponential</i> estimasi momen pertama (β_1)	Nilai <i>exponential</i> estimasi momen kedua (β_2)
0.001	0.001	0.00000001	0.9	0.999

Perhitungan pada *epoch* pertama dilakukan pada data pertama (*id number*) dengan 9 (sembilan) *neuron* pada *input layer*, 6 (enam) *neuron* pada *hidden layer* dan 1 (satu) *neuron* pada *output layer* yang memiliki 2 (dua) kelas yaitu kelas jinak dan kelas ganas. Berikut ini proses perhitungan pada *epoch* pertama:

Tahap Propagasi Maju (*feed forward propagation*)

1. Proses perhitungan jaringan pada setiap *unit hidden layer* dengan menjumlahkan *weight* dan *bias* yang dikirim dari *neuron unit input layer* dengan menggunakan persamaan 2.3:

$$\begin{aligned}
 net_1^{hidden} &= \sum_{i=9} X_i W_{ij}^h + w_{bias_j}^h \\
 &= w_{bias1} + w_{11}.x_1 + w_{21}.x_2 + w_{31}.x_3 + w_{41}.x_4 + w_{51}.x_5 + w_{61}.x_6 + w_{71}.x_7 + \\
 &\quad w_{81}.x_8 + w_{91}.x_9 \\
 &= 0.111 + 0.111 * 0.444 + 0.111 * 0 + 0.111 * 0 + 0.111 * 0 + 0.111 * 0 \\
 &\quad * 0.111 + 0.111 * 0 + 0.111 * 0.222 + 0.111 * 0 + 0.111 * 0 \\
 &= 0.111 + 0.0493 + 0 + 0 + 0 + 0.0123 + 0 + 0.0246 + 0 + 0 \\
 &= 0.197
 \end{aligned}$$

$$\begin{aligned}
 net_2^{hidden} &= \sum_{i=9} X_i W_{ij}^h + w_{bias_j}^h \\
 &= w_{bias2} + w_{12}.x_1 + w_{22}.x_2 + w_{32}.x_3 + w_{42}.x_4 + w_{52}.x_5 + w_{62}.x_6 + w_{72}.x_7 + \\
 &\quad w_{82}.x_8 + w_{92}.x_9 \\
 &= 0.222 + 0.222 * 0.444 + 0.222 * 0 + 0.222 * 0 + 0.222 * 0 + 0.222 * 0 \\
 &\quad * 0.111 + 0.222 * 0 + 0.222 * 0.222 + 0.222 * 0 + 0.222 * 0 \\
 &= 0.222 + 0.0987 + 0 + 0 + 0 + 0.0246 + 0 + 0.0493 + 0 + 0 \\
 &= 0.395
 \end{aligned}$$

Untuk keseluruhan hasil perhitungan nilai pada *neuron input layer* ke *neuron hidden layer* dapat dijabarkan pada tabel 3.11:

Tabel 3.11. Nilai jaringan pada *neuron input layer* ke *neuron hidden layer*

Jaringan <i>neuron input layer</i> ke <i>Neuron Hidden Layer</i>	Nilai jaringan <i>neuron input layer</i> ke <i>Neuron Hidden Layer</i>
net_1^{hidden}	0.197
net_2^{hidden}	0.395
net_3^{hidden}	0.592
net_4^{hidden}	0.790
net_5^{hidden}	0.987
net_6^{hidden}	1.185

2. Proses perhitungan fungsi aktivasi pada *unit hidden layer* setelah mendapatkan hasil perhitungan pada tabel 3.11 di setiap jaringan *neuron input layer* ke *neuron hidden layer* menggunakan persamaan 2.4:

$$\begin{aligned}
 f_{in_net\ n} &= \frac{1}{1 + e^{-net_{pj}^{hidden}}} \\
 f_{in_net\ 1} &= \frac{1}{1 + e^{(-0.197)}} = 0.549 \\
 f_{in_net\ 2} &= \frac{1}{1 + e^{(-0.395)}} = 0.597 \\
 f_{in_net\ 3} &= \frac{1}{1 + e^{(-0.592)}} = 0.643 \\
 f_{in_net\ 4} &= \frac{1}{1 + e^{(-0.790)}} = 0.687 \\
 f_{in_net\ 5} &= \frac{1}{1 + e^{(-0.987)}} = 0.728 \\
 f_{in_net\ 6} &= \frac{1}{1 + e^{(-1.185)}} = 0.765
 \end{aligned}$$

3. Proses perhitungan nilai jaringan pada *output layer* dari *input layer* yang mendapatkan hasil perhitungan fungsi aktivasi pada *neuron hidden layer* dengan menggunakan persamaan 2.5:

$$\begin{aligned}
 net_{ik}^{output} &= \sum_{k=6}^n (f_{in_net\ n} * w_{kj}^0) + w_{bias_k}^0 \\
 &= w_{bias} + w_1 \cdot f_{net\ 1} + w_2 \cdot f_{net\ 2} + w_3 \cdot f_{net\ 3} + w_4 \cdot f_{net\ 4} + w_5 \cdot f_{net\ 5}
 \end{aligned}$$

$$\begin{aligned}
& + w_6 \cdot f_{net\ 6} \\
& = 0.777 + 0.111 * 0.549 + 0.222 * 0.597 + 0.333 * 0.643 + 0.444 \\
& * 0.687 + 0.555 * 0.728 + 0.666 * 0.765 \\
& = 0.777 + 0.0610 + 0.132 + 0.214 + 0.305 + 0.404 + 0.510 \\
& = 1.221
\end{aligned}$$

4. Proses perhitungan fungsi aktivasi pada *output layer* dari *input layer* yang mendapatkan hasil perhitungan nilai jaringan pada *output layer* dengan menggunakan persamaan 2.6:

$$\begin{aligned}
O_{ik} &= f_{out_net} = \frac{1}{1 + e^{-net_k^{output}}} \\
O_{ik} &= f_{out_net} = \frac{1}{1 + e^{(-1.221)}} = 0.772
\end{aligned}$$

5. Proses perhitungan untuk *check error* pada *feedforward propagation* menggunakan persamaan 2.7:

Check error (epoch berhenti bila $|error| \leq 0.01$)

$$Error = Target - O_{ik}$$

$$Error = 0 - 0.772 = -0.772$$

6. Proses perhitungan untuk meminimalisasi *mean square error* pada penurunan gradien (*gradient descent*) pada *output layer* menggunakan persamaan 2.8:

$$\begin{aligned}
Mean\ square\ error &= \frac{1}{2} * (0 - (-0.772))^2 \\
&= 0.298
\end{aligned}$$

Tahap ADAM Pada Gradient Descent Perambatan – Balik (*Gradient Descent Backpropagation*)

7. Proses perhitungan *derivative partial error* jaringan *E* terhadap *weight* dan *bias* pada *output layer* menggunakan persamaan 2.9:

- Perhitungan *derivative partial error* jaringan *E* terhadap *weight* pada *output layer* di *neuron* pertama.

$$\frac{\partial E}{\partial W_{weight\ i}^o} = - \sum_{k=6} (T - f_{out_net}) * f_{out_net} * (1 - f_{out_net}) * f_{in_net\ n}$$

$$\frac{\partial E}{\partial W_{weight\ 1}^o} = - \sum_{k=6} (\text{Target} - O_{ik}) * f_k^o(\text{net}_{ik}^o) * [1 - f_k^o(\text{net}_{ik}^o)] * f_{in_net\ 1}$$

$$\frac{\partial E}{\partial W_{weight\ 1}^o} = -(0 - 0.772) * 0.772 * (1 - 0.772) * 0.549$$

$$\frac{\partial E}{\partial W_{weight\ 1}^o} = 0.772 * 0.772 * 0.227 * 0.549$$

$$\frac{\partial E}{\partial W_{weight\ 1}^o} = 0.0745$$

- Perhitungan *derivative partial error* jaringan E terhadap *weight* pada *output layer* di *neuron* kedua.

$$\frac{\partial E}{\partial W_{weight\ i}^o} = - \sum_{k=6} (T - f_{out_net}) * f_{out_net} * (1 - f_{out_net}) * f_{in_net\ n}$$

$$\frac{\partial E}{\partial W_{weight\ 2}^o} = - \sum_{k=6} (\text{Target} - O_{ik}) * f_k^o(\text{net}_{ik}^o) * [1 - f_k^o(\text{net}_{ik}^o)] * f_{in_net\ 2}$$

$$\frac{\partial E}{\partial W_{weight\ 2}^o} = -(0 - 0.772) * 0.772 * (1 - 0.772) * 0.597$$

$$\frac{\partial E}{\partial W_{weight\ 2}^o} = 0.772 * 0.772 * 0.227 * 0.597$$

$$\frac{\partial E}{\partial W_{weight\ 2}^o} = 0.0811$$

- Perhitungan *derivative partial error* jaringan E terhadap *bias* pada *output layer* di *neuron*.

$$\frac{\partial E}{\partial W_{bias}^o} = - \sum_{k=6} (T - f_{out_net}) * f_{out_net} * (1 - f_{out_net})$$

$$\frac{\partial E}{\partial W_{bias}^o} = - \sum_{k=6} (\text{Target} - O_{ik}) * f_k^o(\text{net}_{ik}^o) * [1 - f_k^o(\text{net}_{ik}^o)]$$

$$\frac{\partial E}{\partial W_{bias}^o} = -(0 - 0.772) * 0.772 * (1 - 0.772)$$

$$\frac{\partial E}{\partial W_{bias}^o} = 0.772 * 0.772 * 0.227$$

$$\frac{\partial E}{\partial W_{bias}^o} = 0.135$$

Untuk keseluruhan hasil nilai perhitungan *derivative partial error* jaringan E terhadap *weight* dan *bias* pada *output layer* di setiap *neuron* dapat dijabarkan pada tabel 3.12:

Tabel 3.12. Nilai *derivative partial error* terhadap *weight* dan *bias* pada *output layer* di setiap *neuron*

<i>Derivative partial weight</i> dan <i>bias</i> pada <i>output layer</i>	Nilai <i>derivative partial weight</i> dan <i>bias</i> pada <i>output layer</i>
$\frac{\partial E}{\partial W_{weight 1}^o}$	0.0745
$\frac{\partial E}{\partial W_{weight 2}^o}$	0.0811
$\frac{\partial E}{\partial W_{weight 3}^o}$	0.0874
$\frac{\partial E}{\partial W_{weight 4}^o}$	0.0934
$\frac{\partial E}{\partial W_{weight 5}^o}$	0.0989
$\frac{\partial E}{\partial W_{weight 6}^o}$	0.1040
$\frac{\partial E}{\partial W_{bias}^o}$	0.135

8. Proses perhitungan *ADAM* pertama pada estimasi momen pertama terhadap *weight* dan *bias* yang menerima hasil pada tabel 3.12 nilai perhitungan *derivative partial error* jaringan E terhadap *weight* dan *bias* pada *output layer* di setiap *neuron* untuk nilai $g_{weight(t)} = \frac{\partial E}{\partial W_{weight i}^o}$ dan $g_{bias(t)} = \frac{\partial E}{\partial W_{bias}^o}$ dengan menggunakan persamaan 2.11 dan persamaan 2.12:

- Perhitungan *ADAM* pertama pada estimasi momen pertama terhadap *weight* yang menerima hasil perhitungan *derivative partial* dari kuadrat *error* pada *output layer* di *weight neuron* pertama.

$$m_{weight(t)} = \beta_1 * m_{weight(t-1)} + (1 - \beta_1) * g_t$$

$$m_{weight(t)} = \beta_1 * m_{weight(t-1)} + (1 - \beta_1) * \frac{\partial E}{\partial W_{weight i}^o}$$

$$m_{weight(1)1} = \beta_1 * m_{weight(0)} + (1 - \beta_1) * \frac{\partial E}{\partial W_{weight 1}^o}$$

$$m_{weight(1)1} = 0.9 * 0 + (1 - 0.9) * (0.0745)$$

$$m_{weight(1)1} = 0 + 0.1 * (0.0745)$$

$$m_{weight(1)1} = 0.00745$$

- Perhitungan ADAM pertama pada estimasi momen pertama terhadap *weight* yang menerima hasil perhitungan *derivative partial* dari kuadrat *error* pada *output layer* di *weight neuron* kedua.

$$m_{weight(t)} = \beta_1 * m_{weight(t-1)} + (1 - \beta_1) * g_t$$

$$m_{weight(1)2} = \beta_1 * m_{weight(t-1)} + (1 - \beta_1) * \frac{\partial E}{\partial W_{weight i}^o}$$

$$m_{weight(1)2} = \beta_1 * m_{weight(0)} + (1 - \beta_1) * \frac{\partial E}{\partial W_{weight 2}^o}$$

$$m_{weight(1)2} = 0.9 * 0 + (1 - 0.9) * (0.0811)$$

$$m_{weight(1)2} = 0 + 0.1 * (0.0811)$$

$$m_{weight(1)2} = 0.00811$$

- Perhitungan ADAM pertama pada estimasi momen pertama terhadap *bias* yang menerima hasil perhitungan *derivative partial* dari kuadrat *error* pada *output layer* di *bias neuron* pertama.

$$m_{bias(t)} = \beta_1 * m_{bias(t-1)} + (1 - \beta_1) * g_t$$

$$m_{bias(t)} = \beta_1 * m_{bias(t-1)} + (1 - \beta_1) * \frac{\partial E}{\partial W_{bias}^o}$$

$$m_{bias(1)} = \beta_1 * m_{bias(0)} + (1 - \beta_1) * \frac{\partial E}{\partial W_{bias}^o}$$

$$m_{bias(1)} = 0.9 * 0 + (1 - 0.9) * (0.135)$$

$$m_{bias(1)} = 0 + 0.1 * (0.135)$$

$$m_{bias(1)} = 0.0135$$

Untuk keseluruhan hasil nilai perhitungan *ADAM* pertama pada estimasi momen pertama terhadap *weight* dan *bias* menerima hasil perhitungan *derivative partial error* jaringan *E* terhadap *weight* dan *bias* pada *output layer* di setiap *neuron* dapat dijabarkan pada tabel 3.13:

Tabel 3.13. Nilai perhitungan *ADAM* pertama pada estimasi momen pertama terhadap *weight* dan *bias* pada *output layer* di setiap *neuron*

<i>ADAM</i> pertama pada estimasi momen pertama terhadap <i>weight</i> dan <i>bias</i> pada <i>output layer</i>	Nilai <i>ADAM</i> pertama pada estimasi momen pertama terhadap <i>weight</i> dan <i>bias</i> pada <i>output layer</i>
$m_{\text{weight}(1)1}$	0.00745
$m_{\text{weight}(1)2}$	0.00811
$m_{\text{weight}(1)3}$	0.00874
$m_{\text{weight}(1)4}$	0.00934
$m_{\text{weight}(1)5}$	0.00989
$m_{\text{weight}(1)6}$	0.01040
$m_{\text{bias}(1)}$	0.0135

9. Pada perhitungan koreksi *weight* dan *bias* yang menerima hasil pada tabel 3.13 nilai perhitungan *ADAM* pertama pada estimasi moment pertama terhadap *weight* dan *bias* pada *output layer* di setiap *neuron* dengan menggunakan persamaan 2.13 dan persamaan 2.14 sebagai berikut :

- Perhitungan koreksi *weight* yang menerima hasil perhitungan *ADAM* pertama pada estimasi moment pertama di *output layer* pada *weight neuron* pertama.

$$\begin{aligned}\hat{m}_{\text{weight}(t)1} &= \frac{m_{\text{weight}(t)1}}{1 - \beta_1^t} \\ \hat{m}_{\text{weight}(1)1} &= \frac{0.00745}{1 - 0.9} \\ \hat{m}_{\text{weight}(1)1} &= \frac{0.00745}{0.1} \\ \hat{m}_{\text{weight}(1)1} &= 0.0745\end{aligned}$$

- Perhitungan koreksi *weight* yang menerima hasil perhitungan *ADAM* pertama pada estimasi moment pertama di *output layer* pada *weight neuron* kedua.

$$\hat{m}_{weight(t)2} = \frac{m_{weight(t)2}}{1 - \beta_1^t}$$

$$\hat{m}_{weight(1)2} = \frac{0.00811}{1 - 0.9}$$

$$\hat{m}_{weight(1)2} = \frac{0.00811}{0.1}$$

$$\hat{m}_{weight(1)2} = 0.0811$$

- Perhitungan koreksi *bias* yang menerima hasil perhitungan *ADAM* pertama pada estimasi moment pertama di *output layer* pada *bias neuron*.

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

$$\hat{m}_{bias(1)} = \frac{0.0135}{1 - 0.9}$$

$$\hat{m}_{bias(1)} = \frac{0.0135}{0.1}$$

$$\hat{m}_{bias(1)} = 0.135$$

Untuk keseluruhan hasil nilai perhitungan koreksi *weight* dan *bias* yang menerima hasil perhitungan *ADAM* pertama pada estimasi moment pertama terhadap *weight* dan *bias* pada *output layer* di setiap *neuron* dapat dijabarkan pada tabel 3.14:

Tabel 3.14. Nilai perhitungan koreksi *weight* dan *bias ADAM* pertama pada estimasi momen pertama pada *output layer* di setiap *neuron*

Koreksi <i>weight</i> dan <i>bias</i> pada <i>ADAM</i> pertama pada estimasi momen pertama pada <i>output layer</i>	Nilai koreksi <i>weight</i> dan <i>bias</i> pada <i>ADAM</i> pertama pada estimasi momen pertama pada <i>output layer</i>
$\hat{m}_{weight(1)1}$	0.0745
$\hat{m}_{weight(1)2}$	0.0811
$\hat{m}_{weight(1)3}$	0.0874
$\hat{m}_{weight(1)4}$	0.0934
$\hat{m}_{weight(1)5}$	0.0989
$\hat{m}_{weight(1)6}$	0.1040
$m_{bias(1)}$	0.0135

10. Pada perhitungan *ADAM* pertama pada estimasi momen kedua terhadap *weight* dan *bias* yang menerima hasil pada tabel 3.12 nilai perhitungan *derivative partial error* jaringan *E* terhadap *weight* dan *bias* pada *output layer* di setiap *neuron* untuk nilai $g_{\text{weight}(t)} = \frac{\partial E}{\partial W_{\text{weight} i}^o}$ dan $g_{\text{bias}(t)} = \frac{\partial E}{\partial W_{\text{bias}}^o}$ dengan menggunakan persamaan 2.15 dan persamaan 2.16 sebagai berikut :

- Perhitungan *ADAM* pertama pada estimasi momen kedua terhadap *weight* yang menerima hasil perhitungan *derivative partial error* jaringan *E* pada *output layer* di *weight neuron* pertama.

$$\begin{aligned} v_{\text{weight}(t)} &= \beta_2 * v_{\text{weight}(t-1)} + (1 - \beta_2) * (g_t)^2 \\ v_{\text{weight}(t)} &= \beta_2 * v_{\text{weight}(t-1)} + (1 - \beta_2) * \left(\frac{\partial E}{\partial W_{\text{weight} i}^o}\right)^2 \\ v_{\text{weight}(1)1} &= \beta_2 * v_{\text{weight}(0)} + (1 - \beta_2) * \left(\frac{\partial E}{\partial W_{\text{weight} 1}^o}\right)^2 \end{aligned}$$

$$\begin{aligned} v_{\text{weight}(1)1} &= 0.999 * 0 + (1 - 0.999) * (0.0745)^2 \\ v_{\text{weight}(1)1} &= 0 + 0.001 * (0.0745)^2 \\ v_{\text{weight}(1)1} &= 0.001 * (0.00555025) \\ v_{\text{weight}(1)1} &= 0.00000555025 \end{aligned}$$

- Perhitungan *ADAM* pertama pada estimasi momen kedua terhadap *weight* yang menerima hasil perhitungan *derivative partial error* jaringan *E* pada *output layer* di *weight neuron* kedua.

$$\begin{aligned} v_{\text{weight}(t)} &= \beta_2 * v_{\text{weight}(t-1)} + (1 - \beta_2) * (g_t)^2 \\ v_{\text{weight}(t)} &= \beta_2 * v_{\text{weight}(t-1)} + (1 - \beta_2) * \left(\frac{\partial E}{\partial W_{\text{weight} i}^o}\right)^2 \\ v_{\text{weight}(1)2} &= \beta_2 * v_{\text{weight}(0)} + (1 - \beta_2) * \left(\frac{\partial E}{\partial W_{\text{weight} 2}^o}\right)^2 \end{aligned}$$

$$\begin{aligned} v_{\text{weight}(1)2} &= 0.999 * 0 + (1 - 0.999) * (0.0811)^2 \\ v_{\text{weight}(1)2} &= 0 + 0.001 * (0.0811)^2 \\ v_{\text{weight}(1)2} &= 0.001 * (0.00657721) \\ v_{\text{weight}(1)2} &= 0.00000657721 \end{aligned}$$

- Perhitungan *ADAM* pertama pada estimasi momen kedua terhadap *bias* yang menerima hasil perhitungan *derivative partial error* jaringan *E* pada *output layer* di *bias neuron*.

$$\begin{aligned} v_{bias(t)} &= \beta_2 * v_{bias(t-1)} + (1 - \beta_2) * (g_t)^2 \\ v_{bias(t)} &= \beta_2 * v_{bias(1-1)} + (1 - \beta_2) * \left(\frac{\partial E}{\partial W_{bias}^o}\right)^2 \\ v_{bias(1)} &= \beta_2 * v_{weight(0)} + (1 - \beta_2) * \left(\frac{\partial E}{\partial W_{bias}^o}\right)^2 \\ v_{bias(1)} &= 0.999 * 0 + (1 - 0.999) * (0.135)^2 \\ v_{bias(1)} &= 0 + 0.001 * (0.135)^2 \\ v_{bias(1)} &= 0.001 * (0.018225) \\ v_{bias(1)} &= 0.000018225 \end{aligned}$$

Untuk keseluruhan hasil nilai perhitungan *ADAM* pertama pada estimasi momen kedua terhadap *weight* dan *bias* yang menerima hasil perhitungan *derivative partial error* jaringan *E* terhadap *weight* dan *bias* pada *output layer* di setiap *neuron* dapat dijabarkan pada tabel 3.15:

Tabel 3.15. Nilai perhitungan *ADAM* pertama pada estimasi momen kedua terhadap *weight* dan *bias* pada *output layer* di setiap *neuron*

<i>ADAM</i> pertama pada estimasi momen kedua terhadap <i>weight</i> dan <i>bias</i> pada <i>output layer</i>	Nilai <i>ADAM</i> pertama pada estimasi momen kedua terhadap <i>weight</i> dan <i>bias</i> pada <i>output layer</i>
$v_{weight(1)1}$	0.00000555025
$v_{weight(1)2}$	0.00000657721
$v_{weight(1)3}$	0.00000763876
$v_{weight(1)4}$	0.00000872356
$v_{weight(1)5}$	0.00000978121
$v_{weight(1)6}$	0.000010816
$v_{bias(1)}$	0.000018225

- Pada perhitungan koreksi *weight* dan *bias* yang menerima hasil pada tabel 3.15 nilai perhitungan *ADAM* pertama pada estimasi momen kedua terhadap *weight* dan *bias* pada *output layer* di setiap *neuron* menggunakan persamaan 2.17 dan persamaan 2.18 sebagai berikut :

- Perhitungan koreksi *weight* yang menerima hasil perhitungan *ADAM* pertama pada estimasi moment kedua di *output layer* pada *weight neuron* pertama.

$$\tilde{V}_{\text{weight}(t)} = \frac{v_{\text{weight}(t)}}{1 - \beta_2^t}$$

$$\tilde{V}_{\text{weight}(t)1} = \frac{v_{\text{weight}(t)1}}{1 - \beta_2^t}$$

$$\tilde{V}_{\text{weight}(t)1} = \frac{0.00000555025}{1 - 0.999}$$

$$\tilde{V}_{\text{weight}(t)1} = \frac{0.00000555025}{0.001}$$

$$\tilde{V}_{\text{weight}(t)1} = 0.00555025$$

- Perhitungan koreksi *weight* yang menerima hasil perhitungan *ADAM* pertama pada estimasi moment kedua di *output layer* pada *weight neuron* kedua.

$$\tilde{V}_{\text{weight}(t)} = \frac{v_{\text{weight}(t)}}{1 - \beta_2^t}$$

$$\tilde{V}_{\text{weight}(t)2} = \frac{v_{\text{weight}(t)2}}{1 - \beta_2^t}$$

$$\tilde{V}_{\text{weight}(t)2} = \frac{0.00000657721}{1 - 0.999}$$

$$\tilde{V}_{\text{weight}(t)2} = \frac{0.00000657721}{0.001}$$

$$\tilde{V}_{\text{weight}(t)2} = 0.00657721$$

- Perhitungan koreksi *bias* yang menerima hasil perhitungan *ADAM* pertama pada estimasi moment kedua di *output layer* pada *bias neuron*.

$$\tilde{V}_{\text{bias}(t)} = \frac{v_{\text{bias}(t)}}{1 - \beta_2^t}$$

$$\tilde{V}_{\text{bias}(t)} = \frac{v_{\text{bias}(1)}}{1 - \beta_2^t}$$

$$\tilde{V}_{\text{bias}(1)} = \frac{0.000018225}{1 - 0.999}$$

$$\tilde{V}_{\text{bias}(1)} = \frac{0.000018225}{0.001}$$

$$\tilde{V}_{\text{bias}(1)} = 0.018225$$

Untuk keseluruhan hasil nilai perhitungan koreksi *weight* dan *bias* yang menerima hasil perhitungan *ADAM* pertama pada estimasi moment kedua terhadap *weight* dan *bias* pada *output layer* di setiap *neuron* dapat dijabarkan pada tabel 3.16

Tabel 3.16. Nilai perhitungan koreksi *weight* dan *bias ADAM* pertama pada estimasi momen kedua pada *output layer* di setiap *neuron*

Koreksi <i>weight</i> dan <i>bias ADAM</i> pertama pada estimasi momen kedua pada <i>output layer</i>	Nilai koreksi <i>weight</i> dan <i>bias ADAM</i> pertama pada estimasi momen kedua pada <i>output layer</i>
$\tilde{V}_{weight(t)1}$	0.00555025
$\tilde{V}_{weight(t)2}$	0.00657721
$\tilde{V}_{weight(t)3}$	0.00763876
$\tilde{V}_{weight(t)4}$	0.00872356
$\tilde{V}_{weight(t)5}$	0.00978121
$\tilde{V}_{weight(t)6}$	0.010816
$\tilde{V}_{bias(1)}$	0.018225

12. Proses perhitungan *derivative partial error* jaringan *E* terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron* menggunakan persamaan 2.10:

- Perhitungan *derivative partial error* jaringan *E* terhadap *weight* pada *input layer neuron* pertama dan *hidden layer neuron* pertama.

$$\begin{aligned} \frac{\partial E}{\partial W_{ij}^h} &= - \sum_k (\text{Target} - O_{ik}) * f_k^o(net_{ik}^o) * \left[1 - f_k^o(net_{ik}^o)\right] * w_{kj}^o * f_j^h(net_{ij}^h) \\ &\quad * [1 - f_j^h(net_{ij}^h)] * X_i \\ \frac{\partial E}{\partial W_{11}^h} &= - \sum_k (\text{Target} - O_{ik}) * f_k^o(net_{ik}^o) * \left[1 - f_k^o(net_{ik}^o)\right] * w_1^o * f_j^h(net_{ij}^h) \\ &\quad * [1 - f_j^h(net_{ij}^h)] * X_1 \\ \frac{\partial E}{\partial W_{11}^h} &= -(0 - 0.772) * 0.772 * [1 - 0.772] * 0.111 * 0.549 * [1 - 0.549] * 0.444 \\ \frac{\partial E}{\partial W_{11}^h} &= 0.772 * 0.772 * 0.227 * 0.111 * 0.549 * 0.450 * 0.444 \\ \frac{\partial E}{\partial W_{11}^h} &= 0.00166 \end{aligned}$$

- Perhitungan *derivative partial error* jaringan *E* terhadap *weight* pada *input layer neuron* pertama dan *hidden layer neuron* kedua.

$$\begin{aligned}
\frac{\partial E}{\partial W_{ij}^h} &= - \sum_k (\text{Target} - O_{ik}) * f_k^0(\text{net}_{ik}^0) * \left[1 - f_k^0(\text{net}_{ik}^0)\right] * w_{kj}^0 * f_j^h(\text{net}_{ij}^h) \\
&\quad * [1 - f_j^h(\text{net}_{ij}^h)] * X_i \\
\frac{\partial E}{\partial W_{12}^h} &= - \sum_k (\text{Target} - O_{ik}) * f_k^0(\text{net}_{ik}^0) * \left[1 - f_k^0(\text{net}_{ik}^0)\right] * w_2^0 * f_j^h(\text{net}_{ij}^h) \\
&\quad * [1 - f_j^h(\text{net}_{ij}^h)] * X_1 \\
\frac{\partial E}{\partial W_{12}^h} &= -(0 - 0.772) * 0.772 * [1 - 0.772] * 0.222 * 0.597 * [1 - 0.597] * 0.444 \\
\frac{\partial E}{\partial W_{12}^h} &= 0.772 * 0.772 * 0.227 * 0.222 * 0.597 * 0.402 * 0.444 \\
\frac{\partial E}{\partial W_{12}^h} &= 0.00322
\end{aligned}$$

- Perhitungan *derivative partial error* jaringan E terhadap *bias* pada *input layer* di *hidden layer neuron* pertama.

$$\begin{aligned}
\frac{\partial E}{\partial W_{bias\ i}^h} &= - \sum_k (\text{Target} - O_{ik}) * f_k^0(\text{net}_{ik}^0) * \left[1 - f_k^0(\text{net}_{ik}^0)\right] * w_{ik}^0 * f_j^h(\text{net}_{ij}^h) \\
&\quad * [1 - f_j^h(\text{net}_{ij}^h)] \\
\frac{\partial E}{\partial W_{bias\ 1}^h} &= - \sum_k (\text{Target} - O_{ik}) * O_{ik} * [1 - O_{ik}] * w_{bias\ 1}^0 * f_{in_net\ 1} * [1 - f_{in_net\ 1}] \\
\frac{\partial E}{\partial W_{bias\ 1}^h} &= -(0 - 0.772) * 0.772 * [1 - 0.772] * 0.111 * 0.549 * [1 - 0.549] \\
\frac{\partial E}{\partial W_{bias\ 1}^h} &= 0.772 * 0.772 * 0.227 * 0.111 * 0.549 * 0.450 \\
\frac{\partial E}{\partial W_{bias\ 1}^h} &= 0.00373
\end{aligned}$$

- Perhitungan *derivative partial error* jaringan E terhadap *bias* pada *input layer* di *hidden layer neuron* kedua.

$$\begin{aligned}
\frac{\partial E}{\partial W_{bias\ i}^h} &= - \sum_k (\text{Target} - O_{ik}) * f_k^0(\text{net}_{ik}^0) * \left[1 - f_k^0(\text{net}_{ik}^0)\right] * w_{ik}^0 * f_j^h(\text{net}_{ij}^h) \\
&\quad * [1 - f_j^h(\text{net}_{ij}^h)] \\
\frac{\partial E}{\partial W_{bias\ 2}^h} &= - \sum_k (\text{Target} - O_{ik}) * O_{ik} * [1 - Y_o] * w_{bias\ 2}^0 * f_{in_net\ 2} * [1 - f_{in_net\ 2}] \\
\frac{\partial E}{\partial W_{bias\ 2}^h} &= -(0 - 0.772) * 0.772 * [1 - 0.772] * 0.222 * 0.597 * [1 - 0.597]
\end{aligned}$$

$$\frac{\partial E}{\partial W_{bias \ 2}^h} = 0.772 * 0.772 * 0.227 * 0.222 * 0.597 * 0.402$$

$$\frac{\partial E}{\partial W_{bias \ 2}^h} = 0.00725$$

Untuk keseluruhan hasil nilai perhitungan *derivative partial error* jaringan E terhadap *weight* dan *bias* pada *input layer* proses *feedbackward* dari *output layer* ke *input layer* di setiap *neuron* dapat dijabarkan pada tabel 3.17:

Tabel 3.17. Nilai perhitungan *derivative partial error* jaringan E terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron*.

<i>Derivative partial error</i> terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>	Nilai <i>derivative partial error</i> terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>
Weight Pada Hidden Layer	
$\frac{\partial E}{\partial W_{11}^h}$	0.00166
$\frac{\partial E}{\partial W_{12}^h}$	0.00322
$\frac{\partial E}{\partial W_{13}^h}$	0.00461
$\frac{\partial E}{\partial W_{14}^h}$	0.00576
$\frac{\partial E}{\partial W_{15}^h}$	0.00663
$\frac{\partial E}{\partial W_{16}^h}$	0.00721
$\frac{\partial E}{\partial W_{21}^h}$	0
$\frac{\partial E}{\partial W_{22}^h}$	0
$\frac{\partial E}{\partial W_{23}^h}$	0
$\frac{\partial E}{\partial W_{24}^h}$	0
$\frac{\partial E}{\partial W_{25}^h}$	0

Tabel 3.17. Nilai perhitungan *derivative partial error* jaringan E terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron* (Lanjutan).

<i>Derivative partial error</i> terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>	Nilai <i>derivative partial error</i> terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>
$\frac{\partial E}{\partial W_{26}^h}$	0
$\frac{\partial E}{\partial W_{31}^h}$	0
$\frac{\partial E}{\partial W_{32}^h}$	0
$\frac{\partial E}{\partial W_{33}^h}$	0
$\frac{\partial E}{\partial W_{34}^h}$	0
$\frac{\partial E}{\partial W_{35}^h}$	0
$\frac{\partial E}{\partial W_{36}^h}$	0
$\frac{\partial E}{\partial W_{41}^h}$	0
$\frac{\partial E}{\partial W_{42}^h}$	0
$\frac{\partial E}{\partial W_{43}^h}$	0
$\frac{\partial E}{\partial W_{44}^h}$	0
$\frac{\partial E}{\partial W_{45}^h}$	0
$\frac{\partial E}{\partial W_{46}^h}$	0
$\frac{\partial E}{\partial W_{51}^h}$	0.000415
$\frac{\partial E}{\partial W_{52}^h}$	0.000806
$\frac{\partial E}{\partial W_{53}^h}$	0.00115

Tabel 3.17. Nilai perhitungan *derivative partial error* jaringan E terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron* (Lanjutan).

<i>Derivative partial error</i> terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>	Nilai <i>derivative partial error</i> terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>
$\frac{\partial E}{\partial W_{54}^h}$	0.001441
$\frac{\partial E}{\partial W_{55}^h}$	0.00165
$\frac{\partial E}{\partial W_{56}^h}$	0.00180
$\frac{\partial E}{\partial W_{61}^h}$	0
$\frac{\partial E}{\partial W_{62}^h}$	0
$\frac{\partial E}{\partial W_{63}^h}$	0
$\frac{\partial E}{\partial W_{64}^h}$	0
$\frac{\partial E}{\partial W_{65}^h}$	0
$\frac{\partial E}{\partial W_{66}^h}$	0
$\frac{\partial E}{\partial W_{71}^h}$	0.000830
$\frac{\partial E}{\partial W_{72}^h}$	0.00161
$\frac{\partial E}{\partial W_{73}^h}$	0.00230
$\frac{\partial E}{\partial W_{74}^h}$	0.00288
$\frac{\partial E}{\partial W_{75}^h}$	0.00331
$\frac{\partial E}{\partial W_{76}^h}$	0.00360
$\frac{\partial E}{\partial W_{81}^h}$	0

Tabel 3.17. Nilai perhitungan *derivative partial error* jaringan E terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron* (Lanjutan).

<i>Derivative partial error</i> terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>	Nilai <i>derivative partial error</i> terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>
$\frac{\partial E}{\partial W_{82}^h}$	0
$\frac{\partial E}{\partial W_{83}^h}$	0
$\frac{\partial E}{\partial W_{84}^h}$	0
$\frac{\partial E}{\partial W_{85}^h}$	0
$\frac{\partial E}{\partial W_{86}^h}$	0
$\frac{\partial E}{\partial W_{91}^h}$	0
$\frac{\partial E}{\partial W_{92}^h}$	0
$\frac{\partial E}{\partial W_{93}^h}$	0
$\frac{\partial E}{\partial W_{94}^h}$	0
$\frac{\partial E}{\partial W_{95}^h}$	0
$\frac{\partial E}{\partial W_{96}^h}$	0
<i>Bias Pada Hidden Layer</i>	
$\frac{\partial E}{\partial W_{bias \ 1}^h}$	0.00373
$\frac{\partial E}{\partial W_{bias \ 2}^h}$	0.00725
$\frac{\partial E}{\partial W_{bias \ 3}^h}$	0.0103
$\frac{\partial E}{\partial W_{bias \ 4}^h}$	0.0129

Tabel 3.17. Nilai perhitungan *derivative partial error* jaringan E terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron* (Lanjutan).

<i>Derivative partial error</i> terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>	Nilai <i>derivative partial error</i> terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>
<i>Bias Pada Hidden Layer</i>	
$\frac{\partial E}{\partial W_{bias \ 5}^h}$	0.0149
$\frac{\partial E}{\partial W_{bias \ 6}^h}$	0.0162

10. Pada perhitungan *ADAM* kedua pada estimasi momen pertama terhadap *weight* dan *bias* yang menerima hasil pada tabel 3.17 nilai perhitungan *derivative partial error* jaringan E terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron* untuk nilai $g_{weight(t)} = \frac{\partial E}{\partial W_{weight \ ij}^h}$ dan $g_{bias(t)} = \frac{\partial E}{\partial W_{bias}^h}$ dengan menggunakan persamaan 2.11 dan persamaan 2.12 sebagai berikut :

- Perhitungan *ADAM* kedua pada estimasi momen pertama terhadap *weight* yang menerima hasil perhitungan *derivative partial* dari kuadrat *error* pada *input layer neuron* pertama dan *hidden layer neuron* pertama.

$$\begin{aligned}m_{weight(t)} &= \beta_1 * m_{weight(t-1)} + (1 - \beta_1) * g_t \\m_{weight(t)} &= \beta_1 * m_{weight(1-1)} + (1 - \beta_1) * \frac{\partial E}{\partial W_{weight \ ij}^h} \\m_{weight(1)ij} &= \beta_1 * m_{weight(0)} + (1 - \beta_1) * \frac{\partial E}{\partial W_{weight \ 11}^h}\end{aligned}$$

$$m_{weight(1)11} = 0.9 * 0 + (1 - 0.9) * (0.00166)$$

$$m_{weight(1)11} = 0 + 0.1 * (0.00166)$$

$$m_{weight(1)11} = 0.000166$$

- Perhitungan *ADAM* kedua pada estimasi momen pertama terhadap *weight* yang menerima hasil perhitungan *derivative partial* dari kuadrat *error* pada *input layer neuron* pertama dan *hidden layer neuron* kedua.

$$m_{weight(t)} = \beta_1 * m_{weight(t-1)} + (1 - \beta_1) * g_t$$

$$m_{weight(t)} = \beta_1 * m_{weight(1-1)} + (1 - \beta_1) * \frac{\partial E}{\partial W_{weight ij}^h}$$

$$m_{weight(1)ij} = \beta_1 * m_{weight(0)} + (1 - \beta_1) * \frac{\partial E}{\partial W_{weight 12}^h}$$

$$m_{weight(1)12} = 0.9 * 0 + (1 - 0.9) * (0.00322)$$

$$m_{weight(1)12} = 0 + 0.1 * (0.00322)$$

$$m_{weight(1)12} = 0.000322$$

- Perhitungan ADAM kedua pada estimasi momen pertama terhadap *bias* yang menerima hasil perhitungan *derivative partial* dari kuadrat *error* pada *hidden layer* di *neuron bias* pertama.

$$m_{bias(t)} = \beta_1 * m_{bias(t-1)} + (1 - \beta_1) * g_t$$

$$m_{bias(t)} = \beta_1 * m_{bias(1-1)} + (1 - \beta_1) * \frac{\partial E}{\partial W_{bias i}^h}$$

$$m_{bias(1)1} = \beta_1 * m_{bias(0)} + (1 - \beta_1) * \frac{\partial E}{\partial W_{bias 1}^h}$$

$$m_{bias(1)1} = 0.9 * 0 + (1 - 0.9) * (0.00373)$$

$$m_{bias(1)1} = 0 + 0.1 * (0.00373)$$

$$m_{bias(1)1} = 0.000373$$

- Perhitungan ADAM kedua pada estimasi momen pertama terhadap *bias* yang menerima hasil perhitungan *derivative partial* dari kuadrat *error* pada *hidden layer* di *neuron bias* kedua.

$$m_{bias(t)} = \beta_1 * m_{bias(t-1)} + (1 - \beta_1) * g_t$$

$$m_{bias(t)} = \beta_1 * m_{bias(1-1)} + (1 - \beta_1) * \frac{\partial E}{\partial W_{bias i}^h}$$

$$m_{bias(1)2} = \beta_1 * m_{bias(0)} + (1 - \beta_1) * \frac{\partial E}{\partial W_{bias 2}^h}$$

$$m_{bias(1)2} = 0.9 * 0 + (1 - 0.9) * (0.00725)$$

$$m_{bias(1)2} = 0 + 0.1 * (0.00725)$$

$$m_{bias(1)2} = 0.000725$$

Untuk keseluruhan hasil nilai perhitungan *ADAM* kedua pada estimasi momen pertama terhadap *weight* dan *bias* yang menerima hasil perhitungan *derivative partial error* jaringan *E* terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron* dapat dijabarkan pada tabel 3.18:

Tabel 3.18. Nilai perhitungan *ADAM* kedua pada estimasi momen pertama terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron*

<i>ADAM</i> pertama pada estimasi momen pertama terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i>	Nilai <i>ADAM</i> pertama pada estimasi momen pertama terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i>
<i>Weight Pada Hidden Layer</i>	
$m_{weight(1)11}$	0.000166
$m_{weight(1)12}$	0.000322
$m_{weight(1)13}$	0.000461
$m_{weight(1)14}$	0.000576
$m_{weight(1)15}$	0.000663
$m_{weight(1)16}$	0.000721
$m_{weight(1)21}$	0
$m_{weight(1)22}$	0
$m_{weight(1)23}$	0
$m_{weight(1)24}$	0
$m_{weight(1)25}$	0
$m_{weight(1)26}$	0
$m_{weight(1)31}$	0
$m_{weight(1)32}$	0
$m_{weight(1)33}$	0
$m_{weight(1)34}$	0
$m_{weight(1)35}$	0
$m_{weight(1)36}$	0
$m_{weight(1)41}$	0
$m_{weight(1)42}$	0
$m_{weight(1)43}$	0
$m_{weight(1)44}$	0
$m_{weight(1)45}$	0

Tabel 3.18. Nilai perhitungan *ADAM* kedua pada estimasi momen pertama terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron* (Lanjutan).

