

**PREDIKSI JUMLAH PERMINTAAN SEMEN MENGGUNAKAN
JARINGAN SYARAF TIRUAN *BACKPROPAGATION***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Mahendro Agni Giri Pawoko

NIM: 165150209111010



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PREDIKSI JUMLAH PERMINTAAN SEMEN MENGGUNAKAN JARINGAN SYARAF
TIRUAN *BACKPROPAGATION*

SKRIPSI

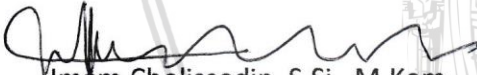
Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer


Disusun Oleh:
Mahendro Agni Giri Pawoko
NIM: 165150209111010

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Agustus 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I


Dosen Pembimbing II


Imam Cholissodin, S.Si., M.Kom.
NIK: 201201 850719 1 001


Ratih Kartika Dewi, S.T., M.Kom.
NIK: 201503 890520 2 001



Mengetahui
Ketua Jurusan Teknik Informatika


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



PERNYATAAN ORISINALITAS

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

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



Malang, 2 Agustus 2018

Mahendro Agni Giri Pawoko
NIM. 165150209111010



KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT yang telah memberikan rahmat, karunia dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul **“PREDIKSI JUMLAH PERMINTAAN SEMEN MENGGUNAKAN JARINGAN SYARAF TIRUAN BACKPROPAGATION”**. Skripsi ini diajukan sebagai ujian seminar skripsi dalam rangka untuk memperoleh gelar Sarjana Komputer di Fakultas Ilmu Komputer (FILKOM), Program Studi Teknik Informatika, Universitas Brawijaya Malang. Atas terselesaikannya skripsi ini, penulis mengucapkan rasa terima kasih kepada:

1. Imam Cholissodin, S.Si., M.Kom. selaku dosen pembimbing pertama skripsi yang telah meluangkan waktu dan juga memberikan banyak pengarahan bagi penulis.
2. Ratih Kartika Dewi, S.T., M.Kom. selaku dosen pembimbing kedua skripsi yang telah meluangkan waktu dan juga memberikan banyak pengarahan bagi penulis.
3. Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. selaku dekan Fakultas Ilmu Komputer Universitas Brawijaya.
4. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
5. Orang tua yang telah memberikan semangat dan juga memberikan banyak bantuan moril maupun materiil selama perkuliahan hingga penyelesaian tugas akhir ini.
6. Seluruh teman – teman SAP 2016 Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberi dorongan dan semangat.
7. Segenap staff dan karyawan di Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak membantu penulis dalam pelaksanaan penyusunan skripsi.

Penulis menyadari bahwa dalam menyelesaikan tugas akhir ini, tentunya masih jauh dari kesempurnaan. Maka dari itu kritik dan saran yang sifatnya membangun sangat diharapkan untuk kesempurnaan karya berikutnya.

Semoga penyusunan skripsi ini dapat berguna, bermanfaat bagi penulis maupun pembaca.

Malang, 2 Agustus 2018

Penulis

mahendrabagoes@gmail.com

ABSTRAK

Mahendro Agni Giri Pawoko, 2018. Prediksi Jumlah Permintaan Semen Menggunakan Jaringan Syaraf Tiruan *Backpropagation*. Fakultas Ilmu Komputer, Universitas Brawijaya, Malang.

Dosen Pembimbing : Imam Cholissodin, S.Si., M.Kom. dan Ratih Kartika Dewi, S.T., M.Kom.

Semen merupakan bahan yang penting pada proses pembangunan. Produksi semen di Indonesia terbilang cukup tinggi jika dibandingkan dengan jumlah konsumsinya. Kondisi ini mengakibatkan *oversupply*, yaitu kondisi dimana jumlah produksi lebih besar dari jumlah konsumsinya. Hal ini mengakibatkan turunnya harga semen dan penuhnya gudang penyimpanan oleh semen yang belum laku terjual. Hal ini membuat Asosiasi Semen Indonesia (ASI) mengeluarkan regulasinya untuk penghentian sementara produksi semen. Penelitian ini bertujuan untuk melakukan prediksi jumlah permintaan semen pada waktu berikutnya agar regulasi dapat dikeluarkan lebih cepat sehingga pabrik dapat menyesuaikan kapasitas produksinya tanpa harus menghentikan produksi. Banyak metode yang dapat digunakan untuk melakukan prediksi, salah satunya adalah Jaringan Syaraf Tiruan *Backpropagation* yang terbukti dapat memberikan hasil yang baik dalam melakukan prediksi, seperti prediksi jumlah permintaan semen dan produksi gula. Penelitian ini menggunakan Jaringan Syaraf Tiruan *Backpropagation* dengan arsitektur jaringan berupa 6 *neuron input*, 4 *neuron hidden* dan 1 *neuron output*. Parameter terbaik yang digunakan berupa *learning rate* sebesar 0,8, iterasi maksimum sebesar 200 dan interval bobot awal antara -1,4 sampai dengan 1,4. Nilai MSE prediksi terbaik yang didapat adalah sebesar 0,049064.

Kata kunci: semen, prediksi, jaringan syaraf tiruan, *backpropagation*, *oversupply*

ABSTRACT

Mahendro Agni Giri Pawoko, 2018. *Prediction the Amount of Cement Request Using Backpropagation Neural Network*. Fakultas Ilmu Komputer, Universitas Brawijaya, Malang.

Dosen Pembimbing : Imam Cholissodin, S.Si., M.Kom. dan Ratih Kartika Dewi, S.T., M.Kom.

Cement is an important material in the development process. Cement production in Indonesia is quite high compared to the amount of consumption. This condition results in oversupply, a condition where the amount of production is greater than the amount of consumption. This resulted in falling cement prices and warehouse full of cement that has not been sold. This makes the Indonesian Cement Association (ASI) to issue its regulation to temporary stop producing cement. This study aims to predict the amount of cement demand in the next time so that regulations can be issued more quickly, so the factory can adjust its production capacity without having to stop production. Many methods can be used to make predictions, for example is Backpropagation Neural Network which is proven to provide good results in predicting, such as predicting the amount of newspaper demand and sugar production. This research uses Backpropagation Neural Network with network architecture of 6 input neurons, 4 hidden neurons and 1 output neuron. The best parameters used are the learning rate of 0.8, the maximum iteration of 200 and the initial weight interval between -1,4 to 1,4. The MSE best predictive value is 0.049064.

Keywords: *cement, forecasting, neural networks, backpropagation, oversupply*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xii
DAFTAR KODE PROGRAM	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Kajian Pustaka	4
2.2 Semen	6
2.2.1 Produksi Semen.....	6
2.2.2 Permintaan Semen.....	7
2.3 Prediksi Data <i>Time Series</i>	7
2.4 Jaringan Syaraf Tiruan.....	9
2.5 <i>Backpropagation</i>	9
2.5.1 Arsitektur <i>Backpropagation</i>	10
2.5.2 Pelatihan <i>Backpropagation</i>	10
2.5.3 <i>Mean Square Error</i>	14
2.5.4 Normalisasi dan Denormalisasi	15
2.5.5 Fungsi Aktivasi.....	15
BAB 3 METODOLOGI	16



3.1 Tipe Penelitian	16
3.2 Strategi Penelitian.....	16
3.3 Partisipan Penelitian	16
3.4 Lokasi Penelitian	16
3.5 Teknik Pengumpulan Data	16
3.6 Data Penelitian.....	17
3.7 Teknik Analisis Data	17
3.8 Implementasi Algoritme	17
3.9 Jadwal Penelitian	18
BAB 4 PERANCANGAN.....	19
4.1 Deskripsi Masalah	19
4.2 Siklus Algoritme <i>Backpropagation</i>	20
4.2.1 Proses <i>Normalisasi</i>	22
4.2.2 Proses <i>Feedforward</i>	23
4.2.3 Proses <i>Backpropagation</i>	26
4.2.4 Proses <i>Update Bobot dan Bias</i>	31
4.2.5 Proses <i>Denormalisasi</i>	34
4.3 Perhitungan Manual Algoritme <i>Backpropagation</i>	35
4.3.1 <i>Normalisasi Data</i>	35
4.3.2 <i>Inisialisasi Bobot</i>	36
4.3.3 Proses <i>Feedforward</i>	37
4.3.4 Proses <i>Backpropagation</i>	39
4.3.5 Proses <i>Update Bobot</i>	41
4.3.6 Proses Perhitungan <i>Mean Square Error (MSE)</i>	42
4.3.7 Hasil Perhitungan Algoritme <i>Backpropagation</i> Iterasi 2	44
4.3.8 Proses Pengujian	46
4.4 Perancangan Antarmuka	51
4.4.1 Perancangan Halaman <i>Load Data</i>	51
4.4.2 Perancangan Halaman <i>Training</i>	52
4.4.3 Perancangan Halaman <i>Prediksi</i>	52
4.5 Perancangan Pengujian Sistem.....	53
4.5.1 Pengujian <i>Konvergensi</i>	53



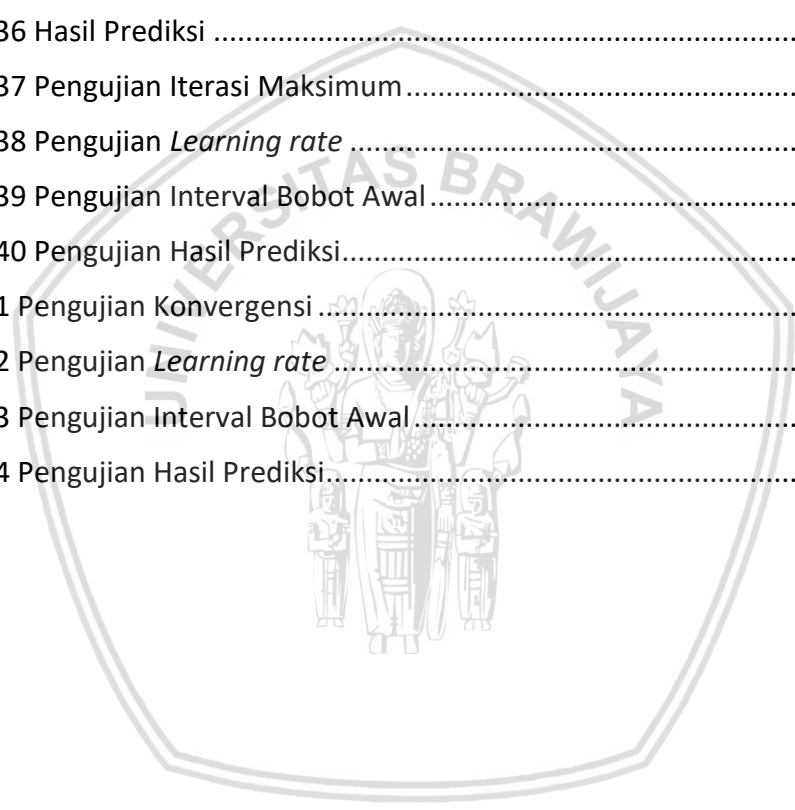
4.5.2 Pengujian <i>Learning rate</i>	54
4.5.3 Pengujian Interval Bobot Awal.....	54
4.5.4 Pengujian Hasil Prediksi	55
4.6 Pengambilan Kesimpulan.....	55
BAB 5 IMPLEMENTASI	56
5.1 Implementasi Program	56
5.1.1 Proses <i>Generate Bobot</i>	56
5.1.2 Proses Pelatihan	57
5.1.3 Proses Pengujian	60
5.2 Implementasi <i>Interface</i>	61
BAB 6 PENGUJIAN DAN ANALISIS.....	64
6.1 Pengujian	64
6.2 Hasil Pengujian.....	64
6.2.1 Pengujian Konvergensi.....	64
6.2.2 Pengujian <i>Learning rate</i>	66
6.2.3 Pengujian Interval Bobot Awal.....	67
6.2.4 Pengujian Hasil Prediksi	68
BAB 7 PENUTUP	70
7.1 Kesimpulan.....	70
7.2 Saran	70
DAFTAR PUSTAKA.....	71
LAMPIRAN	73