<i>ADAM</i> pertama pada estimasi momen pertama terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i>	Nilai <i>ADAM</i> pertama pada estimasi momen pertama terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i>
$m_{weight(1)46}$	0
$m_{weight(1)51}$	0.0000415
$m_{weight(1)52}$	0.0000806
$m_{weight(1)53}$	0.000115
$m_{weight(1)54}$	0.0001441
$m_{weight(1)55}$	0.000165
$m_{weight(1)56}$	0.00018
$m_{weight(1)61}$	0
$m_{weight(1)62}$	0
$m_{weight(1)63}$	0
$m_{weight(1)64}$	0
$m_{weight(1)65}$	0
$m_{weight(1)66}$	0
$m_{weight(1)71}$	0.000083
$m_{weight(1)72}$	0.000161
$m_{weight(1)73}$	0.00023
$m_{weight(1)74}$	0.000288
$m_{weight(1)75}$	0.000331
$m_{weight(1)76}$	0.00036
$m_{weight(1)81}$	0
$m_{weight(1)82}$	0
$m_{weight(1)83}$	0
$m_{weight(1)84}$	0
$m_{weight(1)85}$	0
$m_{weight(1)86}$	0
$m_{weight(1)91}$	0
$m_{weight(1)92}$	0
$m_{weight(1)93}$	0
$m_{weight(1)94}$	0

Tabel 3.18. Nilai perhitungan *ADAM* kedua pada estimasi momen pertama terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron* (Lanjutan).

<i>ADAM</i> pertama pada estimasi momen pertama terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i>	Nilai <i>ADAM</i> pertama pada estimasi momen pertama terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i>
$m_{\text{weight}(1)95}$	0
$m_{\text{weight}(1)96}$	0
<i>Bias Pada Hidden Layer</i>	
$m_{\text{bias}(1)1}$	0.000373
$m_{\text{bias}(1)2}$	0.000725
$m_{\text{bias}(1)3}$	0.00103
$m_{\text{bias}(1)4}$	0.00129
$m_{\text{bias}(1)5}$	0.00149
$m_{\text{bias}(1)6}$	0.00162

11. Proses perhitungan koreksi *weight* dan *bias* yang menerima hasil pada tabel 3.18 nilai perhitungan *ADAM* kedua pada estimasi moment pertama terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron* dengan menggunakan persamaan 2.13 dan persamaan 2.14:

- Perhitungan koreksi *weight* yang menerima hasil perhitungan *ADAM* kedua pada estimasi moment pertama terhadap *weight* pada *input layer neuron* pertama dan *hidden layer neuron* pertama.

$$\bar{m}_{\text{weight}(1)ij} = \frac{m_{\text{weight}(t)ij}}{1 - \beta_1^t}$$

$$\bar{m}_{\text{weight}(t)11} = \frac{m_{\text{weight}(t)11}}{1 - \beta_1^t}$$

$$\bar{m}_{\text{weight}(1)11} = \frac{0.000166}{1 - 0.9}$$

$$\bar{m}_{\text{weight}(1)11} = \frac{0.000166}{0.1}$$

$$\bar{m}_{\text{weight}(1)11} = 0.00166$$

- Perhitungan koreksi *weight* yang menerima hasil perhitungan *ADAM* kedua pada estimasi moment pertama terhadap *weight* pada *input layer neuron* pertama dan *hidden layer neuron* kedua.

$$\hat{m}_{weight(1)ij} = \frac{m_{weight(t)ij}}{1 - \beta_1^t}$$

$$\hat{m}_{weight(t)12} = \frac{m_{weight(t)12}}{1 - \beta_1^t}$$

$$\hat{m}_{weight(1)12} = \frac{0.000322}{1 - 0.9}$$

$$\hat{m}_{weight(1)12} = \frac{0.000322}{0.1}$$

$$\hat{m}_{weight(1)12} = 0.00322$$

- Perhitungan koreksi *bias* yang menerima hasil perhitungan *ADAM* kedua pada estimasi moment pertama terhadap *bias* pada *hidden layer neuron* pertama.

$$\hat{m}_{bias(1)ij} = \frac{m_{bias(t)ij}}{1 - \beta_1^t}$$

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

$$\hat{m}_{bias(t)1} = \frac{0.000373}{1 - 0.9}$$

$$\hat{m}_{bias(t)1} = \frac{0.000373}{0.1}$$

$$\hat{m}_{bias(t)1} = 0.00373$$

- Perhitungan koreksi *bias* yang menerima hasil perhitungan *ADAM* kedua pada estimasi moment pertama terhadap *bias* pada *hidden layer neuron* kedua.

$$\hat{m}_{bias(1)ij} = \frac{m_{bias(t)ij}}{1 - \beta_1^t}$$

$$\hat{m}_{bias(t)2} = \frac{m_{bias(t)2}}{1 - \beta_1^t}$$

$$\hat{m}_{bias(t)2} = \frac{0.000725}{1 - 0.9}$$

$$\hat{m}_{bias(t)2} = \frac{0.000725}{0.1}$$

$$\hat{m}_{bias(t)2} = 0.00725$$

Untuk keseluruhan hasil nilai perhitungan koreksi *weight* dan *bias* yang menerima hasil perhitungan *ADAM* kedua pada estimasi moment pertama terhadap *weight* pada *hidden layer* di setiap *neuron* dapat dijabarkan pada tabel 3.19:

Tabel 3.19. Nilai perhitungan koreksi *weight* dan *bias ADAM* kedua pada estimasi momen pertama pada *hidden layer* di setiap *neuron*.

Koreksi <i>weight</i> dan <i>bias ADAM</i> kedua pada estimasi momen pertama pada <i>hidden layer</i>	Nilai koreksi <i>weight</i> dan <i>bias ADAM</i> kedua pada estimasi momen pertama pada <i>hidden layer</i>
Weight Pada Hidden Layer	
$\bar{m}_{\text{weight}(1)11}$	0.00166
$\bar{m}_{\text{weight}(1)12}$	0.00322
$\bar{m}_{\text{weight}(1)13}$	0.00461
$\bar{m}_{\text{weight}(1)14}$	0.00576
$\bar{m}_{\text{weight}(1)15}$	0.00663
$\bar{m}_{\text{weight}(1)16}$	0.00721
$\bar{m}_{\text{weight}(1)21}$	0
$\bar{m}_{\text{weight}(1)22}$	0
$\bar{m}_{\text{weight}(1)23}$	0
$\bar{m}_{\text{weight}(1)24}$	0
$\bar{m}_{\text{weight}(1)25}$	0
$\bar{m}_{\text{weight}(1)26}$	0
$\bar{m}_{\text{weight}(1)31}$	0
$\bar{m}_{\text{weight}(1)32}$	0
$\bar{m}_{\text{weight}(1)33}$	0
$\bar{m}_{\text{weight}(1)34}$	0
$\bar{m}_{\text{weight}(1)35}$	0
$\bar{m}_{\text{weight}(1)36}$	0
$\bar{m}_{\text{weight}(1)41}$	0
$\bar{m}_{\text{weight}(1)42}$	0
$\bar{m}_{\text{weight}(1)43}$	0
$\bar{m}_{\text{weight}(1)44}$	0
$\bar{m}_{\text{weight}(1)45}$	0
$\bar{m}_{\text{weight}(1)46}$	0
$\bar{m}_{\text{weight}(1)51}$	0.000415
$\bar{m}_{\text{weight}(1)52}$	0.000806

Tabel 3.19. Nilai perhitungan koreksi *weight* dan *bias ADAM* kedua pada estimasi momen pertama pada *hidden layer* di setiap *neuron* (Lanjutan).

Koreksi <i>weight</i> dan <i>bias ADAM</i> kedua pada estimasi momen pertama pada <i>hidden layer</i>	Nilai koreksi <i>weight</i> dan <i>bias ADAM</i> kedua pada estimasi momen pertama pada <i>hidden layer</i>
$\bar{m}_{\text{weight}(1)53}$	0.00115
$\bar{m}_{\text{weight}(1)54}$	0.001441
$\bar{m}_{\text{weight}(1)55}$	0.00165
$\bar{m}_{\text{weight}(1)56}$	0.0018
$\bar{m}_{\text{weight}(1)61}$	0
$\bar{m}_{\text{weight}(1)62}$	0
$\bar{m}_{\text{weight}(1)63}$	0
$\bar{m}_{\text{weight}(1)64}$	0
$\bar{m}_{\text{weight}(1)65}$	0
$\bar{m}_{\text{weight}(1)66}$	0
$\bar{m}_{\text{weight}(1)71}$	0.00083
$\bar{m}_{\text{weight}(1)72}$	0.00161
$\bar{m}_{\text{weight}(1)73}$	0.0023
$\bar{m}_{\text{weight}(1)74}$	0.00288
$\bar{m}_{\text{weight}(1)75}$	0.00331
$\bar{m}_{\text{weight}(1)76}$	0.0036
$\bar{m}_{\text{weight}(1)81}$	0
$\bar{m}_{\text{weight}(1)82}$	0
$\bar{m}_{\text{weight}(1)83}$	0
$\bar{m}_{\text{weight}(1)84}$	0
$\bar{m}_{\text{weight}(1)85}$	0
$\bar{m}_{\text{weight}(1)86}$	0
$\bar{m}_{\text{weight}(1)91}$	0
$\bar{m}_{\text{weight}(1)92}$	0
$\bar{m}_{\text{weight}(1)93}$	0
$\bar{m}_{\text{weight}(1)94}$	0
$\bar{m}_{\text{weight}(1)95}$	0
$\bar{m}_{\text{weight}(1)96}$	0
<i>Bias Pada Hidden Layer</i>	
$\bar{m}_{\text{bias } (t)1}$	0.00373
$\bar{m}_{\text{bias } (t)2}$	0.00725

Tabel 3.19. Nilai perhitungan koreksi *weight* dan *bias ADAM* kedua pada estimasi momen pertama pada *hidden layer* di setiap *neuron* (Lanjutan).

Koreksi <i>weight</i> dan <i>bias ADAM</i> kedua pada estimasi momen pertama pada <i>hidden layer</i>	Nilai koreksi <i>weight</i> dan <i>bias ADAM</i> kedua pada estimasi momen pertama pada <i>hidden layer</i>
Bias Pada Hidden Layer	
$\bar{m}_{bias(t)3}$	0.0103
$\bar{m}_{bias(t)4}$	0.0129
$\bar{m}_{bias(t)5}$	0.0149
$\bar{m}_{bias(t)6}$	0.0162

12. Pada perhitungan *ADAM* kedua pada estimasi momen kedua terhadap *weight* dan *bias* yang menerima hasil pada tabel 3.17 nilai perhitungan *derivative partial error* jaringan *E* terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron* untuk nilai $g_{weight(t)} = \frac{\partial E}{\partial W_{weight\ ij}^h}$ dan $g_{bias(t)} = \frac{\partial E}{\partial W_{bias}^h}$ dengan menggunakan persamaan 2.15 dan persamaan 2.16 sebagai berikut :

- Perhitungan *ADAM* kedua pada estimasi momen kedua terhadap *weight* yang menerima hasil perhitungan *derivative partial error* jaringan *E* pada *input layer neuron* pertama dan *hidden layer neuron* pertama.

$$v_{weight(t)} = \beta_2 * v_{weight(t-1)} + (1 - \beta_2) * (g_t)^2$$

$$v_{weight(t)} = \beta_2 * v_{weight(1-1)} + (1 - \beta_2) * \left(\frac{\partial E}{\partial W_{ij}^h}\right)^2$$

$$v_{weight(1)11} = \beta_2 * v_{weight(0)} + (1 - \beta_2) * \left(\frac{\partial E}{\partial W_{11}^h}\right)^2$$

$$v_{weight(1)11} = 0.999 * 0 + (1 - 0.999) * (0.00166)^2$$

$$v_{weight(1)11} = 0 + 0.001 * (0.00166)^2$$

$$v_{weight(1)11} = 0.001 * (0.00000275)$$

$$v_{weight(1)11} = 0.0000000275$$

- Perhitungan *ADAM* kedua pada estimasi momen kedua terhadap *weight* yang menerima hasil perhitungan *derivative partial error* jaringan *E* pada *input layer neuron* pertama dan *hidden layer neuron* kedua.

$$\begin{aligned}
v_{weight(t)} &= \beta_2 * v_{weight(t-1)} + (1 - \beta_2) * (g_t)^2 \\
v_{weight(t)} &= \beta_2 * v_{weight(1-1)} + (1 - \beta_2) * \left(\frac{\partial E}{\partial W_{ij}^h}\right)^2 \\
v_{weight(1)12} &= \beta_2 * v_{weight(0)} + (1 - \beta_2) * \left(\frac{\partial E}{\partial W_{12}^h}\right)^2 \\
v_{weight(1)12} &= 0.999 * 0 + (1 - 0.999) * (0.00322)^2 \\
v_{weight(1)12} &= 0 + 0.001 * (0.00322)^2 \\
v_{weight(1)12} &= 0.001 * (0.0000104) \\
v_{weight(1)12} &= 0.0000000104
\end{aligned}$$

- Perhitungan ADAM kedua pada estimasi momen kedua terhadap *bias* yang menerima hasil perhitungan *derivative partial error* jaringan *E* pada *input layer neuron* pertama.

$$\begin{aligned}
v_{bias(t)} &= \beta_2 * v_{bias(t-1)} + (1 - \beta_2) * (g_t)^2 \\
v_{bias(t)} &= \beta_2 * v_{bias(1-1)} + (1 - \beta_2) * \left(\frac{\partial E}{\partial W_{ij}^h}\right)^2 \\
v_{bias(1)1} &= \beta_2 * v_{bias(0)} + (1 - \beta_2) * \left(\frac{\partial E}{\partial W_{bias1}^h}\right)^2 \\
v_{bias(1)1} &= 0.999 * 0 + (1 - 0.999) * (0.00373)^2 \\
v_{bias(1)1} &= 0 + 0.001 * (0.00373)^2 \\
v_{bias(1)1} &= 0.001 * (0.0000139) \\
v_{bias(1)1} &= 0.0000000139
\end{aligned}$$

- Perhitungan ADAM kedua pada estimasi momen kedua terhadap *bias* yang menerima hasil perhitungan *derivative partial error* jaringan *E* pada *input layer neuron* kedua.

$$\begin{aligned}
v_{bias(t)} &= \beta_2 * v_{bias(t-1)} + (1 - \beta_2) * (g_t)^2 \\
v_{bias(t)} &= \beta_2 * v_{bias(1-1)} + (1 - \beta_2) * \left(\frac{\partial E}{\partial W_{ij}^h}\right)^2 \\
v_{bias(1)2} &= \beta_2 * v_{bias(0)} + (1 - \beta_2) * \left(\frac{\partial E}{\partial W_{bias2}^h}\right)^2
\end{aligned}$$

$$\nu_{bias(1)2} = 0.999 * 0 + (1 - 0.999) * (0.00725)^2$$

$$\nu_{bias(1)2} = 0 + 0.001 * (0.00725)^2$$

$$\nu_{bias(1)2} = 0.001 * (0.0000526)$$

$$\nu_{bias(1)2} = 0.0000000526$$

Untuk keseluruhan hasil nilai perhitungan *ADAM* kedua pada estimasi momen kedua terhadap *weight* dan *bias* yang menerima hasil perhitungan perhitungan *derivative partial error* jaringan *E* terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron* dapat dijabarkan pada tabel 3.20:

Tabel 3.20. Nilai perhitungan *ADAM* kedua pada estimasi momen kedua terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron*

<i>ADAM</i> kedua pada estimasi momen kedua terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>	Nilai <i>ADAM</i> kedua pada estimasi momen kedua terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>
<i>Weight Pada Hidden Layer</i>	
$\nu_{weight(1)11}$	0.00000000275
$\nu_{weight(1)12}$	0.0000000104
$\nu_{weight(1)13}$	0.0000000212
$\nu_{weight(1)14}$	0.0000000331
$\nu_{weight(1)15}$	0.0000000439
$\nu_{weight(1)16}$	0.0000000520
$\nu_{weight(1)21}$	0
$\nu_{weight(1)22}$	0
$\nu_{weight(1)23}$	0
$\nu_{weight(1)24}$	0
$\nu_{weight(1)25}$	0
$\nu_{weight(1)26}$	0
$\nu_{weight(1)31}$	0
$\nu_{weight(1)32}$	0
$\nu_{weight(1)33}$	0
$\nu_{weight(1)34}$	0
$\nu_{weight(1)35}$	0
$\nu_{weight(1)36}$	0
$\nu_{weight(1)41}$	0
$\nu_{weight(1)42}$	0

Tabel 3.20. Nilai perhitungan *ADAM* kedua pada estimasi momen kedua terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron* (Lanjutan).

<i>ADAM</i> kedua pada estimasi momen kedua terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>	Nilai <i>ADAM</i> kedua pada estimasi momen kedua terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>
$v_{weight(1)43}$	0
$v_{weight(1)44}$	0
$v_{weight(1)45}$	0
$v_{weight(1)46}$	0
$v_{weight(1)51}$	0.000000000172
$v_{weight(1)52}$	0.000000000650
$v_{weight(1)53}$	0.000000000133
$v_{weight(1)54}$	0.000000000207
$v_{weight(1)55}$	0.000000000274
$v_{weight(1)56}$	0.000000000325
$v_{weight(1)61}$	0
$v_{weight(1)62}$	0
$v_{weight(1)63}$	0
$v_{weight(1)64}$	0
$v_{weight(1)65}$	0
$v_{weight(1)66}$	0
$v_{weight(1)71}$	0.000000000689
$v_{weight(1)72}$	0.000000000260
$v_{weight(1)73}$	0.000000000532
$v_{weight(1)74}$	0.000000000829
$v_{weight(1)75}$	0.00000000109
$v_{weight(1)76}$	0.00000000130
$v_{weight(1)81}$	0
$v_{weight(1)82}$	0
$v_{weight(1)83}$	0
$v_{weight(1)84}$	0
$v_{weight(1)85}$	0
$v_{weight(1)86}$	0
$v_{weight(1)91}$	0
$v_{weight(1)92}$	0
$v_{weight(1)93}$	0
$v_{weight(1)94}$	0

Tabel 3.20. Nilai perhitungan *ADAM* kedua pada estimasi momen kedua terhadap *weight* dan *bias* pada *hidden layer* di setiap *neuron* (Lanjutan).

<i>ADAM</i> kedua pada estimasi momen kedua terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>	Nilai <i>ADAM</i> kedua pada estimasi momen kedua terhadap <i>weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>
$v_{\text{weight}(1)95}$	0
$v_{\text{weight}(1)96}$	0
<i>Bias Pada Hidden Layer</i>	
$v_{\text{bias}(1)1}$	0.0000000139
$v_{\text{bias}(1)2}$	0.0000000526
$v_{\text{bias}(1)3}$	0.000000107
$v_{\text{bias}(1)4}$	0.000000167
$v_{\text{bias}(1)5}$	0.000000222
$v_{\text{bias}(1)6}$	0.000000263

13. Proses perhitungan koreksi *weight* dan *bias* yang menerima hasil pada tabel 3.20 nilai perhitungan *ADAM* kedua pada estimasi moment kedua terhadap *weight* dan *bias* dari perhitungan *derivative partial error* jaringan *E* pada *hidden layer* di setiap *neuron* dengan menggunakan persamaan 2.17 dan persamaan 2.18:

- Perhitungan koreksi *weight* yang menerima hasil perhitungan *ADAM* kedua pada estimasi moment kedua di *input layer neuron* pertama dan *hidden layer neuron* pertama.

$$\tilde{V}_{\text{weight}(t)} = \frac{v_{\text{weight}(t)}}{1 - \beta_2^t}$$

$$\tilde{V}_{\text{weight}(t)11} = \frac{v_{\text{weight}(t)11}}{1 - \beta_2^t}$$

$$\tilde{V}_{\text{weight}(t)11} = \frac{0.00000000275}{1 - 0.999}$$

$$\tilde{V}_{\text{weight}(t)11} = \frac{0.00000000275}{0.001}$$

$$\tilde{V}_{\text{weight}(t)11} = 0.00000275$$

- Perhitungan koreksi *weight* yang menerima hasil perhitungan *ADAM* kedua pada estimasi moment kedua di *input layer neuron* pertama dan *hidden layer neuron* kedua.

$$\begin{aligned}\tilde{V}_{weight(t)} &= \frac{v_{weight(t)}}{1 - \beta_2^t} \\ \tilde{V}_{weight(t)12} &= \frac{v_{weight(t)12}}{1 - \beta_2^t} \\ \tilde{V}_{weight(t)12} &= \frac{0.0000000104}{1 - 0.999} \\ \tilde{V}_{weight(t)12} &= \frac{0.0000000104}{0.001} \\ \tilde{V}_{weight(t)12} &= 0.0000104\end{aligned}$$

- Perhitungan koreksi *bias* yang menerima hasil perhitungan *ADAM* kedua pada estimasi moment kedua di *input layer neuron* pertama.

$$\begin{aligned}\tilde{V}_{bias(t)} &= \frac{v_{bias(t)}}{1 - \beta_2^t} \\ \tilde{V}_{bias(t)1} &= \frac{v_{bias(t)1}}{1 - \beta_2^t} \\ \tilde{V}_{bias(t)1} &= \frac{0.0000000139}{1 - 0.999} \\ \tilde{V}_{bias(t)1} &= \frac{0.0000000139}{0.001} \\ \tilde{V}_{bias(t)1} &= 0.0000139\end{aligned}$$

- Perhitungan koreksi *bias* yang menerima hasil perhitungan *ADAM* kedua pada estimasi moment kedua di *input layer neuron* kedua.

$$\begin{aligned}\tilde{V}_{bias(t)} &= \frac{v_{bias(t)}}{1 - \beta_2^t} \\ \tilde{V}_{bias(t)2} &= \frac{v_{bias(t)2}}{1 - \beta_2^t} \\ \tilde{V}_{bias(t)2} &= \frac{0.0000000526}{1 - 0.999} \\ \tilde{V}_{bias(t)2} &= \frac{0.0000000526}{0.001} \\ \tilde{V}_{bias(t)2} &= 0.0000526\end{aligned}$$

Untuk keseluruhan hasil nilai perhitungan koreksi *weight* dan *bias* yang menerima hasil perhitungan *ADAM* kedua pada estimasi moment kedua terhadap

weight dan *bias* dari perhitungan *derivative partial error* jaringan *E* pada *hidden layer* di setiap *neuron* dapat dijabarkan pada tabel 3.21:

Tabel 3.21. Nilai perhitungan koreksi *weight* dan *bias ADAM* kedua pada estimasi moment kedua pada *hidden layer* di setiap *neuron*

Koreksi <i>weight</i> dan <i>bias ADAM</i> kedua pada estimasi moment kedua pada <i>hidden layer</i> di setiap <i>neuron</i>	Nilai koreksi <i>weight</i> dan <i>bias ADAM</i> kedua pada estimasi moment kedua pada <i>hidden layer</i> di setiap <i>neuron</i>
Weight Pada Hidden Layer	
$\tilde{V}_{weight(t)11}$	0.00000275
$\tilde{V}_{weight(t)12}$	0.0000104
$\tilde{V}_{weight(t)13}$	0.0000212
$\tilde{V}_{weight(t)14}$	0.0000331
$\tilde{V}_{weight(t)15}$	0.0000439
$\tilde{V}_{weight(t)16}$	0.0000520
$\tilde{V}_{weight(t)21}$	0
$\tilde{V}_{weight(t)22}$	0
$\tilde{V}_{weight(t)23}$	0
$\tilde{V}_{weight(t)24}$	0
$\tilde{V}_{weight(t)25}$	0
$\tilde{V}_{weight(t)26}$	0
$\tilde{V}_{weight(t)31}$	0
$\tilde{V}_{weight(t)32}$	0
$\tilde{V}_{weight(t)33}$	0
$\tilde{V}_{weight(t)34}$	0
$\tilde{V}_{weight(t)35}$	0
$\tilde{V}_{weight(t)36}$	0
$\tilde{V}_{weight(t)41}$	0
$\tilde{V}_{weight(t)42}$	0
$\tilde{V}_{weight(t)43}$	0
$\tilde{V}_{weight(t)44}$	0
$\tilde{V}_{weight(t)45}$	0
$\tilde{V}_{weight(t)46}$	0
$\tilde{V}_{weight(t)51}$	0.000000172
$\tilde{V}_{weight(t)52}$	0.000000650
$\tilde{V}_{weight(t)53}$	0.00000133
$\tilde{V}_{weight(t)54}$	0.00000207

Tabel 3.21. Nilai perhitungan koreksi *weight* dan *bias ADAM* kedua pada estimasi moment kedua pada *hidden layer* di setiap *neuron* (Lanjutan).

Koreksi <i>weight</i> dan <i>bias ADAM</i> kedua pada estimasi moment kedua pada <i>hidden layer</i> di setiap <i>neuron</i>	Nilai koreksi <i>weight</i> dan <i>bias ADAM</i> kedua pada estimasi moment kedua pada <i>hidden layer</i> di setiap <i>neuron</i>
$\tilde{V}_{\text{weight}(t)55}$	0.00000274
$\tilde{V}_{\text{weight}(t)56}$	0.00000325
$\tilde{V}_{\text{weight}(t)61}$	0
$\tilde{V}_{\text{weight}(t)62}$	0
$\tilde{V}_{\text{weight}(t)63}$	0
$\tilde{V}_{\text{weight}(t)64}$	0
$\tilde{V}_{\text{weight}(t)65}$	0
$\tilde{V}_{\text{weight}(t)66}$	0
$\tilde{V}_{\text{weight}(t)71}$	0.000000689
$\tilde{V}_{\text{weight}(t)72}$	0.00000260
$\tilde{V}_{\text{weight}(t)73}$	0.00000532
$\tilde{V}_{\text{weight}(t)74}$	0.00000829
$\tilde{V}_{\text{weight}(t)75}$	0.0000109
$\tilde{V}_{\text{weight}(t)76}$	0.0000130
$\tilde{V}_{\text{weight}(t)81}$	0
$\tilde{V}_{\text{weight}(t)82}$	0
$\tilde{V}_{\text{weight}(t)83}$	0
$\tilde{V}_{\text{weight}(t)84}$	0
$\tilde{V}_{\text{weight}(t)85}$	0
$\tilde{V}_{\text{weight}(t)86}$	0
$\tilde{V}_{\text{weight}(t)91}$	0
$\tilde{V}_{\text{weight}(t)92}$	0
$\tilde{V}_{\text{weight}(t)93}$	0
$\tilde{V}_{\text{weight}(t)94}$	0
$\tilde{V}_{\text{weight}(t)95}$	0
$\tilde{V}_{\text{weight}(t)96}$	0
Bias Pada Hidden Layer	
$\tilde{V}_{\text{bias}(t)1}$	0.0000139
$\tilde{V}_{\text{bias}(t)2}$	0.0000526
$\tilde{V}_{\text{bias}(t)3}$	0.000107
$\tilde{V}_{\text{bias}(t)4}$	0.000167

Tabel 3.21. Nilai perhitungan koreksi *weight* dan *bias ADAM* kedua pada estimasi moment kedua pada *hidden layer* di setiap *neuron* (Lanjutan).

Koreksi <i>weight</i> dan <i>bias ADAM</i> kedua pada estimasi moment kedua pada <i>hidden layer</i> di setiap <i>neuron</i>	Nilai koreksi <i>weight</i> dan <i>bias ADAM</i> kedua pada estimasi moment kedua pada <i>hidden layer</i> di setiap <i>neuron</i>
Bias Pada Hidden Layer	
$\tilde{V}_{bias(t)5}$	0.000222
$\tilde{V}_{bias(t)6}$	0.000263

14. Pada tahapan proses perhitungan perbaruan parameter *weight* dan *bias* yang menerima proses hasil perhitungan di tabel 3.14 nilai koreksi *weight* dan *bias* pada *ADAM* pertama pada estimasi moment pertama dan *ADAM* pertama pada estimasi moment kedua di tabel 3.16 pada *output layer* yang dijumlahkan dengan *weight* awal dan *bias* awal pada *input layer* dengan menggunakan persamaan 2.19 dan persamaan 2.20 sebagai berikut :

- Proses perhitungan perbaruan parameter *weight* yang menerima hasil proses perhitungan dari koreksi *weight* pada *output layer* di *weight neuron* pertama.

$$\begin{aligned}
 w_{weight(t)} &= w_{weight(t-1)} - \alpha * \frac{\dot{m}_{weight(t)}}{\sqrt{\tilde{V}_{weight(t)} + \varepsilon}} \\
 w_{(weight(t))1}^{output (Update)} &= w_{weight(t-1)1} - \alpha * \frac{\dot{m}_{weight(1)1}}{\sqrt{\tilde{V}_{weight(1)1} + \varepsilon}} \\
 w_{(weight(t))1}^{output (Update)} &= 0.111 - 0.001 * \frac{0.0745}{\sqrt{0.00555025 + 0.00000001}} \\
 w_{(weight(t))1}^{output (Update)} &= 0.111 - 0.001 * \frac{0.0745}{\sqrt{0.00555026}} \\
 w_{(weight(t))1}^{output (Update)} &= 0.111 - 0.001 * \frac{0.0745}{0.07450006} \\
 w_{(weight(t))1}^{output (Update)} &= 0.111 - 0.001 * 0.99999919 \\
 w_{(weight(t))1}^{output (Update)} &= 0.111 - 0.00099999919 \\
 w_{(weight(t))1}^{output (Update)} &= 0.11000000081
 \end{aligned}$$

- Proses perhitungan perbaruan parameter *weight* yang menerima hasil proses perhitungan dari koreksi *weight* pada *output layer* di *weight neuron* kedua.

$$\begin{aligned}
 w_{weight(t)} &= w_{weight(t-1)} - \alpha * \frac{\bar{m}_{weight(t)}}{\sqrt{\tilde{V}_{weight(t)} + \varepsilon}} \\
 w_{(weight(t))2}^{output(Update)} &= w_{weight(t-1)2} - \alpha * \frac{\bar{m}_{weight(t)2}}{\sqrt{\tilde{V}_{weight(t)2} + \varepsilon}} \\
 w_{(weight(t))2}^{output(Update)} &= 0.222 - 0.001 * \frac{0.0811}{\sqrt{0.00657721 + 0.00000001}} \\
 w_{(weight(t))2}^{output(Update)} &= 0.222 - 0.001 * \frac{0.0811}{\sqrt{0.00657722}} \\
 w_{(weight(t))2}^{output(Update)} &= 0.222 - 0.001 * \frac{0.0811}{0.08110006} \\
 w_{(weight(t))2}^{output(Update)} &= 0.222 - 0.001 * 0.99999926 \\
 w_{(weight(t))2}^{output(Update)} &= 0.222 - 0.00099999926 \\
 w_{(weight(t))2}^{output(Update)} &= 0.22100000074
 \end{aligned}$$

- Proses perhitungan perbaruan parameter *bias* yang menerima hasil proses perhitungan dari koreksi *bias* pada *output layer* di *bias neuron*.

$$\begin{aligned}
 w_{bias(t)} &= w_{bias(t)} - \alpha * \frac{\bar{m}_{bias(t)}}{\sqrt{\tilde{V}_{bias(t)} + \varepsilon}} \\
 w_{bias(t)}^{output(Update)} &= w_{bias(t)1} - \alpha * \frac{\bar{m}_{bias(1)}}{\sqrt{\tilde{V}_{bias(1)1} + \varepsilon}} \\
 w_{bias(t)}^{output(Update)} &= 0.777 - 0.001 * \frac{0.135}{\sqrt{0.018225 + 0.00000001}} \\
 w_{bias(t)}^{output(Update)} &= 0.777 - 0.001 * \frac{0.135}{\sqrt{0.01822501}} \\
 w_{bias(t)}^{output(Update)} &= 0.777 - 0.001 * \frac{0.135}{0.13500004} \\
 w_{bias(t)}^{output(Update)} &= 0.777 - 0.001 * 0.99999970 \\
 w_{bias(t)}^{output(Update)} &= 0.777 - 0.0009999997 \\
 w_{bias(t)}^{output(Update)} &= 0.7760000003
 \end{aligned}$$

Untuk keseluruhan hasil nilai perhitungan perbaruan parameter *weight* dan *bias* menerima proses hasil perhitungan dari koreksi *weight* dan *bias* pada *output layer* yang dijumlahkan dengan *weight* awal dan *bias* awal di setiap *neuron* dapat dijabarkan pada tabel 3.22:

Tabel 3.22. Nilai perhitungan *update weight* dan *bias* pada *output layer* di setiap *neuron*

<i>Update weight</i> dan <i>bias</i> pada <i>output layer</i> di setiap <i>neuron</i>	Nilai <i>update weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>
$w_{(weight(t))1}^{output (Update)}$	0.11000000081
$w_{(weight(t))2}^{output (Update)}$	0.22100000074
$w_{(weight(t))3}^{output (Update)}$	0.33200000058
$w_{(weight(t))4}^{output (Update)}$	0.44300000054
$w_{(weight(t))5}^{output (Update)}$	0.55400000051
$w_{(weight(t))6}^{output (Update)}$	0.66500000049
$w_{bias(t)}^{output (Update)}$	0.7760000003

15. Pada tahapan proses perhitungan perbaruan parameter *weight* dan *bias* menerima proses hasil perhitungan pada *ADAM* kedua pada estimasi moment pertama di tabel 3.19 dan *ADAM* kedua pada estimasi momen kedua di tabel 3.21 dari koreksi *weight* dan *bias* untuk dijumlahkan dengan *weight* awal dan *bias* awal pada *input layer* yang ada jaringan *feedforward* dilapisan *hidden layer* dengan menggunakan persamaan 2.21 dan persamaan 2.22 sebagai berikut :
- Proses perhitungan perbaruan parameter *weight* yang menerima hasil proses perhitungan dari koreksi *weight* pada *input layer neuron* pertama dan *hidden layer neuron* pertama.

$$\begin{aligned}
 w_{weight(t)} &= w_{weight(t-1)} - \alpha * \frac{\hat{m}_{weight(t) ij}}{\sqrt{\hat{V}_{weight(t) ij} + \varepsilon}} \\
 w_{(weight(t))ij}^{Update} &= w_{weight(t-1)11} - \alpha * \frac{\hat{m}_{weight(1)11}}{\sqrt{\hat{V}_{weight(t)11} + \varepsilon}} \\
 w_{(weight(t))11}^{Update} &= 0.111 - 0.001 * \frac{0.00166}{\sqrt{0.00000275 + 0.00000001}} \\
 w_{(weight(t))11}^{Update} &= 0.111 - 0.001 * \frac{0.00166}{\sqrt{0.00000276}}
 \end{aligned}$$

$$\begin{aligned}
 w_{(weight(t))11}^{Update} &= 0.111 - 0.001 * \frac{0.00166}{0.00166132} \\
 w_{(weight(t))11}^{Update} &= 0.111 - 0.001 * 0.99920545 \\
 w_{(weight(t))11}^{Update} &= 0.111 - 0.00099920545 \\
 w_{(weight(t))11}^{Update} &= 0.11000079455
 \end{aligned}$$

- Proses perhitungan perbaruan parameter *weight* yang menerima hasil proses perhitungan dari koreksi *weight* pada *input layer neuron* pertama dan *hidden layer neuron* kedua.

$$\begin{aligned}
 w_{weight(t)} &= w_{weight(t-1)} - \alpha * \frac{\hat{m}_{weight(t)ij}}{\sqrt{\tilde{V}_{weight(t)ij} + \varepsilon}} \\
 w_{(weight(t))ij}^{Update} &= w_{weight(t-1)12} - \alpha * \frac{\hat{m}_{weight(1)12}}{\sqrt{\tilde{V}_{weight(t)12} + \varepsilon}} \\
 w_{(weight(t))12}^{Update} &= 0.222 - 0.001 * \frac{0.00322}{\sqrt{0.0000104 + 0.00000001}} \\
 w_{(weight(t))12}^{Update} &= 0.222 - 0.001 * \frac{0.00322}{\sqrt{0.00001041}} \\
 w_{(weight(t))12}^{Update} &= 0.222 - 0.001 * \frac{0.00166}{0.00322645} \\
 w_{(weight(t))12}^{Update} &= 0.222 - 0.001 * 0.51449735 \\
 w_{(weight(t))12}^{Update} &= 0.222 - 0.00051449735 \\
 w_{(weight(t))12}^{Update} &= 0.22148550265
 \end{aligned}$$

- Proses perhitungan perbaruan parameter *bias* yang menerima hasil proses perhitungan dari koreksi *bias* pada *hidden layer neuron* pertama.

$$\begin{aligned}
 w_{bias(t)} &= w_{bias(t-1)} - \alpha * \frac{\hat{m}_{bias(t)ij}}{\sqrt{\tilde{V}_{bias(t)ij} + \varepsilon}} \\
 w_{(bias(t))ij}^{Update} &= w_{bias(t-1)1} - \alpha * \frac{\hat{m}_{bias(1)1}}{\sqrt{\tilde{V}_{bias(t)1} + \varepsilon}} \\
 w_{(bias(t))1}^{Update} &= 0.111 - 0.001 * \frac{0.00373}{\sqrt{0.0000139 + 0.00000001}} \\
 w_{(bias(t))1}^{Update} &= 0.111 - 0.001 * \frac{0.00373}{\sqrt{0.00001391}}
 \end{aligned}$$

$$\begin{aligned}
 w_{(bias(t))1}^{Update} &= 0.111 - 0.001 * \frac{0.00373}{0.00372961} \\
 w_{(bias(t))1}^{Update} &= 0.111 - 0.001 * 1.00010457 \\
 w_{(bias(t))1}^{Update} &= 0.111 - 0.00100010457 \\
 w_{(bias(t))1}^{Update} &= 0.10999989543
 \end{aligned}$$

- Proses perhitungan perbaruan parameter *bias* yang menerima hasil proses perhitungan dari koreksi *bias* pada *hidden layer neuron* kedua.

$$\begin{aligned}
 w_{bias(t)} &= w_{bias(t-1)} - \alpha * \frac{\tilde{m}_{bias(t)ij}}{\sqrt{\tilde{V}_{bias(t)ij} + \varepsilon}} \\
 w_{(bias(t))1j}^{Update} &= w_{bias(t-1)2} - \alpha * \frac{\tilde{m}_{bias(1)2}}{\sqrt{\tilde{V}_{bias(t)2} + \varepsilon}} \\
 w_{(bias(t))2}^{Update} &= 0.222 - 0.001 * \frac{0.00725}{\sqrt{0.0000526 + 0.00000001}} \\
 w_{(bias(t))2}^{Update} &= 0.222 - 0.001 * \frac{0.00725}{\sqrt{0.00005261}} \\
 w_{(bias(t))2}^{Update} &= 0.222 - 0.001 * \frac{0.00725}{0.00725327} \\
 w_{(bias(t))2}^{Update} &= 0.222 - 0.001 * 0.99954917 \\
 w_{(bias(t))2}^{Update} &= 0.222 - 0.00099954917 \\
 w_{(bias(t))2}^{Update} &= 0.22100045083
 \end{aligned}$$

Untuk keseluruhan hasil nilai perhitungan perbaruan parameter *weight* dan *bias* menerima proses hasil perhitungan dari koreksi *weight* dan *bias* pada *hidden layer* yang dijumlahkan dengan *weight* awal dan *bias* awal di setiap *neuron* dapat dijabarkan pada tabel 3.24:

Tabel 3.23. Nilai perhitungan *update weight* dan *bias* pada *hidden layer* di setiap *neuron*

<i>Update weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>	Nilai <i>update weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>
<i>Weight Pada Hidden Layer</i>	
$W_{(weight(t))11}^{hidden(update)}$	0.11000079455
$W_{(weight(t))12}^{hidden(update)}$	0.22148550265

Tabel 3.23. Nilai perhitungan *update weight* dan *bias* pada *hidden layer* di setiap *neuron* (Lanjutan).

<i>Update weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>	Nilai <i>update weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>
$W_{(weight(t))13}^{hidden (update)}$	0.3319990077
$W_{(weight(t))14}^{hidden (update)}$	0.44299897813
$W_{(weight(t))15}^{hidden (update)}$	0.55399946578
$W_{(weight(t))16}^{hidden (update)}$	0.66500024821
$W_{(weight(t))21}^{hidden (update)}$	0.111
$W_{(weight(t))22}^{hidden (update)}$	0.222
$W_{(weight(t))23}^{hidden (update)}$	0.333
$W_{(weight(t))24}^{hidden (update)}$	0.444
$W_{(weight(t))25}^{hidden (update)}$	0.555
$W_{(weight(t))26}^{hidden (update)}$	0.666
$W_{(weight(t))31}^{hidden (update)}$	0.111
$W_{(weight(t))32}^{hidden (update)}$	0.222
$W_{(weight(t))33}^{hidden (update)}$	0.333
$W_{(weight(t))34}^{hidden (update)}$	0.444
$W_{(weight(t))35}^{hidden (update)}$	0.555
$W_{(weight(t))36}^{hidden (update)}$	0.666
$W_{(weight(t))41}^{hidden (update)}$	0.111
$W_{(weight(t))42}^{hidden (update)}$	0.222
$W_{(weight(t))43}^{hidden (update)}$	0.333
$W_{(weight(t))44}^{hidden (update)}$	0.444
$W_{(weight(t))45}^{hidden (update)}$	0.555
$W_{(weight(t))46}^{hidden (update)}$	0.666
$W_{(weight(t))51}^{hidden (update)}$	0.10127214552
$W_{(weight(t))52}^{hidden (update)}$	0.22100788522
$W_{(weight(t))53}^{hidden (update)}$	0.33200654815
$W_{(weight(t))54}^{hidden (update)}$	0.44300084592
$W_{(weight(t))55}^{hidden (update)}$	0.55400501113

Tabel 3.23. Nilai perhitungan *update weight* dan *bias* pada *hidden layer* di setiap *neuron* (Lanjutan).

<i>Update weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>	Nilai <i>update weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>
$W_{(weight(t))56}^{hidden (update)}$	0.66500306834
$W_{(weight(t))61}^{hidden (update)}$	0.111
$W_{(weight(t))62}^{hidden (update)}$	0.222
$W_{(weight(t))63}^{hidden (update)}$	0.333
$W_{(weight(t))64}^{hidden (update)}$	0.444
$W_{(weight(t))65}^{hidden (update)}$	0.555
$W_{(weight(t))66}^{hidden (update)}$	0.666
$W_{(weight(t))71}^{hidden (update)}$	0.11000725078
$W_{(weight(t))72}^{hidden (update)}$	0.22100343536
$W_{(weight(t))73}^{hidden (update)}$	0.33200375541
$W_{(weight(t))74}^{hidden (update)}$	0.4430003367
$W_{(weight(t))75}^{hidden (update)}$	0.55399788982
$W_{(weight(t))76}^{hidden (update)}$	0.66500192131
$W_{(weight(t))81}^{hidden (update)}$	0.111
$W_{(weight(t))82}^{hidden (update)}$	0.222
$W_{(weight(t))83}^{hidden (update)}$	0.333
$W_{(weight(t))84}^{hidden (update)}$	0.444
$W_{(weight(t))85}^{hidden (update)}$	0.555
$W_{(weight(t))86}^{hidden (update)}$	0.666
$W_{(weight(t))91}^{hidden (update)}$	0.111
$W_{(weight(t))92}^{hidden (update)}$	0.222
$W_{(weight(t))93}^{hidden (update)}$	0.333
$W_{(weight(t))94}^{hidden (update)}$	0.444
$W_{(weight(t))95}^{hidden (update)}$	0.555
$W_{(weight(t))96}^{hidden (update)}$	0.666

Tabel 3.23. Nilai perhitungan *update weight* dan *bias* pada *hidden layer* di setiap *neuron* (Lanjutan).

<i>Update weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>	Nilai <i>update weight</i> dan <i>bias</i> pada <i>hidden layer</i> di setiap <i>neuron</i>
<i>Bias Pada Hidden Layer</i>	
$w_{bias(1)1}^{hidden (update)}$	0.10999989543
$w_{bias(1)2}^{hidden (update)}$	0.22100045083
$w_{bias(1)3}^{hidden (update)}$	0.33200430758
$w_{bias(1)4}^{hidden (update)}$	0.44300179754
$w_{bias(1)5}^{hidden (update)}$	0.554
$w_{bias(1)6}^{hidden (update)}$	0.66500108401

BAB 4

HASIL DAN PEMBAHASAN

Pada bab ini dijelaskan hasil penelitian dari penerapan metode *ADAM* sebagai peningkatan proses pembelajaran metode *gradient descent* pada *backpropagation* untuk meminimalkan nilai kuadrat *error* di dalam setiap *epoch* pada pengujian *neural network*. Didalam pengujian dilakukan pengujian yaitu, pengujian pertama *maximum epoch* 1000 dan pengujian kedua *maximum epoch* 2000. Kuadrat *error* yang digunakan adalam *mean square error (MSE)*. Pada *ADAM* menggunakan *target error* yaitu 0.001, *learning rate* yaitu 0.001, nilai ϵ yaitu 10^{-8} (0.00000001), nilai *exponential* untuk estimasi momen pertama (β_1) yaitu 0.9, nilai *exponential* untuk estimasi momen kedua (β_2) yaitu 0.999. Analisis *ADAM* untuk pengoptimasian metode *gradient descent backpropagation* pada setiap *epoch* yang dihasilkan digunakan untuk proses pembelajaran tersebut dimulai, sampai proses pembelajaran tersebut selesai.

4.1. Hasil Penelitian

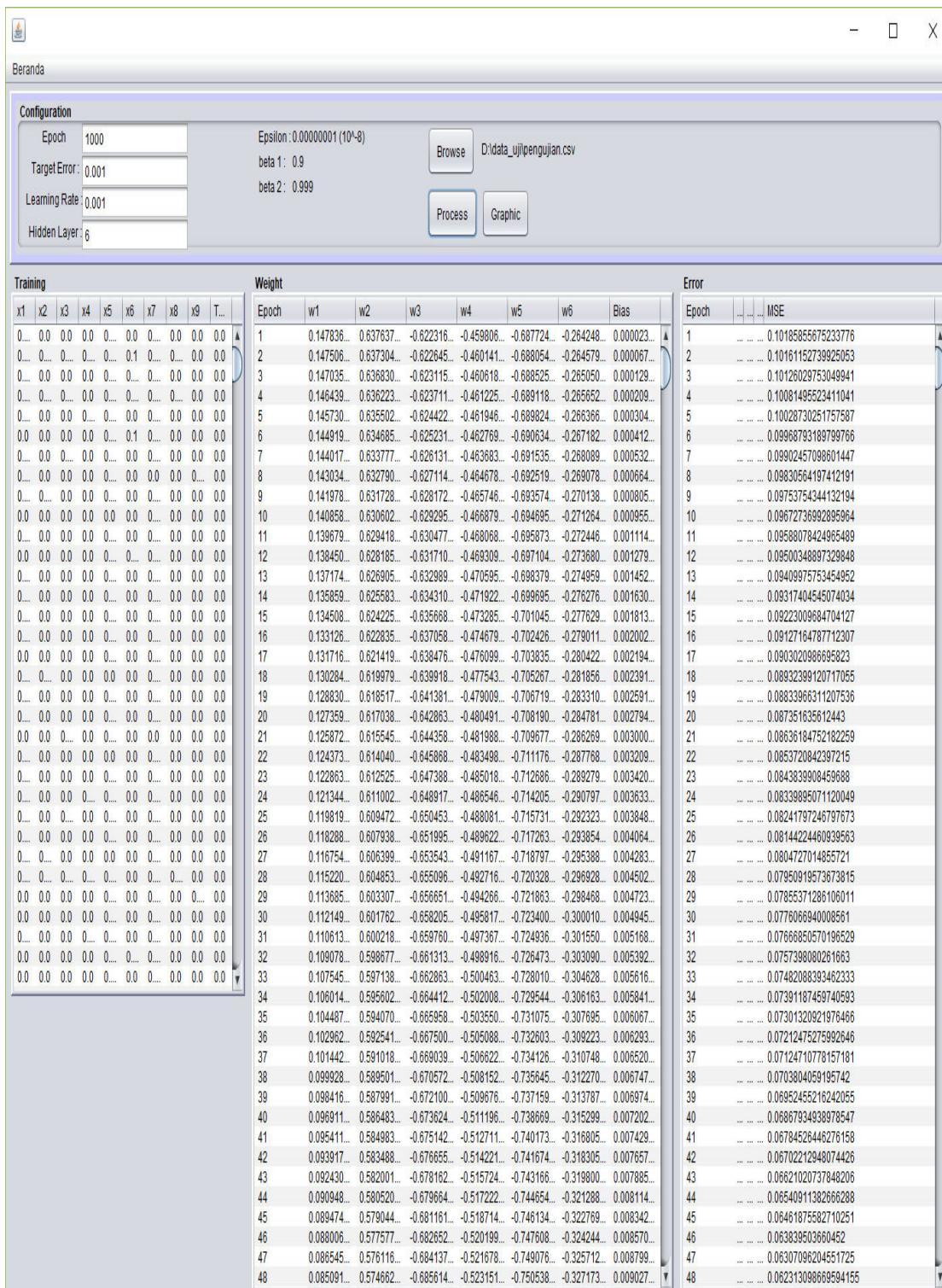
Dari pengujian pertama sampai dengan pengujian kelima pada penjelasan diatas dalam melakukan proses pembelajaran *ADAM* pada arsitektur algoritma *backpropagation*, simulasi dilakukan dengan menggunakan komputer spesifikasi berikut:

- Processor Amd A10 - 5745M APU 2.10 GHz.
- RAM : 4 GB.
- Sistem Operasi Windows 10 Pro 64Bit
- Aplikasi yang digunakan pada penelitian ini adalah NetBens IDE 8.0 64-bit.

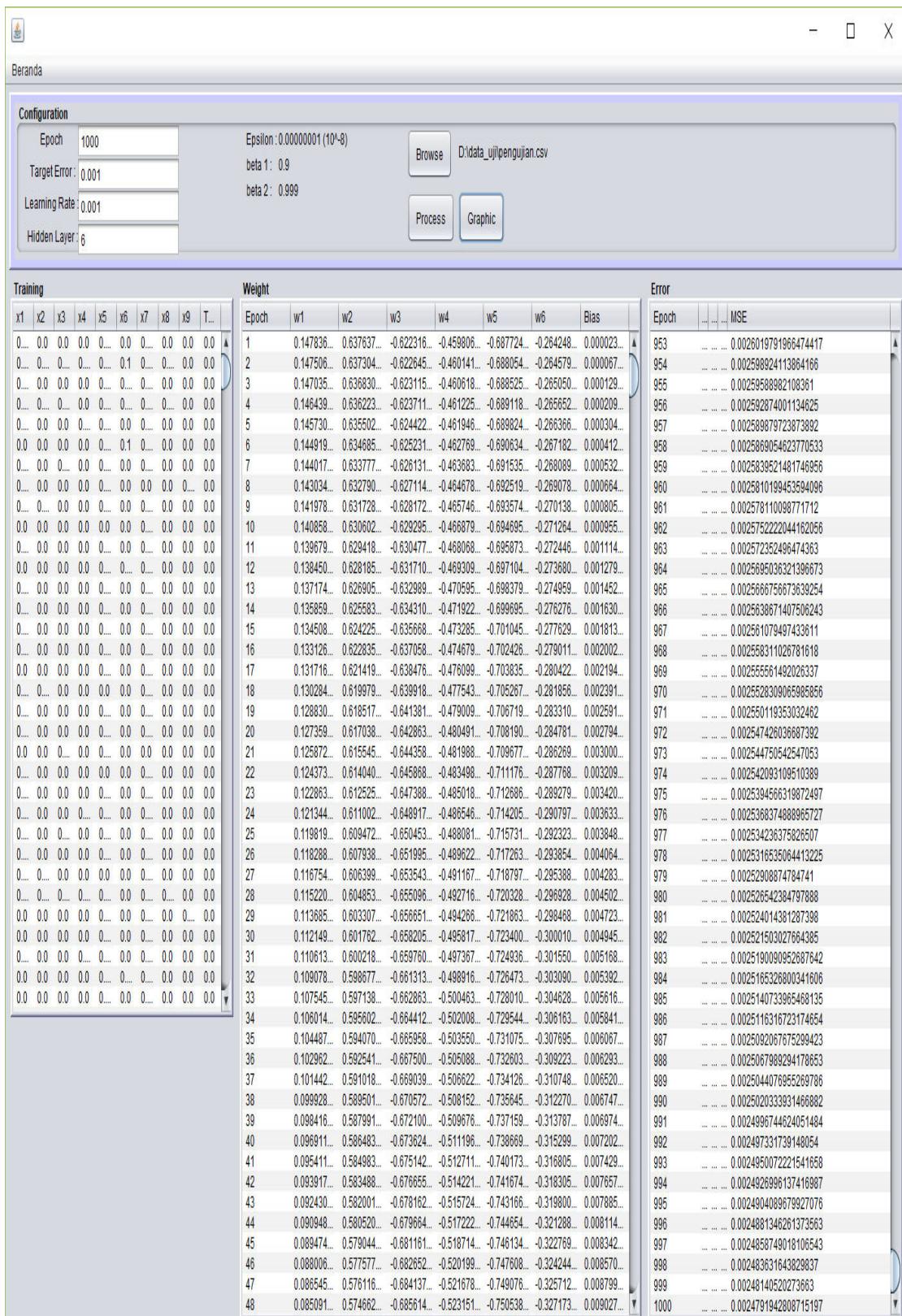
4.1.1 Pengujian pertama maximum epoch 1000

Dari pengujian pertama pada nilai kuadrat *error* di lapisan *output*, proses perhitungan dimulai dari *epoch* 1 sampai dengan *maximum epoch* 1000 dalam melakukan proses pembelajaran *ADAM* untuk meningkatkan proses pembelajaran metode *gradient*

descent pada *backpropagation*. Hasil proses nilai *MSE* pada lapisan *output* dapat dilihat pada penelitian ini seperti gambar 4.1:

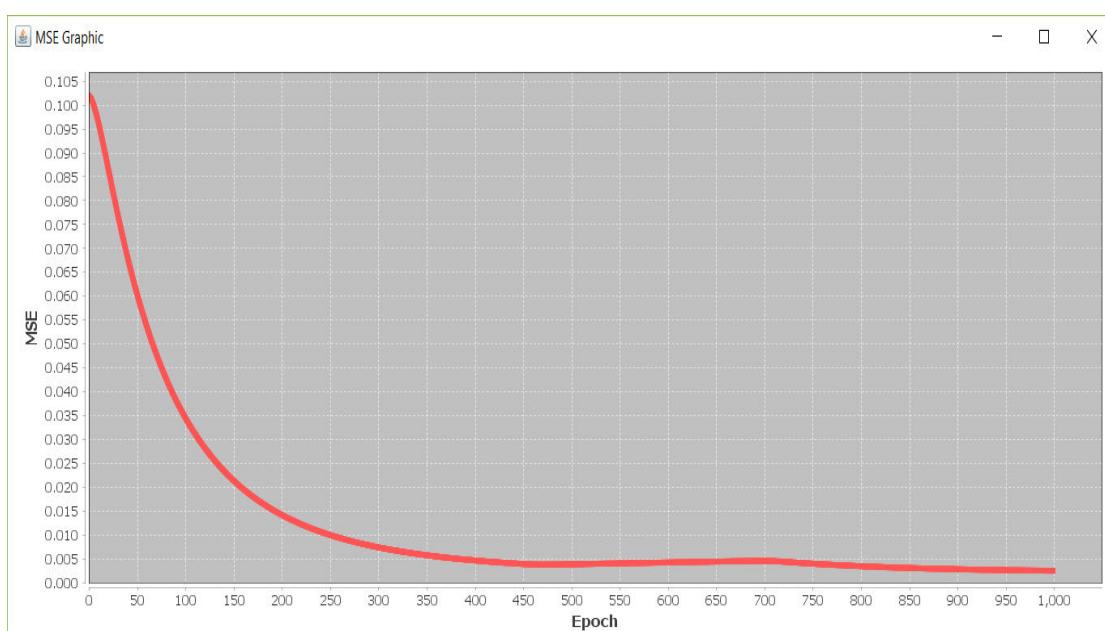


Gambar 4.1. Nilai MSE dalam lapisan *output* pada pengujian pertama.



Gambar 4.1. Nilai MSE dalam lapisan *output* pada pengujian pertama (Lanjutan).

Gambar 4.1. merupakan hasil nilai *MSE* dalam lapisan *output* pada pengujian pertama yang dilakukan perhitungan keseluruhan data dataset *breast cancer*. Pengujian pertama penurunan hasil dari nilai *MSE* untuk perhitungan keseluruhan pada *epoch* dalam penerapan metode *ADAM* untuk meningkatkan proses pembelajaran metode *gradient descent* pada *backpropagation*. Untuk gambar 4.1. nilai *MSE* dalam lapisan *output* pada pengujian pertama dapat dilihat selengkapnya pada lampiran 4. Penurunan hasil dari nilai *MSE* dapat dilihat grafik pada gambar 4.2 :



Gambar 4.2. Grafik pengujian pertama nilai *MSE*

Gambar grafik 4.2 diatas menunjukkan penurunan hasil dari nilai *MSE* pada pengujian pertama untuk untuk *maximum epoch* 1000. Hasil dari *ADAM* dijelaskan pada *output* dari setiap *epoch* untuk maksimum *epoch* 1000. Nilai *output* untuk nilai *mean square error* (*MSE*) menunjukkan bahwa implementasi *ADAM* telah dilakukan dengan benar dan setiap jaringan saraf di *ADAM* dapat beradaptasi dengan set data pelatihan yang digunakan. Untuk *epoch* 1000, nilai *mean square error* pada setiap *epoch* di *output* menurun dari 0.10185855675233776 menjadi 0.0024791942808715197, yang berarti untuk peminimalisasian / penurunan kuadrat *error* dalam *output* yang dihasilkan oleh *ADAM*.

4.1.2 Pengujian kedua maximum epoch 2000

Dari pengujian kedua pada nilai *MSE* di lapisan *output* yang dilakukan dengan *maximum epoch* 2000. Dalam melakukan proses pembelajaran *ADAM* untuk meningkatkan proses pembelajaran metode *gradient descent* pada *backpropagation*. Hasil proses nilai *MSE* pada lapisan *output* dapat dilihat pada penelitian ini seperti gambar 4.3:

The screenshot shows a software window titled "Beranda". The "Configuration" tab is active, displaying the following parameters:

- Epoch: 2000
- Target Error: 0.001
- Learning Rate: 0.001
- Hidden Layer: 6
- Epsilon: 0.0000001 (10^{-8})
- beta 1: 0.9
- beta 2: 0.999
- Browse button pointing to D:\data_uji\pengujian.csv

The "Process" and "Graphic" buttons are also visible.

The "Training" section shows a table with columns x1, x2, x3, x4, x5, x6, x7, x8, x9, T... containing binary values (0 or 1).

The "Weight" section shows a large table with columns Epoch, w1, w2, w3, w4, w5, w6, Bias. The first few rows of weights are:

Epoch	w1	w2	w3	w4	w5	w6	Bias
1	0.14783...	0.63763...	-0.62231...	-0.45980...	-0.68772...	-0.28424...	0.00023...
2	0.147506...	0.637304...	-0.622645...	-0.460141...	-0.68805...	-0.264579...	0.00067...
3	0.147035...	0.636830...	-0.623115...	-0.460618...	-0.688525...	-0.265050...	0.00129...
4	0.148439...	0.636223...	-0.623711...	-0.461225...	-0.689118...	-0.265652...	0.00209...
5	0.145730...	0.635502...	-0.624222...	-0.461946...	-0.689824...	-0.266366...	0.00304...

The "Error" section shows a table with columns Epoch, MSE. The first few rows of errors are:

Epoch	MSE
1	0.101855675233776
2	0.10161152739925053
3	0.10126029753049941
4	0.10081495523411041
5	0.10028730251757587

The table continues with many more rows of data, showing a general downward trend in the error value over time.

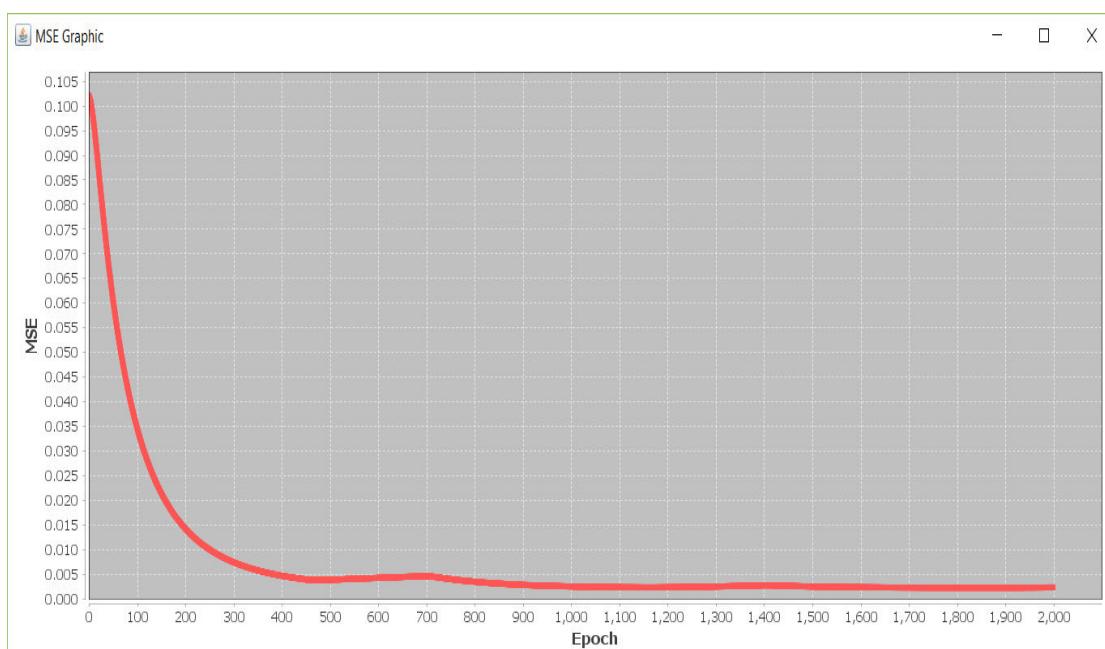
Gambar 4.3. Nilai *MSE* dalam lapisan *output* pada pengujian kedua.

The screenshot shows a software window titled "Beranda". The "Configuration" panel includes fields for Epoch (2000), Epsilon (0.0000001 (10^-8)), Target Error (0.001), Learning Rate (0.001), and Hidden Layer (6). It also has "Browse" and "D:\data_uji\pengujian.csv" buttons, and "Process" and "Graphic" tabs. The "Training" section displays a table of input data (x1 to T) with all values set to 0. The "Weight" section shows two tables: one for weights (w1 to w6, bias) and one for errors (Epoch, MSE). The "Error" table lists 2000 rows of MSE values.

Epoch	w1	w2	w3	w4	w5	w6	Bias	Epoch	MSE
1	0.14783...	0.63763...	-0.622316...	-0.45800...	-0.687724...	-0.264248...	0.00023...	1953	0.002246185176828806
2	0.147506...	0.637304...	-0.622645...	-0.460141...	-0.688054...	-0.264579...	0.000067...	1954	0.002246714982854511
3	0.147035...	0.636830...	-0.623115...	-0.460618...	-0.688525...	-0.265050...	0.00129...	1955	0.0022472537239120946
4	0.146439...	0.636223...	-0.623711...	-0.461225...	-0.689118...	-0.265652...	0.000209...	1956	0.0022478010450808395
5	0.145730...	0.635502...	-0.624242...	-0.461946...	-0.689824...	-0.266366...	0.000304...	1957	0.0022483563391910567
6	0.144919...	0.634685...	-0.625231...	-0.462769...	-0.690634...	-0.267182...	0.000412...	1958	0.0022489210820596196
7	0.144017...	0.633777...	-0.626131...	-0.463683...	-0.691535...	-0.268089...	0.000532...	1959	0.0022494948200673147
8	0.143034...	0.632790...	-0.627114...	-0.464678...	-0.692519...	-0.269078...	0.000664...	1960	0.0022500786624125284
9	0.141978...	0.631728...	-0.628172...	-0.465746...	-0.693574...	-0.270138...	0.000805...	1961	0.0022508734542980818
10	0.140858...	0.630602...	-0.629295...	-0.466879...	-0.694695...	-0.271264...	0.000955...	1962	0.002251276873594636
11	0.139679...	0.629418...	-0.630477...	-0.468068...	-0.695873...	-0.272446...	0.001114...	1963	0.002251890467773653
12	0.138450...	0.628185...	-0.631710...	-0.469309...	-0.697104...	-0.273680...	0.001279...	1964	0.0022525147195384372
13	0.137174...	0.626905...	-0.632989...	-0.470595...	-0.698379...	-0.274959...	0.001452...	1965	0.0022531470116916724
14	0.135859...	0.625583...	-0.634310...	-0.471922...	-0.699895...	-0.276276...	0.001630...	1966	0.002253790302536463
15	0.134508...	0.624225...	-0.635668...	-0.472825...	-0.701045...	-0.277629...	0.001813...	1967	0.002254443023297078
16	0.133126...	0.622285...	-0.637058...	-0.474679...	-0.702428...	-0.279011...	0.002002...	1968	0.0022551054145915597
17	0.131716...	0.621419...	-0.638476...	-0.476099...	-0.703835...	-0.280422...	0.002194...	1969	0.0022557765290249867
18	0.130284...	0.619979...	-0.639918...	-0.477543...	-0.705267...	-0.281856...	0.002391...	1970	0.002256457721028514
19	0.128830...	0.618517...	-0.641381...	-0.479009...	-0.706719...	-0.283310...	0.002591...	1971	0.0022571484228852733
20	0.127359...	0.617038...	-0.642863...	-0.480491...	-0.708190...	-0.284781...	0.002794...	1972	0.00225784791259135
21	0.125872...	0.615545...	-0.644358...	-0.481988...	-0.709677...	-0.286269...	0.003000...	1973	0.00225855597790911
22	0.124373...	0.614040...	-0.645868...	-0.483498...	-0.711176...	-0.287768...	0.003209...	1974	0.0022592729885078933
23	0.122863...	0.612525...	-0.647388...	-0.485018...	-0.712686...	-0.289279...	0.003420...	1975	0.00225999864756578
24	0.121344...	0.611002...	-0.648917...	-0.486546...	-0.714205...	-0.290797...	0.003633...	1976	0.0022607344903337492
25	0.119819...	0.609472...	-0.650453...	-0.488081...	-0.715731...	-0.292323...	0.003848...	1977	0.0022614813165623
26	0.118288...	0.607938...	-0.651995...	-0.489622...	-0.717263...	-0.293854...	0.004064...	1978	0.00226222384192219356
27	0.116754...	0.606399...	-0.653543...	-0.491167...	-0.718797...	-0.295388...	0.004283...	1979	0.0022630054818275963
28	0.115220...	0.604853...	-0.655096...	-0.492716...	-0.720328...	-0.296928...	0.004502...	1980	0.0022637833088103676
29	0.113685...	0.603307...	-0.656651...	-0.494266...	-0.721863...	-0.298468...	0.004723...	1981	0.002264571216742212
30	0.112149...	0.601762...	-0.658205...	-0.495817...	-0.723400...	-0.300010...	0.004945...	1982	0.002265370575533247
31	0.110613...	0.600218...	-0.659760...	-0.497367...	-0.724936...	-0.301550...	0.005168...	1983	0.0022661793340410692
32	0.109078...	0.598677...	-0.661313...	-0.498916...	-0.726473...	-0.303090...	0.005392...	1984	0.002266990896391456
33	0.107545...	0.597138...	-0.662863...	-0.500463...	-0.728010...	-0.304628...	0.005616...	1985	0.0022678259421092957
34	0.106014...	0.595602...	-0.664412...	-0.502008...	-0.729544...	-0.306163...	0.005841...	1986	0.002268660873339987
35	0.104487...	0.594070...	-0.665958...	-0.503550...	-0.731075...	-0.307695...	0.006067...	1987	0.0022695051133110014
36	0.102962...	0.592541...	-0.667500...	-0.505088...	-0.732603...	-0.309223...	0.006293...	1988	0.0022703587824943312
37	0.101442...	0.591018...	-0.669039...	-0.506622...	-0.734126...	-0.310748...	0.006520...	1989	0.0022712217922124123
38	0.099928...	0.589501...	-0.670572...	-0.508152...	-0.735645...	-0.312270...	0.006747...	1990	0.0022720925958320893
39	0.098416...	0.587991...	-0.672100...	-0.509876...	-0.737159...	-0.313787...	0.006974...	1991	0.002272973779882145
40	0.096911...	0.586483...	-0.673624...	-0.511196...	-0.738669...	-0.315299...	0.007022...	1992	0.002273863735567669
41	0.095411...	0.584983...	-0.675142...	-0.512711...	-0.740173...	-0.316805...	0.007429...	1993	0.0022747636858974213
42	0.093917...	0.583488...	-0.676655...	-0.514221...	-0.741674...	-0.318305...	0.007657...	1994	0.002275670677185821
43	0.092430...	0.582001...	-0.678162...	-0.515724...	-0.743166...	-0.319800...	0.007885...	1995	0.0022765876389716487
44	0.090948...	0.580520...	-0.679664...	-0.517222...	-0.744654...	-0.321288...	0.008114...	1996	0.002277514682772539
45	0.089474...	0.579044...	-0.681161...	-0.518714...	-0.746134...	-0.322789...	0.008342...	1997	0.0022784521928143394
46	0.088006...	0.577577...	-0.682652...	-0.520199...	-0.747608...	-0.324244...	0.008570...	1998	0.0022793981770593717
47	0.086545...	0.576116...	-0.684137...	-0.521678...	-0.749076...	-0.325712...	0.008799...	1999	0.0022803512143308633
48	0.085091...	0.574662...	-0.685614...	-0.523151...	-0.750538...	-0.327173...	0.009027...	2000	0.0022813152592056

Gambar 4.3. Nilai MSE dalam lapisan output pada pengujian kedua (Lanjutan).

Gambar 4.3. merupakan hasil nilai *MSE* dalam lapisan *output* pada pengujian kedua yang dilakukan perhitungan keseluruhan data dataset *breast cancer*. Pengujian kedua penurunan hasil dari nilai *MSE* untuk perhitungan keseluruhan pada *epoch* dalam penerapan metode *ADAM* untuk meningkatkan proses pembelajaran metode *gradient descent* pada *backpropagation*. Untuk gambar 4.3. nilai *MSE* dalam lapisan *output* pada pengujian kedua dapat dilihat selengkapnya pada lampiran 5. Penurunan hasil dari nilai *MSE* dapat dilihat grafik pada gambar 4.4 :



Gambar 4.4. Grafik pengujian kedua nilai *MSE*.

Gambar grafik 4.4 diatas menunjukkan penurunan tidak begitu tinggi dari hasil nilai *MSE* pada pengujian kedua untuk untuk *maximum epoch* 2000. Hasil dari *ADAM* pada *output* dari setiap *epoch* untuk *maximum epoch* 2000. Untuk nilai *mean square error* (*MSE*) menunjukkan bahwa implementasi *ADAM* telah dilakukan pada *epoch* 2000 di *ADAM* dapat beradaptasi lagi dengan set data pelatihan yang digunakan. Untuk *epoch* 2000, nilai *mean square error* di *epoch* pada *output* untuk setiap *epoch* menurun dari 0.0024769988314805587 menjadi 0.00228131525892056, yang berarti untuk peminimalisasian / penurunan kuadrat *error* dalam *output* yang dihasilkan oleh *ADAM*.

4.2. Pembahasan

Dari grafik pengujian pertama *maximum epoch* 1000 dan pengujian kedua *maximum epoch* 2000 mendapatkan peminimalisasian / penurunan hasil dari nilai *MSE* di setiap pengujian pada keseluruhan *epoch*. Hasil pengujian yang telah dilakukan pada jaringan *neural network* yaitu melakukan proses pembelajaran *ADAM* untuk meningkatkan proses pembelajaran metode *gradient descent* pada *backpropagation* dapat membantu kinerja pembelajaran dari jaringan *neural network* pada *gradient descent backpropagation*.

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan dari hasil penelitian yang dilakukan yaitu dengan penerapan metode *ADAM* sebagai peningkatan proses pembelajaran metode *gradient descent* pada *backpropagation* untuk membantu kinerja pembelajaran *gradient descent* pada *backpropagation* dengan melihat proses pada pengujian dengan *maximum epoch* 1000 dan pengujian *maximum epoch* 2000 di setiap perhitungan pada *epoch* dengan pengujian berbeda *maximum* pada *epoch* dapat meminimalkan nilai MSE. Namun pada semua pengujian, untuk mencapai target *error* dengan nilai 0.001 yang sudah ditentukan masih membutuhkan jumlah *epoch* yang lebih tinggi untuk bisa mendapatkan target *error* yang ditentukan.

5.2. Saran

Saran yang diberikan oleh penulis adalah untuk penggunaan metode *adaptive moment estimation* sebagai peningkatan proses pembelajaran metode *gradient descent* pada *backpropagation* dibutuhkan pola dan metode baru untuk tingkat pembelajaran ke dalam arsitektur *neural network*, sehingga untuk mencapai target *error* yang ditentukan, jumlah *epoch* yang didapat sedikit untuk mencapai target *error*.

DAFTAR PUSTAKA

- Abraham, A., 2005. Artificial Neural Networks. *Handbook of Measuring System Design*, ISBN 0-470-02143-8: 901 – 908.
- Achkar, R., Geagea, R., Mehio., & Kmeish, W., 2016. SmartCoach personal gym trainer: An Adaptive Modified Backpropagation approach. *Proceedings of the IEEE International Multidisciplinary Conference on Engineering Technology*, pp. 1 – 6.
- Ahmad, F., Isa, N., A., M., Osman M., K., & Hussain, Z., 2010. Performance comparison of gradient descent and Genetic Algorithm based Artificial Neural Networks training. *Proceedings of the IEEE International Conference on Intelligent Systems Design and Applications*, pp. 604 – 609.
- Andayani, U., Nababan, E., B., Siregar, B., Muchtar, M., A., Nasution, T., H., & Siregar, I., 2017. Optimization backpropagation algorithm based on Nguyen–Widrow adaptive weight and adaptive learning rate. *Proceeding of the IEEE International Conference on Industrial Engineering and Application*, pp. 363 – 367.
- Chen, C., C., Kuo, C., Kuo, S., Y., & Chou, Y., H., 2015. Dynamic Normalization BPN for Stock Price Forecasting. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2855 - 2860.
- Chen, L., & Wang, J., 2012. Dictionary learning with weight stochastic gradient descent. *Proceedings of the IEEE International Conference on Computational Problem-Solving*, pp. 9 – 12.
- Endah, S., N., Widodo, A., P., Fariq, M., L., Nadianada, S., I., & Maulana, F., 2017. Beyond back-propagation learning for diabetic: Convergence comparison of gradient descent, momentum and Adaptive Learning Rate. *Proceedings of the IEEE International Conference on Informatics and Computational Sciences*, pp. 189 – 194.
- Gao, T., Chai, Y., & Liu, Y., 2017. Applying long short term momory neural networks for predicting stock closing price. *Proceedings of the IEEE International Conference on Software Engineering and Service Science*,pp. 575 – 578.
- Guan, N., Shan, L., Yang, C., Xu W., & Zhang, M., 2017. Delay Compensated Asynchronous Adam Algorithm for Deep Neural Networks. *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications*, pp. 852 – 859.
- Heaton, J., 2008. *Introduction to Neural Networks with Java Second Edition*. United States of America: Heaton Research, Inc.
- Heravi, A., R., & Hodtani, G., A., 2018. A New Correntropy-Based Conjugate Gradient Backpropagation Algorithm for Improving Training in Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* **29**(12) : 6252 – 6263.

- Indolia, S., Goswami, A., K., Mishra, S., P., & Asopa, P., 2018. Conceptual Understanding of Convolutional Neural Network – A Deep Learning Approach. *Proceedings of the Elsevier International Conference on Computational Intelligence and Data Science* **132**: 679 - 688.
- Lenka, S., K., & Mohapatra, A., G., 2015. Gradient Descent with Momentum based Neural Network Pattern Classification for the Prediction of Soil Moisture Content in Precision Agriculture. *Proceedings of the IEEE International Symposium on Nanoelectronic and Information Systems*, pp. 63 – 66.
- Li, Y., Zhu, L., Zhou, L.-j., & Jiang, J., 2011. Study on the BP–GA model and its application in water quality assessment. *Proceedings of the IEEE International Symposium on Water Resource and Environmental Protection*, pp. 2781 – 2784.
- Mahasagara, S., P., Alamsya A., Rikumahu B., 2017. Indonesia infrastructure and consumer stock portfolio prediction using artificial neural network backpropagation. *Proceedings of the International Conference on Information and Communication Technology (IcoIC7)*.
- Mammadli, S., 2017. Financial time series prediction using artificial neural network based on Levenberg–Marquardt algorithm. *Proceedings of the Elsevier International Conference on Theory and Application of Soft Computing, Computing with Words and Perception* **120**: 602 – 607.
- Paulin, F., & Santhakumaran, A., 2010. Back Propagation Neural Network by Comparing Hidden Neuron: Case study on Breast Cancer Diagnosis. *International Journal of Computer Application* **2**(4): 40 – 44.
- Popa, C-A., 2014. Enhanced Gradient Descent Algorithms for Complex-Valued Neural Networks. *Proceedings of the IEEE International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pp. 272 – 279.
- Rahmat, R., F., Silviani, S., Nababan, E., B., Sitompul, O., S., Anugrahwaty, R., Silmi, S., 2017. Identification of molar and premolar teeth in dental panoramic radiograph image. *Proceedings of the IEEE Second International Conference on Informatics and Computing (ICIC)*.
- Seniman, Arisandi, D., Rahmat, R., F., William, Nababan, E., B., 2016. Chinese chess character recognition using Direction Feature Extraction and backpropagation. *Proceedings of the IEEE International Conference on Data and Software Engineering (ICoDSE)*.
- Singh, B., K., Verma, K., & Thoke, A., S., 2015. Adaptive Gradient Descent Backpropagation for Classification of Breast Tumor in Ultrasound Imaging. *Proceedings of the Elsevier International Conference on Information and Communication Technologies* **46**: 1601 – 1609.
- Srinivasan, N., Ravichandran, V., Chan, K., L., Vidhya, J., R., Ramakirishnan, S., & Krishnan, S., M., 2002. Exponentiated backpropagation algorithm for multilayer feedforward neural networks. *Proceedings of the IEEE Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02*, pp. 327 – 331.
- Teixeira, J., P., & Fernandes, P., O., 2012. Tourism Time Series Forecast–Different ANN Architectures with Time Index Input. *Proceedings of the Elsevier Conference on ENTERprise Information Systems / HCIST 2012 – International Conference on Health and Social Care Information Systems and Technologies* **5**: 445 – 454.