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	5
Tabel 2.2 Permintaan Semen Tahun 2007 - 2017	7
Tabel 3.1 Jadwal Penelitian	18
Tabel 4.1 Permintaan Semen	20
Tabel 4.2 Data Latih dan Data Uji	35
Tabel 4.3 Hasil Normalisasi Data Latih dan Data Uji	36
Tabel 4.4 Inisialisasi Bobot dan Bias v	37
Tabel 4.5 Inisialisasi Bobot dan Bias w	37
Tabel 4.6 Nilai z_{net_j}	38
Tabel 4.7 Nilai z_j	38
Tabel 4.8 Nilai y_{net_k}	39
Tabel 4.9 Nilai y_k	39
Tabel 4.10 Nilai δ_k	40
Tabel 4.11 Nilai Δw_{jk}	40
Tabel 4.12 Nilai δ_{net_j}	40
Tabel 4.13 Nilai δ_j	41
Tabel 4.14 Nilai Δv_{ij} dan Δv_{0j}	41
Tabel 4.15 Nilai Bobot dan Bias w (baru)	42
Tabel 4.16 Nilai Bobot dan Bias v (baru)	42
Tabel 4.17 Nilai <i>Error</i>	43
Tabel 4.18 Nilai <i>MSE</i>	43
Tabel 4.19 Nilai z_{net_j}	44
Tabel 4.20 Nilai z_j	44
Tabel 4.21 y_{net_k}	44
Tabel 4.22 Nilai y_k	45
Tabel 4.23 Nilai δ_k	45
Tabel 4.24 Nilai Δw_{jk}	45
Tabel 4.25 Nilai δ_{net_j}	45
Tabel 4.26 Nilai δ_j	45
Tabel 4.27 Nilai Δv_{ij} dan Δv_{0j}	46

Tabel 4.28 Nilai Bobot dan Bias w (baru).....	46
Tabel 4.29 Nilai Bobot dan Bias v (baru).....	46
Tabel 4.30 Bobot v baru	47
Tabel 4.31 Bobot w baru	47
Tabel 4.32 Nilai z_{net_j}	48
Tabel 4.33 Nilai z_j	48
Tabel 4.34 Nilai y_{net_k}	49
Tabel 4.35 Nilai y_k	50
Tabel 4.36 Hasil Prediksi	50
Tabel 4.37 Pengujian Iterasi Maksimum.....	53
Tabel 4.38 Pengujian <i>Learning rate</i>	54
Tabel 4.39 Pengujian Interval Bobot Awal.....	54
Tabel 4.40 Pengujian Hasil Prediksi.....	55
Tabel 6.1 Pengujian Konvergensi	64
Tabel 6.2 Pengujian <i>Learning rate</i>	66
Tabel 6.3 Pengujian Interval Bobot Awal.....	67
Tabel 6.4 Pengujian Hasil Prediksi.....	68



DAFTAR GAMBAR

Gambar 2.1 Grafik Pola Trend.....	8
Gambar 2.2 Grafik Pola Siklis	8
Gambar 2.3 Grafik Pola Musiman	9
Gambar 2.4 Grafik Pola Irregular	9
Gambar 2.5 Arsitektur <i>Backpropagation</i>	10
Gambar 4.1 Arsitektur Jaringan <i>Backpropagation</i>	19
Gambar 4.2 Siklus Algoritme <i>Backpropagation</i>	21
Gambar 4.3 Proses <i>Normalisasi</i> Data.....	22
Gambar 4.4 Proses <i>Feedforward</i>	23
Gambar 4.5 Proses Perhitungan z_j	24
Gambar 4.6 Proses Perhitungan y_k	25
Gambar 4.7 Proses <i>Backpropagation</i>	26
Gambar 4.8 Proses Perhitungan δ_k	27
Gambar 4.9 Proses Perhitungan Δw_{jk} dan Δw_{ok}	28
Gambar 4.10 Proses Perhitungan δ_{net_j} dan δ_j	29
Gambar 4.11 Proses Perhitungan Δv_{ij} dan Δv_{oj}	30
Gambar 4.12 Proses <i>Update</i> Bobot dan Bias	31
Gambar 4.13 Proses <i>Update</i> w_{jk}	32
Gambar 4.14 Proses <i>Update</i> v_{ij}	33
Gambar 4.15 Proses <i>Denormalisasi</i>	34
Gambar 4.16 Perancangan Halaman <i>Load Data</i>	51
Gambar 4.17 Perancangan Halaman <i>Training</i>	52
Gambar 4.18 Perancangan Halaman <i>Prediksi</i>	52
Gambar 5.1 Implementasi Halaman <i>Load Data</i>	62
Gambar 5.2 Implementasi Halaman <i>Pelatihan</i>	63
Gambar 5.3 Implementasi Halaman <i>Pengujian</i>	63
Gambar 6.1 Grafik <i>Pengujian Konvergensi</i>	65
Gambar 6.2 Grafik <i>Pengujian Learning rate</i>	66
Gambar 6.3 Grafik <i>Pengujian Interval Bobot Awal</i>	68
Gambar 6.4 Grafik <i>Pengujian Hasil Prediksi</i>	69



DAFTAR KODE PROGRAM

Kode Program 5.1 Proses <i>Generate</i> Bobot	57
Kode Program 5.2 Proses Pelatihan	59
Kode Program 5.3 Proses Pengujian	61



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Semen merupakan salah satu bahan yang penting pada proses pembangunan. Semen berfungsi untuk memadatkan material – material lainnya sehingga menjadi satu kesatuan dan dapat tetap mempertahankan kekuatannya meskipun terkena air. Semen digunakan untuk pembuatan hampir di keseluruhan bangunan seperti pondasi, tembok dan lantai (Mahfud & Zakir, 2018). Produksi semen di Indonesia tercatat tinggi mengingat banyaknya perusahaan semen dan munculnya pemain baru di industri ini. Pada tahun 2017 jumlah produksi semen tanah air mencapai 107 juta ton (Asosiasi Semen Indonesia, 2017). Banyaknya produksi semen ini lantas tidak diimbangi oleh permintaan semen sehingga terjadi ketimpangan antara kapasitas produksi dan banyaknya permintaan. Asosiasi Semen Indonesia (ASI) mencatatkan bahwa kenaikan permintaan semen pada tahun 2011 sebesar 20% dari 40 juta menjadi 48 juta ton. Selanjutnya pertumbuhan permintaan semen turun menjadi 14,6% saja pada tahun 2012 dan terus turun hingga tahun 2015 yang menyentuh angka pertumbuhan permintaan sebesar 2,5%. Pada tahun 2017 terjadi kenaikan permintaan semen sebesar 7% dari 6,2 juta ton menjadi 6,8 ton, sehingga total permintaan semen pada tahun 2017 hanya mencapai 66 juta ton.

Perbedaan kapasitas produksi dan permintaan semen ini mengakibatkan terjadinya *oversupply*. Keadaan *oversupply* adalah kondisi yang mana angka produksi lebih besar daripada angka konsumsi sehingga kelebihan produksi akan berpengaruh pada kelangsungan hidup pabrik semen. Pada tahun 2017, Indonesia mengalami kelebihan pasokan semen kurang lebih sebesar 40 juta ton. Hal ini menyebabkan penurunan harga semen hingga 12% sehingga mengakibatkan kerugian produksi. Beberapa pabrik semen juga terpaksa menghentikan sementara produksi semen karena stok di gudang penyimpanan sudah penuh. Bahkan beberapa pabrik diantaranya telah memiliki klinker (produk setengah jadi) di luar gedung karena tidak ada tempat lagi di ruang penyimpanan.

Informasi terkait pertumbuhan permintaan semen di Indonesia dari waktu ke waktu sangat diperlukan bagi produsen semen di Indonesia agar dapat memenuhi permintaan semen di Indonesia namun juga tidak memproduksi semen dengan berlebihan. Selain itu, informasi ini juga diperlukan oleh Asosiasi Semen Indonesia (ASI) untuk mengeluarkan regulasinya mengenai industri semen di Indonesia. ASI sempat mengajukan permohonan untuk tidak membangun pabrik semen baru sampai tahun 2019 atau sampai ketimpangan antara produksi dan permintaan semen setara. Hal ini dilakukan karena produksi semen yang terlalu banyak di tahun 2017 mengakibatkan kinerja industri semen menurun. ASI juga berharap pemerintah dapat melindungi industri semen di Indonesia dengan kondisi *oversupply* tersebut.

Berdasarkan uraian yang telah dijelaskan maka dibutuhkan sebuah sistem yang mampu melakukan prediksi permintaan semen di Indonesia sehingga dapat

dijadikan acuan untuk dilakukannya produksi dan menganalisis pertumbuhan industri semen di Indonesia. Metode yang dapat digunakan untuk melakukan prediksi sangat banyak, salah diantaranya adalah Jaringan Syaraf Tiruan (JST). Jaringan Syaraf tiruan merupakan sebuah cara memproses sebuah data yang mengambil konsep seperti yang dilakukan oleh otak manusia. Model jaringan ini mampu mengenali pola dari data dengan melakukan pelatihan terlebih dahulu (Bustomi, et al., 2014). Salah satu model Jaringan Syaraf Tiruan yang sering dipakai untuk melakukan prediksi adalah metode *Backpropagation*. *Backpropagation* merupakan sebuah metode yang dapat memperkecil tingkat *error* dengan mengubah bobot berdasarkan *input* dan *outputnya*. Metode ini juga baik digunakan untuk *dataset time-series* atau data yang diterbitkan secara konsisten dalam suatu waktu (Haviluddin & Alfred, 2015).

Metode *Backpropagation* telah banyak digunakan dalam penelitian sebelumnya seperti memprediksi jumlah permintaan koran yang dilakukan oleh Sakinah (2018) yang mendapatkan tingkat *error* sebesar 0,0162. Selanjutnya adalah prediksi ketersediaan komoditi pangan di Provinsi Riau oleh Cynthia (2017) dengan memperoleh tingkat *error* sebesar 0,00334 dengan persentase akurasi sebesar 99%. Prediksi jumlah pengangguran terbuka di Indonesia oleh Adwandha (2017) dengan memperoleh nilai AFER sebesar 4,715198444%. Dari latar belakang yang telah dijabarkan dan beberapa contoh kasus prediksi yang memiliki nilai *error* yang kecil tersebut maka penulis akan menggunakan metode *Backpropagation* untuk melakukan prediksi permintaan semen di Indonesia.