- Wakitani, S., Yamamoto, T., & Ishimura, A., 2017. Study on an adaptive GMDH-PID controller using adaptive moment estimation. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1587 – 1591.
- Wijnhoven, R., G., J., & With P., H., N., D., 2010. Fast Training of Object Detection Using Stochastic Gradient Descent. *Proceedings of the IEEE International Conference on Pattern Recognition*, pp. 424 – 427.
- Wu, M., Guo, S., Chen, X., Xing, N., & Zhong, C., 2016. LM–BP based operation quality assessment method for OTN in Smart grid. *Proceedings of the IEEE Network Operations and Management Symposium*.
- Zhao, C., Shi, D., & Gao, Y., 2012. Antenna Recognition Based On BP Neural Network. *Proceedings of the IEEE Asia-Pacific Conference on Environmental Electromagnetics*, pp. 2855 – 2860.

LAMPIRAN 1

TABEL 3.2 *Breast cancer* dataset

No.	id number	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
1.	1000025	5	1	1	1	2	1	3	1	1	2
2.	1002945	5	4	4	5	7	10	3	2	1	2
3.	1015425	3	1	1	1	2	2	3	1	1	2
4.	1016277	6	8	8	1	3	4	3	7	1	2
5.	1017023	4	1	1	3	2	1	3	1	1	2
6.	1017122	8	10	10	8	7	10	9	7	1	4
7.	1018099	1	1	1	1	2	10	3	1	1	2
8.	1018561	2	1	2	1	2	1	3	1	1	2
9.	1033078	2	1	1	1	2	1	1	1	5	2
10.	1033078	4	2	1	1	2	1	2	1	1	2
11.	1035283	1	1	1	1	1	1	3	1	1	2
12.	1036172	2	1	1	1	2	1	2	1	1	2
13.	1041801	5	3	3	3	2	3	4	4	1	4
14.	1043999	1	1	1	1	2	3	3	1	1	2
15.	1044572	8	7	5	10	7	9	5	5	4	4
16.	1047630	7	4	6	4	6	1	4	3	1	4
17.	1048672	4	1	1	1	2	1	2	1	1	2
18.	1049815	4	1	1	1	2	1	3	1	1	2
19.	1050670	10	7	7	6	4	10	4	1	2	4
20.	1050718	6	1	1	1	2	1	3	1	1	2
21.	1054590	7	3	2	10	5	10	5	4	4	4
22.	1054593	10	5	5	3	6	7	7	10	1	4
23.	1056784	3	1	1	1	2	1	2	1	1	2
24.	1057013	8	4	5	1	2	?	7	3	1	4
25.	1059552	1	1	1	1	2	1	3	1	1	2
26.	1065726	5	2	3	4	2	7	3	6	1	4
27.	1066373	3	2	1	1	1	1	2	1	1	2
28.	1066979	5	1	1	1	2	1	2	1	1	2
29.	1067444	2	1	1	1	2	1	2	1	1	2
30.	1070935	1	1	3	1	2	1	1	1	1	2
31.	1070935	3	1	1	1	1	1	2	1	1	2
32.	1071760	2	1	1	1	2	1	3	1	1	2
33.	1072179	10	7	7	3	8	5	7	4	3	4
34.	1074610	2	1	1	2	2	1	3	1	1	2
35.	1075123	3	1	2	1	2	1	2	1	1	2
36.	1079304	2	1	1	1	2	1	2	1	1	2
37.	1080185	10	10	10	8	6	1	8	9	1	4
38.	1081791	6	2	1	1	1	1	7	1	1	2
39.	1084584	5	4	4	9	2	10	5	6	1	4
40.	1091262	2	5	3	3	6	7	7	5	1	4
41.	1096800	6	6	6	9	6	?	7	8	1	2
42.	1099510	10	4	3	1	3	3	6	5	2	4
43.	1100524	6	10	10	2	8	10	7	3	3	4
44.	1102573	5	6	5	6	10	1	3	1	1	4
45.	1103608	10	10	10	4	8	1	8	10	1	4
46.	1103722	1	1	1	1	2	1	2	1	2	2
47.	1105257	3	7	7	4	4	9	4	8	1	4
48.	1105524	1	1	1	1	2	1	2	1	1	2

49.	1106095	4	1	1	3	2	1	3	1	1	2
50.	1106829	7	8	7	2	4	8	3	8	2	4
51.	1108370	9	5	8	1	2	3	2	1	5	4
52.	1108449	5	3	3	4	2	4	3	4	1	4
53.	1110102	10	3	6	2	3	5	4	10	2	4
54.	1110503	5	5	5	8	10	8	7	3	7	4
55.	1110524	10	5	5	6	8	8	7	1	1	4
56.	1111249	10	6	6	3	4	5	3	6	1	4
57.	1112209	8	10	10	1	3	6	3	9	1	4
58.	1113038	8	2	4	1	5	1	5	4	4	4
59.	1113483	5	2	3	1	6	10	5	1	1	4
60.	1113906	9	5	5	2	2	2	5	1	1	4
61.	1115282	5	3	5	5	3	3	4	10	1	4
62.	1115293	1	1	1	1	2	2	2	1	1	2
63.	1116116	9	10	10	1	10	8	3	3	1	4
64.	1116132	6	3	4	1	5	2	3	9	1	4
65.	1116192	1	1	1	1	2	1	2	1	1	2
66.	1116998	10	4	2	1	3	2	4	3	10	4
67.	1117152	4	1	1	1	2	1	3	1	1	2
68.	1118039	5	3	4	1	8	10	4	9	1	4
69.	1120559	8	3	8	3	4	9	8	9	8	4
70.	1121732	1	1	1	1	2	1	3	2	1	2
71.	1121919	5	1	3	1	2	1	2	1	1	2
72.	1123061	6	10	2	8	10	2	7	8	10	4
73.	1124651	1	3	3	2	2	1	7	2	1	2
74.	1125035	9	4	5	10	6	10	4	8	1	4
75.	1126417	10	6	4	1	3	4	3	2	3	4
76.	1131294	1	1	2	1	2	2	4	2	1	2
77.	1132347	1	1	4	1	2	1	2	1	1	2
78.	1133041	5	3	1	2	2	1	2	1	1	2
79.	1133136	3	1	1	1	2	3	3	1	1	2
80.	1136142	2	1	1	1	3	1	2	1	1	2
81.	1137156	2	2	2	1	1	1	7	1	1	2
82.	1143978	4	1	1	2	2	1	2	1	1	2
83.	1143978	5	2	1	1	2	1	3	1	1	2
84.	1147044	3	1	1	1	2	2	7	1	1	2
85.	1147699	3	5	7	8	8	9	7	10	7	4
86.	1147748	5	10	6	1	10	4	4	10	10	4
87.	1148278	3	3	6	4	5	8	4	4	1	4
88.	1148873	3	6	6	6	5	10	6	8	3	4
89.	1152331	4	1	1	1	2	1	3	1	1	2
90.	1155546	2	1	1	2	3	1	2	1	1	2
91.	1156272	1	1	1	1	2	1	3	1	1	2
92.	1156948	3	1	1	2	2	1	1	1	1	2
93.	1157734	4	1	1	1	2	1	3	1	1	2
94.	1158247	1	1	1	1	2	1	2	1	1	2
95.	1160476	2	1	1	1	2	1	3	1	1	2
96.	1164066	1	1	1	1	2	1	3	1	1	2

97.	1165297	2	1	1	2	2	1	1	1	1	2
98.	1165790	5	1	1	1	2	1	3	1	1	2
99.	1165926	9	6	9	2	10	6	2	9	10	4
100.	1166630	7	5	6	10	5	10	7	9	4	4
101.	1166654	10	3	5	1	10	5	3	10	2	4
102.	1167439	2	3	4	4	2	5	2	5	1	4
103.	1167471	4	1	2	1	2	1	3	1	1	2
104.	1168359	8	2	3	1	6	3	7	1	1	4
105.	1168736	10	10	10	10	10	1	8	8	8	4
106.	1169049	7	3	4	4	3	3	3	2	7	4
107.	1170419	10	10	10	8	2	10	4	1	1	4
108.	1170420	1	6	8	10	8	10	5	7	1	4
109.	1171710	1	1	1	1	2	1	2	3	1	2
110.	1171710	6	5	4	4	3	9	7	8	3	4
111.	1171795	1	3	1	2	2	2	5	3	2	2
112.	1171845	8	6	4	3	5	9	3	1	1	4
113.	1172152	10	3	3	10	2	10	7	3	3	4
114.	1173216	10	10	10	3	10	8	8	1	1	4
115.	1173235	3	3	2	1	2	3	3	1	1	2
116.	1173347	1	1	1	1	2	5	1	1	1	2
117.	1173347	8	3	3	1	2	2	3	2	1	2
118.	1173509	4	5	5	10	4	10	7	5	8	4
119.	1173514	1	1	1	1	4	3	1	1	1	2
120.	1173681	3	2	1	1	2	2	3	1	1	2
121.	1174057	1	1	2	2	2	1	3	1	1	2
122.	1174057	4	2	1	1	2	2	3	1	1	2
123.	1174131	10	10	10	2	10	10	5	3	3	4
124.	1174428	5	3	5	1	8	10	5	3	1	4
125.	1175937	5	4	6	7	9	7	8	10	1	4
126.	1176406	1	1	1	1	2	1	2	1	1	2
127.	1176881	7	5	3	7	4	10	7	5	5	4
128.	1177027	3	1	1	1	2	1	3	1	1	2
129.	1177399	8	3	5	4	5	10	1	6	2	4
130.	1177512	1	1	1	1	10	1	1	1	1	2
131.	1178580	5	1	3	1	2	1	2	1	1	2
132.	1179818	2	1	1	1	2	1	3	1	1	2
133.	1180194	5	10	8	10	8	10	3	6	3	4
134.	1180523	3	1	1	1	2	1	2	2	1	2
135.	1180831	3	1	1	1	3	1	2	1	1	2
136.	1181356	5	1	1	1	2	2	3	3	1	2
137.	1182404	4	1	1	1	2	1	2	1	1	2
138.	1182410	3	1	1	1	2	1	1	1	1	2
139.	1183240	4	1	2	1	2	1	2	1	1	2
140.	1183246	1	1	1	1	1	?	2	1	1	2
141.	1183516	3	1	1	1	2	1	1	1	1	2
142.	1183911	2	1	1	1	2	1	1	1	1	2
143.	1183983	9	5	5	4	4	5	4	3	3	4
144.	1184184	1	1	1	1	2	5	1	1	1	2

145.	1184241	2	1	1	1	2	1	2	1	1	2
146.	1184840	1	1	3	1	2	?	2	1	1	2
147.	1185609	3	4	5	2	6	8	4	1	1	4
148.	1185610	1	1	1	1	3	2	2	1	1	2
149.	1187457	3	1	1	3	8	1	5	8	1	2
150.	1187805	8	8	7	4	10	10	7	8	7	4
151.	1188472	1	1	1	1	1	1	3	1	1	2
152.	1189266	7	2	4	1	6	10	5	4	3	4
153.	1189286	10	10	8	6	4	5	8	10	1	4
154.	1190394	4	1	1	1	2	3	1	1	1	2
155.	1190485	1	1	1	1	2	1	1	1	1	2
156.	1192325	5	5	5	6	3	10	3	1	1	4
157.	1193091	1	2	2	1	2	1	2	1	1	2
158.	1193210	2	1	1	1	2	1	3	1	1	2
159.	1193683	1	1	2	1	3	?	1	1	1	2
160.	1196295	9	9	10	3	6	10	7	10	6	4
161.	1196915	10	7	7	4	5	10	5	7	2	4
162.	1197080	4	1	1	1	2	1	3	2	1	2
163.	1197270	3	1	1	1	2	1	3	1	1	2
164.	1197440	1	1	1	2	1	3	1	1	7	2
165.	1197510	5	1	1	1	2	?	3	1	1	2
166.	1197979	4	1	1	1	2	2	3	2	1	2
167.	1197993	5	6	7	8	8	10	3	10	3	4
168.	1198128	10	8	10	10	6	1	3	1	10	4
169.	1198641	3	1	1	1	2	1	3	1	1	2
170.	1199219	1	1	1	2	1	1	1	1	1	2
171.	1199731	3	1	1	1	2	1	1	1	1	2
172.	1199983	1	1	1	1	2	1	3	1	1	2
173.	1200772	1	1	1	1	2	1	2	1	1	2
174.	1200847	6	10	10	10	8	10	10	10	7	4
175.	1200892	8	6	5	4	3	10	6	1	1	4
176.	1200952	5	8	7	7	10	10	5	7	1	4
177.	1201834	2	1	1	1	2	1	3	1	1	2
178.	1201936	5	10	10	3	8	1	5	10	3	4
179.	1202125	4	1	1	1	2	1	3	1	1	2
180.	1202812	5	3	3	3	6	10	3	1	1	4
181.	1203096	1	1	1	1	1	1	3	1	1	2
182.	1204242	1	1	1	1	2	1	1	1	1	2
183.	1204898	6	1	1	1	2	1	3	1	1	2
184.	1205138	5	8	8	8	5	10	7	8	1	4
185.	1205579	8	7	6	4	4	10	5	1	1	4
186.	1206089	2	1	1	1	1	1	3	1	1	2
187.	1206695	1	5	8	6	5	8	7	10	1	4
188.	1206841	10	5	6	10	6	10	7	7	10	4
189.	1207986	5	8	4	10	5	8	9	10	1	4
190.	1208301	1	2	3	1	2	1	3	1	1	2
191.	1210963	10	10	10	8	6	8	7	10	1	4
192.	1211202	7	5	10	10	10	10	4	10	3	4

193.	1212232	5	1	1	1	2	1	2	1	1	2
194.	1212251	1	1	1	1	2	1	3	1	1	2
195.	1212422	3	1	1	1	2	1	3	1	1	2
196.	1212422	4	1	1	1	2	1	3	1	1	2
197.	1213375	8	4	4	5	4	7	7	8	2	2
198.	1213383	5	1	1	4	2	1	3	1	1	2
199.	1214092	1	1	1	1	2	1	1	1	1	2
200.	1214556	3	1	1	1	2	1	2	1	1	2
201.	1214966	9	7	7	5	5	10	7	8	3	4
202.	1216694	10	8	8	4	10	10	8	1	1	4
203.	1216947	1	1	1	1	2	1	3	1	1	2
204.	1217051	5	1	1	1	2	1	3	1	1	2
205.	1217264	1	1	1	1	2	1	3	1	1	2
206.	1218105	5	10	10	9	6	10	7	10	5	4
207.	1218741	10	10	9	3	7	5	3	5	1	4
208.	1218860	1	1	1	1	1	1	3	1	1	2
209.	1218860	1	1	1	1	1	1	3	1	1	2
210.	1219406	5	1	1	1	1	1	3	1	1	2
211.	1219525	8	10	10	10	5	10	8	10	6	4
212.	1219859	8	10	8	8	4	8	7	7	1	4
213.	1220330	1	1	1	1	2	1	3	1	1	2
214.	1221863	10	10	10	10	7	10	7	10	4	4
215.	1222047	10	10	10	10	3	10	10	6	1	4
216.	1222936	8	7	8	7	5	5	5	10	2	4
217.	1223282	1	1	1	1	2	1	2	1	1	2
218.	1223426	1	1	1	1	2	1	3	1	1	2
219.	1223793	6	10	7	7	6	4	8	10	2	4
220.	1223967	6	1	3	1	2	1	3	1	1	2
221.	1224329	1	1	1	2	2	1	3	1	1	2
222.	1225799	10	6	4	3	10	10	9	10	1	4
223.	1226012	4	1	1	3	1	5	2	1	1	4
224.	1226612	7	5	6	3	3	8	7	4	1	4
225.	1227210	10	5	5	6	3	10	7	9	2	4
226.	1227244	1	1	1	1	2	1	2	1	1	2
227.	1227481	10	5	7	4	4	10	8	9	1	4
228.	1228152	8	9	9	5	3	5	7	7	1	4
229.	1228311	1	1	1	1	1	1	3	1	1	2
230.	1230175	10	10	10	3	10	10	9	10	1	4
231.	1230688	7	4	7	4	3	7	7	6	1	4
232.	1231387	6	8	7	5	6	8	8	9	2	4
233.	1231706	8	4	6	3	3	1	4	3	1	2
234.	1232225	10	4	5	5	5	10	4	1	1	4
235.	1236043	3	3	2	1	3	1	3	6	1	2
236.	1241232	3	1	4	1	2	?	3	1	1	2
237.	1241559	10	8	8	2	8	10	4	8	10	4
238.	1241679	9	8	8	5	6	2	4	10	4	4
239.	1242364	8	10	10	8	6	9	3	10	10	4
240.	1243256	10	4	3	2	3	10	5	3	2	4

241.	1270479	5	1	3	3	2	2	2	3	1	2
242.	1276091	3	1	1	3	1	1	3	1	1	2
243.	1277018	2	1	1	1	2	1	3	1	1	2
244.	128059	1	1	1	1	2	5	5	1	1	2
245.	1285531	1	1	1	1	2	1	3	1	1	2
246.	1287775	5	1	1	2	2	2	3	1	1	2
247.	144888	8	10	10	8	5	10	7	8	1	4
248.	145447	8	4	4	1	2	9	3	3	1	4
249.	167528	4	1	1	1	2	1	3	6	1	2
250.	169356	3	1	1	1	2	?	3	1	1	2
251.	183913	1	2	2	1	2	1	1	1	1	2
252.	191250	10	4	4	10	2	10	5	3	3	4
253.	1017023	6	3	3	5	3	10	3	5	3	2
254.	1100524	6	10	10	2	8	10	7	3	3	4
255.	1116116	9	10	10	1	10	8	3	3	1	4
256.	1168736	5	6	6	2	4	10	3	6	1	4
257.	1182404	3	1	1	1	2	1	1	1	1	2
258.	1182404	3	1	1	1	2	1	2	1	1	2
259.	1198641	3	1	1	1	2	1	3	1	1	2
260.	242970	5	7	7	1	5	8	3	4	1	2
261.	255644	10	5	8	10	3	10	5	1	3	4
262.	263538	5	10	10	6	10	10	10	6	5	4
263.	274137	8	8	9	4	5	10	7	8	1	4
264.	303213	10	4	4	10	6	10	5	5	1	4
265.	314428	7	9	4	10	10	3	5	3	3	4
266.	1182404	5	1	4	1	2	1	3	2	1	2
267.	1198641	10	10	6	3	3	10	4	3	2	4
268.	320675	3	3	5	2	3	10	7	1	1	4
269.	324427	10	8	8	2	3	4	8	7	8	4
270.	385103	1	1	1	1	2	1	3	1	1	2
271.	390840	8	4	7	1	3	10	3	9	2	4
272.	411453	5	1	1	1	2	1	3	1	1	2
273.	320675	3	3	5	2	3	10	7	1	1	4
274.	428903	7	2	4	1	3	4	3	3	1	4
275.	431495	3	1	1	1	2	1	3	2	1	2
276.	432809	3	1	3	1	2	?	2	1	1	2
277.	434518	3	1	1	1	2	1	2	1	1	2
278.	452264	1	1	1	1	2	1	2	1	1	2
279.	456282	1	1	1	1	2	1	3	1	1	2
280.	476903	10	5	7	3	3	7	3	3	8	4
281.	486283	3	1	1	1	2	1	3	1	1	2
282.	486662	2	1	1	2	2	1	3	1	1	2
283.	488173	1	4	3	10	4	10	5	6	1	4
284.	492268	10	4	6	1	2	10	5	3	1	4
285.	508234	7	4	5	10	2	10	3	8	2	4
286.	527363	8	10	10	10	8	10	10	7	3	4
287.	529329	10	10	10	10	10	10	4	10	10	4
288.	535331	3	1	1	1	3	1	2	1	1	2

289.	543558	6	1	3	1	4	5	5	10	1	4
290.	555977	5	6	6	8	6	10	4	10	4	4
291.	560680	1	1	1	1	2	1	1	1	1	2
292.	561477	1	1	1	1	2	1	3	1	1	2
293.	563649	8	8	8	1	2	?	6	10	1	4
294.	601265	10	4	4	6	2	10	2	3	1	4
295.	606140	1	1	1	1	2	?	2	1	1	2
296.	606722	5	5	7	8	6	10	7	4	1	4
297.	616240	5	3	4	3	4	5	4	7	1	2
298.	61634	5	4	3	1	2	?	2	3	1	2
299.	625201	8	2	1	1	5	1	1	1	1	2
300.	63375	9	1	2	6	4	10	7	7	2	4
301.	635844	8	4	10	5	4	4	7	10	1	4
302.	636130	1	1	1	1	2	1	3	1	1	2
303.	640744	10	10	10	7	9	10	7	10	10	4
304.	646904	1	1	1	1	2	1	3	1	1	2
305.	653777	8	3	4	9	3	10	3	3	1	4
306.	659642	10	8	4	4	4	10	3	10	4	4
307.	666090	1	1	1	1	2	1	3	1	1	2
308.	666942	1	1	1	1	2	1	3	1	1	2
309.	667204	7	8	7	6	4	3	8	8	4	4
310.	673637	3	1	1	1	2	5	5	1	1	2
311.	684955	2	1	1	1	3	1	2	1	1	2
312.	688033	1	1	1	1	2	1	1	1	1	2
313.	691628	8	6	4	10	10	1	3	5	1	4
314.	693702	1	1	1	1	2	1	1	1	1	2
315.	704097	1	1	1	1	1	1	2	1	1	2
316.	704168	4	6	5	6	7	?	4	9	1	2
317.	706426	5	5	5	2	5	10	4	3	1	4
318.	709287	6	8	7	8	6	8	8	9	1	4
319.	718641	1	1	1	1	5	1	3	1	1	2
320.	721482	4	4	4	4	6	5	7	3	1	2
321.	730881	7	6	3	2	5	10	7	4	6	4
322.	733639	3	1	1	1	2	?	3	1	1	2
323.	733639	3	1	1	1	2	1	3	1	1	2
324.	733823	5	4	6	10	2	10	4	1	1	4
325.	740492	1	1	1	1	2	1	3	1	1	2
326.	743348	3	2	2	1	2	1	2	3	1	2
327.	752904	10	1	1	1	2	10	5	4	1	4
328.	756136	1	1	1	1	2	1	2	1	1	2
329.	760001	8	10	3	2	6	4	3	10	1	4
330.	760239	10	4	6	4	5	10	7	1	1	4
331.	76389	10	4	7	2	2	8	6	1	1	4
332.	764974	5	1	1	1	2	1	3	1	2	2
333.	770066	5	2	2	2	2	1	2	2	1	2
334.	785208	5	4	6	6	4	10	4	3	1	4
335.	785615	8	6	7	3	3	10	3	4	2	4
336.	792744	1	1	1	1	2	1	1	1	1	2

337.	797327	6	5	5	8	4	10	3	4	1	4
338.	798429	1	1	1	1	2	1	3	1	1	2
339.	704097	1	1	1	1	1	1	2	1	1	2
340.	806423	8	5	5	5	2	10	4	3	1	4
341.	809912	10	3	3	1	2	10	7	6	1	4
342.	810104	1	1	1	1	2	1	3	1	1	2
343.	814265	2	1	1	1	2	1	1	1	1	2
344.	814911	1	1	1	1	2	1	1	1	1	2
345.	822829	7	6	4	8	10	10	9	5	3	4
346.	826923	1	1	1	1	2	1	1	1	1	2
347.	830690	5	2	2	2	3	1	1	3	1	2
348.	831268	1	1	1	1	1	1	1	3	1	2
349.	832226	3	4	4	10	5	1	3	3	1	4
350.	832567	4	2	3	5	3	8	7	6	1	4
351.	836433	5	1	1	3	2	1	1	1	1	2
352.	837082	2	1	1	1	2	1	3	1	1	2
353.	846832	3	4	5	3	7	3	4	6	1	2
354.	850831	2	7	10	10	7	10	4	9	4	4
355.	855524	1	1	1	1	2	1	2	1	1	2
356.	857774	4	1	1	1	3	1	2	2	1	2
357.	859164	5	3	3	1	3	3	3	3	3	4
358.	859350	8	10	10	7	10	10	7	3	8	4
359.	866325	8	10	5	3	8	4	4	10	3	4
360.	873549	10	3	5	4	3	7	3	5	3	4
361.	877291	6	10	10	10	10	10	8	10	10	4
362.	877943	3	10	3	10	6	10	5	1	4	4
363.	888169	3	2	2	1	4	3	2	1	1	2
364.	888523	4	4	4	2	2	3	2	1	1	2
365.	896404	2	1	1	1	2	1	3	1	1	2
366.	897172	2	1	1	1	2	1	2	1	1	2
367.	95719	6	10	10	10	8	10	7	10	7	4
368.	160296	5	8	8	10	5	10	8	10	3	4
369.	342245	1	1	3	1	2	1	1	1	1	2
370.	428598	1	1	3	1	1	1	2	1	1	2
371.	492561	4	3	2	1	3	1	2	1	1	2
372.	493452	1	1	3	1	2	1	1	1	1	2
373.	493452	4	1	2	1	2	1	2	1	1	2
374.	521441	5	1	1	2	2	1	2	1	1	2
375.	560680	3	1	2	1	2	1	2	1	1	2
376.	636437	1	1	1	1	2	1	1	1	1	2
377.	640712	1	1	1	1	2	1	2	1	1	2
378.	654244	1	1	1	1	1	1	2	1	1	2
379.	657753	3	1	1	4	3	1	2	2	1	2
380.	685977	5	3	4	1	4	1	3	1	1	2
381.	805448	1	1	1	1	2	1	1	1	1	2
382.	846423	10	6	3	6	4	10	7	8	4	4
383.	1002504	3	2	2	2	2	1	3	2	1	2
384.	1022257	2	1	1	1	2	1	1	1	1	2

385.	1026122	2	1	1	1	2	1	1	1	1	2
386.	1071084	3	3	2	2	3	1	1	2	3	2
387.	1080233	7	6	6	3	2	10	7	1	1	4
388.	1114570	5	3	3	2	3	1	3	1	1	2
389.	1114570	2	1	1	1	2	1	2	2	1	2
390.	1116715	5	1	1	1	3	2	2	2	1	2
391.	1131411	1	1	1	2	2	1	2	1	1	2
392.	1151734	10	8	7	4	3	10	7	9	1	4
393.	1156017	3	1	1	1	2	1	2	1	1	2
394.	1158247	1	1	1	1	1	1	1	1	1	2
395.	1158405	1	2	3	1	2	1	2	1	1	2
396.	1168278	3	1	1	1	2	1	2	1	1	2
397.	1176187	3	1	1	1	2	1	3	1	1	2
398.	1196263	4	1	1	1	2	1	1	1	1	2
399.	1196475	3	2	1	1	2	1	2	2	1	2
400.	1206314	1	2	3	1	2	1	1	1	1	2
401.	1211265	3	10	8	7	6	9	9	3	8	4
402.	1213784	3	1	1	1	2	1	1	1	1	2
403.	1223003	5	3	3	1	2	1	2	1	1	2
404.	1223306	3	1	1	1	2	4	1	1	1	2
405.	1223543	1	2	1	3	2	1	1	2	1	2
406.	1229929	1	1	1	1	2	1	2	1	1	2
407.	1231853	4	2	2	1	2	1	2	1	1	2
408.	1234554	1	1	1	1	2	1	2	1	1	2
409.	1236837	2	3	2	2	2	2	3	1	1	2
410.	1237674	3	1	2	1	2	1	2	1	1	2
411.	1238021	1	1	1	1	2	1	2	1	1	2
412.	1238464	1	1	1	1	1	?	2	1	1	2
413.	1238633	10	10	10	6	8	4	8	5	1	4
414.	1238915	5	1	2	1	2	1	3	1	1	2
415.	1238948	8	5	6	2	3	10	6	6	1	4
416.	1239232	3	3	2	6	3	3	3	5	1	2
417.	1239347	8	7	8	5	10	10	7	2	1	4
418.	1239967	1	1	1	1	2	1	2	1	1	2
419.	1240337	5	2	2	2	2	2	3	2	2	2
420.	1253505	2	3	1	1	5	1	1	1	1	2
421.	1255384	3	2	2	3	2	3	3	1	1	2
422.	1257200	10	10	10	7	10	10	8	2	1	4
423.	1257648	4	3	3	1	2	1	3	3	1	2
424.	1257815	5	1	3	1	2	1	2	1	1	2
425.	1257938	3	1	1	1	2	1	1	1	1	2
426.	1258549	9	10	10	10	10	10	10	10	1	4
427.	1258556	5	3	6	1	2	1	1	1	1	2
428.	1266154	8	7	8	2	4	2	5	10	1	4
429.	1272039	1	1	1	1	2	1	2	1	1	2
430.	1276091	2	1	1	1	2	1	2	1	1	2
431.	1276091	1	3	1	1	2	1	2	2	1	2
432.	1276091	5	1	1	3	4	1	3	2	1	2

433.	1277629	5	1	1	1	2	1	2	2	1	2
434.	1293439	3	2	2	3	2	1	1	1	1	2
435.	1293439	6	9	7	5	5	8	4	2	1	2
436.	1294562	10	8	10	1	3	10	5	1	1	4
437.	1295186	10	10	10	1	6	1	2	8	1	4
438.	527337	4	1	1	1	2	1	1	1	1	2
439.	558538	4	1	3	3	2	1	1	1	1	2
440.	566509	5	1	1	1	2	1	1	1	1	2
441.	608157	10	4	3	10	4	10	10	1	1	4
442.	677910	5	2	2	4	2	4	1	1	1	2
443.	734111	1	1	1	3	2	3	1	1	1	2
444.	734111	1	1	1	1	2	2	1	1	1	2
445.	780555	5	1	1	6	3	1	2	1	1	2
446.	827627	2	1	1	1	2	1	1	1	1	2
447.	1049837	1	1	1	1	2	1	1	1	1	2
448.	1058849	5	1	1	1	2	1	1	1	1	2
449.	1182404	1	1	1	1	1	1	1	1	1	2
450.	1193544	5	7	9	8	6	10	8	10	1	4
451.	1201870	4	1	1	3	1	1	2	1	1	2
452.	1202253	5	1	1	1	2	1	1	1	1	2
453.	1227081	3	1	1	3	2	1	1	1	1	2
454.	1230994	4	5	5	8	6	10	10	7	1	4
455.	1238410	2	3	1	1	3	1	1	1	1	2
456.	1246562	10	2	2	1	2	6	1	1	2	4
457.	1257470	10	6	5	8	5	10	8	6	1	4
458.	1259008	8	8	9	6	6	3	10	10	1	4
459.	1266124	5	1	2	1	2	1	1	1	1	2
460.	1267898	5	1	3	1	2	1	1	1	1	2
461.	1268313	5	1	1	3	2	1	1	1	1	2
462.	1268804	3	1	1	1	2	5	1	1	1	2
463.	1276091	6	1	1	3	2	1	1	1	1	2
464.	1280258	4	1	1	1	2	1	1	2	1	2
465.	1293966	4	1	1	1	2	1	1	1	1	2
466.	1296572	10	9	8	7	6	4	7	10	3	4
467.	1298416	10	6	6	2	4	10	9	7	1	4
468.	1299596	6	6	6	5	4	10	7	6	2	4
469.	1105524	4	1	1	1	2	1	1	1	1	2
470.	1181685	1	1	2	1	2	1	2	1	1	2
471.	1211594	3	1	1	1	1	1	2	1	1	2
472.	1238777	6	1	1	3	2	1	1	1	1	2
473.	1257608	6	1	1	1	1	1	1	1	1	2
474.	1269574	4	1	1	1	2	1	1	1	1	2
475.	1277145	5	1	1	1	2	1	1	1	1	2
476.	1287282	3	1	1	1	2	1	1	1	1	2
477.	1296025	4	1	2	1	2	1	1	1	1	2
478.	1296263	4	1	1	1	2	1	1	1	1	2
479.	1296593	5	2	1	1	2	1	1	1	1	2
480.	1299161	4	8	7	10	4	10	7	5	1	4

481.	1301945	5	1	1	1	1	1	1	1	1	2
482.	1302428	5	3	2	4	2	1	1	1	1	2
483.	1318169	9	10	10	10	10	5	10	10	10	4
484.	474162	8	7	8	5	5	10	9	10	1	4
485.	787451	5	1	2	1	2	1	1	1	1	2
486.	1002025	1	1	1	3	1	3	1	1	1	2
487.	1070522	3	1	1	1	1	1	2	1	1	2
488.	1073960	10	10	10	10	6	10	8	1	5	4
489.	1076352	3	6	4	10	3	3	3	4	1	4
490.	1084139	6	3	2	1	3	4	4	1	1	4
491.	1115293	1	1	1	1	2	1	1	1	1	2
492.	1119189	5	8	9	4	3	10	7	1	1	4
493.	1133991	4	1	1	1	1	1	2	1	1	2
494.	1142706	5	10	10	10	6	10	6	5	2	4
495.	1155967	5	1	2	10	4	5	2	1	1	2
496.	1170945	3	1	1	1	1	1	2	1	1	2
497.	1181567	1	1	1	1	1	1	1	1	1	2
498.	1182404	4	2	1	1	2	1	1	1	1	2
499.	1204558	4	1	1	1	2	1	2	1	1	2
500.	1217952	4	1	1	1	2	1	2	1	1	2
501.	1224565	6	1	1	1	2	1	3	1	1	2
502.	1238186	4	1	1	1	2	1	2	1	1	2
503.	1253917	4	1	1	2	2	1	2	1	1	2
504.	1265899	4	1	1	1	2	1	3	1	1	2
505.	1268766	1	1	1	1	2	1	1	1	1	2
506.	1277268	3	3	1	1	2	1	1	1	1	2
507.	1286943	8	10	10	10	7	5	4	8	7	4
508.	1295508	1	1	1	1	2	4	1	1	1	2
509.	1297327	5	1	1	1	2	1	1	1	1	2
510.	1297522	2	1	1	1	2	1	1	1	1	2
511.	1298360	1	1	1	1	2	1	1	1	1	2
512.	1299924	5	1	1	1	2	1	2	1	1	2
513.	1299994	5	1	1	1	2	1	1	1	1	2
514.	1304595	3	1	1	1	1	1	2	1	1	2
515.	1306282	6	6	7	10	3	10	8	10	2	4
516.	1313325	4	10	4	7	3	10	9	10	1	4
517.	1320077	1	1	1	1	1	1	1	1	1	2
518.	1320077	1	1	1	1	1	1	2	1	1	2
519.	1320304	3	1	2	2	2	1	1	1	1	2
520.	1330439	4	7	8	3	4	10	9	1	1	4
521.	333093	1	1	1	1	3	1	1	1	1	2
522.	369565	4	1	1	1	3	1	1	1	1	2
523.	412300	10	4	5	4	3	5	7	3	1	4
524.	672113	7	5	6	10	4	10	5	3	1	4
525.	749653	3	1	1	1	2	1	2	1	1	2
526.	769612	3	1	1	2	2	1	1	1	1	2
527.	769612	4	1	1	1	2	1	1	1	1	2
528.	798429	4	1	1	1	2	1	3	1	1	2

529.	807657	6	1	3	2	2	1	1	1	1	2
530.	8233704	4	1	1	1	1	1	2	1	1	2
531.	837480	7	4	4	3	4	10	6	9	1	4
532.	867392	4	2	2	1	2	1	2	1	1	2
533.	869828	1	1	1	1	1	1	3	1	1	2
534.	1043068	3	1	1	1	2	1	2	1	1	2
535.	1056171	2	1	1	1	2	1	2	1	1	2
536.	1061990	1	1	3	2	2	1	3	1	1	2
537.	1113061	5	1	1	1	2	1	3	1	1	2
538.	1116192	5	1	2	1	2	1	3	1	1	2
539.	1135090	4	1	1	1	2	1	2	1	1	2
540.	1145420	6	1	1	1	2	1	2	1	1	2
541.	1158157	5	1	1	1	2	2	2	1	1	2
542.	1171578	3	1	1	1	2	1	1	1	1	2
543.	1174841	5	3	1	1	2	1	1	1	1	2
544.	1184586	4	1	1	1	2	1	2	1	1	2
545.	1186936	2	1	3	2	2	1	2	1	1	2
546.	1197527	5	1	1	1	2	1	2	1	1	2
547.	1222464	6	10	10	10	4	10	7	10	1	4
548.	1240603	2	1	1	1	1	1	1	1	1	2
549.	1240603	3	1	1	1	1	1	1	1	1	2
550.	1241035	7	8	3	7	4	5	7	8	2	4
551.	1287971	3	1	1	1	2	1	2	1	1	2
552.	1289391	1	1	1	1	2	1	3	1	1	2
553.	1299924	3	2	2	2	2	1	4	2	1	2
554.	1306339	4	4	2	1	2	5	2	1	2	2
555.	1313658	3	1	1	1	2	1	1	1	1	2
556.	1313982	4	3	1	1	2	1	4	8	1	2
557.	1321264	5	2	2	2	1	1	2	1	1	2
558.	1321321	5	1	1	3	2	1	1	1	1	2
559.	1321348	2	1	1	1	2	1	2	1	1	2
560.	1321931	5	1	1	1	2	1	2	1	1	2
561.	1321942	5	1	1	1	2	1	3	1	1	2
562.	1321942	5	1	1	1	2	1	3	1	1	2
563.	1328331	1	1	1	1	2	1	3	1	1	2
564.	1328755	3	1	1	1	2	1	2	1	1	2
565.	1331405	4	1	1	1	2	1	3	2	1	2
566.	1331412	5	7	10	10	5	10	10	10	1	4
567.	1333104	3	1	2	1	2	1	3	1	1	2
568.	1334071	4	1	1	1	2	3	2	1	1	2
569.	1343068	8	4	4	1	6	10	2	5	2	4
570.	1343374	10	10	8	10	6	5	10	3	1	4
571.	1344121	8	10	4	4	8	10	8	2	1	4
572.	142932	7	6	10	5	3	10	9	10	2	4
573.	183936	3	1	1	1	2	1	2	1	1	2
574.	324382	1	1	1	1	2	1	2	1	1	2
575.	378275	10	9	7	3	4	2	7	7	1	4
576.	385103	5	1	2	1	2	1	3	1	1	2

577.	690557	5	1	1	1	2	1	2	1	1	2
578.	695091	1	1	1	1	2	1	2	1	1	2
579.	695219	1	1	1	1	2	1	2	1	1	2
580.	824249	1	1	1	1	2	1	3	1	1	2
581.	871549	5	1	2	1	2	1	2	1	1	2
582.	878358	5	7	10	6	5	10	7	5	1	4
583.	1107684	6	10	5	5	4	10	6	10	1	4
584.	1115762	3	1	1	1	2	1	1	1	1	2
585.	1217717	5	1	1	6	3	1	1	1	1	2
586.	1239420	1	1	1	1	2	1	1	1	1	2
587.	1254538	8	10	10	10	6	10	10	10	1	4
588.	1261751	5	1	1	1	2	1	2	2	1	2
589.	1268275	9	8	8	9	6	3	4	1	1	4
590.	1272166	5	1	1	1	2	1	1	1	1	2
591.	1294261	4	10	8	5	4	1	10	1	1	4
592.	1295529	2	5	7	6	4	10	7	6	1	4
593.	1298484	10	3	4	5	3	10	4	1	1	4
594.	1311875	5	1	2	1	2	1	1	1	1	2
595.	1315506	4	8	6	3	4	10	7	1	1	4
596.	1320141	5	1	1	1	2	1	2	1	1	2
597.	1325309	4	1	2	1	2	1	2	1	1	2
598.	1333063	5	1	3	1	2	1	3	1	1	2
599.	1333495	3	1	1	1	2	1	2	1	1	2
600.	1334659	5	2	4	1	1	1	1	1	1	2
601.	1336798	3	1	1	1	2	1	2	1	1	2
602.	1344449	1	1	1	1	1	1	2	1	1	2
603.	1350568	4	1	1	1	2	1	2	1	1	2
604.	1352663	5	4	6	8	4	1	8	10	1	4
605.	188336	5	3	2	8	5	10	8	1	2	4
606.	352431	10	5	10	3	5	8	7	8	3	4
607.	353098	4	1	1	2	2	1	1	1	1	2
608.	411453	1	1	1	1	2	1	1	1	1	2
609.	557583	5	10	10	10	10	10	10	1	1	4
610.	636375	5	1	1	1	2	1	1	1	1	2
611.	736150	10	4	3	10	3	10	7	1	2	4
612.	803531	5	10	10	10	5	2	8	5	1	4
613.	822829	8	10	10	10	6	10	10	10	10	4
614.	1016634	2	3	1	1	2	1	2	1	1	2
615.	1031608	2	1	1	1	1	1	2	1	1	2
616.	1041043	4	1	3	1	2	1	2	1	1	2
617.	1042252	3	1	1	1	2	1	2	1	1	2
618.	1057067	1	1	1	1	1	?	1	1	1	2
619.	1061990	4	1	1	1	2	1	2	1	1	2
620.	1073836	5	1	1	1	2	1	2	1	1	2
621.	1083817	3	1	1	1	2	1	2	1	1	2
622.	1096352	6	3	3	3	3	2	6	1	1	2
623.	1140597	7	1	2	3	2	1	2	1	1	2
624.	1149548	1	1	1	1	2	1	1	1	1	2

625.	1174009	5	1	1	2	1	1	2	1	1	2
626.	1183596	3	1	3	1	3	4	1	1	1	2
627.	1190386	4	6	6	5	7	6	7	7	3	4
628.	1190546	2	1	1	1	2	5	1	1	1	2
629.	1213273	2	1	1	1	2	1	1	1	1	2
630.	1218982	4	1	1	1	2	1	1	1	1	2
631.	1225382	6	2	3	1	2	1	1	1	1	2
632.	1235807	5	1	1	1	2	1	2	1	1	2
633.	1238777	1	1	1	1	2	1	1	1	1	2
634.	1253955	8	7	4	4	5	3	5	10	1	4
635.	1257366	3	1	1	1	2	1	1	1	1	2
636.	1260659	3	1	4	1	2	1	1	1	1	2
637.	1268952	10	10	7	8	7	1	10	10	3	4
638.	1275807	4	2	4	3	2	2	2	1	1	2
639.	1277792	4	1	1	1	2	1	1	1	1	2
640.	1277792	5	1	1	3	2	1	1	1	1	2
641.	1285722	4	1	1	3	2	1	1	1	1	2
642.	1288608	3	1	1	1	2	1	2	1	1	2
643.	1290203	3	1	1	1	2	1	2	1	1	2
644.	1294413	1	1	1	1	2	1	1	1	1	2
645.	1299596	2	1	1	1	2	1	1	1	1	2
646.	1303489	3	1	1	1	2	1	2	1	1	2
647.	1311033	1	2	2	1	2	1	1	1	1	2
648.	1311108	1	1	1	3	2	1	1	1	1	2
649.	1315807	5	10	10	10	10	2	10	10	10	4
650.	1318671	3	1	1	1	2	1	2	1	1	2
651.	1319609	3	1	1	2	3	4	1	1	1	2
652.	1323477	1	2	1	3	2	1	2	1	1	2
653.	1324572	5	1	1	1	2	1	2	2	1	2
654.	1324681	4	1	1	1	2	1	2	1	1	2
655.	1325159	3	1	1	1	2	1	3	1	1	2
656.	1326892	3	1	1	1	2	1	2	1	1	2
657.	1330361	5	1	1	1	2	1	2	1	1	2
658.	1333877	5	4	5	1	8	1	3	6	1	2
659.	1334015	7	8	8	7	3	10	7	2	3	4
660.	1334667	1	1	1	1	2	1	1	1	1	2
661.	1339781	1	1	1	1	2	1	2	1	1	2
662.	1339781	4	1	1	1	2	1	3	1	1	2
663.	1.3E+07	1	1	3	1	2	1	2	1	1	2
664.	1345452	1	1	3	1	2	1	2	1	1	2
665.	1345593	3	1	1	3	2	1	2	1	1	2
666.	1347749	1	1	1	1	2	1	1	1	1	2
667.	1347943	5	2	2	2	2	1	1	1	2	2
668.	1348851	3	1	1	1	2	1	3	1	1	2
669.	1350319	5	7	4	1	6	1	7	10	3	4
670.	1350423	5	10	10	8	5	5	7	10	1	4
671.	1352848	3	10	7	8	5	8	7	4	1	4
672.	1353092	3	2	1	2	2	1	3	1	1	2

673.	1354840	2	1	1	1	2	1	3	1	1	2
674.	1354840	5	3	2	1	3	1	1	1	1	2
675.	1355260	1	1	1	1	2	1	2	1	1	2
676.	1365075	4	1	4	1	2	1	1	1	1	2
677.	1365328	1	1	2	1	2	1	2	1	1	2
678.	1368267	5	1	1	1	2	1	1	1	1	2
679.	1368273	1	1	1	1	2	1	1	1	1	2
680.	1368882	2	1	1	1	2	1	1	1	1	2
681.	1369821	10	10	10	10	5	10	10	10	7	4
682.	1371026	5	10	10	10	4	10	5	6	3	4
683.	1371920	5	1	1	1	2	1	3	2	1	2
684.	466906	1	1	1	1	2	1	1	1	1	2
685.	466906	1	1	1	1	2	1	1	1	1	2
686.	534555	1	1	1	1	2	1	1	1	1	2
687.	536708	1	1	1	1	2	1	1	1	1	2
688.	566346	3	1	1	1	2	1	2	3	1	2
689.	603148	4	1	1	1	2	1	1	1	1	2
690.	654546	1	1	1	1	2	1	1	1	8	2
691.	654546	1	1	1	3	2	1	1	1	1	2
692.	695091	5	10	10	5	4	5	4	4	1	4
693.	714039	3	1	1	1	2	1	1	1	1	2
694.	763235	3	1	1	1	2	1	2	1	2	2
695.	776715	3	1	1	1	3	2	1	1	1	2
696.	841769	2	1	1	1	2	1	1	1	1	2
697.	888820	5	10	10	3	7	3	8	10	2	4
698.	897471	4	8	6	4	3	4	10	6	1	4
699.	897471	4	8	8	5	4	5	10	4	1	4

LAMPIRAN 2

Pada TABEL 3.3 dan TABEL 3.4. *Dataset* pada kelas kanker jinak dan kelas ganas

No.	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
TABEL 3.3 KELAS JINAK										
1.	5	1	1	1	2	1	3	1	1	2
2.	5	4	4	5	7	10	3	2	1	2
3.	3	1	1	1	2	2	3	1	1	2
4.	6	8	8	1	3	4	3	7	1	2
5.	4	1	1	3	2	1	3	1	1	2
6.	1	1	1	1	2	10	3	1	1	2
7.	2	1	2	1	2	1	3	1	1	2
8.	2	1	1	1	2	1	1	1	5	2
9.	4	2	1	1	2	1	2	1	1	2
10.	1	1	1	1	1	1	3	1	1	2
11.	2	1	1	1	2	1	2	1	1	2
12.	1	1	1	1	2	3	3	1	1	2
13.	4	1	1	1	2	1	2	1	1	2
14.	4	1	1	1	2	1	3	1	1	2
15.	6	1	1	1	2	1	3	1	1	2
16.	3	1	1	1	2	1	2	1	1	2
17.	1	1	1	1	2	1	3	1	1	2
18.	3	2	1	1	1	1	2	1	1	2
19.	5	1	1	1	2	1	2	1	1	2
20.	2	1	1	1	2	1	2	1	1	2
21.	1	1	3	1	2	1	1	1	1	2
22.	3	1	1	1	1	1	2	1	1	2
23.	2	1	1	1	2	1	3	1	1	2
24.	2	1	1	2	2	1	3	1	1	2
25.	3	1	2	1	2	1	2	1	1	2
26.	2	1	1	1	2	1	2	1	1	2
27.	6	2	1	1	1	1	7	1	1	2
28.	6	6	6	9	6	1	7	8	1	2
29.	1	1	1	1	2	1	2	1	2	2
30.	1	1	1	1	2	1	2	1	1	2
31.	4	1	1	3	2	1	3	1	1	2
32.	1	1	1	1	2	2	2	1	1	2
33.	1	1	1	1	2	1	2	1	1	2
34.	4	1	1	1	2	1	3	1	1	2
35.	1	1	1	1	2	1	3	2	1	2
36.	5	1	3	1	2	1	2	1	1	2
37.	1	3	3	2	2	1	7	2	1	2
38.	1	1	2	1	2	2	4	2	1	2
39.	1	1	4	1	2	1	2	1	1	2
40.	5	3	1	2	2	1	2	1	1	2
41.	3	1	1	1	2	3	3	1	1	2
42.	2	1	1	1	3	1	2	1	1	2
43.	2	2	2	1	1	1	7	1	1	2
44.	4	1	1	2	2	1	2	1	1	2

45.	5	2	1	1	2	1	3	1	1	2
46.	3	1	1	1	2	2	7	1	1	2
47.	4	1	1	1	2	1	3	1	1	2
48.	2	1	1	2	3	1	2	1	1	2
49.	1	1	1	1	2	1	3	1	1	2
50.	3	1	1	2	2	1	1	1	1	2
51.	4	1	1	1	2	1	3	1	1	2
52.	1	1	1	1	2	1	2	1	1	2
53.	2	1	1	1	2	1	3	1	1	2
54.	1	1	1	1	2	1	3	1	1	2
55.	2	1	1	2	2	1	1	1	1	2
56.	5	1	1	1	2	1	3	1	1	2
57.	4	1	2	1	2	1	3	1	1	2
58.	1	1	1	1	2	1	2	3	1	2
59.	1	3	1	2	2	2	5	3	2	2
60.	3	3	2	1	2	3	3	1	1	2
61.	1	1	1	1	2	5	1	1	1	2
62.	8	3	3	1	2	2	3	2	1	2
63.	1	1	1	1	4	3	1	1	1	2
64.	3	2	1	1	2	2	3	1	1	2
65.	1	1	2	2	2	1	3	1	1	2
66.	4	2	1	1	2	2	3	1	1	2
67.	1	1	1	1	2	1	2	1	1	2
68.	3	1	1	1	2	1	3	1	1	2
69.	1	1	1	1	10	1	1	1	1	2
70.	5	1	3	1	2	1	2	1	1	2
71.	2	1	1	1	2	1	3	1	1	2
72.	3	1	1	1	2	1	2	2	1	2
73.	3	1	1	1	3	1	2	1	1	2
74.	5	1	1	1	2	2	3	3	1	2
75.	4	1	1	1	2	1	2	1	1	2
76.	3	1	1	1	2	1	1	1	1	2
77.	4	1	2	1	2	1	2	1	1	2
78.	1	1	1	1	1	1	2	1	1	2
79.	3	1	1	1	2	1	1	1	1	2
80.	2	1	1	1	2	1	1	1	1	2
81.	1	1	1	1	2	5	1	1	1	2
82.	2	1	1	1	2	1	2	1	1	2
83.	1	1	3	1	2	1	2	1	1	2
84.	1	1	1	1	3	2	2	1	1	2
85.	3	1	1	3	8	1	5	8	1	2
86.	1	1	1	1	1	1	3	1	1	2
87.	4	1	1	1	1	2	3	1	1	2
88.	1	1	1	1	2	1	1	1	1	2

89.	1	2	2	1	2	1	2	1	1	2
90.	2	1	1	1	2	1	3	1	1	2
91.	1	1	2	1	3	1	1	1	1	2
92.	4	1	1	1	2	1	3	2	1	2
93.	3	1	1	1	2	1	3	1	1	2
94.	1	1	1	2	1	3	1	1	7	2
95.	5	1	1	1	2	1	3	1	1	2
96.	4	1	1	1	2	2	3	2	1	2
97.	3	1	1	1	2	1	3	1	1	2
98.	1	1	1	2	1	1	1	1	1	2
99.	3	1	1	1	2	1	1	1	1	2
100.	1	1	1	1	2	1	3	1	1	2
101.	1	1	1	1	2	1	2	1	1	2
102.	2	1	1	1	2	1	3	1	1	2
103.	4	1	1	1	2	1	3	1	1	2
104.	1	1	1	1	1	1	3	1	1	2
105.	1	1	1	1	2	1	1	1	1	2
106.	6	1	1	1	2	1	3	1	1	2
107.	2	1	1	1	1	1	3	1	1	2
108.	1	2	3	1	2	1	3	1	1	2
109.	5	1	1	1	2	1	2	1	1	2
110.	1	1	1	1	2	1	3	1	1	2
111.	3	1	1	1	2	1	3	1	1	2
112.	4	1	1	1	2	1	3	1	1	2
113.	8	4	4	5	4	7	7	8	2	2
114.	5	1	1	4	2	1	3	1	1	2
115.	1	1	1	1	2	1	1	1	1	2
116.	3	1	1	1	2	1	2	1	1	2
117.	1	1	1	1	2	1	3	1	1	2
118.	5	1	1	1	2	1	3	1	1	2
119.	1	1	1	1	2	1	3	1	1	2
120.	1	1	1	1	1	1	3	1	1	2
121.	1	1	1	1	1	1	3	1	1	2
122.	5	1	1	1	1	1	3	1	1	2
123.	1	1	1	1	2	1	3	1	1	2
124.	1	1	1	1	2	1	2	1	1	2
125.	1	1	1	1	2	1	3	1	1	2
126.	6	1	3	1	2	1	3	1	1	2
127.	1	1	1	2	2	1	3	1	1	2
128.	1	1	1	1	2	1	2	1	1	2
129.	1	1	1	1	1	1	3	1	1	2
130.	8	4	6	3	3	1	4	3	1	2
131.	3	3	2	1	3	1	3	6	1	2
132.	3	1	4	1	2	1	3	1	1	2

133.	5	1	3	3	2	2	2	3	1	2
134.	3	1	1	3	1	1	3	1	1	2
135.	2	1	1	1	2	1	3	1	1	2
136.	1	1	1	1	2	5	5	1	1	2
137.	1	1	1	1	2	1	3	1	1	2
138.	5	1	1	2	2	2	3	1	1	2
139.	4	1	1	1	2	1	3	6	1	2
140.	3	1	1	1	2	1	3	1	1	2
141.	1	2	2	1	2	1	1	1	1	2
142.	6	3	3	5	3	10	3	5	3	2
143.	3	1	1	1	2	1	1	1	1	2
144.	3	1	1	1	2	1	2	1	1	2
145.	3	1	1	1	2	1	3	1	1	2
146.	5	7	7	1	5	8	3	4	1	2
147.	5	1	4	1	2	1	3	2	1	2
148.	1	1	1	1	2	1	3	1	1	2
149.	5	1	1	1	2	1	3	1	1	2
150.	3	1	1	1	2	1	3	2	1	2
151.	3	1	3	1	2	1	2	1	1	2
152.	3	1	1	1	2	1	2	1	1	2
153.	1	1	1	1	2	1	2	1	1	2
154.	1	1	1	1	2	1	3	1	1	2
155.	3	1	1	1	2	1	3	1	1	2
156.	2	1	1	2	2	1	3	1	1	2
157.	3	1	1	1	3	1	2	1	1	2
158.	1	1	1	1	2	1	1	1	1	2
159.	1	1	1	1	2	1	3	1	1	2
160.	1	1	1	1	2	1	2	1	1	2
161.	5	3	4	3	4	5	4	7	1	2
162.	5	4	3	1	2	1	2	3	1	2
163.	8	2	1	1	5	1	1	1	1	2
164.	1	1	1	1	2	1	3	1	1	2
165.	1	1	1	1	2	1	3	1	1	2
166.	1	1	1	1	2	1	3	1	1	2
167.	1	1	1	1	2	1	3	1	1	2
168.	3	1	1	1	2	5	5	1	1	2
169.	2	1	1	1	3	1	2	1	1	2
170.	1	1	1	1	2	1	1	1	1	2
171.	1	1	1	1	2	1	1	1	1	2
172.	1	1	1	1	1	1	2	1	1	2
173.	4	6	5	6	7	1	4	9	1	2
174.	1	1	1	1	5	1	3	1	1	2
175.	4	4	4	4	6	5	7	3	1	2
176.	3	1	1	1	2	1	3	1	1	2

177.	3	1	1	1	2	1	3	1	1	2
178.	1	1	1	1	2	1	3	1	1	2
179.	3	2	2	1	2	1	2	3	1	2
180.	1	1	1	1	2	1	2	1	1	2
181.	5	1	1	1	2	1	3	1	2	2
182.	5	2	2	2	2	1	2	2	1	2
183.	1	1	1	1	2	1	1	1	1	2
184.	1	1	1	1	2	1	3	1	1	2
185.	1	1	1	1	1	1	2	1	1	2
186.	1	1	1	1	2	1	3	1	1	2
187.	2	1	1	1	2	1	1	1	1	2
188.	1	1	1	1	2	1	1	1	1	2
189.	1	1	1	1	2	1	1	1	1	2
190.	5	2	2	2	3	1	1	3	1	2
191.	1	1	1	1	1	1	1	3	1	2
192.	5	1	1	3	2	1	1	1	1	2
193.	2	1	1	1	2	1	3	1	1	2
194.	3	4	5	3	7	3	4	6	1	2
195.	1	1	1	1	2	1	2	1	1	2
196.	4	1	1	1	3	1	2	2	1	2
197.	3	2	2	1	4	3	2	1	1	2
198.	4	4	4	2	2	3	2	1	1	2
199.	2	1	1	1	2	1	3	1	1	2
200.	2	1	1	1	2	1	2	1	1	2
201.	1	1	3	1	2	1	1	1	1	2
202.	1	1	3	1	1	1	2	1	1	2
203.	4	3	2	1	3	1	2	1	1	2
204.	1	1	3	1	2	1	1	1	1	2
205.	4	1	2	1	2	1	2	1	1	2
206.	5	1	1	2	2	1	2	1	1	2
207.	3	1	2	1	2	1	2	1	1	2
208.	1	1	1	1	2	1	1	1	1	2
209.	1	1	1	1	2	1	2	1	1	2
210.	1	1	1	1	1	1	2	1	1	2
211.	3	1	1	4	3	1	2	2	1	2
212.	5	3	4	1	4	1	3	1	1	2
213.	1	1	1	1	2	1	1	1	1	2
214.	3	2	2	2	2	1	3	2	1	2
215.	2	1	1	1	2	1	1	1	1	2
216.	2	1	1	1	2	1	1	1	1	2
217.	3	3	2	2	3	1	1	2	3	2
218.	5	3	3	2	3	1	3	1	1	2
219.	2	1	1	1	2	1	2	2	1	2
220.	5	1	1	1	3	2	2	2	1	2

221.	1	1	1	2	2	1	2	1	1	2
222.	3	1	1	1	2	1	2	1	1	2
223.	1	1	1	1	1	1	1	1	1	2
224.	1	2	3	1	2	1	2	1	1	2
225.	3	1	1	1	2	1	2	1	1	2
226.	3	1	1	1	2	1	3	1	1	2
227.	4	1	1	1	2	1	1	1	1	2
228.	3	2	1	1	2	1	2	2	1	2
229.	1	2	3	1	2	1	1	1	1	2
230.	3	1	1	1	2	1	1	1	1	2
231.	5	3	3	1	2	1	2	1	1	2
232.	3	1	1	1	2	4	1	1	1	2
233.	1	2	1	3	2	1	1	2	1	2
234.	1	1	1	1	2	1	2	1	1	2
235.	4	2	2	1	2	1	2	1	1	2
236.	1	1	1	1	2	1	2	1	1	2
237.	2	3	2	2	2	2	3	1	1	2
238.	3	1	2	1	2	1	2	1	1	2
239.	1	1	1	1	2	1	2	1	1	2
240.	1	1	1	1	1	1	2	1	1	2
241.	5	1	2	1	2	1	3	1	1	2
242.	3	3	2	6	3	3	3	5	1	2
243.	1	1	1	1	2	1	2	1	1	2
244.	5	2	2	2	2	2	3	2	2	2
245.	2	3	1	1	5	1	1	1	1	2
246.	3	2	2	3	2	3	3	1	1	2
247.	4	3	3	1	2	1	3	3	1	2
248.	5	1	3	1	2	1	2	1	1	2
249.	3	1	1	1	2	1	1	1	1	2
250.	5	3	6	1	2	1	1	1	1	2
251.	1	1	1	1	2	1	2	1	1	2
252.	2	1	1	1	2	1	2	1	1	2
253.	1	3	1	1	2	1	2	2	1	2
254.	5	1	1	3	4	1	3	2	1	2
255.	5	1	1	1	2	1	2	2	1	2
256.	3	2	2	3	2	1	1	1	1	2
257.	6	9	7	5	5	8	4	2	1	2
258.	4	1	1	1	2	1	1	1	1	2
259.	4	1	3	3	2	1	1	1	1	2
260.	5	1	1	1	2	1	1	1	1	2
261.	5	2	2	4	2	4	1	1	1	2
262.	1	1	1	3	2	3	1	1	1	2
263.	1	1	1	1	2	2	1	1	1	2
264.	5	1	1	6	3	1	2	1	1	2

265.	2	1	1	1	2	1	1	1	1	2
266.	1	1	1	1	2	1	1	1	1	2
267.	5	1	1	1	2	1	1	1	1	2
268.	1	1	1	1	1	1	1	1	1	2
269.	4	1	1	3	1	1	2	1	1	2
270.	5	1	1	1	2	1	1	1	1	2
271.	3	1	1	3	2	1	1	1	1	2
272.	2	3	1	1	3	1	1	1	1	2
273.	5	1	2	1	2	1	1	1	1	2
274.	5	1	3	1	2	1	1	1	1	2
275.	5	1	1	3	2	1	1	1	1	2
276.	3	1	1	1	2	5	1	1	1	2
277.	6	1	1	3	2	1	1	1	1	2
278.	4	1	1	1	2	1	1	2	1	2
279.	4	1	1	1	2	1	1	1	1	2
280.	4	1	1	1	2	1	1	1	1	2
281.	1	1	2	1	2	1	2	1	1	2
282.	3	1	1	1	1	1	2	1	1	2
283.	6	1	1	3	2	1	1	1	1	2
284.	6	1	1	1	1	1	1	1	1	2
285.	4	1	1	1	2	1	1	1	1	2
286.	5	1	1	1	2	1	1	1	1	2
287.	3	1	1	1	2	1	1	1	1	2
288.	4	1	2	1	2	1	1	1	1	2
289.	4	1	1	1	2	1	1	1	1	2
290.	5	2	1	1	2	1	1	1	1	2
291.	5	1	1	1	1	1	1	1	1	2
292.	5	3	2	4	2	1	1	1	1	2
293.	5	1	2	1	2	1	1	1	1	2
294.	1	1	1	3	1	3	1	1	1	2
295.	3	1	1	1	1	1	2	1	1	2
296.	1	1	1	1	2	1	1	1	1	2
297.	4	1	1	1	1	1	2	1	1	2
298.	5	1	2	10	4	5	2	1	1	2
299.	3	1	1	1	1	1	2	1	1	2
300.	1	1	1	1	1	1	1	1	1	2
301.	4	2	1	1	2	1	1	1	1	2
302.	4	1	1	1	2	1	2	1	1	2
303.	4	1	1	1	2	1	2	1	1	2
304.	6	1	1	1	2	1	3	1	1	2
305.	4	1	1	1	2	1	2	1	1	2
306.	4	1	1	2	2	1	2	1	1	2
307.	4	1	1	1	2	1	3	1	1	2
308.	1	1	1	1	2	1	1	1	1	2

309.	3	3	1	1	2	1	1	1	1	2
310.	1	1	1	1	2	4	1	1	1	2
311.	5	1	1	1	2	1	1	1	1	2
312.	2	1	1	1	2	1	1	1	1	2
313.	1	1	1	1	2	1	1	1	1	2
314.	5	1	1	1	2	1	2	1	1	2
315.	5	1	1	1	2	1	1	1	1	2
316.	3	1	1	1	1	1	2	1	1	2
317.	1	1	1	1	1	1	1	1	1	2
318.	1	1	1	1	1	1	2	1	1	2
319.	3	1	2	2	2	1	1	1	1	2
320.	1	1	1	1	3	1	1	1	1	2
321.	4	1	1	1	3	1	1	1	1	2
322.	3	1	1	1	2	1	2	1	1	2
323.	3	1	1	2	2	1	1	1	1	2
324.	4	1	1	1	2	1	1	1	1	2
325.	4	1	1	1	2	1	3	1	1	2
326.	6	1	3	2	2	1	1	1	1	2
327.	4	1	1	1	1	1	2	1	1	2
328.	4	2	2	1	2	1	2	1	1	2
329.	1	1	1	1	1	1	3	1	1	2
330.	3	1	1	1	2	1	2	1	1	2
331.	2	1	1	1	2	1	2	1	1	2
332.	1	1	3	2	2	1	3	1	1	2
333.	5	1	1	1	2	1	3	1	1	2
334.	5	1	2	1	2	1	3	1	1	2
335.	4	1	1	1	2	1	2	1	1	2
336.	6	1	1	1	2	1	2	1	1	2
337.	5	1	1	1	2	2	2	1	1	2
338.	3	1	1	1	2	1	1	1	1	2
339.	5	3	1	1	2	1	1	1	1	2
340.	4	1	1	1	2	1	2	1	1	2
341.	2	1	3	2	2	1	2	1	1	2
342.	5	1	1	1	2	1	2	1	1	2
343.	2	1	1	1	1	1	1	1	1	2
344.	3	1	1	1	1	1	1	1	1	2
345.	3	1	1	1	2	1	2	1	1	2
346.	1	1	1	1	2	1	3	1	1	2
347.	3	2	2	2	2	1	4	2	1	2
348.	4	4	2	1	2	5	2	1	2	2
349.	3	1	1	1	2	1	1	1	1	2
350.	4	3	1	1	2	1	4	8	1	2
351.	5	2	2	2	1	1	2	1	1	2
352.	5	1	1	3	2	1	1	1	1	2

353.	2	1	1	1	2	1	2	1	1	2
354.	5	1	1	1	2	1	2	1	1	2
355.	5	1	1	1	2	1	3	1	1	2
356.	5	1	1	1	2	1	3	1	1	2
357.	1	1	1	1	2	1	3	1	1	2
358.	3	1	1	1	2	1	2	1	1	2
359.	4	1	1	1	2	1	3	2	1	2
360.	3	1	2	1	2	1	3	1	1	2
361.	4	1	1	1	2	3	2	1	1	2
362.	3	1	1	1	2	1	2	1	1	2
363.	1	1	1	1	2	1	2	1	1	2
364.	5	1	2	1	2	1	3	1	1	2
365.	5	1	1	1	2	1	2	1	1	2
366.	1	1	1	1	2	1	2	1	1	2
367.	1	1	1	1	2	1	2	1	1	2
368.	1	1	1	1	2	1	3	1	1	2
369.	5	1	2	1	2	1	2	1	1	2
370.	3	1	1	1	2	1	1	1	1	2
371.	5	1	1	6	3	1	1	1	1	2
372.	1	1	1	1	2	1	1	1	1	2
373.	5	1	1	1	2	1	2	2	1	2
374.	5	1	1	1	2	1	1	1	1	2
375.	5	1	2	1	2	1	1	1	1	2
376.	5	1	1	1	2	1	2	1	1	2
377.	4	1	2	1	2	1	2	1	1	2
378.	5	1	3	1	2	1	3	1	1	2
379.	3	1	1	1	2	1	2	1	1	2
380.	5	2	4	1	1	1	1	1	1	2
381.	3	1	1	1	2	1	2	1	1	2
382.	1	1	1	1	1	1	2	1	1	2
383.	4	1	1	1	2	1	2	1	1	2
384.	4	1	1	2	2	1	1	1	1	2
385.	1	1	1	1	2	1	1	1	1	2
386.	5	1	1	1	2	1	1	1	1	2
387.	2	3	1	1	2	1	2	1	1	2
388.	2	1	1	1	1	1	2	1	1	2
389.	4	1	3	1	2	1	2	1	1	2
390.	3	1	1	1	2	1	2	1	1	2
391.	1	1	1	1	1	1	1	1	1	2
392.	4	1	1	1	2	1	2	1	1	2
393.	5	1	1	1	2	1	2	1	1	2
394.	3	1	1	1	2	1	2	1	1	2
395.	6	3	3	3	3	2	6	1	1	2
396.	7	1	2	3	2	1	2	1	1	2

397.	1	1	1	1	2	1	1	1	1	2
398.	5	1	1	2	1	1	2	1	1	2
399.	3	1	3	1	3	4	1	1	1	2
400.	2	1	1	1	2	5	1	1	1	2
401.	2	1	1	1	2	1	1	1	1	2
402.	4	1	1	1	2	1	1	1	1	2
403.	6	2	3	1	2	1	1	1	1	2
404.	5	1	1	1	2	1	2	1	1	2
405.	1	1	1	1	2	1	1	1	1	2
406.	3	1	1	1	2	1	1	1	1	2
407.	3	1	4	1	2	1	1	1	1	2
408.	4	2	4	3	2	2	2	1	1	2
409.	4	1	1	1	2	1	1	1	1	2
410.	5	1	1	3	2	1	1	1	1	2
411.	4	1	1	3	2	1	1	1	1	2
412.	3	1	1	1	2	1	2	1	1	2
413.	3	1	1	1	2	1	2	1	1	2
414.	1	1	1	1	2	1	1	1	1	2
415.	2	1	1	1	2	1	1	1	1	2
416.	3	1	1	1	2	1	2	1	1	2
417.	1	2	2	1	2	1	1	1	1	2
418.	1	1	1	3	2	1	1	1	1	2
419.	3	1	1	1	2	1	2	1	1	2
420.	3	1	1	2	3	4	1	1	1	2
421.	1	2	1	3	2	1	2	1	1	2
422.	5	1	1	1	2	1	2	2	1	2
423.	4	1	1	1	2	1	2	1	1	2
424.	3	1	1	1	2	1	3	1	1	2
425.	3	1	1	1	2	1	2	1	1	2
426.	5	1	1	1	2	1	2	1	1	2
427.	5	4	5	1	8	1	3	6	1	2
428.	1	1	1	1	2	1	1	1	1	2
429.	1	1	1	1	2	1	2	1	1	2
430.	4	1	1	1	2	1	3	1	1	2
431.	1	1	3	1	2	1	2	1	1	2
432.	1	1	3	1	2	1	2	1	1	2
433.	3	1	1	3	2	1	2	1	1	2
434.	1	1	1	1	2	1	1	1	1	2
435.	5	2	2	2	2	1	1	1	2	2
436.	3	1	1	1	2	1	3	1	1	2
437.	3	2	1	2	2	1	3	1	1	2
438.	2	1	1	1	2	1	3	1	1	2
439.	5	3	2	1	3	1	1	1	1	2
440.	1	1	1	1	2	1	2	1	1	2

441.	4	1	4	1	2	1	1	1	1	2
442.	1	1	2	1	2	1	2	1	1	2
443.	5	1	1	1	2	1	1	1	1	2
444.	1	1	1	1	2	1	1	1	1	2
445.	2	1	1	1	2	1	1	1	1	2
446.	5	1	1	1	2	1	3	2	1	2
447.	1	1	1	1	2	1	1	1	1	2
448.	1	1	1	1	2	1	1	1	1	2
449.	1	1	1	1	2	1	1	1	1	2
450.	1	1	1	1	2	1	1	1	1	2
451.	3	1	1	1	2	1	2	3	1	2
452.	4	1	1	1	2	1	1	1	1	2
453.	1	1	1	1	2	1	1	1	8	2
454.	1	1	1	3	2	1	1	1	1	2
455.	3	1	1	1	2	1	1	1	1	2
456.	3	1	1	1	2	1	2	1	2	2
457.	3	1	1	1	3	2	1	1	1	2
458.	2	1	1	1	2	1	1	1	1	2
No.	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10

TABEL 3.4. KELAS GANAS

459.	8	10	10	8	7	10	9	7	1	4
460.	5	3	3	3	2	3	4	4	1	4
461.	8	7	5	10	7	9	5	5	4	4
462.	7	4	6	4	6	1	4	3	1	4
463.	10	7	7	6	4	10	4	1	2	4
464.	7	3	2	10	5	10	5	4	4	4
465.	10	5	5	3	6	7	7	10	1	4
466.	8	4	5	1	2	1	7	3	1	4
467.	5	2	3	4	2	7	3	6	1	4
468.	10	7	7	3	8	5	7	4	3	4
469.	10	10	10	8	6	1	8	9	1	4
470.	5	4	4	9	2	10	5	6	1	4
471.	2	5	3	3	6	7	7	5	1	4
472.	10	4	3	1	3	3	6	5	2	4
473.	6	10	10	2	8	10	7	3	3	4
474.	5	6	5	6	10	1	3	1	1	4
475.	10	10	10	4	8	1	8	10	1	4
476.	3	7	7	4	4	9	4	8	1	4
477.	7	8	7	2	4	8	3	8	2	4
478.	9	5	8	1	2	3	2	1	5	4
479.	5	3	3	4	2	4	3	4	1	4
480.	10	3	6	2	3	5	4	10	2	4
481.	5	5	5	8	10	8	7	3	7	4
482.	10	5	5	6	8	8	7	1	1	4
483.	10	6	6	3	4	5	3	6	1	4

484.	8	10	10	1	3	6	3	9	1	4
485.	8	2	4	1	5	1	5	4	4	4
486.	5	2	3	1	6	10	5	1	1	4
487.	9	5	5	2	2	2	5	1	1	4
488.	5	3	5	5	3	3	4	10	1	4
489.	9	10	10	1	10	8	3	3	1	4
490.	6	3	4	1	5	2	3	9	1	4
491.	10	4	2	1	3	2	4	3	10	4
492.	5	3	4	1	8	10	4	9	1	4
493.	8	3	8	3	4	9	8	9	8	4
494.	6	10	2	8	10	2	7	8	10	4
495.	9	4	5	10	6	10	4	8	1	4
496.	10	6	4	1	3	4	3	2	3	4
497.	3	5	7	8	8	9	7	10	7	4
498.	5	10	6	1	10	4	4	10	10	4
499.	3	3	6	4	5	8	4	4	1	4
500.	3	6	6	6	5	10	6	8	3	4
501.	9	6	9	2	10	6	2	9	10	4
502.	7	5	6	10	5	10	7	9	4	4
503.	10	3	5	1	10	5	3	10	2	4
504.	2	3	4	4	2	5	2	5	1	4
505.	8	2	3	1	6	3	7	1	1	4
506.	10	10	10	10	10	1	8	8	8	4
507.	7	3	4	4	3	3	3	2	7	4
508.	10	10	10	8	2	10	4	1	1	4
509.	1	6	8	10	8	10	5	7	1	4
510.	6	5	4	4	3	9	7	8	3	4
511.	8	6	4	3	5	9	3	1	1	4
512.	10	3	3	10	2	10	7	3	3	4
513.	10	10	10	3	10	8	8	1	1	4
514.	4	5	5	10	4	10	7	5	8	4
515.	10	10	10	2	10	10	5	3	3	4
516.	5	3	5	1	8	10	5	3	1	4
517.	5	4	6	7	9	7	8	10	1	4
518.	7	5	3	7	4	10	7	5	5	4
519.	8	3	5	4	5	10	1	6	2	4
520.	5	10	8	10	8	10	3	6	3	4
521.	9	5	5	4	4	5	4	3	3	4
522.	3	4	5	2	6	8	4	1	1	4
523.	8	8	7	4	10	10	7	8	7	4
524.	7	2	4	1	6	10	5	4	3	4
525.	10	10	8	6	4	5	8	10	1	4
526.	5	5	5	6	3	10	3	1	1	4
527.	9	9	10	3	6	10	7	10	6	4

528.	10	7	7	4	5	10	5	7	2	4
529.	5	6	7	8	8	10	3	10	3	4
530.	10	8	10	10	6	1	3	1	10	4
531.	6	10	10	10	8	10	10	10	7	4
532.	8	6	5	4	3	10	6	1	1	4
533.	5	8	7	7	10	10	5	7	1	4
534.	5	10	10	3	8	1	5	10	3	4
535.	5	3	3	3	6	10	3	1	1	4
536.	5	8	8	8	5	10	7	8	1	4
537.	8	7	6	4	4	10	5	1	1	4
538.	1	5	8	6	5	8	7	10	1	4
539.	10	5	6	10	6	10	7	7	10	4
540.	5	8	4	10	5	8	9	10	1	4
541.	10	10	10	8	6	8	7	10	1	4
542.	7	5	10	10	10	10	4	10	3	4
543.	9	7	7	5	5	10	7	8	3	4
544.	10	8	8	4	10	10	8	1	1	4
545.	5	10	10	9	6	10	7	10	5	4
546.	10	10	9	3	7	5	3	5	1	4
547.	8	10	10	10	5	10	8	10	6	4
548.	8	10	8	8	4	8	7	7	1	4
549.	10	10	10	10	7	10	7	10	4	4
550.	10	10	10	10	3	10	10	6	1	4
551.	8	7	8	7	5	5	5	10	2	4
552.	6	10	7	7	6	4	8	10	2	4
553.	10	6	4	3	10	10	9	10	1	4
554.	4	1	1	3	1	5	2	1	1	4
555.	7	5	6	3	3	8	7	4	1	4
556.	10	5	5	6	3	10	7	9	2	4
557.	10	5	7	4	4	10	8	9	1	4
558.	8	9	9	5	3	5	7	7	1	4
559.	10	10	10	3	10	10	9	10	1	4
560.	7	4	7	4	3	7	7	6	1	4
561.	6	8	7	5	6	8	8	9	2	4
562.	10	4	5	5	5	10	4	1	1	4
563.	10	8	8	2	8	10	4	8	10	4
564.	9	8	8	5	6	2	4	10	4	4
565.	8	10	10	8	6	9	3	10	10	4
566.	10	4	3	2	3	10	5	3	2	4
567.	8	10	10	8	5	10	7	8	1	4
568.	8	4	4	1	2	9	3	3	1	4
569.	10	4	4	10	2	10	5	3	3	4
570.	6	10	10	2	8	10	7	3	3	4
571.	9	10	10	1	10	8	3	3	1	4

572.	5	6	6	2	4	10	3	6	1	4
573.	10	5	8	10	3	10	5	1	3	4
574.	5	10	10	6	10	10	10	6	5	4
575.	8	8	9	4	5	10	7	8	1	4
576.	10	4	4	10	6	10	5	5	1	4
577.	7	9	4	10	10	3	5	3	3	4
578.	10	10	6	3	3	10	4	3	2	4
579.	3	3	5	2	3	10	7	1	1	4
580.	10	8	8	2	3	4	8	7	8	4
581.	8	4	7	1	3	10	3	9	2	4
582.	3	3	5	2	3	10	7	1	1	4
583.	7	2	4	1	3	4	3	3	1	4
584.	10	5	7	3	3	7	3	3	8	4
585.	1	4	3	10	4	10	5	6	1	4
586.	10	4	6	1	2	10	5	3	1	4
587.	7	4	5	10	2	10	3	8	2	4
588.	8	10	10	10	8	10	10	7	3	4
589.	10	10	10	10	10	10	4	10	10	4
590.	6	1	3	1	4	5	5	10	1	4
591.	5	6	6	8	6	10	4	10	4	4
592.	8	8	8	1	2	1	6	10	1	4
593.	10	4	4	6	2	10	2	3	1	4
594.	5	5	7	8	6	10	7	4	1	4
595.	9	1	2	6	4	10	7	7	2	4
596.	8	4	10	5	4	4	7	10	1	4
597.	10	10	10	7	9	10	7	10	10	4
598.	8	3	4	9	3	10	3	3	1	4
599.	10	8	4	4	4	10	3	10	4	4
600.	7	8	7	6	4	3	8	8	4	4
601.	8	6	4	10	10	1	3	5	1	4
602.	5	5	5	2	5	10	4	3	1	4
603.	6	8	7	8	6	8	8	9	1	4
604.	7	6	3	2	5	10	7	4	6	4
605.	5	4	6	10	2	10	4	1	1	4
606.	10	1	1	1	2	10	5	4	1	4
607.	8	10	3	2	6	4	3	10	1	4
608.	10	4	6	4	5	10	7	1	1	4
609.	10	4	7	2	2	8	6	1	1	4
610.	5	4	6	6	4	10	4	3	1	4
611.	8	6	7	3	3	10	3	4	2	4
612.	6	5	5	8	4	10	3	4	1	4
613.	8	5	5	5	2	10	4	3	1	4
614.	10	3	3	1	2	10	7	6	1	4
615.	7	6	4	8	10	10	9	5	3	4