1.2 Rumusan Masalah

Dari latar belakang yang telah dijabarkan sebelumnya maka rumusan masalah yang akan dibahas penelitian ini adalah:

1. Bagaimana menerapkan metode *Backpropagation* untuk melakukan prediksi jumlah permintaan semen di Indonesia?
2. Bagaimana nilai MSE dari prediksi jumlah permintaan semen di Indonesia menggunakan metode *Backpropagation*?

1.3 Tujuan

Tujuan dilakukannya penelitian ini adalah:

1. Menerapkan metode *Backpropagation* untuk melakukan prediksi jumlah permintaan semen di Indonesia.
2. Menguji nilai MSE untuk melakukan prediksi jumlah permintaan semen di Indonesia menggunakan metode *Backpropagation*.

1.4 Manfaat

Manfaat yang dapat diperoleh dari penelitian ini ialah membantu instansi terkait untuk meramalkan banyaknya permintaan semen sehingga dapat menjadi acuan untuk mengeluarkan kebijakan mengenai perkembangan industri semen.

1.5 Batasan Masalah

Penelitian ini memiliki batasan-batasan, antara lain:

1. Data yang digunakan adalah laporan permintaan dan penjualan PT. Semen Indonesia yang diterbitkan mulai dari tahun 2007 sampai 2017.
2. Prediksi yang dilakukan hanya untuk permintaan semen di Indonesia.

1.6 Sistematika Pembahasan

Gambaran umum serta pembahasan laporan skripsi ini akan diulas pada sistematika pembahasan sistematis sebagai berikut :

BAB I Pendahuluan

Membahas tentang penentuan latar belakang masalah, tujuan dari penelitian itu sendiri, manfaat yang didapatkan berdasarkan penelitian yang dilakukan, serta sistematika penulisan.

BAB II Landasan Kepustakaan

Landasan kepastakaan akan membahas terkait literatur penelitian dan teori terkait metodologi penelitian yang telah dipublikasikan sebelumnya. Hal ini dimaksudkan untuk memperkuat hepotesis dan dapat digunakan sebagai acuan dalam penulisan dan penelitian ini.

BAB III Metodologi

Bab metodologi akan membahas terkait ekosistem penitian ini sendiri, yang terdiri dari struktur penelitian, mulai dari pengumpulan data sampai bentuk stuktur kesimpulan laporan.

BAB IV Perancangan

Sedangkan pada bab pembahasan terkait spesifikasi kebutuhan dan perancangan sistem akan dibahas dalam bab ini untuk dapat melakukan prediksi.

BAB V Implementasi

Pada bab ini difokuskan mengimplementasikan apa yang telah dirancang pada bab sebelumnya.

BAB VI Pengujian dan Analisis

Bab pengujian dan analisis bertujuan untuk membuktikan tingkat keberhasilan sistem berdasarkan rancangan pengujian seperti yang telah dijelaskan pada bab sebelumnya.

BAB VII Penutup

Bab ini berisikan kesimpulan dari penelitian yang mengenai prediksi jumlah permintaan semen menggunakan metode *Backpropagation* beserta saran pengembangan untuk dilakukannya penelitian lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini akan menjelaskan mengenai landasan kepastakaan seperti metode Jaringan Syaraf Tiruan *Backpropagation*, peramalan, semen dan permintaan semen. Selanjutnya pada kajian pustaka akan dibahas mengenai penelitian sebelumnya yang pernah dilakukan terkait penggunaan metode Jaringan Syaraf Tiruan *Backpropagation*.

2.1 Kajian Pustaka

Penelitian ini akan membahas kajian pustaka berdasarkan penelitian yang pernah dilakukan sebelumnya. Penelitian pertama berjudul “Prediksi Jumlah Pengangguran Terbuka di Indonesia menggunakan Metode Genetic-Based *Backpropagation*” (Adwandha, et al., 2017). Dalam penelitian tersebut membahas tentang prediksi banyaknya pengangguran terbuka berdasarkan data pengangguran pada tahun – tahun sebelumnya. Penelitian ini bertujuan untuk memberikan gambaran kepada pemerintah tentang semakin banyaknya pertumbuhan pengangguran namun tidak dibarengi dengan pertumbuhan lahan pekerjaan. Penelitian ini menghasilkan nilai *Average Forecast Error Rate* (AFER) yang cukup kecil yaitu sebesar 4.715198444% sehingga dapat dikatakan dalam penelitian tersebut metode *Backpropagation* berhasil melakukan prediksi.

Penelitian selanjutnya yang berjudul “Peramalan Dosis Pupuk Berdasarkan Karakteristik dan Lingkungan Tanaman Jeruk Siam Menggunakan Metode *Backpropagation*” (Ridhani, et al., 2017). Pada penelitian tersebut dilakukan prediksi dosis pupuk yang sesuai terhadap beberapa faktor seperti iklim, lingkungan dan penyakit. Hal ini diperlukan agar produksi jeruk siam terus meningkat karena jeruk siam adalah salah satu buah yang populer di Indonesia. Dalam penelitian ini menggunakan 3 neuron *input*, 5 neuron *hidden layer* dan 3 neuron *output*. Parameter yang digunakan yaitu *learning rate* sebesar 0,3, maksimum iterasi sebesar 500 dan data latih 56 serta data uji 8 menghasilkan nilai evaluasi *Mean Absolute Percentage Error* (MAPE) sebesar 9,178%.

Kemudian penelitian yang berjudul “Prediksi Tingkat Keuntungan Usaha Peternakan Itik Alabio Petelur menggunakan Jaringan Syaraf Tiruan *Backpropagation* (Kasus di Kabupaten Hulu Sungai Utara Kalimantan Selatan)” (Diputra, et al., 2018) membahas tentang prediksi keuntungan usaha peternakan itik alabio petelur berdasarkan 7 parameter yang dipilih yaitu banyaknya induk, harga pakan, penyusutan itik, penyusutan kandang, biaya obat dan vaksin dan biaya tenaga kerja. Dari proses peramalan digunakan data latih sebesar 90% dari total data latih dengan *learning rate* sebesar 0,8 dan *hidden layer* sebanyak 17 menghasilkan nilai *Mean Absolute Percentage Error* (MAPE) peramalan terbaik sebesar 25,7852%.

Kemudian penelitian yang berjudul “Peramalan Produksi Gula Menggunakan Metode Jaringan Syaraf Tiruan *Backpropagation* Pada PG Candi Baru Sidoarjo” (Rachman, et al., 2018). Pada penelitian ini menggunakan arsitektur

jaringan berupa 4 neuron *input*, 3 neuron *hidden* dan 1 neuron target. Nilai *Mean Absolute Percentage Error* (MAPE) terbaik didapatkan dengan menggunakan *learning rate* sebesar 0,4 dengan nilai iterasi maksimum sebesar 800 menghasilkan nilai MAPE sebesar 16,98%.

Penelitian lain yang berjudul “Prediksi Jumlah Permintaan Koran Menggunakan Metode Jaringan Syaraf Tiruan *Backpropagation*” (Sakinah, et al., 2018). Pada penelitian ini membahas tentang peramalan koran per harinya menggunakan data penjualan satu tahun sebelumnya. Penelitian ini berhasil mendapatkan peramalan yang baik dengan *error rate* sebesar 0,0162 menggunakan *learning rate* sebesar 0,6, jumlah iterasi sebesar 200, 100 data latih dan 10 data uji.

Tabel 2.1 Kajian Pustaka

No	Judul	Objek	Metode	Hasil
1	Prediksi Jumlah Pengangguran Terbuka di Indonesia menggunakan Metode Genetic-Based <i>Backpropagation</i> (Adwandha, et al., 2017)	Pengangguran terbuka di Indonesia	<i>Genetic-Based Backpropagation</i>	Memberikan prediksi pengangguran terbuka di Indonesia untuk pemerataan lapangan pekerjaan.
2	Peramalan Dosis Pupuk Berdasarkan Karakteristik dan Lingkungan Tanaman Jeruk Siam Menggunakan Metode <i>Backpropagation</i> (Ridhani, et al., 2017)	Jeruk siam	<i>Backpropagation</i>	Memberikan prediksi pemberian pupuk yang optimal berdasarkan keadaan lingkungan.
3	Prediksi Tingkat Keuntungan Usaha Peternakan Itik Alabio Petelur menggunakan Jaringan Syaraf Tiruan <i>Backpropagation</i> (Kasus di Kabupaten Hulu Sungai Utara Kalimantan Selatan) (Diputra, et al., 2018)	Itik alabio	<i>Backpropagation</i>	Memberikan prediksi keuntungan penjualan telur itik alabio berdasarkan banyaknya induk, harga pakan, penyusutan itik, penyusutan kandang, biaya obat dan vaksin

				dan biaya tenaga kerja.
4	Peramalan Produksi Gula Menggunakan Metode Jaringan Syaraf Tiruan <i>Backpropagation</i> Pada PG Candi Baru Sidoarjo (Rachman, et al., 2018)	Produksi gula	<i>Backpropagation</i>	Memberikan informasi prediksi produksi gula dengan nilai <i>error</i> sebesar 16,98%.
5	Prediksi Jumlah Permintaan Koran Menggunakan Metode Jaringan Syaraf Tiruan <i>Backpropagation</i> (Sakinah, et al., 2018)	Permintaan koran	<i>Backpropagation</i>	Memberikan prediksi jumlah permintaan koran pada waktu selanjutnya dengan nilai <i>error</i> sebesar 0,0162.

2.2 Semen

Semen adalah material yang dapat mengeras jika dicampur dengan air melalui proses kimia (hidrasi), dan jika telah mengeras akan tetap mempertahankan kekuatan dan stabilitasnya meskipun terkena air. Sehingga semen digunakan sebagai pengikat bahan bangunan seperti batu, pasir, batu bata dan keramik untuk pembuatan beton atau konstruksi bangunan seperti lantai, dinding, dan lain – lain (Mahfud & Zakir, 2018).

Kata semen diadopsi dari bahasa inggris yaitu *cement* dan kata "*cement*" berasal dari masyarakat romawi yang menggunakan kata "*opus caementitium*" untuk menggambarkan bangunan batu yang menyerupai beton dan dibuat dari batu yang dihancurkan dengan batu kapur yang dibakar sebagai pengikat. Debu vulkanis dan bubuk batu bata yang ditambahkan pada batu kapur yang dibakar untuk membentuk pengikat hidrolik kemudian disebut sebagai *cementium*, *cimentium*, *caement* dan *cement*.

2.2.1 Produksi Semen

Kementrian Perindustrian (Kemenperin, 2017) mengatakan bahwa total kapasitas produksi semen di tanah air mencapai 107 juta ton. Hal ini terjadi karena besarnya realisasi investasi pada industri semen di Indonesia. Direktur Jenderal Industri Kimia mengatakan bahwa akan terjadi ledakan pasokan semen pada tahun 2017 sehingga kelebihan produksi diharapkan mampu memenuhi permintaan pasar ekspor. Di sisi lain kelebihan produksi semen dapat menurunkan harga semen hingga 12%.