616.	3	4	4	10	5	1	3	3	1	4
617.	4	2	3	5	3	8	7	6	1	4
618.	2	7	10	10	7	10	4	9	4	4
619.	5	3	3	1	3	3	3	3	3	4
620.	8	10	10	7	10	10	7	3	8	4
621.	8	10	5	3	8	4	4	10	3	4
622.	10	3	5	4	3	7	3	5	3	4
623.	6	10	10	10	10	10	8	10	10	4
624.	3	10	3	10	6	10	5	1	4	4
625.	6	10	10	10	8	10	7	10	7	4
626.	5	8	8	10	5	10	8	10	3	4
627.	10	6	3	6	4	10	7	8	4	4
628.	7	6	6	3	2	10	7	1	1	4
629.	10	8	7	4	3	10	7	9	1	4
630.	3	10	8	7	6	9	9	3	8	4
631.	10	10	10	6	8	4	8	5	1	4
632.	8	5	6	2	3	10	6	6	1	4
633.	8	7	8	5	10	10	7	2	1	4
634.	10	10	10	7	10	10	8	2	1	4
635.	9	10	10	10	10	10	10	10	1	4
636.	8	7	8	2	4	2	5	10	1	4
637.	10	8	10	1	3	10	5	1	1	4
638.	10	10	10	1	6	1	2	8	1	4
639.	10	4	3	10	4	10	10	1	1	4
640.	5	7	9	8	6	10	8	10	1	4
641.	4	5	5	8	6	10	10	7	1	4
642.	10	2	2	1	2	6	1	1	2	4
643.	10	6	5	8	5	10	8	6	1	4
644.	8	8	9	6	6	3	10	10	1	4
645.	10	9	8	7	6	4	7	10	3	4
646.	10	6	6	2	4	10	9	7	1	4
647.	6	6	6	5	4	10	7	6	2	4
648.	4	8	7	10	4	10	7	5	1	4
649.	9	10	10	10	10	5	10	10	10	4
650.	8	7	8	5	5	10	9	10	1	4
651.	10	10	10	10	6	10	8	1	5	4
652.	3	6	4	10	3	3	3	4	1	4
653.	6	3	2	1	3	4	4	1	1	4
654.	5	8	9	4	3	10	7	1	1	4
655.	5	10	10	10	6	10	6	5	2	4
656.	8	10	10	10	7	5	4	8	7	4
657.	6	6	7	10	3	10	8	10	2	4
658.	4	10	4	7	3	10	9	10	1	4
659.	4	7	8	3	4	10	9	1	1	4

660.	10	4	5	4	3	5	7	3	1	4
661.	7	5	6	10	4	10	5	3	1	4
662.	7	4	4	3	4	10	6	9	1	4
663.	6	10	10	10	4	10	7	10	1	4
664.	7	8	3	7	4	5	7	8	2	4
665.	5	7	10	10	5	10	10	10	1	4
666.	8	4	4	1	6	10	2	5	2	4
667.	10	10	8	10	6	5	10	3	1	4
668.	8	10	4	4	8	10	8	2	1	4
669.	7	6	10	5	3	10	9	10	2	4
670.	10	9	7	3	4	2	7	7	1	4
671.	5	7	10	6	5	10	7	5	1	4
672.	6	10	5	5	4	10	6	10	1	4
673.	8	10	10	10	6	10	10	10	1	4
674.	9	8	8	9	6	3	4	1	1	4
675.	4	10	8	5	4	1	10	1	1	4
676.	2	5	7	6	4	10	7	6	1	4
677.	10	3	4	5	3	10	4	1	1	4
678.	4	8	6	3	4	10	7	1	1	4
679.	5	4	6	8	4	1	8	10	1	4
680.	5	3	2	8	5	10	8	1	2	4
681.	10	5	10	3	5	8	7	8	3	4
682.	5	10	10	10	10	10	10	1	1	4
683.	10	4	3	10	3	10	7	1	2	4
684.	5	10	10	10	5	2	8	5	1	4
685.	8	10	10	10	6	10	10	10	10	4
686.	4	6	6	5	7	6	7	7	3	4
687.	8	7	4	4	5	3	5	10	1	4
688.	10	10	7	8	7	1	10	10	3	4
689.	5	10	10	10	10	2	10	10	10	4
690.	7	8	8	7	3	10	7	2	3	4
691.	5	7	4	1	6	1	7	10	3	4
692.	5	10	10	8	5	5	7	10	1	4
693.	3	10	7	8	5	8	7	4	1	4
694.	10	10	10	10	5	10	10	10	7	4
695.	5	10	10	10	4	10	5	6	3	4
696.	5	10	10	5	4	5	4	4	1	4
697.	5	10	10	3	7	3	8	10	2	4
698.	4	8	6	4	3	4	10	6	1	4
699.	4	8	8	5	4	5	10	4	1	4

LAMPIRAN 3

TABEL 3.5 dan TABEL 3.6 Normalisasi data pada kelas kanker jinak dan kelas ganas

No.	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
TABEL 3.5 KELAS JINAK										
1.	0.444444	0	0	0	0.111111	0	0.222222	0	0	0
2.	0.444444	0.333333	0.333333	0.444444	0.666667	1	0.222222	0.111111	0	0
3.	0.222222	0	0	0	0.111111	0.111111	0.222222	0	0	0
4.	0.555556	0.777778	0.777778	0	0.222222	0.333333	0.222222	0.666667	0	0
5.	0.333333	0	0	0.222222	0.111111	0	0.222222	0	0	0
6.	0	0	0	0	0.111111	1	0.222222	0	0	0
7.	0.111111	0	0.111111	0	0.111111	0	0.222222	0	0	0
8.	0.111111	0	0	0	0.111111	0	0	0	0.444444	0
9.	0.333333	0.111111	0	0	0.111111	0	0.111111	0	0	0
10.	0	0	0	0	0	0	0.222222	0	0	0
11.	0.111111	0	0	0	0.111111	0	0.111111	0	0	0
12.	0	0	0	0	0.111111	0.222222	0.222222	0	0	0
13.	0.333333	0	0	0	0.111111	0	0.111111	0	0	0
14.	0.333333	0	0	0	0.111111	0	0.222222	0	0	0
15.	0.555556	0	0	0	0.111111	0	0.222222	0	0	0
16.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0
17.	0	0	0	0	0.111111	0	0.222222	0	0	0
18.	0.222222	0.111111	0	0	0	0	0.111111	0	0	0
19.	0.444444	0	0	0	0.111111	0	0.111111	0	0	0
20.	0.111111	0	0	0	0.111111	0	0.111111	0	0	0
21.	0	0	0.222222	0	0.111111	0	0	0	0	0
22.	0.222222	0	0	0	0	0	0.111111	0	0	0
23.	0.111111	0	0	0	0.111111	0	0.222222	0	0	0
24.	0.111111	0	0	0.111111	0.111111	0	0.222222	0	0	0
25.	0.222222	0	0.111111	0	0.111111	0	0.111111	0	0	0
26.	0.111111	0	0	0	0.111111	0	0.111111	0	0	0
27.	0.555556	0.111111	0	0	0	0	0.666667	0	0	0
28.	0.555556	0.555556	0.555556	0.888889	0.555556	0	0.666667	0.777778	0	0
29.	0	0	0	0	0.111111	0	0.111111	0	0.111111	0
30.	0	0	0	0	0.111111	0	0.111111	0	0	0
31.	0.333333	0	0	0.222222	0.111111	0	0.222222	0	0	0
32.	0	0	0	0	0.111111	0.111111	0.111111	0	0	0
33.	0	0	0	0	0.111111	0	0.111111	0	0	0
34.	0.333333	0	0	0	0.111111	0	0.222222	0	0	0
35.	0	0	0	0	0.111111	0	0.222222	0.111111	0	0
36.	0.444444	0	0.222222	0	0.111111	0	0.111111	0	0	0
37.	0	0.222222	0.222222	0.111111	0.111111	0	0.666667	0.111111	0	0
38.	0	0	0.111111	0	0.111111	0.111111	0.333333	0.111111	0	0
39.	0	0	0.333333	0	0.111111	0	0.111111	0	0	0
40.	0.444444	0.222222	0	0.111111	0.111111	0	0.111111	0	0	0
41.	0.222222	0	0	0	0.111111	0.222222	0.222222	0	0	0
42.	0.111111	0	0	0	0.222222	0	0.111111	0	0	0
43.	0.111111	0.111111	0.111111	0	0	0	0.666667	0	0	0
44.	0.333333	0	0	0.111111	0.111111	0	0.111111	0	0	0

45.	0.444444	0.111111	0	0	0.111111	0	0.222222	0	0	0	0
46.	0.222222	0	0	0	0.111111	0.111111	0.666667	0	0	0	0
47.	0.333333	0	0	0	0.111111	0	0.222222	0	0	0	0
48.	0.111111	0	0	0.111111	0.222222	0	0.111111	0	0	0	0
49.	0	0	0	0	0.111111	0	0.222222	0	0	0	0
50.	0.222222	0	0	0.111111	0.111111	0	0	0	0	0	0
51.	0.333333	0	0	0	0.111111	0	0.222222	0	0	0	0
52.	0	0	0	0	0.111111	0	0.111111	0	0	0	0
53.	0.111111	0	0	0	0.111111	0	0.222222	0	0	0	0
54.	0	0	0	0	0.111111	0	0.222222	0	0	0	0
55.	0.111111	0	0	0.111111	0.111111	0	0	0	0	0	0
56.	0.444444	0	0	0	0.111111	0	0.222222	0	0	0	0
57.	0.333333	0	0.111111	0	0.111111	0	0.222222	0	0	0	0
58.	0	0	0	0	0.111111	0	0.111111	0.222222	0	0	0
59.	0	0.222222	0	0.111111	0.111111	0.111111	0.444444	0.222222	0.111111	0	0
60.	0.222222	0.222222	0.111111	0	0.111111	0.222222	0.222222	0	0	0	0
61.	0	0	0	0	0.111111	0.444444	0	0	0	0	0
62.	0.777778	0.222222	0.222222	0	0.111111	0.111111	0.222222	0.111111	0	0	0
63.	0	0	0	0	0.333333	0.222222	0	0	0	0	0
64.	0.222222	0.111111	0	0	0.111111	0.111111	0.222222	0	0	0	0
65.	0	0	0.111111	0.111111	0.111111	0	0.222222	0	0	0	0
66.	0.333333	0.111111	0	0	0.111111	0.111111	0.222222	0	0	0	0
67.	0	0	0	0	0.111111	0	0.111111	0	0	0	0
68.	0.222222	0	0	0	0.111111	0	0.222222	0	0	0	0
69.	0	0	0	0	1	0	0	0	0	0	0
70.	0.444444	0	0.222222	0	0.111111	0	0.111111	0	0	0	0
71.	0.111111	0	0	0	0.111111	0	0.222222	0	0	0	0
72.	0.222222	0	0	0	0.111111	0	0.111111	0.111111	0	0	0
73.	0.222222	0	0	0	0.222222	0	0.111111	0	0	0	0
74.	0.444444	0	0	0	0.111111	0.111111	0.222222	0.222222	0	0	0
75.	0.333333	0	0	0	0.111111	0	0.111111	0	0	0	0
76.	0.222222	0	0	0	0.111111	0	0	0	0	0	0
77.	0.333333	0	0.111111	0	0.111111	0	0.111111	0	0	0	0
78.	0	0	0	0	0	0	0.111111	0	0	0	0
79.	0.222222	0	0	0	0.111111	0	0	0	0	0	0
80.	0.111111	0	0	0	0.111111	0	0	0	0	0	0
81.	0	0	0	0	0.111111	0.444444	0	0	0	0	0
82.	0.111111	0	0	0	0.111111	0	0.111111	0	0	0	0
83.	0	0	0.222222	0	0.111111	0	0.111111	0	0	0	0
84.	0	0	0	0	0.222222	0.111111	0.111111	0	0	0	0
85.	0.222222	0	0	0.222222	0.777778	0	0.444444	0.777778	0	0	0
86.	0	0	0	0	0	0	0.222222	0	0	0	0
87.	0.333333	0	0	0	0.111111	0.222222	0	0	0	0	0
88.	0	0	0	0	0.111111	0	0	0	0	0	0

89.	0	0.111111	0.111111	0	0.111111	0	0.111111	0	0	0	0
90.	0.111111	0	0	0	0.111111	0	0.222222	0	0	0	0
91.	0	0	0.111111	0	0.222222	0	0	0	0	0	0
92.	0.333333	0	0	0	0.111111	0	0.222222	0.111111	0	0	0
93.	0.222222	0	0	0	0.111111	0	0.222222	0	0	0	0
94.	0	0	0	0.111111	0	0.222222	0	0	0.666667	0	0
95.	0.444444	0	0	0	0.111111	0	0.222222	0	0	0	0
96.	0.333333	0	0	0	0.111111	0.111111	0.222222	0.111111	0	0	0
97.	0.222222	0	0	0	0.111111	0	0.222222	0	0	0	0
98.	0	0	0	0.111111	0	0	0	0	0	0	0
99.	0.222222	0	0	0	0.111111	0	0	0	0	0	0
100.	0	0	0	0	0.111111	0	0.222222	0	0	0	0
101.	0	0	0	0	0.111111	0	0.111111	0	0	0	0
102.	0.111111	0	0	0	0.111111	0	0.222222	0	0	0	0
103.	0.333333	0	0	0	0.111111	0	0.222222	0	0	0	0
104.	0	0	0	0	0	0	0.222222	0	0	0	0
105.	0	0	0	0	0.111111	0	0	0	0	0	0
106.	0.555556	0	0	0	0.111111	0	0.222222	0	0	0	0
107.	0.111111	0	0	0	0	0	0.222222	0	0	0	0
108.	0	0.111111	0.222222	0	0.111111	0	0.222222	0	0	0	0
109.	0.444444	0	0	0	0.111111	0	0.111111	0	0	0	0
110.	0	0	0	0	0.111111	0	0.222222	0	0	0	0
111.	0.222222	0	0	0	0.111111	0	0.222222	0	0	0	0
112.	0.333333	0	0	0	0.111111	0	0.222222	0	0	0	0
113.	0.777778	0.333333	0.333333	0.444444	0.333333	0.666667	0.666667	0.777778	0.111111	0	0
114.	0.444444	0	0	0.333333	0.111111	0	0.222222	0	0	0	0
115.	0	0	0	0	0.111111	0	0	0	0	0	0
116.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0	0
117.	0	0	0	0	0.111111	0	0.222222	0	0	0	0
118.	0.444444	0	0	0	0.111111	0	0.222222	0	0	0	0
119.	0	0	0	0	0.111111	0	0.222222	0	0	0	0
120.	0	0	0	0	0	0	0.222222	0	0	0	0
121.	0	0	0	0	0	0	0.222222	0	0	0	0
122.	0.444444	0	0	0	0	0	0.222222	0	0	0	0
123.	0	0	0	0	0.111111	0	0.222222	0	0	0	0
124.	0	0	0	0	0.111111	0	0.111111	0	0	0	0
125.	0	0	0	0	0.111111	0	0.222222	0	0	0	0
126.	0.555556	0	0.222222	0	0.111111	0	0.222222	0	0	0	0
127.	0	0	0	0.111111	0.111111	0	0.222222	0	0	0	0
128.	0	0	0	0	0.111111	0	0.111111	0	0	0	0
129.	0	0	0	0	0	0	0.222222	0	0	0	0
130.	0.777778	0.333333	0.555556	0.222222	0.222222	0	0.333333	0.222222	0	0	0
131.	0.222222	0.222222	0.111111	0	0.222222	0	0.222222	0.555556	0	0	0
132.	0.222222	0	0.333333	0	0.111111	0	0.222222	0	0	0	0

133.	0.444444	0	0.222222	0.222222	0.111111	0.111111	0.111111	0.222222	0	0
134.	0.222222	0	0	0.222222	0	0	0.222222	0	0	0
135.	0.111111	0	0	0	0.111111	0	0.222222	0	0	0
136.	0	0	0	0	0.111111	0.444444	0.444444	0	0	0
137.	0	0	0	0	0.111111	0	0.222222	0	0	0
138.	0.444444	0	0	0.111111	0.111111	0.111111	0.222222	0	0	0
139.	0.333333	0	0	0	0.111111	0	0.222222	0.555556	0	0
140.	0.222222	0	0	0	0.111111	0	0.222222	0	0	0
141.	0	0.111111	0.111111	0	0.111111	0	0	0	0	0
142.	0.555556	0.222222	0.222222	0.444444	0.222222	1	0.222222	0.444444	0.222222	0
143.	0.222222	0	0	0	0.111111	0	0	0	0	0
144.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0
145.	0.222222	0	0	0	0.111111	0	0.222222	0	0	0
146.	0.444444	0.666667	0.666667	0	0.444444	0.777778	0.222222	0.333333	0	0
147.	0.444444	0	0.333333	0	0.111111	0	0.222222	0.111111	0	0
148.	0	0	0	0	0.111111	0	0.222222	0	0	0
149.	0.444444	0	0	0	0.111111	0	0.222222	0	0	0
150.	0.222222	0	0	0	0.111111	0	0.222222	0.111111	0	0
151.	0.222222	0	0.222222	0	0.111111	0	0.111111	0	0	0
152.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0
153.	0	0	0	0	0.111111	0	0.111111	0	0	0
154.	0	0	0	0	0.111111	0	0.222222	0	0	0
155.	0.222222	0	0	0	0.111111	0	0.222222	0	0	0
156.	0.111111	0	0	0.111111	0.111111	0	0.222222	0	0	0
157.	0.222222	0	0	0	0.222222	0	0.111111	0	0	0
158.	0	0	0	0	0.111111	0	0	0	0	0
159.	0	0	0	0	0.111111	0	0.222222	0	0	0
160.	0	0	0	0	0.111111	0	0.111111	0	0	0
161.	0.444444	0.222222	0.333333	0.222222	0.333333	0.444444	0.333333	0.666667	0	0
162.	0.444444	0.333333	0.222222	0	0.111111	0	0.111111	0.222222	0	0
163.	0.777778	0.111111	0	0	0.444444	0	0	0	0	0
164.	0	0	0	0	0.111111	0	0.222222	0	0	0
165.	0	0	0	0	0.111111	0	0.222222	0	0	0
166.	0	0	0	0	0.111111	0	0.222222	0	0	0
167.	0	0	0	0	0.111111	0	0.222222	0	0	0
168.	0.222222	0	0	0	0.111111	0.444444	0.444444	0	0	0
169.	0.111111	0	0	0	0.222222	0	0.111111	0	0	0
170.	0	0	0	0	0.111111	0	0	0	0	0
171.	0	0	0	0	0.111111	0	0	0	0	0
172.	0	0	0	0	0	0	0.111111	0	0	0
173.	0.333333	0.555556	0.444444	0.555556	0.666667	0	0.333333	0.888889	0	0
174.	0	0	0	0	0.444444	0	0.222222	0	0	0
175.	0.333333	0.333333	0.333333	0.333333	0.555556	0.444444	0.666667	0.222222	0	0
176.	0.222222	0	0	0	0.111111	0	0.222222	0	0	0

177.	0.222222	0	0	0	0.111111	0	0.222222	0	0	0
178.	0	0	0	0	0.111111	0	0.222222	0	0	0
179.	0.222222	0.111111	0.111111	0	0.111111	0	0.111111	0.222222	0	0
180.	0	0	0	0	0.111111	0	0.111111	0	0	0
181.	0.444444	0	0	0	0.111111	0	0.222222	0	0.111111	0
182.	0.444444	0.111111	0.111111	0.111111	0.111111	0	0.111111	0.111111	0	0
183.	0	0	0	0	0.111111	0	0	0	0	0
184.	0	0	0	0	0.111111	0	0.222222	0	0	0
185.	0	0	0	0	0	0	0.111111	0	0	0
186.	0	0	0	0	0.111111	0	0.222222	0	0	0
187.	0.111111	0	0	0	0.111111	0	0	0	0	0
188.	0	0	0	0	0.111111	0	0	0	0	0
189.	0	0	0	0	0.111111	0	0	0	0	0
190.	0.444444	0.111111	0.111111	0.111111	0.222222	0	0	0.222222	0	0
191.	0	0	0	0	0	0	0	0.222222	0	0
192.	0.444444	0	0	0.222222	0.111111	0	0	0	0	0
193.	0.111111	0	0	0	0.111111	0	0.222222	0	0	0
194.	0.222222	0.333333	0.444444	0.222222	0.666667	0.222222	0.333333	0.555556	0	0
195.	0	0	0	0	0.111111	0	0.111111	0	0	0
196.	0.333333	0	0	0	0.222222	0	0.111111	0.111111	0	0
197.	0.222222	0.111111	0.111111	0	0.333333	0.222222	0.111111	0	0	0
198.	0.333333	0.333333	0.333333	0.111111	0.111111	0.222222	0.111111	0	0	0
199.	0.111111	0	0	0	0.111111	0	0.222222	0	0	0
200.	0.111111	0	0	0	0.111111	0	0.111111	0	0	0
201.	0	0	0.222222	0	0.111111	0	0	0	0	0
202.	0	0	0.222222	0	0	0	0.111111	0	0	0
203.	0.333333	0.222222	0.111111	0	0.222222	0	0.111111	0	0	0
204.	0	0	0.222222	0	0.111111	0	0	0	0	0
205.	0.333333	0	0.111111	0	0.111111	0	0.111111	0	0	0
206.	0.444444	0	0	0.111111	0.111111	0	0.111111	0	0	0
207.	0.222222	0	0.111111	0	0.111111	0	0.111111	0	0	0
208.	0	0	0	0	0.111111	0	0	0	0	0
209.	0	0	0	0	0.111111	0	0.111111	0	0	0
210.	0	0	0	0	0	0	0.111111	0	0	0
211.	0.222222	0	0	0.333333	0.222222	0	0.111111	0.111111	0	0
212.	0.444444	0.222222	0.333333	0	0.333333	0	0.222222	0	0	0
213.	0	0	0	0	0.111111	0	0	0	0	0
214.	0.222222	0.111111	0.111111	0.111111	0.111111	0	0.222222	0.111111	0	0
215.	0.111111	0	0	0	0.111111	0	0	0	0	0
216.	0.111111	0	0	0	0.111111	0	0	0	0	0
217.	0.222222	0.222222	0.111111	0.111111	0.222222	0	0	0.111111	0.222222	0
218.	0.444444	0.222222	0.222222	0.111111	0.222222	0	0.222222	0	0	0
219.	0.111111	0	0	0	0.111111	0	0.111111	0.111111	0	0
220.	0.444444	0	0	0	0.222222	0.111111	0.111111	0.111111	0	0

221.	0	0	0	0.111111	0.111111	0	0.111111	0	0	0	0
222.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0	0
223.	0	0	0	0	0	0	0	0	0	0	0
224.	0	0.111111	0.222222	0	0.111111	0	0.111111	0	0	0	0
225.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0	0
226.	0.222222	0	0	0	0.111111	0	0.222222	0	0	0	0
227.	0.333333	0	0	0	0.111111	0	0	0	0	0	0
228.	0.222222	0.111111	0	0	0.111111	0	0.111111	0.111111	0	0	0
229.	0	0.111111	0.222222	0	0.111111	0	0	0	0	0	0
230.	0.222222	0	0	0	0.111111	0	0	0	0	0	0
231.	0.444444	0.222222	0.222222	0	0.111111	0	0.111111	0	0	0	0
232.	0.222222	0	0	0	0.111111	0.333333	0	0	0	0	0
233.	0	0.111111	0	0.222222	0.111111	0	0	0.111111	0	0	0
234.	0	0	0	0	0.111111	0	0.111111	0	0	0	0
235.	0.333333	0.111111	0.111111	0	0.111111	0	0.111111	0	0	0	0
236.	0	0	0	0	0.111111	0	0.111111	0	0	0	0
237.	0.111111	0.222222	0.111111	0.111111	0.111111	0.111111	0.222222	0	0	0	0
238.	0.222222	0	0.111111	0	0.111111	0	0.111111	0	0	0	0
239.	0	0	0	0	0.111111	0	0.111111	0	0	0	0
240.	0	0	0	0	0	0	0.111111	0	0	0	0
241.	0.444444	0	0.111111	0	0.111111	0	0.222222	0	0	0	0
242.	0.222222	0.222222	0.111111	0.555556	0.222222	0.222222	0.222222	0.444444	0	0	0
243.	0	0	0	0	0.111111	0	0.111111	0	0	0	0
244.	0.444444	0.111111	0.111111	0.111111	0.111111	0.111111	0.222222	0.111111	0.111111	0	0
245.	0.111111	0.222222	0	0	0.444444	0	0	0	0	0	0
246.	0.222222	0.111111	0.111111	0.222222	0.111111	0.222222	0.222222	0	0	0	0
247.	0.333333	0.222222	0.222222	0	0.111111	0	0.222222	0.222222	0	0	0
248.	0.444444	0	0.222222	0	0.111111	0	0.111111	0	0	0	0
249.	0.222222	0	0	0	0.111111	0	0	0	0	0	0
250.	0.444444	0.222222	0.555556	0	0.111111	0	0	0	0	0	0
251.	0	0	0	0	0.111111	0	0.111111	0	0	0	0
252.	0.111111	0	0	0	0.111111	0	0.111111	0	0	0	0
253.	0	0.222222	0	0	0.111111	0	0.111111	0.111111	0	0	0
254.	0.444444	0	0	0.222222	0.333333	0	0.222222	0.111111	0	0	0
255.	0.444444	0	0	0	0.111111	0	0.111111	0.111111	0	0	0
256.	0.222222	0.111111	0.111111	0.222222	0.111111	0	0	0	0	0	0
257.	0.555556	0.888889	0.666667	0.444444	0.444444	0.777778	0.333333	0.111111	0	0	0
258.	0.333333	0	0	0	0.111111	0	0	0	0	0	0
259.	0.333333	0	0.222222	0.222222	0.111111	0	0	0	0	0	0
260.	0.444444	0	0	0	0.111111	0	0	0	0	0	0
261.	0.444444	0.111111	0.111111	0.333333	0.111111	0.333333	0	0	0	0	0
262.	0	0	0	0.222222	0.111111	0.222222	0	0	0	0	0
263.	0	0	0	0	0.111111	0.111111	0	0	0	0	0
264.	0.444444	0	0	0.555556	0.222222	0	0.111111	0	0	0	0

265.	0.111111	0	0	0	0.111111	0	0	0	0	0	0
266.	0	0	0	0	0.111111	0	0	0	0	0	0
267.	0.444444	0	0	0	0.111111	0	0	0	0	0	0
268.	0	0	0	0	0	0	0	0	0	0	0
269.	0.333333	0	0	0.222222	0	0	0.111111	0	0	0	0
270.	0.444444	0	0	0	0.111111	0	0	0	0	0	0
271.	0.222222	0	0	0.222222	0.111111	0	0	0	0	0	0
272.	0.111111	0.222222	0	0	0.222222	0	0	0	0	0	0
273.	0.444444	0	0.111111	0	0.111111	0	0	0	0	0	0
274.	0.444444	0	0.222222	0	0.111111	0	0	0	0	0	0
275.	0.444444	0	0	0.222222	0.111111	0	0	0	0	0	0
276.	0.222222	0	0	0	0.111111	0.444444	0	0	0	0	0
277.	0.555556	0	0	0.222222	0.111111	0	0	0	0	0	0
278.	0.333333	0	0	0	0.111111	0	0	0.111111	0	0	0
279.	0.333333	0	0	0	0.111111	0	0	0	0	0	0
280.	0.333333	0	0	0	0.111111	0	0	0	0	0	0
281.	0	0	0.111111	0	0.111111	0	0.111111	0	0	0	0
282.	0.222222	0	0	0	0	0	0.111111	0	0	0	0
283.	0.555556	0	0	0.222222	0.111111	0	0	0	0	0	0
284.	0.555556	0	0	0	0	0	0	0	0	0	0
285.	0.333333	0	0	0	0.111111	0	0	0	0	0	0
286.	0.444444	0	0	0	0.111111	0	0	0	0	0	0
287.	0.222222	0	0	0	0.111111	0	0	0	0	0	0
288.	0.333333	0	0.111111	0	0.111111	0	0	0	0	0	0
289.	0.333333	0	0	0	0.111111	0	0	0	0	0	0
290.	0.444444	0.111111	0	0	0.111111	0	0	0	0	0	0
291.	0.444444	0	0	0	0	0	0	0	0	0	0
292.	0.444444	0.222222	0.111111	0.333333	0.111111	0	0	0	0	0	0
293.	0.444444	0	0.111111	0	0.111111	0	0	0	0	0	0
294.	0	0	0	0.222222	0	0.222222	0	0	0	0	0
295.	0.222222	0	0	0	0	0	0.111111	0	0	0	0
296.	0	0	0	0	0.111111	0	0	0	0	0	0
297.	0.333333	0	0	0	0	0	0.111111	0	0	0	0
298.	0.444444	0	0.111111	1	0.333333	0.444444	0.111111	0	0	0	0
299.	0.222222	0	0	0	0	0	0.111111	0	0	0	0
300.	0	0	0	0	0	0	0	0	0	0	0
301.	0.333333	0.111111	0	0	0.111111	0	0	0	0	0	0
302.	0.333333	0	0	0	0.111111	0	0.111111	0	0	0	0
303.	0.333333	0	0	0	0.111111	0	0.111111	0	0	0	0
304.	0.555556	0	0	0	0.111111	0	0.222222	0	0	0	0
305.	0.333333	0	0	0	0.111111	0	0.111111	0	0	0	0
306.	0.333333	0	0	0.111111	0.111111	0	0.111111	0	0	0	0
307.	0.333333	0	0	0	0.111111	0	0.222222	0	0	0	0
308.	0	0	0	0	0.111111	0	0	0	0	0	0

309.	0.222222	0.222222	0	0	0.111111	0	0	0	0	0	0
310.	0	0	0	0	0.111111	0.333333	0	0	0	0	0
311.	0.444444	0	0	0	0.111111	0	0	0	0	0	0
312.	0.111111	0	0	0	0.111111	0	0	0	0	0	0
313.	0	0	0	0	0.111111	0	0	0	0	0	0
314.	0.444444	0	0	0	0.111111	0	0.111111	0	0	0	0
315.	0.444444	0	0	0	0.111111	0	0	0	0	0	0
316.	0.222222	0	0	0	0	0	0.111111	0	0	0	0
317.	0	0	0	0	0	0	0	0	0	0	0
318.	0	0	0	0	0	0	0.111111	0	0	0	0
319.	0.222222	0	0.111111	0.111111	0.111111	0	0	0	0	0	0
320.	0	0	0	0	0.222222	0	0	0	0	0	0
321.	0.333333	0	0	0	0.222222	0	0	0	0	0	0
322.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0	0
323.	0.222222	0	0	0.111111	0.111111	0	0	0	0	0	0
324.	0.333333	0	0	0	0.111111	0	0	0	0	0	0
325.	0.333333	0	0	0	0.111111	0	0.222222	0	0	0	0
326.	0.555556	0	0.222222	0.111111	0.111111	0	0	0	0	0	0
327.	0.333333	0	0	0	0	0	0.111111	0	0	0	0
328.	0.333333	0.111111	0.111111	0	0.111111	0	0.111111	0	0	0	0
329.	0	0	0	0	0	0	0.222222	0	0	0	0
330.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0	0
331.	0.111111	0	0	0	0.111111	0	0.111111	0	0	0	0
332.	0	0	0.222222	0.111111	0.111111	0	0.222222	0	0	0	0
333.	0.444444	0	0	0	0.111111	0	0.222222	0	0	0	0
334.	0.444444	0	0.111111	0	0.111111	0	0.222222	0	0	0	0
335.	0.333333	0	0	0	0.111111	0	0.111111	0	0	0	0
336.	0.555556	0	0	0	0.111111	0	0.111111	0	0	0	0
337.	0.444444	0	0	0	0.111111	0.111111	0.111111	0	0	0	0
338.	0.222222	0	0	0	0.111111	0	0	0	0	0	0
339.	0.444444	0.222222	0	0	0.111111	0	0	0	0	0	0
340.	0.333333	0	0	0	0.111111	0	0.111111	0	0	0	0
341.	0.111111	0	0.222222	0.111111	0.111111	0	0.111111	0	0	0	0
342.	0.444444	0	0	0	0.111111	0	0.111111	0	0	0	0
343.	0.111111	0	0	0	0	0	0	0	0	0	0
344.	0.222222	0	0	0	0	0	0	0	0	0	0
345.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0	0
346.	0	0	0	0	0.111111	0	0.222222	0	0	0	0
347.	0.222222	0.111111	0.111111	0.111111	0.111111	0	0.333333	0.111111	0	0	0
348.	0.333333	0.333333	0.111111	0	0.111111	0.444444	0.111111	0	0.111111	0	0
349.	0.222222	0	0	0	0.111111	0	0	0	0	0	0
350.	0.333333	0.222222	0	0	0.111111	0	0.333333	0.777778	0	0	0
351.	0.444444	0.111111	0.111111	0.111111	0	0	0.111111	0	0	0	0
352.	0.444444	0	0	0.222222	0.111111	0	0	0	0	0	0

353.	0.111111	0	0	0	0.111111	0	0.111111	0	0	0	0
354.	0.444444	0	0	0	0.111111	0	0.111111	0	0	0	0
355.	0.444444	0	0	0	0.111111	0	0.222222	0	0	0	0
356.	0.444444	0	0	0	0.111111	0	0.222222	0	0	0	0
357.	0	0	0	0	0.111111	0	0.222222	0	0	0	0
358.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0	0
359.	0.333333	0	0	0	0.111111	0	0.222222	0.111111	0	0	0
360.	0.222222	0	0.111111	0	0.111111	0	0.222222	0	0	0	0
361.	0.333333	0	0	0	0.111111	0.222222	0.111111	0	0	0	0
362.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0	0
363.	0	0	0	0	0.111111	0	0.111111	0	0	0	0
364.	0.444444	0	0.111111	0	0.111111	0	0.222222	0	0	0	0
365.	0.444444	0	0	0	0.111111	0	0.111111	0	0	0	0
366.	0	0	0	0	0.111111	0	0.111111	0	0	0	0
367.	0	0	0	0	0.111111	0	0.111111	0	0	0	0
368.	0	0	0	0	0.111111	0	0.222222	0	0	0	0
369.	0.444444	0	0.111111	0	0.111111	0	0.111111	0	0	0	0
370.	0.222222	0	0	0	0.111111	0	0	0	0	0	0
371.	0.444444	0	0	0.555556	0.222222	0	0	0	0	0	0
372.	0	0	0	0	0.111111	0	0	0	0	0	0
373.	0.444444	0	0	0	0.111111	0	0.111111	0.111111	0	0	0
374.	0.444444	0	0	0	0.111111	0	0	0	0	0	0
375.	0.444444	0	0.111111	0	0.111111	0	0	0	0	0	0
376.	0.444444	0	0	0	0.111111	0	0.111111	0	0	0	0
377.	0.333333	0	0.111111	0	0.111111	0	0.111111	0	0	0	0
378.	0.444444	0	0.222222	0	0.111111	0	0.222222	0	0	0	0
379.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0	0
380.	0.444444	0.111111	0.333333	0	0	0	0	0	0	0	0
381.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0	0
382.	0	0	0	0	0	0	0.111111	0	0	0	0
383.	0.333333	0	0	0	0.111111	0	0.111111	0	0	0	0
384.	0.333333	0	0	0.111111	0.111111	0	0	0	0	0	0
385.	0	0	0	0	0.111111	0	0	0	0	0	0
386.	0.444444	0	0	0	0.111111	0	0	0	0	0	0
387.	0.111111	0.222222	0	0	0.111111	0	0.111111	0	0	0	0
388.	0.111111	0	0	0	0	0	0.111111	0	0	0	0
389.	0.333333	0	0.222222	0	0.111111	0	0.111111	0	0	0	0
390.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0	0
391.	0	0	0	0	0	0	0	0	0	0	0
392.	0.333333	0	0	0	0.111111	0	0.111111	0	0	0	0
393.	0.444444	0	0	0	0.111111	0	0.111111	0	0	0	0
394.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0	0
395.	0.555556	0.222222	0.222222	0.222222	0.111111	0.555556	0	0	0	0	0
396.	0.666667	0	0.111111	0.222222	0.111111	0	0.111111	0	0	0	0

397.	0	0	0	0	0.111111	0	0	0	0	0
398.	0.444444	0	0	0.111111	0	0	0.111111	0	0	0
399.	0.222222	0	0.222222	0	0.222222	0.333333	0	0	0	0
400.	0.111111	0	0	0	0.111111	0.444444	0	0	0	0
401.	0.111111	0	0	0	0.111111	0	0	0	0	0
402.	0.333333	0	0	0	0.111111	0	0	0	0	0
403.	0.555556	0.111111	0.222222	0	0.111111	0	0	0	0	0
404.	0.444444	0	0	0	0.111111	0	0.111111	0	0	0
405.	0	0	0	0	0.111111	0	0	0	0	0
406.	0.222222	0	0	0	0.111111	0	0	0	0	0
407.	0.222222	0	0.333333	0	0.111111	0	0	0	0	0
408.	0.333333	0.111111	0.333333	0.222222	0.111111	0.111111	0.111111	0	0	0
409.	0.333333	0	0	0	0.111111	0	0	0	0	0
410.	0.444444	0	0	0.222222	0.111111	0	0	0	0	0
411.	0.333333	0	0	0.222222	0.111111	0	0	0	0	0
412.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0
413.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0
414.	0	0	0	0	0.111111	0	0	0	0	0
415.	0.111111	0	0	0	0.111111	0	0	0	0	0
416.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0
417.	0	0.111111	0.111111	0	0.111111	0	0	0	0	0
418.	0	0	0	0.222222	0.111111	0	0	0	0	0
419.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0
420.	0.222222	0	0	0.111111	0.222222	0.333333	0	0	0	0
421.	0	0.111111	0	0.222222	0.111111	0	0.111111	0	0	0
422.	0.444444	0	0	0	0.111111	0	0.111111	0.111111	0	0
423.	0.333333	0	0	0	0.111111	0	0.111111	0	0	0
424.	0.222222	0	0	0	0.111111	0	0.222222	0	0	0
425.	0.222222	0	0	0	0.111111	0	0.111111	0	0	0
426.	0.444444	0	0	0	0.111111	0	0.111111	0	0	0
427.	0.444444	0.333333	0.444444	0	0.777778	0	0.222222	0.555556	0	0
428.	0	0	0	0	0.111111	0	0	0	0	0
429.	0	0	0	0	0.111111	0	0.111111	0	0	0
430.	0.333333	0	0	0	0.111111	0	0.222222	0	0	0
431.	0	0	0.222222	0	0.111111	0	0.111111	0	0	0
432.	0	0	0.222222	0	0.111111	0	0.111111	0	0	0
433.	0.222222	0	0	0.222222	0.111111	0	0.111111	0	0	0
434.	0	0	0	0	0.111111	0	0	0	0	0
435.	0.444444	0.111111	0.111111	0.111111	0.111111	0	0	0	0.111111	0
436.	0.222222	0	0	0	0.111111	0	0.222222	0	0	0
437.	0.222222	0.111111	0	0.111111	0.111111	0	0.222222	0	0	0
438.	0.111111	0	0	0	0.111111	0	0.222222	0	0	0
439.	0.444444	0.222222	0.111111	0	0.222222	0	0	0	0	0
440.	0	0	0	0	0.111111	0	0.111111	0	0	0