2.2.2 Permintaan Semen

Menurut Asosiasi Semen Indonesia (ASI), permintaan semen di Indonesia cukup besar yaitu sebesar 66 juta ton semen pada tahun 2017. Namun jumlah permintaan tersebut masihlah lebih rendah daripada kapasitas produksi. Dibawah ini ditunjukkan permintaan semen pada tahun 2007 sampai tahun 2017 Tabel 2.2.

Tabel 2.2 Permintaan Semen Tahun 2007 - 2017

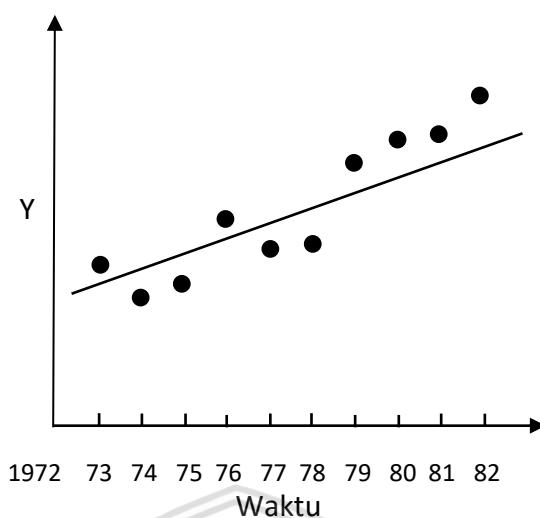
No	Tahun	Permintaan semen (ton)
1	2007	34177727
2	2008	38122610
3	2009	38517205
4	2010	40821460
5	2011	48000347
6	2012	54964407
7	2013	58005037
8	2014	59909501
9	2015	62251690
10	2016	62007529
11	2017	66349945

2.3 Prediksi Data *Time Series*

Prediksi adalah sebuah proses perhitungan selisih antara kejadian yang belum terjadi sebelumnya dengan analisis data yang sedang dilakukan (Watimena, 2008). Data *time series* adalah sebuah data historis yang dikumpulkan pada satuan waktu tertentu (Zulfikar & Budiantara, 2014). Berdasarkan kedua definisi tersebut maka dapat disimpulkan, Prediksi Data *Time Series* merupakan proses mendapatkan solusi berdasarkan data historis dalam kurun waktu tertentu.

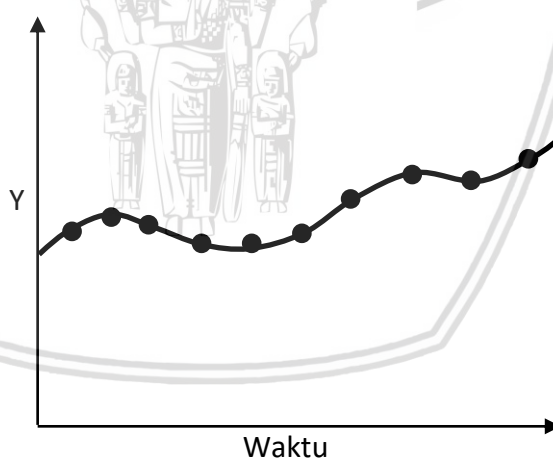
Data *time series* mempunyai beberapa pola, seperti trend, siklis, musiman, dan irregular (Ukhra, 2014).

1. Pola Trend : Yaitu pola yang mempunyai jangka yang panjang. Pola ini biasanya memerlukan waktu yang lama untuk mendapatkan polanya. Contoh pola trend yaitu data penjualan dari suatu perusahaan. Grafik pola trend dapat dilihat pada Gambar 2.1.



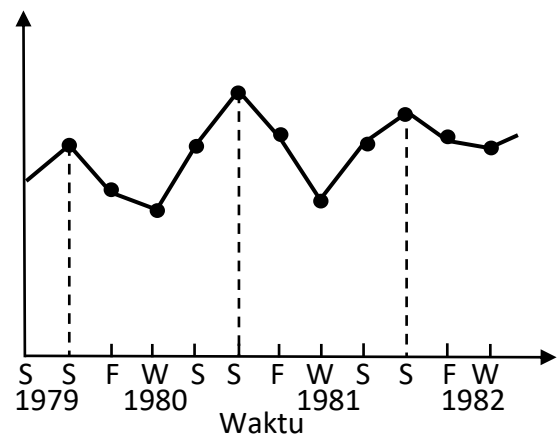
Gambar 2.1 Grafik Pola Trend

2. Pola Siklis : Yaitu data yang dapat berubah berdasarkan perubahan fluktuasi politik dan ekonomi. Data ini kebanyakan berhubungan dengan dunia bisnis. Contoh pola siklis ialah data penjualan barang seperti mobil, baja dan lain – lain. Grafik pola siklis dapat dilihat pada Gambar 2.2.



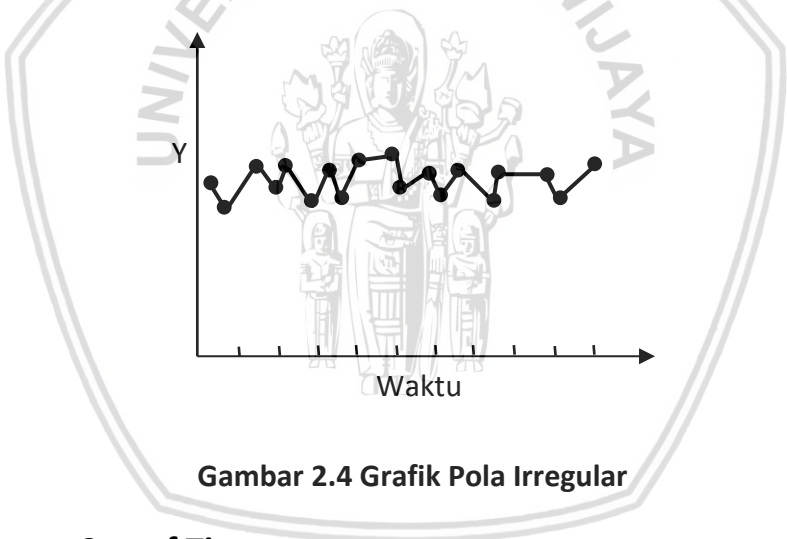
Gambar 2.2 Grafik Pola Siklis

3. Pola Musiman : Yaitu pola data yang didapatkan berdasarkan periode tertentu dalam satuan waktu, misalnya harian sampai tahunan. Contoh pola musiman adalah data curah hujan. Grafik pola musiman dapat dilihat pada Gambar 2.3.



Gambar 2.3 Grafik Pola Musiman

- 4. Pola Irregular : Yaitu sebuah pola data yang sangat acak sehingga tidak mudah ditebak. Contoh dari pola irregular adalah data penjualan yang sifatnya tidak teratur. Grafik pola irregular dapat dilihat pada Gambar 2.4.



Gambar 2.4 Grafik Pola Irregular

2.4 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) adalah contoh metode perhitungan matematis yang meniru seperti struktur jaringan syaraf otak. JST memiliki kelebihan dalam hal mempelajari pola melalui proses pelatihan sebelumnya (Bustomi, et al., 2014).

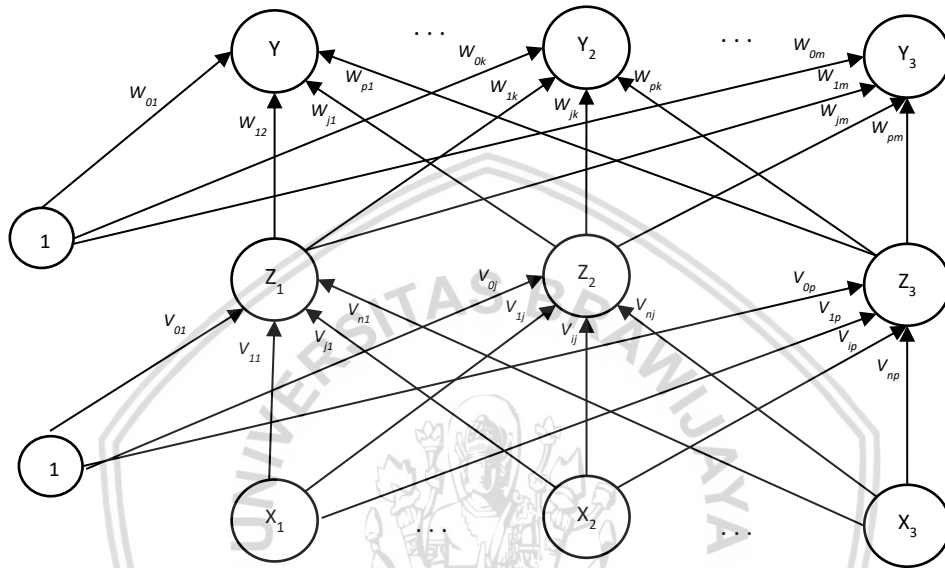
2.5 Backpropagation

Salah satu metode yang populer dalam JST untuk memprediksi data adalah metode *Backpropagation*. Metode *backpropagation* memanfaatkan data sample yang diolah dalam iterasi tertentu untuk menghasilkan solusi dengan cara membandingkan nilai data latih dan data asli. Nilai *error* pada data akan terus dikurangi dengan melakukan modifikasi bobot. Proses utama pada metode

backpropagation adalah menggunakan sistem perambatan mundur untuk mendapatkan solusi terbaik (Kusrini & Luthfi, 2009).

2.5.1 Arsitektur *Backpropagation*

Arsitektur dalam metode *backpropagation* diawali perambatan data dari *input layer* yang disimbolkan dengan variabel x , sedangkan pada *hidden layer* disimbolkan dengan variabel z dan variabel y pada *output layer*nya. Arsitektur *backpropagation* ditunjukkan pada Gambar 2.1.



Gambar 2.5 Arsitektur *Backpropagation*

- x : layer input
- z : layer hidden
- y : layer output
- v_{ij} : bobot dari layer input x_i ke layer hidden z_j
- v_{0j} : bobot dari bias di layer input ke layer hidden z_j
- w_{jk} : bobot dari layer hidden z_j ke layer output y_k
- w_{0k} : bobot dari bias di layer hidden ke layer output y_k

2.5.2 Pelatihan *Backpropagation*

Tingkat keberhasilan prediksi dari metode *backpropagation* dipengaruhi oleh hal – hal berikut ini:

1. Inisialisasi bobot

Rentang nilai acak antara -1 sampai dengan 1 digunakan sebagai bobot awal sehingga proses perhitungan aritmatika dapat dilakukan dengan lebih cepat.

2. *Learning rate*



Pada proses *training* data diperlukan *learning rate* untuk meningkatkan nilai akurasi dan efektifitas pemrosesan data.

3. Penentuan jumlah *hidden layer*

Penentuan *hidden layer* juga dapat mempengaruhi tingkat akurasi prediksi yang dihasilkan sehingga perlu dilakukan uji banyaknya *hidden layer* untuk mendapatkan nilai *error* yang kecil.

Pelatihan dengan metode *backpropagation* terdiri atas tiga fase menurut Siang, dalam Hudyawan (2015):

a. Fase I : *Feedforward*

Fase I dimulai dengan melakukan *feedforward* unit *input layer* (x_i) menuju ke unit *hidden layer* (z_j), selanjutnya dilakukan aktivasi sesuai dengan fungsi aktivasi yang telah ditentukan. Hasil dari unit *hidden* (z_j) *difeedforward* kembali menuju ke unit *output layer* sehingga menghasilkan nilai *output* (y_k). selanjutnya nilai y_k dan target (t_k) akan dibandingkan, sehingga dapat dilakukan perhitungan *Mean Square Error* (MSE).

b. Fase II : *Backpropagation*

Fase II yaitu melakukan propagasi mundur berdasarkan perbandingan antara unit target dan unit *output*. Dengan menggunakan data tersebut dapat dilakukan perhitungan δ_k ($k = 1, 2, \dots, n$). Selanjutnya data tersebut digunakan untuk melakukan koreksi pada *hidden layer*. Selain digunakan untuk mengkoreksi nilai pada *hidden layer*, data tersebut juga digunakan untuk mencari nilai δ_{net} sehingga nilai δ_{net} dapat digunakan untuk mengkoreksi nilai *input layer*.

c. Fase III : *Weight Update*

Fase III yaitu proses melakukan koreksi nilai bobot untuk menghasilkan nilai bobot yang baru dan menghasilkan *error* yang kecil.

Fase I sampai III akan dijalankan berulang - ulang sampai menemui kondisi berhenti. Pelatihan akan selesai jika telah mencapai iterasi yang telah ditentukan sebelumnya. Dibawah ini contoh penjelasan algoritme pembelajaran *backpropagation*:

1. Langkah 0 : Inisialisasi bobot dan bias secara acak (bobot bernilai kecil antara -1 sampai 1). Menentukan banyak iterasi dan *learning rate*.
2. Langkah 1 : Jika kondisi berhenti belum terpenuhi, ulangi kembali langkah 2 sampai dengan 8.
3. Langkah 2 : Proses berjalannya *training* data. Lakukan langkah 3 sampai dengan 8.

Fase I : *Feedforward*

4. Langkah 3 : *layer input* menerima sinyal *input* x_i dari data, kemudian dilanjutkan menuju *layer hidden*.

5. Langkah 4 : Hasil *output* pada *layer hidden* dihitung dengan menggunakan rumus 2.1 sehingga menghasilkan z_net_j ($j=1,2,\dots,p$).

$$z_net_j = v_{0j} + \sum_{i=0}^n x_i \cdot v_{ij} \quad (2.1)$$

Berdasarkan persamaan sebelumnya, kemudian dilakukan aktivasi sigmoid untuk mendapatkan nilai z_j :

$$z_j = f(z_net_j) = \frac{1}{1 + \exp(-z_net_j)} \quad (2.2)$$

Keterangan:

z_net_j : *Input* pada jaringan z_j

v_{0j} : Nilai bobot pada bias dari *layer input* menuju unit z_j

x_i : Unit ke- i pada *layer input*

v_{ij} : Bobot dari *layer input* x_i ke *layer hidden* z_j

z_j : Hasil aktivasi unit ke- j pada *layer hidden*

i : Indeks pada unit *layer input*

j : Indeks pada unit *layer hidden*

p : Jumlah unit *hidden*

Selanjutnya adalah proses pendistribusian sinyal dari *layer hidden* ke semua bagian unit *layer output*.

6. Langkah 5 : Menjumlahkan sinyal dari *layer hidden* dengan bobot w yang menuju *layer output* hingga menghasilkan keluaran y_k ($k=1,2,\dots,m$)

$$y_net_k = w_{0k} + \sum_{j=1}^p z_j \cdot w_{jk} \quad (2.3)$$

Selanjutnya gunakan persamaan fungsi aktivasi sigmoid untuk memproses sinyal *output*:

$$y_k = f(y_net_k) = \frac{1}{1 + \exp(-y_net_k)} \quad (2.4)$$

Keterangan:

y_net_k : *Input* pada jaringan y_k

w_{0k} : Bobot pada bias dari *hidden layer* menuju unit y_k

w_{jk} : Bobot dari *layer hidden* z_j ke *layer output* y_k

z_j : Nilai aktivasi dari z_net_j

y_k : Hasil aktivasi dari unit y_net_k

k : Indeks pada unit *output layer*

m : Jumlah unit *output*

Fase II : Backpropagation

7. Langkah 6

Mencari nilai δ pada *layer output* berdasarkan *error* pada semua *output* y_k ($k=1,2,\dots,m$)

$$\delta_k = (t_k - y_k)f'(y_{net_k}) = (t_k - y_k)y_k(1 - y_k) \quad (2.5)$$

Keterangan:

δ_k : Nilai kesalahan yang terjadi pada *layer output*

y_{net_k} : *Input* pada jaringan y_k

t_k : Nilai target pada *input*

y_k : Nilai pada unit *output*

k : Indeks pada unit *output layer*

δ_k adalah nilai kesalahan yang digunakan untuk mengubah bobot pada *layer output* (langkah 7)

Menghitung perubahan bobot w_{jk} dengan menggunakan *learning rate* yang ditentukan:

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2.6)$$

Keterangan:

Δw_{jk} : Perbedaan nilai $w_{jk}(t)$ dengan $w_{jk}(t+1)$

α : Konstanta laju pelatihan (*learning rate*). Nilai *alpha* diantara 0 - 1

δ_k : Faktor pengendali bobot pada *output* ke unit *hidden*

z_j : Nilai pada unit *hidden*

j : Indeks pada unit *hidden*

8. Langkah 7

Mencari nilai δ pada *layer hidden* berdasarkan *error* pada *layer hidden* z_j ($j=1,2,\dots,p$)

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (2.7)$$

Faktor dari nilai δ pada *layer hidden*:

$$\delta_j = \delta_{net_j} \cdot f'(z_{net_j}) = \delta_{net_j} \cdot z_j(1 - z_j) \quad (2.8)$$

Menghitung perubahan bobot v_{ij} yang nantinya digunakan pada saat melakukan *update* pada bobot v_{ij}

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.9)$$

Keterangan:

δ_{net_j} : Faktor pengendali nilai bobot pada z_j



- δ_k : Faktor pengendali bobot *layer output* menuju *layer hidden*
- w_{jk} : Bobot *layer hidden* menuju *layer output*
- z_{net_j} : *Input* pada jaringan z_j
- z_j : Nilai pada unit *hidden*
- Δv_{ij} : Perbedaan nilai $v_{ij}(t)$ dengan $v_{ij}(t+1)$
- α : Konstanta laju pelatihan (*learning rate*). Nilai *alpha* diantara 0 - 1
- δ_j : Faktor pengendali bobot pada *layer hidden*
- x_i : Unit ke- i pada *layer input*
- p : Jumlah unit *hidden*

Fase III : Weight Update

9. Langkah 8

Menghitung nilai bobot baru w .

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \tag{2.10}$$

Keterangan:

- $w_{jk}(\text{baru})$: bobot dari *layer hidden* ke *layer output* baru
- $w_{jk}(\text{lama})$: bobot dari *layer hidden* ke *layer output* lama
- Δw_{jk} : Perbedaan nilai $w_{jk}(t)$ dengan $w_{jk}(t+1)$

Menghitung nilai bobot baru v .

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \tag{2.11}$$

Keterangan:

- $v_{ij}(\text{baru})$: bobot dari *layer input* ke *layer hidden* baru
- $v_{ij}(\text{lama})$: bobot dari *layer input* ke *layer hidden* lama
- Δv_{ij} : Selisih antara $v_{ij}(t)$ dengan $v_{ij}(t+1)$

2.5.3 Mean Square Error

Untuk mengetahui keberhasilan metode *backpropagation* dalam melakukan pelatihan data, maka dapat dilihat pada *error* yang dihasilkan pada tiap – tiap data. Setelah itu data dijumlahkan dan dihitung nilai rata – rata *error*nya atau biasa disebut dengan *Mean Square Error* (MSE). Dalam prosesnya akan terjadi konvergensi MSE dimana nilai *error* akan mencapai titik terkecil atau tidak berubah menjadi yang lebih baik lagi. Perhitungan MSE dapat dilihat pada Persamaan 2.12.

$$MSE = \frac{\sum(t-y_k)^2}{n} \tag{2.12}$$



Keterangan:

t : Nilai sebenarnya

y_k : Nilai prediksi

k : Indeks untuk unit *output*

n : Jumlah data latih

2.5.4 Normalisasi dan Denormalisasi

Normalisasi merupakan proses penyetaraan data untuk mendapatkan nilai data yang seragam. Selain itu proses normalisasi juga dapat menambah efektifitas proses pelatihan. Kebalikan dari proses normalisasi adalah proses denormalisasi, yaitu proses mengembalikan data hasil normalisasi menjadi nilai yang sebenarnya. Proses normalisasi dijabarkan pada Persamaan 2.13.

$$x' = \frac{0,8(x - \text{min value})}{(\text{max value} - \text{min value})} + 0,1 \quad (2.13)$$

Proses denormalisasi atau pegembalian nilai menuju data sebenarnya dijabarkan pada Persamaan 2.15.

$$x = \frac{(\text{max value} - \text{min value})(x' - 0,1)}{0,8} + \text{min value} \quad (2.14)$$

Keterangan:

x : Data asli

x' : Data normalisasi

2.5.5 Fungsi Aktivasi

Fungsi aktivasi merupakan sebuah fungsi yang diperlukan pada Jaringan Syaraf Tiruan untuk mengaktifkan neuron. Beberapa karakteristik dalam fungsi aktivasi pada jaringan merambat mundur adalah kontinuitas dan tidak menurun secara monotonis (*monotonically non-decreasing*) (Fausett, 1994). Beberapa contoh fungsi aktivasi dalam Jaringan Syaraf Tiruan adalah fungsi aktivasi sigmoid biner dan sigmoid bipolar.

Fungsi aktivasi sigmoid biner mempunyai hasil aktivasi dengan nilai antara 0 sampai dengan 1, didefinisikan dengan rumus:

$$y = f(x) = \frac{1}{1 + \exp(-x)} \quad (2.15)$$

Fungsi aktivasi lainnya adalah sigmoid bipolar. Fungsi aktivasi ini mirip dengan sigmoid biner hanya saja memiliki perbedaan pada hasil aktivasinya dengan rentang -1 sampai dengan 1. Fungsi aktivasi sigmoid bipolar didefinisikan dengan rumus:

$$y = f(x) = \frac{1 - \exp(-x)}{1 + \exp(-x)} \quad (2.16)$$

BAB 3 METODOLOGI

Metodologi penelitian digunakan untuk mengetahui langkah – langkah yang dijalankan pada proses penelitian. Beberapa proses yang dilakukan dibagi menjadi beberapa bab yaitu tipe penelitian, strategi penelitian, partisipan penelitian, lokasi penelitian, teknik pengumpulan data, data penelitian, teknik analisis data, implementasi algoritme dan jadwal penelitian.