441.	0.333333	0	0.333333	0	0.111111	0	0	0	0	0	0
442.	0	0	0.111111	0	0.111111	0	0.111111	0	0	0	0
443.	0.444444	0	0	0	0.111111	0	0	0	0	0	0
444.	0	0	0	0	0.111111	0	0	0	0	0	0
445.	0.111111	0	0	0	0.111111	0	0	0	0	0	0
446.	0.444444	0	0	0	0.111111	0	0.222222	0.111111	0	0	0
447.	0	0	0	0	0.111111	0	0	0	0	0	0
448.	0	0	0	0	0.111111	0	0	0	0	0	0
449.	0	0	0	0	0.111111	0	0	0	0	0	0
450.	0	0	0	0	0.111111	0	0	0	0	0	0
451.	0.222222	0	0	0	0.111111	0	0.111111	0.222222	0	0	0
452.	0.333333	0	0	0	0.111111	0	0	0	0	0	0
453.	0	0	0	0	0.111111	0	0	0	0.777778	0	0
454.	0	0	0	0.222222	0.111111	0	0	0	0	0	0
455.	0.222222	0	0	0	0.111111	0	0	0	0	0	0
456.	0.222222	0	0	0	0.111111	0	0.111111	0	0.111111	0	0
457.	0.222222	0	0	0	0.222222	0.111111	0	0	0	0	0
458.	0.111111	0	0	0	0.111111	0	0	0	0	0	0
No.	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	

TABEL 3.6 KELAS GANAS

459.	0.777778	1	1	0.777778	0.666667	1	0.888889	0.666667	0	1	
460.	0.444444	0.222222	0.222222	0.222222	0.111111	0.222222	0.333333	0.333333	0	1	
461.	0.777778	0.666667	0.444444	1	0.666667	0.888889	0.444444	0.444444	0.333333	1	
462.	0.666667	0.333333	0.555556	0.333333	0.555556	0	0.333333	0.222222	0	1	
463.	1	0.666667	0.666667	0.555556	0.333333	1	0.333333	0	0.111111	1	
464.	0.666667	0.222222	0.111111	1	0.444444	1	0.444444	0.333333	0.333333	1	
465.	1	0.444444	0.444444	0.222222	0.555556	0.666667	0.666667	1	0	1	
466.	0.777778	0.333333	0.444444	0	0.111111	0	0.666667	0.222222	0	1	
467.	0.444444	0.111111	0.222222	0.333333	0.111111	0.666667	0.222222	0.555556	0	1	
468.	1	0.666667	0.666667	0.222222	0.777778	0.444444	0.666667	0.333333	0.222222	1	
469.	1	1	1	0.777778	0.555556	0	0.777778	0.888889	0	1	
470.	0.444444	0.333333	0.333333	0.888889	0.111111	1	0.444444	0.555556	0	1	
471.	0.111111	0.444444	0.222222	0.222222	0.555556	0.666667	0.666667	0.444444	0	1	
472.	1	0.333333	0.222222	0	0.222222	0.222222	0.555556	0.444444	0.111111	1	
473.	0.555556	1	1	0.111111	0.777778	1	0.666667	0.222222	0.222222	1	
474.	0.444444	0.555556	0.444444	0.555556	1	0	0.222222	0	0	1	
475.	1	1	1	0.333333	0.777778	0	0.777778	1	0	1	
476.	0.222222	0.666667	0.666667	0.333333	0.333333	0.888889	0.333333	0.777778	0	1	
477.	0.666667	0.777778	0.666667	0.111111	0.333333	0.777778	0.222222	0.777778	0.111111	1	
478.	0.888889	0.444444	0.777778	0	0.111111	0.222222	0.111111	0	0.444444	1	
479.	0.444444	0.222222	0.222222	0.333333	0.111111	0.333333	0.222222	0.333333	0	1	
480.	1	0.222222	0.555556	0.111111	0.222222	0.444444	0.333333	1	0.111111	1	
481.	0.444444	0.444444	0.444444	0.777778	1	0.777778	0.666667	0.222222	0.666667	1	
482.	1	0.444444	0.444444	0.555556	0.777778	0.777778	0.666667	0	0	1	

483.	1	0.555556	0.555556	0.222222	0.333333	0.444444	0.222222	0.555556	0	1
484.	0.777778	1	1	0	0.222222	0.555556	0.222222	0.888889	0	1
485.	0.777778	0.111111	0.333333	0	0.444444	0	0.444444	0.333333	0.333333	1
486.	0.444444	0.111111	0.222222	0	0.555556	1	0.444444	0	0	1
487.	0.888889	0.444444	0.444444	0.111111	0.111111	0.111111	0.444444	0	0	1
488.	0.444444	0.222222	0.444444	0.444444	0.222222	0.222222	0.333333	1	0	1
489.	0.888889	1	1	0	1	0.777778	0.222222	0.222222	0	1
490.	0.555556	0.222222	0.333333	0	0.444444	0.111111	0.222222	0.888889	0	1
491.	1	0.333333	0.111111	0	0.222222	0.111111	0.333333	0.222222	1	1
492.	0.444444	0.222222	0.333333	0	0.777778	1	0.333333	0.888889	0	1
493.	0.777778	0.222222	0.777778	0.222222	0.333333	0.888889	0.777778	0.888889	0.777778	1
494.	0.555556	1	0.111111	0.777778	1	0.111111	0.666667	0.777778	1	1
495.	0.888889	0.333333	0.444444	1	0.555556	1	0.333333	0.777778	0	1
496.	1	0.555556	0.333333	0	0.222222	0.333333	0.222222	0.111111	0.222222	1
497.	0.222222	0.444444	0.666667	0.777778	0.777778	0.888889	0.666667	1	0.666667	1
498.	0.444444	1	0.555556	0	1	0.333333	0.333333	1	1	1
499.	0.222222	0.222222	0.555556	0.333333	0.444444	0.777778	0.333333	0.333333	0	1
500.	0.222222	0.555556	0.555556	0.555556	0.444444	1	0.555556	0.777778	0.222222	1
501.	0.888889	0.555556	0.888889	0.111111	1	0.555556	0.111111	0.888889	1	1
502.	0.666667	0.444444	0.555556	1	0.444444	1	0.666667	0.888889	0.333333	1
503.	1	0.222222	0.444444	0	1	0.444444	0.222222	1	0.111111	1
504.	0.111111	0.222222	0.333333	0.333333	0.111111	0.444444	0.111111	0.444444	0	1
505.	0.777778	0.111111	0.222222	0	0.555556	0.222222	0.666667	0	0	1
506.	1	1	1	1	1	0	0.777778	0.777778	0.777778	1
507.	0.666667	0.222222	0.333333	0.333333	0.222222	0.222222	0.222222	0.111111	0.666667	1
508.	1	1	1	0.777778	0.111111	1	0.333333	0	0	1
509.	0	0.555556	0.777778	1	0.777778	1	0.444444	0.666667	0	1
510.	0.555556	0.444444	0.333333	0.333333	0.222222	0.888889	0.666667	0.777778	0.222222	1
511.	0.777778	0.555556	0.333333	0.222222	0.444444	0.888889	0.222222	0	0	1
512.	1	0.222222	0.222222	1	0.111111	1	0.666667	0.222222	0.222222	1
513.	1	1	1	0.222222	1	0.777778	0.777778	0	0	1
514.	0.333333	0.444444	0.444444	1	0.333333	1	0.666667	0.444444	0.777778	1
515.	1	1	1	0.111111	1	1	0.444444	0.222222	0.222222	1
516.	0.444444	0.222222	0.444444	0	0.777778	1	0.444444	0.222222	0	1
517.	0.444444	0.333333	0.555556	0.666667	0.888889	0.666667	0.777778	1	0	1
518.	0.666667	0.444444	0.222222	0.666667	0.333333	1	0.666667	0.444444	0.444444	1
519.	0.777778	0.222222	0.444444	0.333333	0.444444	1	0	0.555556	0.111111	1
520.	0.444444	1	0.777778	1	0.777778	1	0.222222	0.555556	0.222222	1
521.	0.888889	0.444444	0.444444	0.333333	0.333333	0.444444	0.333333	0.222222	0.222222	1
522.	0.222222	0.333333	0.444444	0.111111	0.555556	0.777778	0.333333	0	0	1
523.	0.777778	0.777778	0.666667	0.333333	1	1	0.666667	0.777778	0.666667	1
524.	0.666667	0.111111	0.333333	0	0.555556	1	0.444444	0.333333	0.222222	1
525.	1	1	0.777778	0.555556	0.333333	0.444444	0.777778	1	0	1
526.	0.444444	0.444444	0.444444	0.555556	0.222222	1	0.222222	0	0	1

527.	0.888889	0.888889	1	0.222222	0.555556	1	0.666667	1	0.555556	1
528.	1	0.666667	0.666667	0.333333	0.444444	1	0.444444	0.666667	0.111111	1
529.	0.444444	0.555556	0.666667	0.777778	0.777778	1	0.222222	1	0.222222	1
530.	1	0.777778	1	1	0.555556	0	0.222222	0	1	1
531.	0.555556	1	1	1	0.777778	1	1	1	0.666667	1
532.	0.777778	0.555556	0.444444	0.333333	0.222222	1	0.555556	0	0	1
533.	0.444444	0.777778	0.666667	0.666667	1	1	0.444444	0.666667	0	1
534.	0.444444	1	1	0.222222	0.777778	0	0.444444	1	0.222222	1
535.	0.444444	0.222222	0.222222	0.222222	0.555556	1	0.222222	0	0	1
536.	0.444444	0.777778	0.777778	0.777778	0.444444	1	0.666667	0.777778	0	1
537.	0.777778	0.666667	0.555556	0.333333	0.333333	1	0.444444	0	0	1
538.	0	0.444444	0.777778	0.555556	0.444444	0.777778	0.666667	1	0	1
539.	1	0.444444	0.555556	1	0.555556	1	0.666667	0.666667	1	1
540.	0.444444	0.777778	0.333333	1	0.444444	0.777778	0.888889	1	0	1
541.	1	1	1	0.777778	0.555556	0.777778	0.666667	1	0	1
542.	0.666667	0.444444	1	1	1	1	0.333333	1	0.222222	1
543.	0.888889	0.666667	0.666667	0.444444	0.444444	1	0.666667	0.777778	0.222222	1
544.	1	0.777778	0.777778	0.333333	1	1	0.777778	0	0	1
545.	0.444444	1	1	0.888889	0.555556	1	0.666667	1	0.444444	1
546.	1	1	0.888889	0.222222	0.666667	0.444444	0.222222	0.444444	0	1
547.	0.777778	1	1	1	0.444444	1	0.777778	1	0.555556	1
548.	0.777778	1	0.777778	0.777778	0.333333	0.777778	0.666667	0.666667	0	1
549.	1	1	1	1	0.666667	1	0.666667	1	0.333333	1
550.	1	1	1	1	0.222222	1	1	0.555556	0	1
551.	0.777778	0.666667	0.777778	0.666667	0.444444	0.444444	0.444444	1	0.111111	1
552.	0.555556	1	0.666667	0.666667	0.555556	0.333333	0.777778	1	0.111111	1
553.	1	0.555556	0.333333	0.222222	1	1	0.888889	1	0	1
554.	0.333333	0	0	0.222222	0	0.444444	0.111111	0	0	1
555.	0.666667	0.444444	0.555556	0.222222	0.222222	0.777778	0.666667	0.333333	0	1
556.	1	0.444444	0.444444	0.555556	0.222222	1	0.666667	0.888889	0.111111	1
557.	1	0.444444	0.666667	0.333333	0.333333	1	0.777778	0.888889	0	1
558.	0.777778	0.888889	0.888889	0.444444	0.222222	0.444444	0.666667	0.666667	0	1
559.	1	1	1	0.222222	1	1	0.888889	1	0	1
560.	0.666667	0.333333	0.666667	0.333333	0.222222	0.666667	0.666667	0.555556	0	1
561.	0.555556	0.777778	0.666667	0.444444	0.555556	0.777778	0.777778	0.888889	0.111111	1
562.	1	0.333333	0.444444	0.444444	0.444444	1	0.333333	0	0	1
563.	1	0.777778	0.777778	0.111111	0.777778	1	0.333333	0.777778	1	1
564.	0.888889	0.777778	0.777778	0.444444	0.555556	0.111111	0.333333	1	0.333333	1
565.	0.777778	1	1	0.777778	0.555556	0.888889	0.222222	1	1	1
566.	1	0.333333	0.222222	0.111111	0.222222	1	0.444444	0.222222	0.111111	1
567.	0.777778	1	1	0.777778	0.444444	1	0.666667	0.777778	0	1
568.	0.777778	0.333333	0.333333	0	0.111111	0.888889	0.222222	0.222222	0	1
569.	1	0.333333	0.333333	1	0.111111	1	0.444444	0.222222	0.222222	1
570.	0.555556	1	1	0.111111	0.777778	1	0.666667	0.222222	0.222222	1

571.	0.888889	1	1	0	1	0.777778	0.222222	0.222222	0	1
572.	0.444444	0.555556	0.555556	0.111111	0.333333	1	0.222222	0.555556	0	1
573.	1	0.444444	0.777778	1	0.222222	1	0.444444	0	0.222222	1
574.	0.444444	1	1	0.555556	1	1	1	0.555556	0.444444	1
575.	0.777778	0.777778	0.888889	0.333333	0.444444	1	0.666667	0.777778	0	1
576.	1	0.333333	0.333333	1	0.555556	1	0.444444	0.444444	0	1
577.	0.666667	0.888889	0.333333	1	1	0.222222	0.444444	0.222222	0.222222	1
578.	1	1	0.555556	0.222222	0.222222	1	0.333333	0.222222	0.111111	1
579.	0.222222	0.222222	0.444444	0.111111	0.222222	1	0.666667	0	0	1
580.	1	0.777778	0.777778	0.111111	0.222222	0.333333	0.777778	0.666667	0.777778	1
581.	0.777778	0.333333	0.666667	0	0.222222	1	0.222222	0.888889	0.111111	1
582.	0.222222	0.222222	0.444444	0.111111	0.222222	1	0.666667	0	0	1
583.	0.666667	0.111111	0.333333	0	0.222222	0.333333	0.222222	0.222222	0	1
584.	1	0.444444	0.666667	0.222222	0.222222	0.666667	0.222222	0.222222	0.777778	1
585.	0	0.333333	0.222222	1	0.333333	1	0.444444	0.555556	0	1
586.	1	0.333333	0.555556	0	0.111111	1	0.444444	0.222222	0	1
587.	0.666667	0.333333	0.444444	1	0.111111	1	0.222222	0.777778	0.111111	1
588.	0.777778	1	1	1	0.777778	1	1	0.666667	0.222222	1
589.	1	1	1	1	1	1	0.333333	1	1	1
590.	0.555556	0	0.222222	0	0.333333	0.444444	0.444444	1	0	1
591.	0.444444	0.555556	0.555556	0.777778	0.555556	1	0.333333	1	0.333333	1
592.	0.777778	0.777778	0.777778	0	0.111111	0	0.555556	1	0	1
593.	1	0.333333	0.333333	0.555556	0.111111	1	0.111111	0.222222	0	1
594.	0.444444	0.444444	0.666667	0.777778	0.555556	1	0.666667	0.333333	0	1
595.	0.888889	0	0.111111	0.555556	0.333333	1	0.666667	0.666667	0.111111	1
596.	0.777778	0.333333	1	0.444444	0.333333	0.333333	0.666667	1	0	1
597.	1	1	1	0.666667	0.888889	1	0.666667	1	1	1
598.	0.777778	0.222222	0.333333	0.888889	0.222222	1	0.222222	0.222222	0	1
599.	1	0.777778	0.333333	0.333333	0.333333	1	0.222222	1	0.333333	1
600.	0.666667	0.777778	0.666667	0.555556	0.333333	0.222222	0.777778	0.777778	0.333333	1
601.	0.777778	0.555556	0.333333	1	1	0	0.222222	0.444444	0	1
602.	0.444444	0.444444	0.444444	0.111111	0.444444	1	0.333333	0.222222	0	1
603.	0.555556	0.777778	0.666667	0.777778	0.555556	0.777778	0.777778	0.888889	0	1
604.	0.666667	0.555556	0.222222	0.111111	0.444444	1	0.666667	0.333333	0.555556	1
605.	0.444444	0.333333	0.555556	1	0.111111	1	0.333333	0	0	1
606.	1	0	0	0	0.111111	1	0.444444	0.333333	0	1
607.	0.777778	1	0.222222	0.111111	0.555556	0.333333	0.222222	1	0	1
608.	1	0.333333	0.555556	0.333333	0.444444	1	0.666667	0	0	1
609.	1	0.333333	0.666667	0.111111	0.111111	0.777778	0.555556	0	0	1
610.	0.444444	0.333333	0.555556	0.555556	0.333333	1	0.333333	0.222222	0	1
611.	0.777778	0.555556	0.666667	0.222222	0.222222	1	0.222222	0.333333	0.111111	1
612.	0.555556	0.444444	0.444444	0.777778	0.333333	1	0.222222	0.333333	0	1
613.	0.777778	0.444444	0.444444	0.444444	0.111111	1	0.333333	0.222222	0	1
614.	1	0.222222	0.222222	0	0.111111	1	0.666667	0.555556	0	1

615.	0.666667	0.555556	0.333333	0.777778	1	1	0.888889	0.444444	0.222222	1
616.	0.222222	0.333333	0.333333	1	0.444444	0	0.222222	0.222222	0	1
617.	0.333333	0.111111	0.222222	0.444444	0.222222	0.777778	0.666667	0.555556	0	1
618.	0.111111	0.666667	1	1	0.666667	1	0.333333	0.888889	0.333333	1
619.	0.444444	0.222222	0.222222	0	0.222222	0.222222	0.222222	0.222222	0.222222	1
620.	0.777778	1	1	0.666667	1	1	0.666667	0.222222	0.777778	1
621.	0.777778	1	0.444444	0.222222	0.777778	0.333333	0.333333	1	0.222222	1
622.	1	0.222222	0.444444	0.333333	0.222222	0.666667	0.222222	0.444444	0.222222	1
623.	0.555556	1	1	1	1	1	0.777778	1	1	1
624.	0.222222	1	0.222222	1	0.555556	1	0.444444	0	0.333333	1
625.	0.555556	1	1	1	0.777778	1	0.666667	1	0.666667	1
626.	0.444444	0.777778	0.777778	1	0.444444	1	0.777778	1	0.222222	1
627.	1	0.555556	0.222222	0.555556	0.333333	1	0.666667	0.777778	0.333333	1
628.	0.666667	0.555556	0.555556	0.222222	0.111111	1	0.666667	0	0	1
629.	1	0.777778	0.666667	0.333333	0.222222	1	0.666667	0.888889	0	1
630.	0.222222	1	0.777778	0.666667	0.555556	0.888889	0.888889	0.222222	0.777778	1
631.	1	1	1	0.555556	0.777778	0.333333	0.777778	0.444444	0	1
632.	0.777778	0.444444	0.555556	0.111111	0.222222	1	0.555556	0.555556	0	1
633.	0.777778	0.666667	0.777778	0.444444	1	1	0.666667	0.111111	0	1
634.	1	1	1	0.666667	1	1	0.777778	0.111111	0	1
635.	0.888889	1	1	1	1	1	1	1	0	1
636.	0.777778	0.666667	0.777778	0.111111	0.333333	0.111111	0.444444	1	0	1
637.	1	0.777778	1	0	0.222222	1	0.444444	0	0	1
638.	1	1	1	0	0.555556	0	0.111111	0.777778	0	1
639.	1	0.333333	0.222222	1	0.333333	1	1	0	0	1
640.	0.444444	0.666667	0.888889	0.777778	0.555556	1	0.777778	1	0	1
641.	0.333333	0.444444	0.444444	0.777778	0.555556	1	1	0.666667	0	1
642.	1	0.111111	0.111111	0	0.111111	0.555556	0	0	0.111111	1
643.	1	0.555556	0.444444	0.777778	0.444444	1	0.777778	0.555556	0	1
644.	0.777778	0.777778	0.888889	0.555556	0.555556	0.222222	1	1	0	1
645.	1	0.888889	0.777778	0.666667	0.555556	0.333333	0.666667	1	0.222222	1
646.	1	0.555556	0.555556	0.111111	0.333333	1	0.888889	0.666667	0	1
647.	0.555556	0.555556	0.555556	0.444444	0.333333	1	0.666667	0.555556	0.111111	1
648.	0.333333	0.777778	0.666667	1	0.333333	1	0.666667	0.444444	0	1
649.	0.888889	1	1	1	1	0.444444	1	1	1	1
650.	0.777778	0.666667	0.777778	0.444444	0.444444	1	0.888889	1	0	1
651.	1	1	1	1	0.555556	1	0.777778	0	0.444444	1
652.	0.222222	0.555556	0.333333	1	0.222222	0.222222	0.222222	0.333333	0	1
653.	0.555556	0.222222	0.111111	0	0.222222	0.333333	0.333333	0	0	1
654.	0.444444	0.777778	0.888889	0.333333	0.222222	1	0.666667	0	0	1
655.	0.444444	1	1	1	0.555556	1	0.555556	0.444444	0.111111	1
656.	0.777778	1	1	1	0.666667	0.444444	0.333333	0.777778	0.666667	1
657.	0.555556	0.555556	0.666667	1	0.222222	1	0.777778	1	0.111111	1
658.	0.333333	1	0.333333	0.666667	0.222222	1	0.888889	1	0	1

659.	0.333333	0.666667	0.777778	0.222222	0.333333	1	0.888889	0	0	0
660.	1	0.333333	0.444444	0.333333	0.222222	0.444444	0.666667	0.222222	0	0
661.	0.666667	0.444444	0.555556	1	0.333333	1	0.444444	0.222222	0	0
662.	0.666667	0.333333	0.333333	0.222222	0.333333	1	0.555556	0.888889	0	0
663.	0.555556	1	1	1	0.333333	1	0.666667	1	0	0
664.	0.666667	0.777778	0.222222	0.666667	0.333333	0.444444	0.666667	0.777778	0.111111	1
665.	0.444444	0.666667	1	1	0.444444	1	1	1	0	0
666.	0.777778	0.333333	0.333333	0	0.555556	1	0.111111	0.444444	0.111111	1
667.	1	1	0.777778	1	0.555556	0.444444	1	0.222222	0	0
668.	0.777778	1	0.333333	0.333333	0.777778	1	0.777778	0.111111	0	0
669.	0.666667	0.555556	1	0.444444	0.222222	1	0.888889	1	0.111111	1
670.	1	0.888889	0.666667	0.222222	0.333333	0.111111	0.666667	0.666667	0	0
671.	0.444444	0.666667	1	0.555556	0.444444	1	0.666667	0.444444	0	0
672.	0.555556	1	0.444444	0.444444	0.333333	1	0.555556	1	0	0
673.	0.777778	1	1	1	0.555556	1	1	1	0	0
674.	0.888889	0.777778	0.777778	0.888889	0.555556	0.222222	0.333333	0	0	0
675.	0.333333	1	0.777778	0.444444	0.333333	0	1	0	0	0
676.	0.111111	0.444444	0.666667	0.555556	0.333333	1	0.666667	0.555556	0	0
677.	1	0.222222	0.333333	0.444444	0.222222	1	0.333333	0	0	0
678.	0.333333	0.777778	0.555556	0.222222	0.333333	1	0.666667	0	0	0
679.	0.444444	0.333333	0.555556	0.777778	0.333333	0	0.777778	1	0	0
680.	0.444444	0.222222	0.111111	0.777778	0.444444	1	0.777778	0	0.111111	1
681.	1	0.444444	1	0.222222	0.444444	0.777778	0.666667	0.777778	0.222222	1
682.	0.444444	1	1	1	1	1	1	0	0	0
683.	1	0.333333	0.222222	1	0.222222	1	0.666667	0	0.111111	1
684.	0.444444	1	1	1	0.444444	0.111111	0.777778	0.444444	0	0
685.	0.777778	1	1	1	0.555556	1	1	1	1	1
686.	0.333333	0.555556	0.555556	0.444444	0.666667	0.555556	0.666667	0.666667	0.222222	1
687.	0.777778	0.666667	0.333333	0.333333	0.444444	0.222222	0.444444	1	0	0
688.	1	1	0.666667	0.777778	0.666667	0	1	1	0.222222	1
689.	0.444444	1	1	1	1	0.111111	1	1	1	1
690.	0.666667	0.777778	0.777778	0.666667	0.222222	1	0.666667	0.111111	0.222222	1
691.	0.444444	0.666667	0.333333	0	0.555556	0	0.666667	1	0.222222	1
692.	0.444444	1	1	0.777778	0.444444	0.444444	0.666667	1	0	0
693.	0.222222	1	0.666667	0.777778	0.444444	0.777778	0.666667	0.333333	0	0
694.	1	1	1	1	0.444444	1	1	1	0.666667	1
695.	0.444444	1	1	1	0.333333	1	0.444444	0.555556	0.222222	1
696.	0.444444	1	1	0.444444	0.333333	0.444444	0.333333	0.333333	0	0
697.	0.444444	1	1	0.222222	0.666667	0.222222	0.777778	1	0.111111	1
698.	0.333333	0.777778	0.555556	0.333333	0.222222	0.333333	1	0.555556	0	0
699.	0.333333	0.777778	0.777778	0.444444	0.333333	0.444444	1	0.333333	0	0

LAMPIRAN 4

Gambar 4.1. Nilai MSE dalam lapisan *output* pada pengujian pertama

Error			Error			Error			Error		
Epoch	MSE		Epoch	MSE		Epoch	MSE		Epoch	MSE	
1	0.10185855675233776		49	0.0615659735750082		97	0.03582298229144173		145	0.02246127997174527	
2	0.10161512739925053		50	0.0608228169928494		98	0.03454059508043706		146	0.022259837187474135	
3	0.10126029753049941		51	0.060120947060654076		99	0.03508315946297167		147	0.02206073738108097	
4	0.10081495523411041		52	0.0593870329706502		100	0.0347207126242161		148	0.021864085860858442	
5	0.10028730251757587		53	0.0568130721655039		101	0.034363150227256806		149	0.02166978878517704	
6	0.0998879318979766		54	0.05798578991396911		102	0.034010378820170894		150	0.02147774433981359	
7	0.0990245708801447		55	0.05730019125981117		103	0.03366225794107146		151	0.0212880526899759	
8	0.09830564197412191		56	0.0566242943955680		104	0.033118421235818		152	0.021100599209731143	
9	0.09753754344132194		57	0.0559580165634922		105	0.032979953051761684		153	0.02091541476629797	
10	0.09672736992895964		58	0.055301302474939175		106	0.03264551952091143		154	0.02073246812697472	
11	0.09580878424965489		59	0.054564216538546005		107	0.032155059723882		155	0.0205168863917274	
12	0.09500348897329848		60	0.05401644563485983		108	0.0319898374495875		156	0.020373057465757478	
13	0.0940997575345952		61	0.0533880488104055		109	0.031668333784438		157	0.0201956077312338	
14	0.09317404554704034		62	0.052768390284423896		110	0.031351158100533016		158	0.02022054931742056	
15	0.09223009684704127		63	0.05215766575365019		111	0.03103809444994445		159	0.0198496776405178	
16	0.09127164787712307		64	0.051555740740832955		112	0.030729082193724912		160	0.0196793372899344	
17	0.090320986695823		65	0.05086254757537634514		113	0.030423916925030454		161	0.019510846715209113	
18	0.08932399120717055		66	0.05033783320298108		114	0.0301226678087879		162	0.019344206884426143	
19	0.08833966311207536		67	0.049801573059035		115	0.0298537139638985		163	0.019179376643056308	
20	0.0873751635612443		68	0.049233556820726436		116	0.02953192345548016		164	0.01901651573921523	
21	0.08636184752182529		69	0.048673387713983864		117	0.029428113039413		165	0.0188558897608015	
22	0.08563720842397215		70	0.048121080876013855		118	0.028956416948229428		166	0.0186966641622911	
23	0.0843839908459688		71	0.04757674420317856		119	0.0286742403438718		167	0.0185394132909576	
24	0.083398957120497		72	0.04740413420917361		120	0.0283975794561056		168	0.01834415321500396	
25	0.08241797246797673		73	0.04651133705280166		121	0.02812090524684776		169	0.0182306364583933	
26	0.08144224460939563		74	0.04589957489584002		122	0.027849521039263646		170	0.018078944579461652	
27	0.0804727014855721		75	0.04547539478513231		123	0.0275817930208666		171	0.017292895103562	
28	0.07950919573673815		76	0.04496849839104302		124	0.02731719534077223		172	0.017780730856831444	
29	0.07855371281606111		77	0.04446872902057572		125	0.027056089721796928		173	0.0176338780775296	
30	0.077606940008561		78	0.043976162881735057		126	0.027981658899746		174	0.01748869456784103	
31	0.07666850570196529		79	0.04349052776530484		127	0.026543575254638867		175	0.01734505989479947	
32	0.075739808261663		80	0.04301175535022152		128	0.026292190483106517		176	0.017203139965838554	
33	0.074820839349762333		81	0.04253939460203062		129	0.0260440176082066		177	0.017062804167455122	
34	0.07391187459740593		82	0.04207477659668248		130	0.025798635824901533		178	0.01692408548521214	
35	0.07303120291976466		83	0.041616126146441126		131	0.02556173057592103		179	0.0167868831864732	
36	0.0721247575925648		84	0.0411639407194845		132	0.025317713635570486		180	0.016651256125973242	
37	0.0712471077815181		85	0.040717657773331		133	0.025080156567775722		181	0.0165177578595899	
38	0.0703804059195742		86	0.0402771904134583		134	0.024846582854089146		182	0.016384498666077284	
39	0.06952455216242055		87	0.03984395784413401		135	0.024619569488055		183	0.01625334419387344	
40	0.0686793493897547		88	0.039416149008339365		136	0.024388351057440902		184	0.016123677883683524	
41	0.0678452646267158		89	0.038994310586152804		137	0.0241636009780915		185	0.015959548409901946	
42	0.067022129480747428		90	0.03857834519764405		138	0.0239453204055603		186	0.0158807303246887	
43	0.0662102073748206		91	0.0381608745774407		139	0.023722195867404326		187	0.01574337266928507	
44	0.0654091138266288		92	0.037736337058252184		140	0.023505496840222108		188	0.015619411593331243	
45	0.0646187582170251		93	0.0373642158001265		141	0.02391518565680853		189	0.015496818968924088	
46	0.063839503664652		94	0.03697108684776825		142	0.02308019267539804		190	0.015375477290663424	
47	0.06307096204581725		95	0.0363583194600424		143	0.022871318723706104		191	0.015255468508376844	
48	0.062313098669594155		96	0.036200510286694396		144	0.0228665051935279554		192	0.015136731315206764	
Error			Error			Error			Error		
Epoch	MSE		Epoch	MSE		Epoch	MSE		Epoch	MSE	
193	0.0150192998956769		241	0.01061684575321688		289	0.00787956862634205		337	0.006111463078141898	
194	0.014903003568417611		242	0.010546094530761632		290	0.007834542251572316		338	0.0060818327130356745	
195	0.01478800755689889		243	0.01047605384180589		291	0.00778992492865908		339	0.006052443446667553	
196	0.014674216084728822		244	0.0104066951300100497		292	0.007745562785022950		340	0.00602329958293569	
197	0.014561659735560326		245	0.010337923277140622		293	0.007701845112433365		341	0.00599436616352488	
198	0.014450239438958598		246	0.010269967189038555		294	0.0076854282351755431		342	0.005965731721126417	
199	0.0143401475024757457		247	0.0102025688498201692		295	0.0076153984109205		343	0.00593714562087931	
200	0.01423120421261922		248	0.010125805338183078		296	0.0075277125767195		344	0.0059091387973582155	
201	0.01412342465726763		249	0.0100695956753443		297	0.007530526776711524		345	0.00588119843623322	
202	0.014016817235671119		250	0.01000418274942044		298	0.0074886439791944925		346	0.0058534965442219	
203	0.013911306992293319		251	0.009939326580300584		299	0.007447141680431146		347	0.005826016645114271	
204	0.01380862414962293262		252	0.0098751106643257		300	0.00740602284052065		348	0.005798775268239865	
205	0.013703634127269868		253	0.009815120170478517		301	0.00736526283373795		349	0.0057717668107943945	
206	0.013601482070497496		254	0.00974051007916371		302	0.00732460816116375		350	0.0057449051024938	
207	0.013500282272469214		255	0.00966804965768315		303	0.00728413291967921		351	0.0057183018450707	
208	0.01340023465638183		256	0.0096224729283351		304	0.00724510065032128		352	0.005619619217000581	
209	0.01331026205242988		257	0.00956299696606145		305	0.00720572513741766		353	0.005665796837991223	
210	0.0132035773760097		258	0.00950231429562226		306	0.007166740274516336		354	0.00563984322071151	
211	0.013106447283260865		259	0.009442199839609999		307	0.0071208921997125		355	0.00561409998663371	
212	0.0131014827921252152		260	0.00938265752437916		308	0.00708974792492724		356	0.0055855053824816	
213	0.01291553742097398		261	0.0093236905225336		309	0.007051755269201324		357	0.0055632476693825	
214	0.012821564155005218		262	0.009285631612512281		310	0.007014123835336065		358	0.00553813280094947	
215	0.01272857951432794		263	0.0092075212451008							

Error			Error			Error			Error		
Epoch	...	MSE	Epoch	...	MSE	Epoch	...	MSE	Epoch	...	MSE
385	...	0.00493150192705257	433	...	0.00412474189011278	481	...	0.003770961007709716	529	...	0.003945122701153228
386	...	0.0049114468541379425	434	...	0.004110874938420227	482	...	0.00377270654323653	530	...	0.00394901507600254
387	...	0.004891551451259704	435	...	0.00409711119852685	483	...	0.003775432089604933	531	...	0.003952699928679457
388	...	0.004871618492286593	436	...	0.00408345145498911	484	...	0.003782568173627064	532	...	0.003956794220396784
389	...	0.004852237499205699	437	...	0.004068993286784376	485	...	0.003781191413563302	533	...	0.003960687875297334
390	...	0.004832060656561318	438	...	0.004056438740294769	486	...	0.003784203486417198	534	...	0.003964507649732725
391	...	0.004813537013205766	439	...	0.00404307812824093	487	...	0.0037873694821638185	535	...	0.0039683867300669
392	...	0.0047944171152776585	440	...	0.00402982125982757	488	...	0.0037905591772489024	536	...	0.003972214337046666
393	...	0.00477544319281746	441	...	0.004016658051161830	489	...	0.003793030905521856	537	...	0.0039760882994432
394	...	0.004756618222515643	442	...	0.004003595389872795	490	...	0.003797099464627082	538	...	0.0039790866777773
395	...	0.004737927268749609	443	...	0.003990628247907461	491	...	0.003800496867738795	539	...	0.003983779118614293
396	...	0.0047193716997915125	444	...	0.0039777614006663295	492	...	0.003803457954029302	540	...	0.00398762897831363
397	...	0.00470096733177508	445	...	0.003964982211153956	493	...	0.0038074615528866807	541	...	0.00399143643229169
398	...	0.004682705467421921	446	...	0.003952314036308606	494	...	0.0038110271365774636	542	...	0.003995224808924794
399	...	0.00466458914749037	447	...	0.0039375444724753	495	...	0.0038146651717154	543	...	0.00399916170203696
400	...	0.004646829710518809	448	...	0.0039272489125660595	496	...	0.003818238309883742	544	...	0.004002798749906337
401	...	0.004628810705164061	449	...	0.0039148595706146845	497	...	0.0038219017543309193	545	...	0.0040065808707606054
402	...	0.004611127688294428	450	...	0.00390025641993047804	498	...	0.003825080108121453	546	...	0.00401042535289121
403	...	0.004593571142049286	451	...	0.0038890356323003056	499	...	0.003829337882725033	547	...	0.00401410640401981
404	...	0.004576147622091821	452	...	0.0038782370726445116	500	...	0.0038330914241423973	548	...	0.004017854652165369
405	...	0.004558863463278788	453	...	0.003866236244368403	501	...	0.0038376270721253	549	...	0.00402158950107819
406	...	0.004544171284668744	454	...	0.00385432158761394	502	...	0.0038466870224246	550	...	0.00402530680192102
407	...	0.004524689393567954	455	...	0.003842491728651048	503	...	0.003844557954029302	551	...	0.004029008355022231
408	...	0.0045077908521597	456	...	0.0038307499574428711865	504	...	0.003848260284426947	552	...	0.0040326913157867714
409	...	0.004491021648747931	457	...	0.00381909209730360412	505	...	0.0038207203931588	553	...	0.00403638715438304
410	...	0.00447437722498996	458	...	0.0038075185106203965	506	...	0.00385587454508642	554	...	0.004040084932950766
411	...	0.0044578585194785	459	...	0.003797511984374634	507	...	0.0038390304250424	555	...	0.00403813422497853
412	...	0.004441468820415215	460	...	0.00378891985333927	508	...	0.003863532951242694	556	...	0.00404745742333737
413	...	0.004425206340691099	461	...	0.0037816225618016656	509	...	0.003867357736249963	557	...	0.0040512767426707326
414	...	0.004409073279419793	462	...	0.00377543174076662132	510	...	0.003871200412039263	558	...	0.0040549894362285266
415	...	0.004393066252296779	463	...	0.003770264036338016	511	...	0.003875056823235935	559	...	0.004058676721540211
416	...	0.004377182127284776	464	...	0.003766240780908838	512	...	0.0038789321496510895	560	...	0.00402368301497762
417	...	0.004361420526579009	465	...	0.00376529159820956	513	...	0.003838278577491891	561	...	0.00406605732123712
418	...	0.004345780663826568	466	...	0.003759866045203467	514	...	0.0038866757006435378	562	...	0.00406976432848718
419	...	0.0043026003355176	467	...	0.0037578172360054903	515	...	0.003889055946791084	563	...	0.004073513778879822
420	...	0.00431486195933583	468	...	0.003756348372821616	516	...	0.00389448336238625674	564	...	0.004077243842341592
421	...	0.004299580947343068	469	...	0.003753562383894597	517	...	0.00389318151718941	565	...	0.00408099998502226
422	...	0.004284410526986246	470	...	0.003754863867289882	518	...	0.003902209464044655	566	...	0.004084781474580398
423	...	0.004269353560414837	471	...	0.003755480423279103	519	...	0.003906108648443937	567	...	0.00408853884221643
424	...	0.004254410721919541	472	...	0.00375134321597323	520	...	0.0039099851632434	568	...	0.0040923188900858
425	...	0.004239581024152413	473	...	0.00375581780382001	521	...	0.003913891688459411	569	...	0.0040961310216438
426	...	0.0042242860349189035	474	...	0.0037567892820547036	522	...	0.00391777936051043	570	...	0.0040999087617563
427	...	0.004212202159927521	475	...	0.0037508173430976	523	...	0.0039217009125306212	571	...	0.004103865298549773
428	...	0.00419569692751808	476	...	0.0037594886990682	524	...	0.003925639770302918	572	...	0.00410742061858226
429	...	0.0041812887561887	477	...	0.0037612171886641	525	...	0.00392930710805125	573	...	0.00411263426389383
430	...	0.004166989821791892	478	...	0.0037631651988717	526	...	0.00393345270623879	574	...	0.00411504948719832
431	...	0.0041527997978894627	479	...	0.0037653093197771	527	...	0.003937339922902203	575	...	0.004118619216761323
432	...	0.00413871821040362	480	...	0.00376761645639715	528	...	0.003941236135442566	576	...	0.004122579338147531
Error	...	MSE	Error	...	MSE	Error	...	MSE	Error	...	MSE
Epoch	...	MSE	Epoch	...	MSE	Epoch	...	MSE	Epoch	...	MSE
577	...	0.00412631935973881	625	...	0.00430664182205738	673	...	0.00447531724301557	721	...	0.004362149610087341
578	...	0.004130074488604023	626	...	0.00431034616443894	674	...	0.004478656295061939	722	...	0.00434827818144362
579	...	0.00413385491389427	627	...	0.004314057138753848	675	...	0.0044819842239439195	723	...	0.004334347911769307
580	...	0.004137649453130206	628	...	0.00431777245011376	676	...	0.0044845329869817757	724	...	0.004320348505831821
581	...	0.004141446496217151	629	...	0.004322147605813424	677	...	0.004488746242536008	725	...	0.00430564306783283281
582	...	0.004145265160699771	630	...	0.004325201586039836	678	...	0.00449210528551206	726	...	0.004392409148714984
583	...	0.00414908851212273	631	...	0.0043288649296033	679	...	0.004494571265727091	727	...	0.004278115209183799
584	...	0.00415294094082882	632	...	0.00432538617422601	680	...	0.004498886200775501	728	...	0.0040866750002435378
585	...	0.004156790264710113	633	...	0.00436188020697538	681	...	0.00450225982728204	729	...	0.00429907409790478
586	...	0.0041606043914535434	634	...	0.004339804485990502	682	...	0.004505628660892669	730	...	0.00423583214771573
587	...	0.00416489417491865	635	...	0.004343388196041392	683	...	0.004509018424636365	731	...	0.004221793232810254
588	...	0.00416830040317015	636	...	0.0043469012879821	684	...	0.00451236848272721	732	...	0.004027798232947634
589	...	0.00417211122933	637	...	0.00435044529517599	685	...	0.0045154750480238	733	...	0.00419384535040584
590	...	0.00417591806178103	638	...	0.00435950438438325	686	...	0.00451911377405829	734	...	0.00417944574012578
591	...	0.004179709573722842	639	...	0.004378450530393287	687	...	0.0045224593253639287	735	...	0.004166093096468125
592	...	0.00418347102168263	640	...	0.004360961220015075	688	...	0.0045257678707492	736	...	0.004152297978109361
593	...	0.004187241302856962	641	...	0.004364581596963905	689	...	0.004529069753476985	737	...	0.004138699280465085
594	...	0.00419100124389432	642	...	0.004367991749745394	690	...	0.00453239748577007	738	...	0.004142908235571285
595	...	0.00419476250971665	643	...	0.00437151713371971	691	...	0.0045375915397946	739	...	0.00411113282963325
596	...	0.00419849185510963	644	...	0.00437499600021283	692	...	0.0045			