3.1 Tipe Penelitian

Pada penelitian ini menggunakan tipe penelitian non-implementatif yang bersifat analitik. Penelitian ini fokus kepada hubungan antara masalah yang terjadi pada objek penelitian dengan usulan solusi yang dapat digunakan sehingga dapat menghasilkan produk berupa hasil analisis. Jenis penelitian yang digunakan dalam tulisan ini adalah kuantitatif dimana menggunakan data bulanan sebelumnya kemudian digunakan untuk memprediksi data pada bulan berikutnya.

3.2 Strategi Penelitian

Strategi penelitian diperlukan untuk menyelesaikan sebuah studi kasus yang bersifat eksperimental. Metode eksperimental dilakukan untuk menguji variabel – variabel yang saling berhubungan pada proses pengujian. Disamping itu, strategi penelitian diperlukan untuk mengetahui studi kasus pada objek tertentu untuk dilakukan penggalan permasalahan sehingga permasalahan tersebut dapat dipecahkan oleh metode pembelajaran.

3.3 Partisipan Penelitian

Partisipan penelitian yang ikut serta dalam penelitian ini adalah pegawai yang bekerja pada perusahaan semen baik pusat maupun pabrik. Pemilihan partisipan penelitian tersebut dilakukan untuk mengetahui bagaimana cara perusahaan mengetahui jumlah permintaan semen di waktu yang akan datang yang sifatnya tidak menentu.

3.4 Lokasi Penelitian

Penelitian ini dilakukan di Laboratorium Komputasi Cerdas Fakultas Ilmu Komputer Universitas Brawijaya untuk mengimplementasikan algoritme Jaringan Syaraf Tiruan *Backpropagation* pada kasus prediksi permintaan semen di Indonesia.

3.5 Teknik Pengumpulan Data

Pada Teknik Pengumpulan data, peneliti menggunakan data laporan penjualan PT. Semen Indonesia yang diterbitkan tiap bulan mulai tahun 2007 sampai dengan tahun 2017. Data tersebut berisi beberapa parameter seperti data penjualan dan data permintaan semen yang terjadi di tiap pulau besar di

Indonesia. Data permintaan tersebut kemudian diurutkan sehingga menjadi data mentah penelitian yang memiliki pola tertentu.

3.6 Data Penelitian

Penelitian ini menggunakan data yang telah diproses pada teknik pengumpulan data. Data yang digunakan adalah data permintaan semen yang didapatkan dari laporan penjualan PT. Semen Indonesia yang diterbitkan tiap bulan mulai tahun 2007 sampai dengan tahun 2017. Data tersebut memiliki pola data *time-series* dengan panjang data sebanyak 132 data. Dari 132 data yang ada kemudian dibuat pola datanya sehingga menghasilkan 126 data dengan 6 parameter *input* dan 1 target.

3.7 Teknik Analisis Data

Teknik analisis data dilakukan dengan melakukan proses pengujian dimana data akan dilatih menggunakan metode *backpropagation*. Proses perhitungan akan dilakukan dengan parameter – parameter yang juga digunakan sebagai skenario pengujian. Skenario yang digunakan dalam penelitian ini antara lain pengujian konvergensi, pengujian *learning rate* dan pengujian interval bobot awal sehingga dapat menghasilkan hasil prediksi yang baik.

3.8 Implementasi Algoritme

Langkah pertama pada proses implementasi algoritme adalah melakukan perhitungan manual melalui excel, hal ini dilakukan supaya peneliti lebih mengetahui alur dan cara kerja algoritme yang digunakan. Selain itu hasil perhitungan manual digunakan sebagai acuan sistem, apakah keluaran sistem sudah sesuai dengan perhitungan manual yang telah dilakukan. Langkah selanjutnya adalah membuat diagram alir untuk mengetahui alur perhitungan *backpropagation*. Proses implementasi dimulai dari normalisasi data, penentuan bobot awal, proses *feedforward*, proses *backpropagation*, update bobot dan denormalisasi data.

3.9 Jadwal Penelitian

Tabel 3.1 Jadwal Penelitian

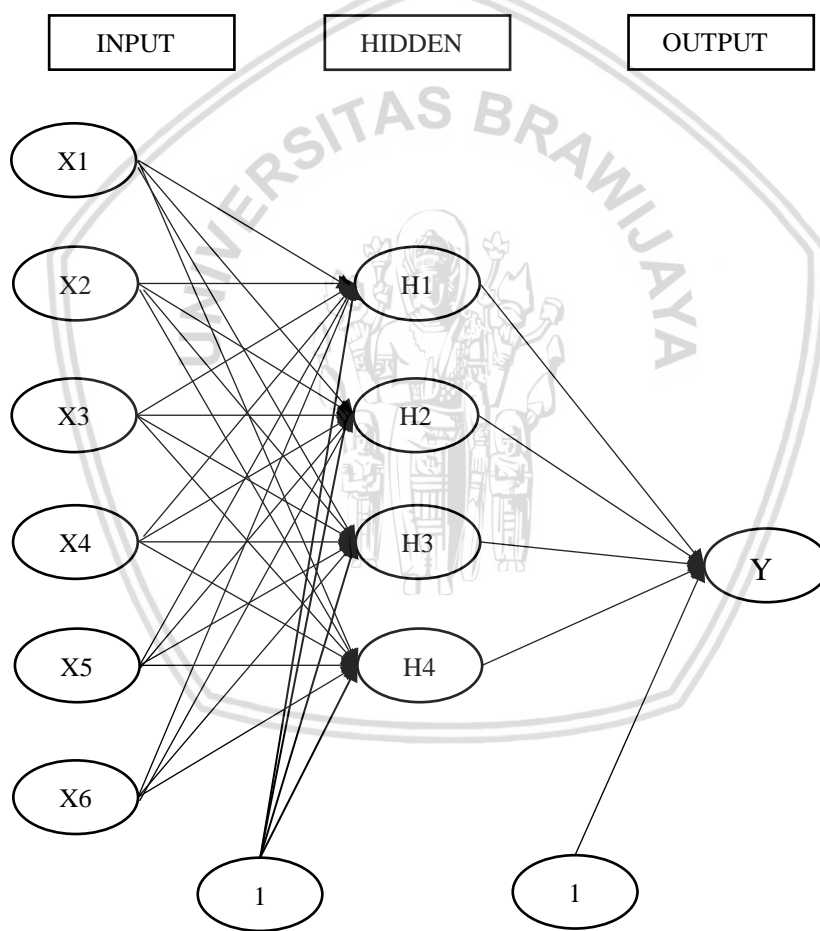
No	Uraian	Februari				Maret				April				Mei				Juni				Juli							
		Minggu ke-																											
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
1	Pengumpulan Data	■	■																										
2	Studi Kepustakaan			■	■																								
3	Perancangan					■	■	■	■																				
4	Implementasi									■	■	■	■	■	■	■													
5	Pengujian dan Analisis																	■	■	■	■	■	■						
6	Kesimpulan dan Saran																									■	■		

BAB 4 PERANCANGAN

Bab ini mengulas formulasi permasalahan, cara penyelesaian prediksi permintaan semen menggunakan Jaringan Syaraf Tiruan *Backpropagation*, contoh perhitungan manual dan perancangan antarmuka.

4.1 Deskripsi Masalah

Permasalahan yang terdapat pada penelitian ini adalah prediksi permintaan semen menggunakan Jaringan Syaraf Tiruan *Backpropagation* dengan harapan untuk mengetahui jumlah permintaan semen pada bulan berikutnya. Arsitektur jaringan yang digunakan dalam penelitian ini disajikan pada Gambar 4.1.



Gambar 4.1 Arsitektur Jaringan *Backpropagation*

Pada Gambar 4.1 menjelaskan arsitektur jaringan berupa 6 *neuron input*, 4 *neuron hidden* dan 1 *neuron target*. Rancangan arsitektur ini mengacu pada penelitian sebelumnya oleh (Sakinah, et al., 2018) yang memprediksi jumlah permintaan koran pada PT. Jawa Pos dengan mendapatkan *error rate* yang baik yaitu sebesar 0,0162. Selain itu, berdasarkan penelitian yang telah dilakukan,

pehitungan *error* yang digunakan adalah menggunakan MSE (*Mean Square Error*). *Input* sistem adalah data permintaan semen yang direkam tiap bulan. Data tersebut kemudian dibuat pola dengan model pola bulan 1 sampai bulan 6 sebagai *input* dan bulan 7 sebagai *output*. Bulan 2 sampai bulan 7 sebagai *input* dan bulan 8 sebagai *output*, begitu seterusnya. Data tersebut kemudian akan melalui proses pelatihan menggunakan metode *Backpropagation* untuk mendapatkan bobot tiap *input* sehingga dapat dilakukan prediksi.

Dalam penelitian ini menggunakan 10 pola terakhir dari semua pola yang terbentuk untuk digunakan pada perhitungan manual. Data yang digunakan disajikan pada Tabel 4.1.

Tabel 4.1 Permintaan Semen

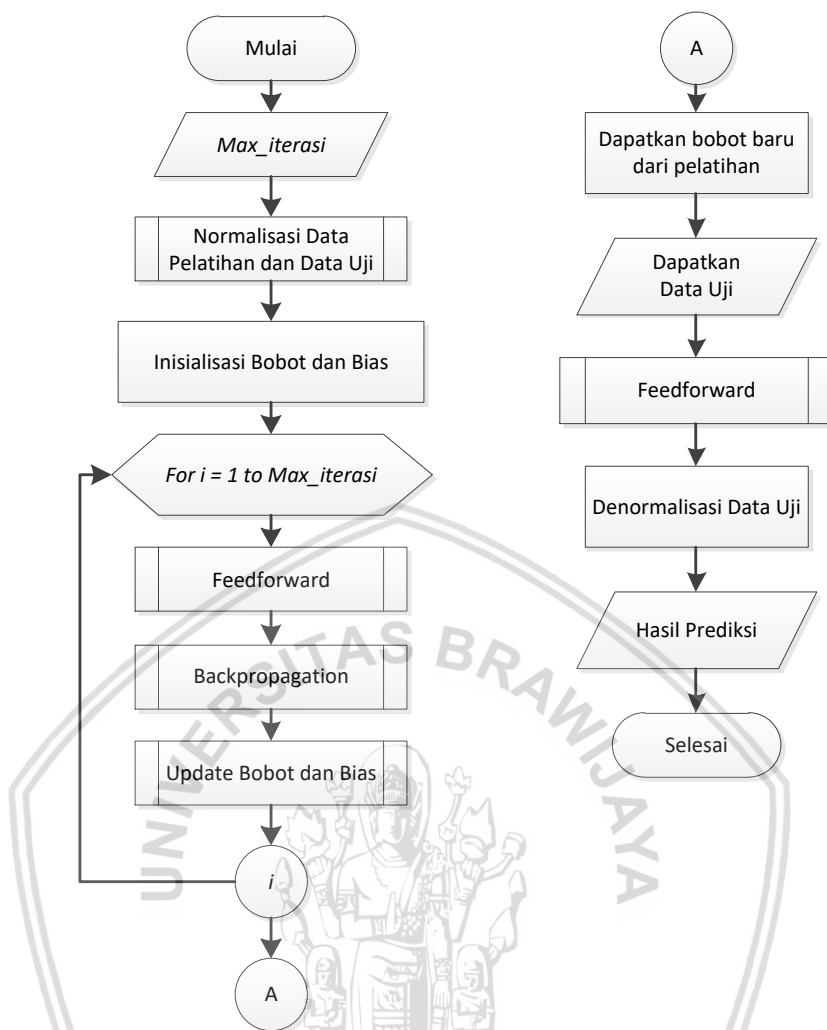
<i>Data</i>	<i>x1</i>	<i>x2</i>	<i>x3</i>	<i>x4</i>	<i>x5</i>	<i>x6</i>	<i>target</i>
1	5633478	6062829	5737245	5503160	5172287	4554653	5027788
2	6062829	5737245	5503160	5172287	4554653	5027788	5033066
3	5737245	5503160	5172287	4554653	5027788	5033066	5475102
4	5503160	5172287	4554653	5027788	5033066	5475102	3731358
5	5172287	4554653	5027788	5033066	5475102	3731358	5638695
6	4554653	5027788	5033066	5475102	3731358	5638695	6495831
7	5027788	5033066	5475102	3731358	5638695	6495831	6303276
8	5033066	5475102	3731358	5638695	6495831	6303276	6754075
9	5475102	3731358	5638695	6495831	6303276	6754075	6365491
10	3731358	5638695	6495831	6303276	6754075	6365491	5798323