Error		Error		Error		Error	
Epoch	MSE	Epoch	MSE	Epoch	MSE	Epoch	MSE
769	0.003741913279739026	817	0.0032927709803745	865	0.0029725822248344576	913	0.0027430135118813375
770	0.003708812678099445	818	0.0032849461414345372	866	0.0029669839521500084	914	0.00273899196430625
771	0.003719927850216568	819	0.0032771791518528084	867	0.002981428454264883	915	0.002734990295372994
772	0.003709052455117721	820	0.0032694692682059	868	0.002955911366633858	916	0.0027310345760009277
773	0.0036982528049508643	821	0.0032618125190970165	869	0.0029504336165899344	917	0.00272709450318064
774	0.0036875307815502434	822	0.00325421086191542	870	0.002949499493193859	918	0.002723182007431468
775	0.003676868247790206	823	0.003246663918420736	871	0.002939564539935	919	0.00271925770988215
776	0.003666317859416716	824	0.0032391709473372303	872	0.002934217359423883	920	0.002714372999866673
777	0.003655829424806952	825	0.003231725607810513	873	0.00292876114067245	921	0.0027116576867987867
778	0.003645417008748877	826	0.0032243332348204	874	0.002923568699767731	922	0.0027078023237025227
779	0.0036350810801356547	827	0.00321699424312956	875	0.0029182988944083336	923	0.002704248636779783
780	0.00362482869653057	828	0.0032097086534652224	876	0.00291306846710536	924	0.0027002735635980493
781	0.0036146503811847665	829	0.0032024601906416447	877	0.002907873214664286	925	0.002696549151426867
782	0.0036045441421745007	830	0.0031952566074650336	878	0.00290207143451119492	926	0.0026928481996363605
783	0.0035945123764786454	831	0.003188121354460103	879	0.0029875932819872902	927	0.002689171202323928
784	0.0035845344980367532	832	0.003180994469610486	880	0.002892500914002793	928	0.0026855234323640356
785	0.0035746324983812207	833	0.003173938172349014	881	0.002887457549937524	929	0.0026818990218299775
786	0.00355648043775086246	834	0.00316693361553574	882	0.002882443012759223	930	0.002678297863938858
787	0.00355504914683033	835	0.003159895429051013	883	0.0028774647421474777	931	0.002674723448433301
788	0.00354536482810074726	836	0.0031530878815239087	884	0.002872520029356588	932	0.0026711735589881333
789	0.003535749642851251	837	0.003146238795412073	885	0.002867146990582766	933	0.002687146990582766
790	0.00352620352860059	838	0.0031394318158572106	886	0.0028627446859588366	934	0.0026641476539605243
791	0.003516723853026537	839	0.0031326739160809107	887	0.0028579075220338496	935	0.002660671295496674
792	0.0035073110774981077	840	0.003125964339407761	888	0.0028531046979091033	936	0.0026572187827709881
793	0.003497989859822866	841	0.0031193036534515024	889	0.0028493308362566148	937	0.002653789577945818
794	0.0034887308153740506	842	0.0031126953771886126	890	0.0028435906494134306	938	0.002650384805065487
795	0.003479533781697552	843	0.003105130446562608	891	0.00283881518953701	939	0.0026470459253089094
796	0.0034704038203525888	844	0.003099611253922739	892	0.0028342055079851928	940	0.002643462748192758
797	0.003461383904258535	845	0.003093132560855253	893	0.0028295537772119106	941	0.002640308051025293
798	0.00345233606589215417	846	0.0030866948762917921	894	0.002824935047754644	942	0.0026369938240936933
799	0.0034433398247122936	847	0.003080304297898507	895	0.0028203492491223115	943	0.00263370239597489
800	0.00343452464169787004	848	0.0030739575129847796	896	0.00281579450791986	944	0.0026304326396541337
801	0.0034257131339244873	849	0.003067654877720273	897	0.0028112701313615808	945	0.002627185727543291
802	0.0034169819716500196	850	0.003061395238355347	898	0.00280842055079851928	946	0.002623959153005797
803	0.003408274641676768	851	0.0030551800785200895	899	0.00280231834730176	947	0.002620753739063718
804	0.0033996495401923694	852	0.0030490165210566	900	0.0027719748430738313	948	0.0026175710538667335
805	0.003391081374119479	853	0.0030428848548375914	901	0.002793493073470019	949	0.00261440783284381
806	0.00338257498685592935	854	0.0030368023616171116	902	0.002789124573793802	950	0.0026112678510613885
807	0.0033741288102731436	855	0.00303076412078939125	903	0.00278478455821193	951	0.002608150465193187
808	0.0033657358931858273	856	0.003024764018645613	904	0.002780477819795102	952	0.002605051426261745
809	0.0033574054380718656	857	0.003018808809460515	905	0.002776198109164394	953	0.0026019791966474417
810	0.003349133093877746	858	0.003012896197398100	906	0.002771947843025685	954	0.002598924113864166
811	0.00334091762093207	859	0.0030070257997638163	907	0.00276772832133485	955	0.0025958892108381
812	0.003332750560507043	860	0.0030011892368588175	908	0.002763539205554034	956	0.002592874001134625
813	0.003324638040798696	861	0.00299538802276187	909	0.002759380583489553	957	0.00258987972373892
814	0.0033165854100122853	862	0.0029886240198794507	910	0.002755246864196416	958	0.0025869054623770533
815	0.003308590140434821	863	0.002983901717545816	911	0.00275114149833053	959	0.0025839521481746956
816	0.0033006533428736585	864	0.002978221304705639	912	0.002747063997261644	960	0.0025810199453594096

Error	
Epoch	MSE
953	0.0026019791966474417
954	0.002598924113864166
955	0.0025958892108381
956	0.002592874001134625
957	0.00258987972387392
958	0.002589054623770533
959	0.0025839521481746956
960	0.0025810194535940906
961	0.00257811098771712
962	0.002575222044162056
963	0.002572352496474363
964	0.002569803632139673
965	0.0025666756673639254
966	0.00256386871407506243
967	0.0025610794743611
968	0.002558311026781618
969	0.00255561492026337
970	0.0025528303095985856
971	0.00255011935032462
972	0.002547426036887392
973	0.002544750542547053
974	0.002542093109510389
975	0.002539456319872497
976	0.0025368374888965727
977	0.002534236375826507
978	0.002531853504413225
979	0.00252908874784741
980	0.002526542384797888
981	0.002524014381287398
982	0.002521503207664385
983	0.0025190090952687642
984	0.0025165326800341608
985	0.0025140733965468135
986	0.0025116316723174654
987	0.0025092067765299423
988	0.0025067989294178653
989	0.00250440476955269786
990	0.002502033931466882
991	0.0024998744624051484
992	0.002497331739148054
993	0.0024950072221541658
994	0.0024926998137416987
995	0.0024904089679927076
996	0.0024881346261373563
997	0.0024858749018106543
998	0.0024838316463829837
999	0.00248140520273663
1000	0.0024791942808715197

LAMPIRAN 5

Gambar 4.3. Nilai MSE dalam lapisan *output* pada pengujian kedua

Error			Error			Error			Error		
Epoch	MSE		Epoch	MSE		Epoch	MSE		Epoch	MSE	
1001	0.0024769988314805587		1049	0.0023881721423892675		1097	0.002359484444577753		1145	0.0022836419660190828	
1002	0.0024748187694509185		1050	0.0023866308976828753		1098	0.0023248835473573813		1146	0.0022829345997769384	
1003	0.002472653503327395		1051	0.002385106871470543		1099	0.00232382678751587		1147	0.0022822338427073894	
1004	0.0024705035087021258		1052	0.0023835822016482233		1100	0.002322781596372593		1148	0.0022815396322593095	
1005	0.0024683685139438089		1053	0.0023820748516762574		1101	0.002321742517245668		1149	0.002280851908824913	
1006	0.00246624855061651245		1054	0.0023805785577988214		1102	0.00232071268890986		1150	0.002280170312321037	
1007	0.00246411434719107		1055	0.0023790932627120744		1103	0.002319960263217387		1151	0.002279494283355477	
1008	0.002462050397639763976		1056	0.002377619532246646		1104	0.00231867439447682		1152	0.002278276859269413	
1009	0.0024598924478575695		1057	0.0023761569684679783		1105	0.002317670502503517		1153	0.002276108442080757	
1010	0.002457924082456568		1058	0.002374705127029196		1106	0.002316673190538363		1154	0.002277511171643575	
1011	0.00245588061303288		1059	0.0023732640892381897		1107	0.002315831556412484		1155	0.0022786820993133754	
1012	0.0024538520994814767		1060	0.002371834248841613		1108	0.0023170472450513422		1156	0.0022762189840815954	
1013	0.0024518375306721994		1061	0.002370315611054354		1109	0.0023137271288443807		1157	0.00227581786212427	
1014	0.002449836907413043		1062	0.0023690078490247544		1110	0.00231276097161332		1158	0.002270508333391075	
1015	0.002447850828941695		1063	0.0023676102234507545		1111	0.002311802736410137		1159	0.0022747538416020845	
1016	0.0024458796294026363		1064	0.0023662229474061986		1112	0.0023108521621423904		1160	0.0022745322412102485	
1017	0.00244392309153571527		1065	0.0023648465509249465		1113	0.002309106616616084		1161	0.002274149435012674	
1018	0.00244197998323566		1066	0.0023638092588093		1114	0.002308976705371751		1162	0.002274405656511581	
1019	0.002440050881777284		1067	0.0023621259175563673		1115	0.002308055655718397		1163	0.002274481536588985	
1020	0.0024381350517851964		1068	0.002360707289563809		1116	0.002307132218243242		1164	0.002274836047582547	
1021	0.002436232187200044		1069	0.002359445941207643		1117	0.002306218141331478		1165	0.002274862799888727	
1022	0.002434344219299584		1070	0.002358120121510047		1118	0.00230531964948264		1166	0.002275156293918427	
1023	0.00243246893764724726		1071	0.0023568049483194495		1119	0.0023044236345296995		1167	0.002275509014477985	
1024	0.00243060685893010485		1072	0.0023554994800177053		1120	0.0023035361153204525		1168	0.002275912774430241	
1025	0.002427566795920587		1073	0.00235420387531685		1121	0.002306557742933234		1169	0.00227637448627295	
1026	0.0024269205862120283		1074	0.002352817919126421		1122	0.0023017827921429685		1170	0.0022768691660468047	
1027	0.002425097050233297		1075	0.002351641721125195		1123	0.00230097412094978245		1171	0.002277414131887096	
1028	0.0024232874077684797		1076	0.0023503751935786505		1124	0.00230056974325474		1172	0.002277998794728573	
1029	0.002421490910671201		1077	0.00234911796020188		1125	0.00229920787954834805		1173	0.0022768190448147187	
1030	0.0024197076552761373		1078	0.002347970674572136		1126	0.0022983622100834606		1174	0.002276927256407	
1031	0.002417937032818893		1079	0.002346632559143689		1127	0.002397524176862221		1175	0.0022799527498138642	
1032	0.002416178893494948		1080	0.002345401944949442		1128	0.0023996386938093982		1176	0.00228066074920311	
1033	0.002414422859020081		1081	0.002344186171835905		1129	0.002395870373395939		1177	0.002281409328633175	
1034	0.002412699673287394		1082	0.002342977530136623		1130	0.002395054246727553		1178	0.002282179699513305	
1035	0.0024109785987594177		1083	0.002341778224384312		1131	0.002394245249228836		1179	0.002282974349153255	
1036	0.0024092702089450048		1084	0.0023405888175925764		1132	0.0023934432438140744		1180	0.002283736072495893	
1037	0.0024075447383454307		1085	0.002339408306507106		1133	0.00239254581265253		1181	0.002284634226389425	
1038	0.002405890836321755		1086	0.002338237209807155		1134	0.002391866747740216		1182	0.002285495195702566	
1039	0.002404219409702395		1087	0.002337058731464094		1135	0.002391079731439002		1183	0.0022803750466558053	
1040	0.002402560108557404		1088	0.002335921773320953		1136	0.00229030084484315		1184	0.0022872748595775534	
1041	0.002400912726032784		1089	0.0023347795869731163		1137	0.002389538743228974		1185	0.0022881948432167845	
1042	0.002399270855234128		1090	0.0023336455160687056		1138	0.0022887746202741984		1186	0.002289132219833566	
1043	0.002397655772653373		1091	0.00233256052037651374		1139	0.00228808480751516		1187	0.00229008509831933	
1044	0.0023960455109439476		1092	0.002331403787397104		1140	0.0022877276069058518		1188	0.002291052833156447	
1045	0.002394447473874697		1093	0.00233029616719232		1141	0.002286537202250477		1189	0.002290355675545017	
1046	0.002392860572658053		1094	0.00232919944641593575		1142	0.00228503640917743		1190	0.0022930376298126356	
1047	0.002391286724520688		1095	0.002328104800574713		1143	0.0022850734599775		1191	0.0022940543270927	
1048	0.002389724495512922		1096	0.002327023100103865		1144	0.00228435007251872		1192	0.00229508744892008	
Error			Error			Error			Error		
Epoch	MSE		Epoch	MSE		Epoch	MSE		Epoch	MSE	
1193	0.00229613350423294		1241	0.002359152444251726		1289	0.00244392574123207		1337	0.002550146751284126	
1194	0.002297191793723106		1242	0.00236069586963034		1290	0.0024459332994085345		1338	0.0025525466738991535	
1195	0.002299264274350696		1243	0.0023622474684614933		1291	0.0024479431398603086		1339	0.002554950942622046	
1196	0.00229934966153886		1244	0.0023638087684593346		1292	0.0024499562851363222		1340	0.0025733367307453668	
1197	0.0023004491687479227		1245	0.0023635773240293		1293	0.0024519946749453603		1341	0.0025597941703943143	
1198	0.002301560849676464		1246	0.0023669595273162093		1294	0.00245044678495717963		1342	0.00256221678542813	
1199	0.002302684443290901		1247	0.0023685407295710458		1295	0.0024580756154756528		1343	0.00254643248203725517	
1200	0.0023038210636434553		1248	0.0023701335482076294		1296	0.002458129830738792		1344	0.0025671066584726713	
1201	0.002304961233956974		1249	0.0023717332291978516		1297	0.002460193620671684		1345	0.00259556998645009	
1202	0.002306124611220687		1250	0.00237334054575855		1298	0.00246226916103017		1346	0.002572018938463065	
1203	0.002307293069564103		1251	0.002374554444591583		1299	0.00246435038932256		1347	0.002574488501883467	
1204	0.0023084723084519814		1252	0.0023765773320971796		1300	0.00246643429363253		1348	0.002578666632019135	
1205	0.0023096591863118214		1253	0.002378215963180398		1301	0.0024685361859645091963		1349	0.0025794504569755165	
1206	0.002310860041693786		1254	0.002379865917613296		1302	0.0024706312401794146		1350	0.00258194502309922	
1207	0.002312070466662814		1255	0.0023816526188668658		1303	0.0024727493850197103		1351	0.002584474958215447	
1208	0.0023128943721983274		1256	0.0023831953040977157		1304	0.00247487221618048		1352	0.002589661833041206	
1209	0.0023145199710323874		1257	0.002384871313620098		1305	0.00247704646638635		1353	0.002589460317314076	
1210	0.002315762202909712		1258	0.002388553716377076		1306	0.0024791653913954346		1354	0.002592034337391795	
1211	0.002317055796418857		1259	0.0023882454689							

Error	Epoch	MSE	Error	Epoch	MSE	Error	Epoch	MSE	Error	Epoch	MSE
	1385	0.0026742727484612107		1433	0.0026412642288347406		1481	0.00250593321393769		1529	0.0024084023885711362
	1386	0.002677043593017717		1434	0.002638075185082674		1482	0.002503545806587558		1530	0.0024067020436032358
	1387	0.002679810898759789		1435	0.0026348942875171513		1483	0.002501107298660937		1531	0.0024050133039192227
	1388	0.002682585193721697		1436	0.00263317192609491424		1484	0.0024988124273010568		1532	0.002403337285912128
	1389	0.002685374164122553		1437	0.002628560630748879		1485	0.00249647196539094		1533	0.0024016741218991634
	1390	0.002688170556225464		1438	0.002625403127288006		1486	0.002494148308954267		1534	0.0024000250028909055
	1391	0.002690965494297859		1439	0.002622264204809231		1487	0.0024918411264811284		1535	0.002398388478418704
	1392	0.002693768988791167		1440	0.0026191386475283		1488	0.002498503105684648		1536	0.0023987638835967198
	1393	0.002695519456543494		1441	0.0026160272591815		1489	0.00249725464001424		1537	0.002395150189719634
	1394	0.002699404025265198		1442	0.002612930104905014		1490	0.002495016010554491		1538	0.00239354583828907
	1395	0.002702231931510757		1443	0.0026098476843092505		1491	0.002482772431239265		1539	0.0023919591021411109
	1396	0.002705060382061032		1444	0.0026063781651843764		1492	0.0024805507158361		1540	0.002390382749485574
	1397	0.002707895567011054		1445	0.0026037317461882998		1493	0.0024783343209574197		1541	0.002388181199837465
	1398	0.0027107376221513816		1446	0.0026006984811330893		1494	0.0024761518662309425		1542	0.002387265092473757
	1399	0.002712972037927127		1447	0.002597683971415		1495	0.00247547434915283		1543	0.002385723552375967
	1400	0.002714523959883657		1448	0.0025946849831403625		1496	0.00247182980438544		1544	0.0023841923718682
	1401	0.0027155048865950824		1449	0.002591704229618978		1497	0.002469663377670866		1545	0.0023826715169648145
	1402	0.0027161542463153407		1450	0.002587847209631282		1498	0.002467534090848464		1546	0.002381162491316339
	1403	0.002716285866219884		1451	0.0025867982469128945		1499	0.0024654186381013873		1547	0.0023796645284919426
	1404	0.0027160316057045254		1452	0.002582873126657611		1500	0.0024633171440126707		1548	0.0023781777139626594
	1405	0.002715423113507515		1453	0.0025799631535898		1501	0.00249175464014883		1549	0.002375723552375967
	1406	0.002714502859700084		1454	0.00257707008092395		1502	0.002459175845609338		1550	0.00237323709560645
	1407	0.0027132969524124		1455	0.002574205488739176		1503	0.0024571005393718513		1551	0.0023737388699235737
	1408	0.00271183747290526		1456	0.00257135233829844		1504	0.002455056585705873		1552	0.002372341938747517
	1409	0.0027101508438760417		1457	0.0025685185695716716		1505	0.002453028056774773		1553	0.002370108793469485
	1410	0.002708235782324067		1458	0.00256570360717708		1506	0.00245103049225662		1554	0.0023694906425329797
	1411	0.0027061954789468585		1459	0.00256299758179561		1507	0.0024490117461335777		1555	0.0023680809735080297
	1412	0.002703969234635707		1460	0.0025601391409323		1508	0.002447024935662888		1556	0.0023668623365496932
	1413	0.0027015958442893827		1461	0.0025573728515483324		1509	0.00244505277413785		1557	0.0023652945484231337
	1414	0.0026990995308215433		1462	0.002554632690237725		1510	0.002443092996622846		1558	0.002363917539886873
	1415	0.002696494495153827		1463	0.002551910424694275		1511	0.002436915324295723		1559	0.002362459434924345
	1416	0.00269379349899888		1464	0.0025459206577268797		1512	0.0024392104084024		1560	0.002381190528492322
	1417	0.0026910075820229946		1465	0.00254562136112026		1513	0.0024370208867390936		1561	0.0023598409923938925
	1418	0.002688148921682415		1466	0.002543854601840915		1514	0.0024353844496487		1562	0.002356502310584864
	1419	0.002685226867520446		1467	0.00254420255061023		1515	0.0024334392694301306		1563	0.002357174349017196
	1420	0.0026822506861257147		1468	0.0025368577796636581		1516	0.002431614214886577		1564	0.002358585681657217
	1421	0.002679228862501344		1469	0.0025359516507460405		1517	0.0024297893390184		1565	0.002354500896791716
	1422	0.00267617605866042		1470	0.0025333529632791344		1518	0.002427890346271834		1566	0.002353254224307238
	1423	0.0026730719526222506		1471	0.00253072056848106		1519	0.00242602632508254404		1567	0.00235196824362224
	1424	0.002669950023533783		1472	0.00252820308669579		1520	0.0024228415260952705		1568	0.0023506924871285594
	1425	0.0026686048440707889		1473	0.00252566221575242		1521	0.0024224482841870023		1569	0.0023494267912125596
	1426	0.0026636303588777536		1474	0.0025231339589204173		1522	0.002420631561078151		1570	0.00234817123103034
	1427	0.002660445912806343		1475	0.002520622730622273		1523	0.002421847200416934		1571	0.002346921494398177
	1428	0.0026572530699576		1476	0.002515183005735431		1524	0.0024170579547147425		1572	0.00234561588145215
	1429	0.00265405477794156		1477	0.00251565408265904		1525	0.00241531690452666		1573	0.002344500637379343
	1430	0.00265085477166527		1478	0.0025151291740333		1526	0.002413578220862233		1574	0.00234228857931502
	1431	0.0026476553626404277		1479	0.002510758984033783		1527	0.00241838434944377		1575	0.0023420169294356666
	1432	0.0026444575922930052		1480	0.002508337843237157		1528	0.0024241014853810754		1576	0.0023408152417173215
Error	Epoch	MSE	Error	Epoch	MSE	Error	Epoch	MSE	Error	Epoch	MSE
	1577	0.0023396228938962954		1625	0.0022924158363252104		1673	0.0022614090990092176		1721	0.002242629523831404
	1578	0.002338439987812009		1626	0.00229162973995109		1674	0.0022689513259105108		1722	0.0022423549363391554
	1579	0.002337266921507358		1627	0.002290832654372099604		1675	0.0022690379484463463		1723	0.0022420632468211417
	1580	0.0023361025415983908		1628	0.00229005145741235		1676	0.002259904186731669		1724	0.002241790525922585
	1581	0.0023334974082043444		1629	0.002289276736784263		1677	0.00225951557763923		1725	0.002241518649913946
	1582	0.0023338025368995703		1630	0.002288509445253		1678	0.00225893213538959		1726	0.002244250673175747
	1583	0.00233266817925749		1631	0.002287744883588592		1679	0.00225845390522706		1727	0.002240966597119196
	1584	0.00233150427987951		1632	0.0022869951311744008		1680	0.002257980390086382		1728	0.002240726517537553
	1585	0.0023304227953015104		1633	0.0022862479220015796		1681	0.0022575126700470384		1729	0.00224074396466463
	1586	0.00233293147544473734		1634	0.002285603562462406		1682	0.002250704521040637		1730	0.002240217511721202
	1587	0.002328215539551684		1635	0.00228877377678579		1683	0.002256591390998734		1731	0.00223990322582523
	1588	0.002327121452138755		1636	0.0022884647287285		1684	0.00225613820631694		1732	0.002239724013505939
	1589	0.00232261496222022212		1637	0.0022833227200759		1685	0.002255609227425927		1733	0.002235822709414614
	1590	0.002324987458204910		1638	0.0022861921896054		1686	0.00225526716664021		1734	0.002239245206994123
	1591	0.0023239020052289093		1639	0.0022881904231954748		1687	0.002254808311009037		1735	0.0022390114115061538
	1592	0.0023228423783005968		1640	0.00228120105744364		1688	0.00225437344336773		1736	0.002238781329224037
	1593	0.00232179319407035		1641	0.002280507190519808		1689	0.00225228310253167691		1737	0.0022385494070707135
	1594	0.0023227033817504		1642	0.0022789135416762948		1690	0.002253210253167691		1738	0.0022383317223034
	1595	0.002319719114203177		1643	0.0022784589954942815		1691	0.0022508637691940427		1739	0.002238112865996988
	1596	0.00231689442342667		1644	0.002278459954942815		1692	0.002256286256239833		1740	0.002237897347464497
	1597	0.00231778324771653		1645	0.002277884039754482						

Error	Epoch	MSE	Error	Epoch	MSE	Error	Epoch	MSE	Error	Epoch	MSE
	1769	0.0022331328669258402		1817	0.002230700037122662		1865	0.002232603351596056		1913	0.0022322696596685557
	1770	0.0022330167356600732		1818	0.002230711622141345		1866	0.0022325171956563685		1914	0.00223244639451955
	1771	0.002232930621513036		1819	0.0022307255147082234		1867	0.002232420288249943		1915	0.002232631394120053
	1772	0.002232793312535481		1820	0.002230741722978905		1868	0.0022324975344786		1916	0.0022328540622854
	1773	0.00223286385468472		1821	0.0022307602116495703		1869	0.002232215193373472		1917	0.0022332398745572592
	1774	0.0022325822373339068		1822	0.0022307809662530094		1870	0.002232115193373472		1918	0.0022332398745572592
	1775	0.002232481048730728		1823	0.00223080396503408		1871	0.002232006122934364		1919	0.002233450368211842
	1776	0.002232382798847819		1824	0.0022308292199587597		1872	0.00223189749364299		1920	0.0022336901309893015
	1777	0.002232287488815622		1825	0.0022308568470301876		1873	0.002231887662523187		1921	0.002233925648842523
	1778	0.0022321950912287435		1826	0.002230868636692108		1874	0.0022316008470169915		1922	0.00223476353562569
	1779	0.0022321056002211717		1827	0.00223091887932096		1875	0.002231574534487375		1923	0.0022344321476490806
	1780	0.002232019007491669		1828	0.0022309531344668733		1876	0.00223147079592151		1924	0.00223469720076682
	1781	0.00223195328026655		1829	0.00223099548426123		1877	0.002231370420056881		1925	0.0022347907879946241
	1782	0.002231954396407558		1830	0.002231028118629465		1878	0.0022312783723406194		1926	0.002235253265224822
	1783	0.0022317763441365482		1831	0.002231068851607123		1879	0.0022311812922742285		1927	0.002235542630517323
	1784	0.002231701102857728		1832	0.0022311116765910675		1880	0.002231039503688914		1928	0.0022358443575949896
	1785	0.002231628679881904		1833	0.002231156613028512		1881	0.002231093928486		1929	0.0022351527766738666
	1786	0.002231559043280695		1834	0.002231203758230596		1882	0.002230933821280097		1930	0.00223674509955101
	1787	0.0022314921683377815		1835	0.00223125298177216		1883	0.00223086200760244		1931	0.00223579667712585
	1788	0.002231428043899937		1836	0.0022313042787717543		1884	0.0022307980445238313		1932	0.0022371311348978007
	1789	0.002231366499581375		1837	0.00223135771917059		1885	0.0022307399563981463		1933	0.00223747537646444
	1790	0.002231307974173564		1838	0.0022314131824509066		1886	0.002230868779773686		1934	0.002237280313574187
	1791	0.002231252003559337		1839	0.002231470747487837		1887	0.0022305441108326477		1935	0.002238190157184844
	1792	0.002231198719854771		1840	0.002231530330799		1888	0.00223068031399894		1936	0.00223566012402889
	1793	0.002231148181775086		1841	0.0022315919530449214		1889	0.00223057645158337		1937	0.00223894071733344
	1794	0.0022311003141593862		1842	0.002231655842702588		1890	0.00223054262356427		1938	0.002239320171784883
	1795	0.00223105076553312		1843	0.002231721107251629		1891	0.0022305390294239262		1939	0.002239725397065683
	1796	0.0022310124741751294		1844	0.002231788723762043		1892	0.0022305131478758423		1940	0.00224012888083217
	1797	0.002230972538317165		1845	0.002231858275132964		1893	0.00223051302996421		1941	0.002240543419469165
	1798	0.0022309352459734734		1846	0.0022319297536309		1894	0.002230539333369485		1942	0.002240957309181096
	1799	0.00223090052459129		1847	0.0022320031513024906		1895	0.002230562107746464		1943	0.002241396798877727
	1800	0.0022308683435229757		1848	0.0022320784600821		1896	0.002230579075251744		1944	0.0022418365468447094
	1801	0.0022308387279420415		1849	0.00223155755566923		1897	0.002230612474170603		1945	0.002242850608539145
	1802	0.002230811640253834		1850	0.002232243984510676		1898	0.0022306515892259035		1946	0.0022427421188204576
	1803	0.0022307870635616164		1851	0.002232315874666964		1899	0.002230705131380118		1947	0.0022432075060427863
	1804	0.0022307649721856845		1852	0.0022323886795091900		1900	0.002230756787318933		1948	0.0022436811648183365
	1805	0.002230745378956464		1853	0.002232492410703854		1901	0.0022308219202304023		1949	0.00224416346944744
	1806	0.0022307328320196945		1854	0.002232599939078225		1902	0.00223089594131323		1950	0.002244654242668483
	1807	0.0022307137190221847		1855	0.002232597815670113		1903	0.0022309781110311042		1951	0.002245159304395141
	1808	0.002230701580560513		1856	0.0022327488012583415		1904	0.002231068594822941		1952	0.00224566467804943
	1809	0.0022306918855113517		1857	0.00223281268217203		1905	0.0022311680757453107		1953	0.002246185176828806
	1810	0.0022306846083849485		1858	0.0022328520729656114		1906	0.0022312757995350886		1954	0.002246714882854511
	1811	0.002230677357685227		1859	0.0022320839391144323		1907	0.002231391811258502		1955	0.0022472537239120946
	1812	0.002230577231678334		1860	0.0022320839413081272		1908	0.002231517419325297		1956	0.0022478010450808395
	1813	0.0022306770912451255		1861	0.0022328412984831385		1909	0.002231649970422143		1957	0.00224833391910567
	1814	0.0022306793174532165		1862	0.00223280146772492		1910	0.002231792158118095		1958	0.002248921020598196
	1815	0.0022306833938620444		1863	0.00223274745675611		1911	0.002231942573629002		1959	0.0022494948200673147
	1816	0.0022306908014138494		1864	0.002232680570326909		1912	0.002232101872144492		1960	0.0022500786624125284
Error											
Epoch											
MSE											
1953	...	0.00224618517682806		1963	...	0.002251890467773653		1973	...	0.0022551054145915597	
1954	...	0.002246714982854511		1964	...	0.0022521547195384372		1974	...	0.0022557765290249867	
1955	...	0.0022472537239120946		1965	...	0.0022531470116916724		1975	...	0.0022544432297078	
1956	...	0.00224780104508395		1966	...	0.0022537900302563463		1976	...	0.00225557675290249867	
1957	...	0.0022483563391910567		1967	...	0.0022544432297078		1977	...	0.00225571484228852733	
1958	...	0.002248921020598196		1968	...	0.0022551054145915597		1978	...	0.002258555977990911	
1959	...	0.0022494948200673147		1969	...	0.002256457721028514		1979	...	0.0022592729865078933	
1960	...	0.0022500786624125284		1970	...	0.0022537900302563463		1980	...	0.002259998584765678	
1961	...	0.0022505734542980618		1971	...	0.0022537900302563463		1981	...	0.002259998584765678	
1962	...	0.002251276873594833		1972	...	0.0022537900302563463		1982	...	0.002259998584765678	
1963	...	0.002251890467773653		1973	...	0.0022537900302563463		1983	...	0.002259998584765678	
1964	...	0.0022521547195384372		1974	...	0.0022544432297078		1984	...	0.002259998584765678	
1965	...	0.0022531470116916724		1975	...	0.0022551054145915597		1985	...	0.002259998584765678	
1966	...	0.0022537900302563463		1976	...	0.00225571484228852733		1986	...	0.002259998584765678	
1967	...	0.0022544432297078		1977	...	0.002256457721028514		1987	...	0.002259998584765678	
1968	...	0.0022551054145915597		1978	...	0.0022571484228852733		1988	...	0.002259998584765678	
1969	...	0.0022557765290249867		1979	...	0.0022578479129591535		1989	...	0.002259998584765678	
1970	...	0.002256457721028514		1980	...	0.002258555977990911		1990	...	0.002259998584765678	
1971	...	0.0022571484228852733		1981	...	0.002259998584765678		1991	...	0.002259998584765678	
1972	...	0.0022578479129591535		1982	...	0.002259998584765678		1992	...	0.002259998584765678	
1973	...	0.002258555977990911		1983	...	0.002259998584765678		1993	...	0.002259998584765678	
1974	...	0.0022592729865078933		1984	...	0.002259998584765678		1994			

LAMPIRAN 6 - SOURCE CODE

-step 1 Proses perhitungan jaringan pada setiap *unit hidden layer*

```
for (int i = 0; i < wNeuron.length; i++) {  
    double tmpNet = 0;  
    for (int j = 0; j < wNeuron[0].length; j++) {  
        tmpNet += (x[h][j] * wNeuron[i][j]);  
    }  
    net[i] = tmpNet + wNeuronBias[i];  
}
```

-step 2 Proses perhitungan fungsi aktivasi pada *unit hidden layer*

```
double fin[] = new double[net.length];  
for (int i = 0; i < net.length; i++) {  
    double exp = 1 + (Math.exp(-net[i]));  
    fin[i] = (double) 1 / (double) exp;  
}
```

-step 3 Perhitungan nilai jaringan pada *unit hidden layer* ke *unit output layer*

```
for (int i = 0; i < fin.length; i++) {  
    tempK += fin[i] * wOutput[i];  
}  
netKOutput = wOutputBias[0] + tempK;
```

-step 4 Proses perhitungan fungsi aktivasi pada *unit output layer*

```
fout_net = 1 / (1 + (double) Math.exp(-netKOutput));
```

-step 5 Proses perhitungan untuk *error* pada *unit output layer*

```
cek_error = t[h] - fout_net;
```

-step 6 Proses perhitungan *mean square error*

```
double kuadrat_error = 0.5 * (Math.pow(t[h] - fout_net, 2));
```

-step 7 Proses perhitungan *derivative partial error* terhadap *weight* dan *bias* pada *unit output layer*

```
for (int i = 0; i < gt.length; i++) {
    gt[i] = (-t[h] - fout_net) * fout_net * (1 - fout_net) * fin[i];
}
double gtBias = -(t[h] - fout_net) * fout_net * (1 - fout_net);
```

-step 8 Proses perhitungan estimasi moment pertama *weight* dan *bias* pada *unit output layer*

```
for (int i = 0; i < m.length; i++) {
    if (i == 0) {
        m[i] = beta1 * 0 + 1 - beta1 * gt[i];
    } else {
        m[i] = beta1 * m[i - 1] + 1 - beta1 * gt[i];
    }
}
mBias = beta1 * mBias + (1 - beta1) * gtBias;
```

-step 9 Proses perhitungan koreksi *weight* dan *bias* estimasi momen pertama pada *unit output layer*

```
for (int i = 0; i < m.length; i++) {
    mKoreksi[i] = m[i] / (double) (1 - Math.pow(beta1, (i + 1)));
}
double mKoreksiBias = mBias / (double) (1 - beta1);
```

-step 10 Proses perhitungan estimasi moment kedua *weight* dan *bias* pada *unit output layer*

```
for (int i = 0; i < v.length; i++) {
    for (int j = 0; j < v[0].length; j++) {
        v[i][j] = beta2 * v[i][j] + (1 - beta2) * Math.pow(gt[i][j],2);
    }
}
for (int i = 0; i < vBias.length; i++) {
    vBias[i] = beta2 * vBias[i] + (1 - beta2) * Math.pow(gtBias[i],2);
}
```

-step 11 Proses perhitungan koreksi *weight* dan *bias* estimasi momen kedua pada *unit output layer*

```
for (int i = 0; i < v.length; i++) {
    for (int j = 0; j < v[i].length; j++) {
        v[i][j] = v[i][j] / (double) (1 - Math.pow(beta2, (j + 1)));
        if (Double.isInfinite(v[i][j])) {
            v[i][j] = 0;
        }
    }
}
for (int i = 0; i < vBias.length; i++) {
    vBias[i] = vBias[i] / (double) (1 - Math.pow(beta2, (i + 1)));
}
```

-step 12 Proses Perhitungan *derivative partial error* terhadap *weight* dan *bias* pada *hidden layer*

```
for (int i = 0; i < x[h].length; i++) {
    for (int j = 0; j < wNeuronBias.length; j++) {
```

```

        DO[i][j] = -(t[h] - fout_net) * fout_net * (1 - fout_net) *
wNeuronBias[j] + (fin[j] * ((1 - fin[j])));

    }

}

for (int i = 0; i < DOBias.length; i++) {

    DOBias[i] = -(t[h] - fout_net) * fout_net * (1 - fout_net) * wNeuronBias[i];

}

```

-step 13 Proses perhitungan estimasi moment pertama *weight* dan *bias* pada *unit hidden layer*

```

for (int i = 0; i < m.length; i++) {

    if (i == 0) {

        m[i] = beta1 * 0 + 1 - beta1 * DO[i][j];

    } else {

        m[i] = beta1 * m[i - 1] + 1 - beta1 * DO[i][j];

    }

}

mBias = beta1 * mBias + (1 - beta1) * DOBias[i];

```

-step 14 Proses perhitungan koreksi *weight* dan *bias* estimasi momen pertama pada *unit output layer*

```

for (int i = 0; i < m.length; i++) {

    mKoreksi[i] = m[i] / (double) (1 - Math.pow(beta1, (i + 1)));

}

```

double mKoreksiBias = mBias / (double) (1 - beta1);

-step 15 Proses perhitungan estimasi moment kedua *weight* dan *bias* pada *unit hidden layer*

```

for (int i = 0; i < v.length; i++) {

    for (int j = 0; j < v[0].length; j++) {

```

```

v[i][j] = beta2 * v[i][j] + (1 - beta2) * Math.pow(DO[i][j], 2);

}

}

for (int i = 0; i < vBias.length; i++) {

    vBias[i] = beta2 * vBias[i] + (1 - beta2) * Math.pow(DOBias[i], 2);

}

```

-step 16 Proses perhitungan koreksi *weight* dan *bias* estimasi momen kedua pada *unit hidden layer*

```

for (int i = 0; i < v.length; i++) {

    for (int j = 0; j < v[i].length; j++) {

        v[i][j] = v[i][j] / (double) (1 - Math.pow(beta2, (j + 1)));

        if (Double.isInfinite(v[i][j])) {

            v[i][j] = 0;

        }

    }

}

for (int i = 0; i < vBias.length; i++) {

    vBias[i] = vBias[i] / (double) (1 - Math.pow(beta2, (i + 1)));

}

```

-step 17 Pproses perhitungan perbaharuan parameter *weight* dan *bias*

```

for (int i = 0; i < wOutput.length; i++) {

    wOutput[i] = wOutput[i] - alfa * ((m[i]) / (double) Math.sqrt(vBias[i] +
epsilon));

}

```