Sumber : Laporan bulanan PT. Semen Indonesia

x1, *x2*, *x3*, *x4*, *x5*, *x6* dan *target* merupakan *input* yang didapat dari data mentah yang kemudian dibuat pola. Data didapat dari laporan penjualan bulanan PT. Semen Indonesia yang kemudian diambil hanya pada data permintaan semen di Indonesia. Data tersebut akan dijadikan sebagai data latih dan data uji pada proses perhitungan manual algoritme *Backpropagation*.

4.2 Siklus Algoritme *Backpropagation*

Dalam sub bab ini akan dijabarkan mengenai siklus atau proses Jaringan Syaraf Tiruan *Backpropagation* untuk melakukan prediksi jumlah permintaan semen di Indonesia. Proses umum algoritme *backpropagation* digambarkan pada diagram alir pada Gambar 4.1.



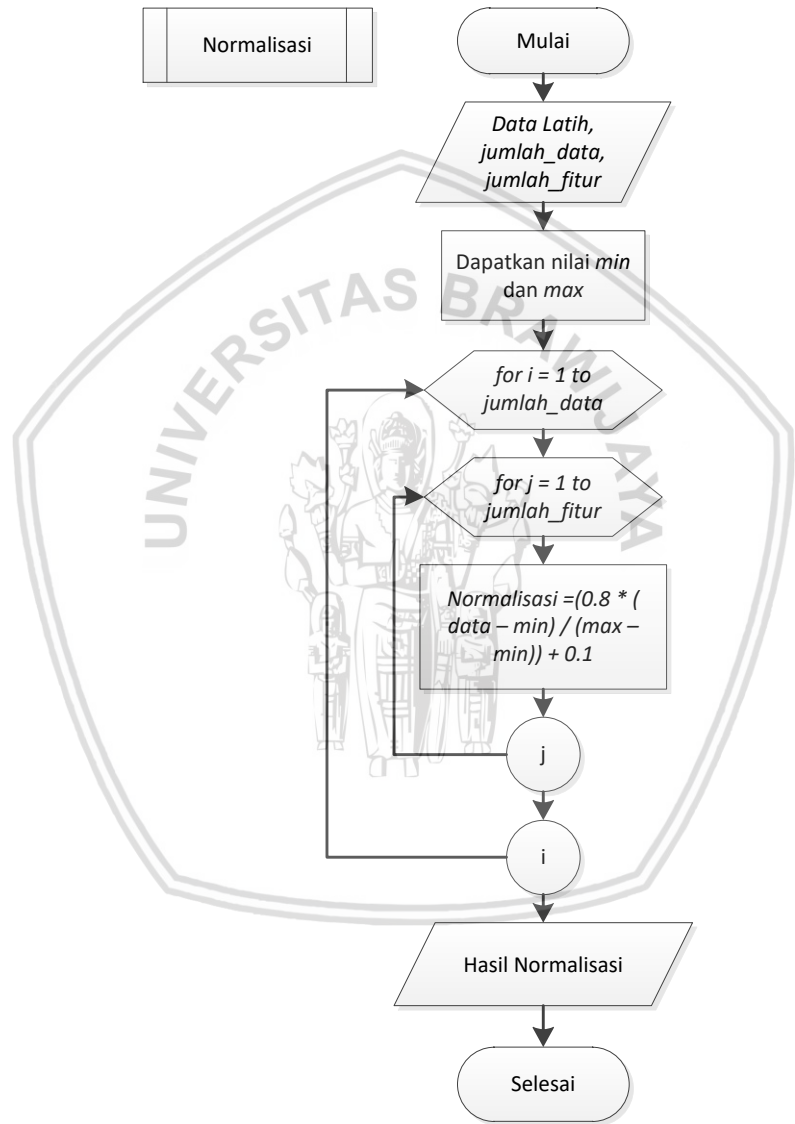
Gambar 4.2 Siklus Algoritme *Backpropagation*

Gambar 4.1 menjelaskan mengenai proses dilakukannya prediksi menggunakan algoritme *backpropagation* untuk memprediksi jumlah permintaan semen di Indonesia. Tahapan – tahapan algoritme *backpropagation* adalah sebagai berikut:

1. Tentukan jumlah iterasi yang diinginkan sebagai kondisi berhenti.
2. Masukkan data latih dan data uji selanjutnya dilakukan normalisasi data.
3. Inisialisasi bobot dan bias secara random dengan nilai antara -1 dan 1.
4. Setelah bobot dan bias didapatkan, langkah selanjutnya adalah dilakukan *feedforward*.
5. Setelah dilakukan proses *feedforward* maka akan didapatkan selisih *error* yang kemudian digunakan untuk melakukan proses *backpropagation*.
6. Tahap selanjutnya adalah mengupdate bobot dan bias.

7. Tahapan tersebut dilakukan berulang – ulang sampai kondisi berhenti tercapai.
8. Setelah ditemukan bobot terakhir, prediksi dapat dilakukan dengan melakukan proses *feedforward* saja pada data uji.
9. Setelah didapatkan hasil prediksi, data tersebut akan dikembalikan nilai aslinya dengan melakukan denormalisasi data.

4.2.1 Proses Normalisasi



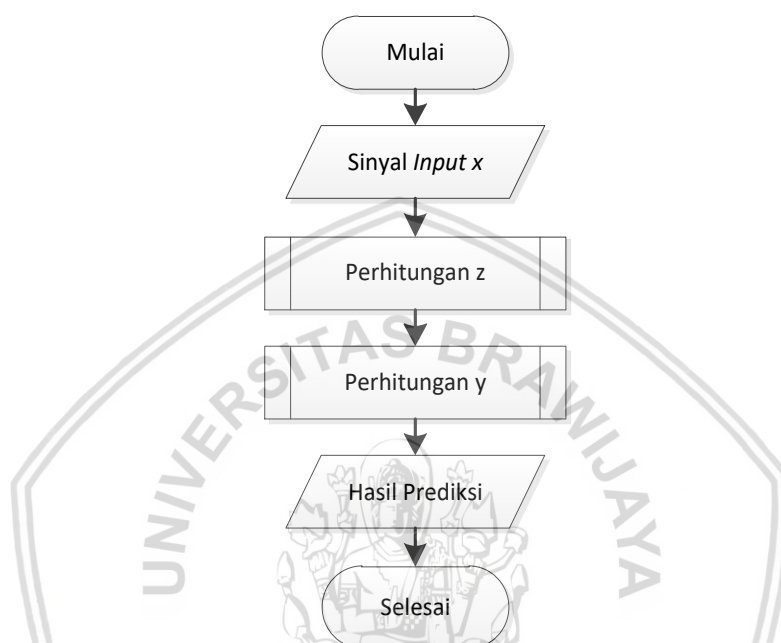
Gambar 4.3 Proses Normalisasi Data

Gambar 4.2 menjelaskan mengenai proses normalisasi data. Tahapan proses normalisasi data adalah sebagai berikut:

1. Sistem mendapatkan data latih dan data uji.

2. Sistem menghitung nilai min dan max dari tiap data dalam data latih dan data uji.
3. Sistem menghitung nilai normalisasi pada tiap data.
4. Hasil dari proses ini adalah menghasilkan data dengan rentang antara 0.1 sampai dengan 0.9.

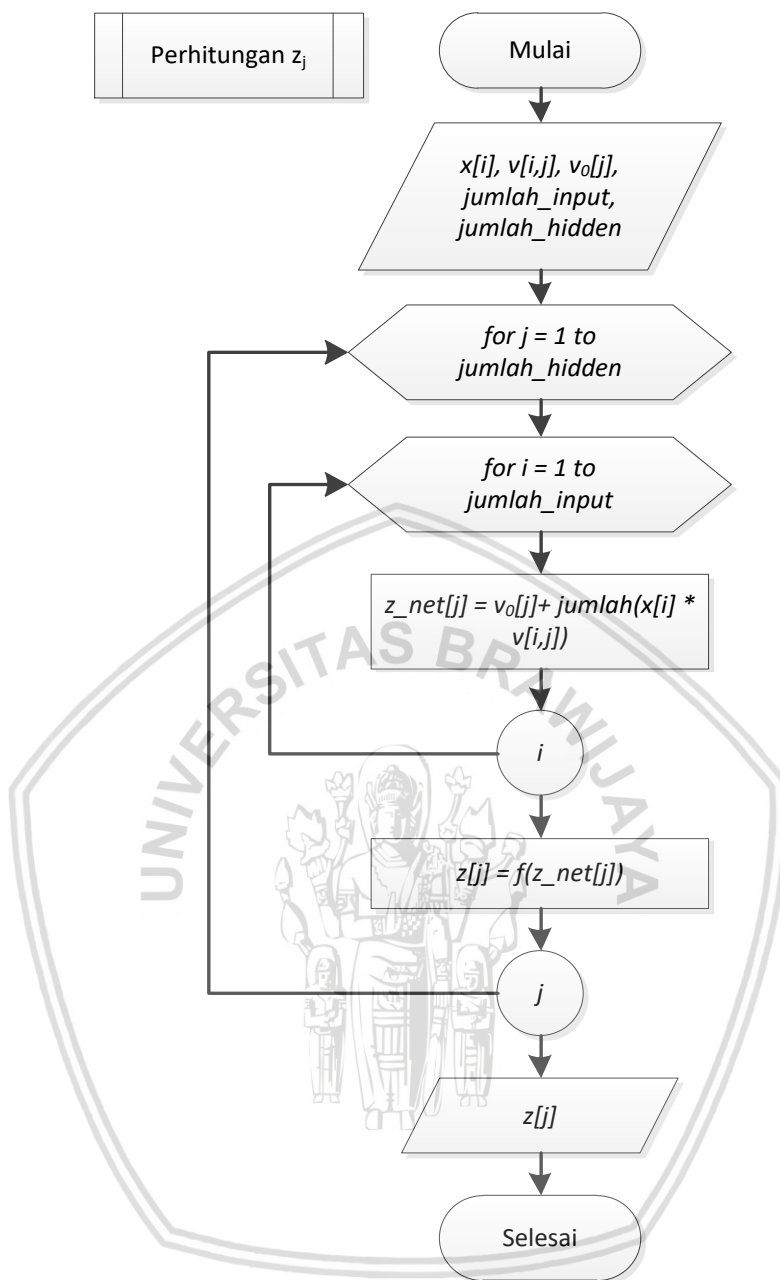
4.2.2 Proses *Feedforward*



Gambar 4.4 Proses *Feedforward*

Gambar 4.3 menjelaskan mengenai gambaran umum proses *feedforward* pada metode *backpropagation*. Tahapan proses *feedforward* adalah sebagai berikut:

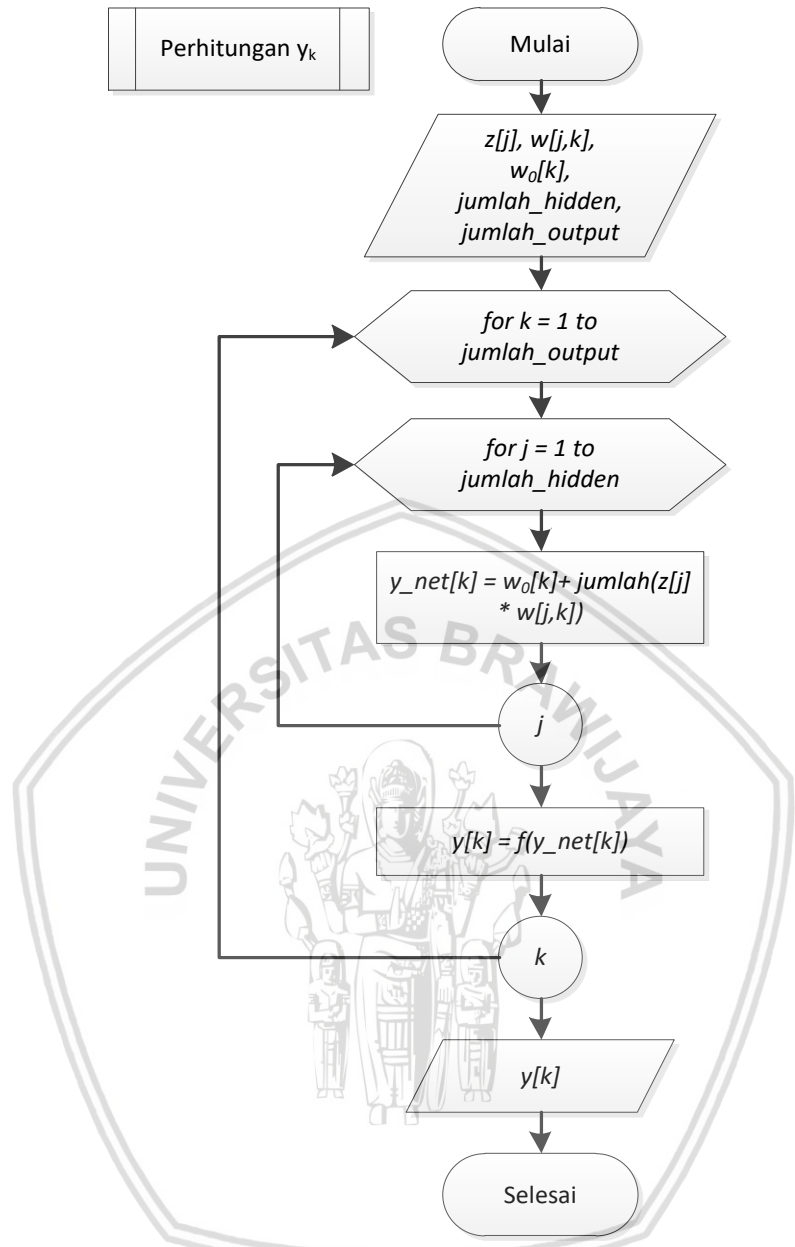
1. Sistem menerima *input x* yang berasal dari jumlah permintaan semen yang telah dibuatkan polanya.
2. Sistem menghitung nilai *z* pada *hidden layer* yang didapatkan dari perkalian *input* dengan bobot *v*.
3. Sistem menghitung nilai *y* pada *output layer* yang didapatkan dari perkalian *hidden layer* dengan bobot *w*.
4. Proses perhitungan akan menghasilkan *output* yaitu nilai *z* dan *y*.



Gambar 4.5 Proses Perhitungan z_j

Gambar 4.4 menjelaskan mengenai perhitungan nilai *input* x dan bobot v hingga menghasilkan *output* z_j pada *hidden layer*. Proses perhitungannya sebagai berikut:

1. Sistem mendapatkan *input* x dari data uji dan bobot v yang didapatkan secara random.
2. Sistem akan mencari nilai z_{netj} dengan mengali dan menjumlahkan nilai x dan bobot w , kemudian dilakukan fungsi aktivasi untuk menghasilkan nilai z_j .
3. Sistem akan menghasilkan nilai z_j pada *hidden layer*.



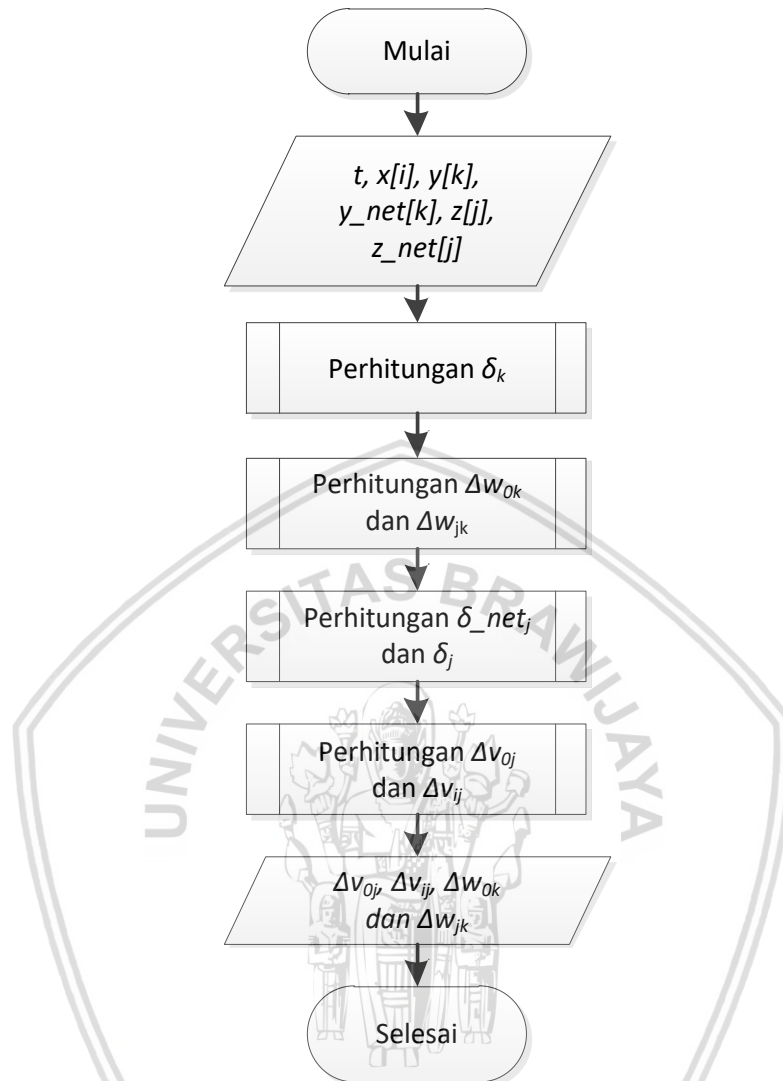
Gambar 4.6 Proses Perhitungan y_k

Gambar 4.5 menjelaskan mengenai perhitungan nilai *hidden* z dan bobot w hingga menghasilkan *output* y_k pada *output layer*. Proses perhitungannya sebagai berikut:

1. Sistem telah mendapatkan nilai z_j pada *hidden layer* yang didapatkan pada proses perhitungan z_j . Selain itu sistem telah mendapatkan bobot w yang dibangkitkan secara random.
2. Sistem akan melakukan perulangan pada tiap *input* pada *hidden layer* dengan bobot w untuk mendapatkan nilai y_net_k . Selanjutnya sistem akan melakukan fungsi aktivasi untuk mendapatkan nilai y_k .
3. Sistem mendapatkan nilai y_k pada *layer output*.



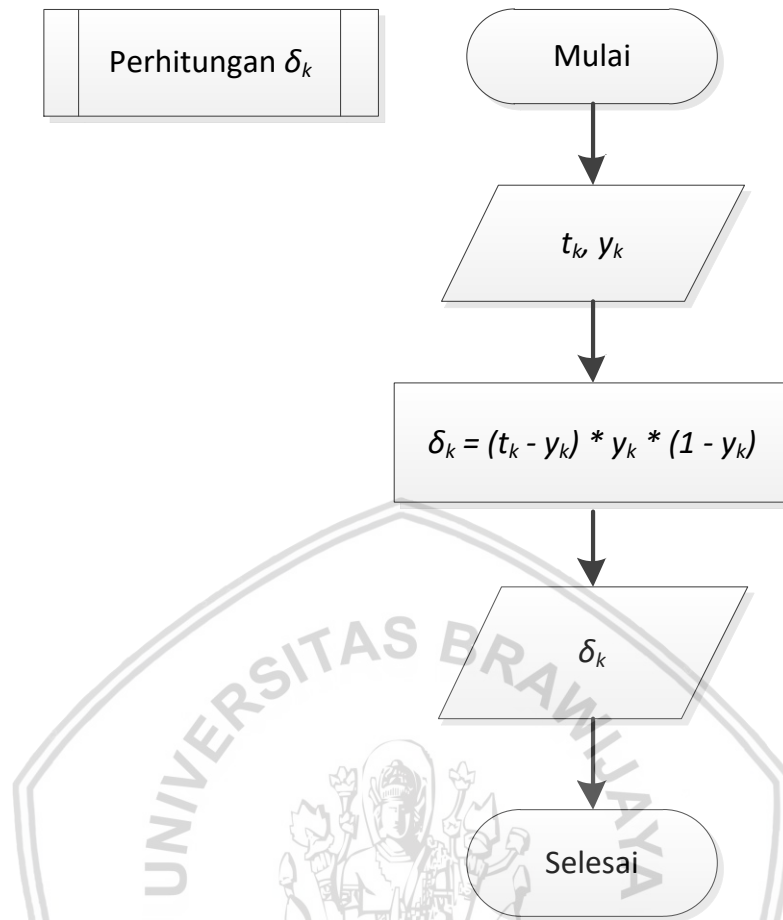
4.2.3 Proses *Backpropagation*



Gambar 4.7 Proses *Backpropagation*

Gambar 4.6 menjelaskan mengenai gambaran umum proses *backpropagation*. Tahapan proses *backpropagation* adalah sebagai berikut:

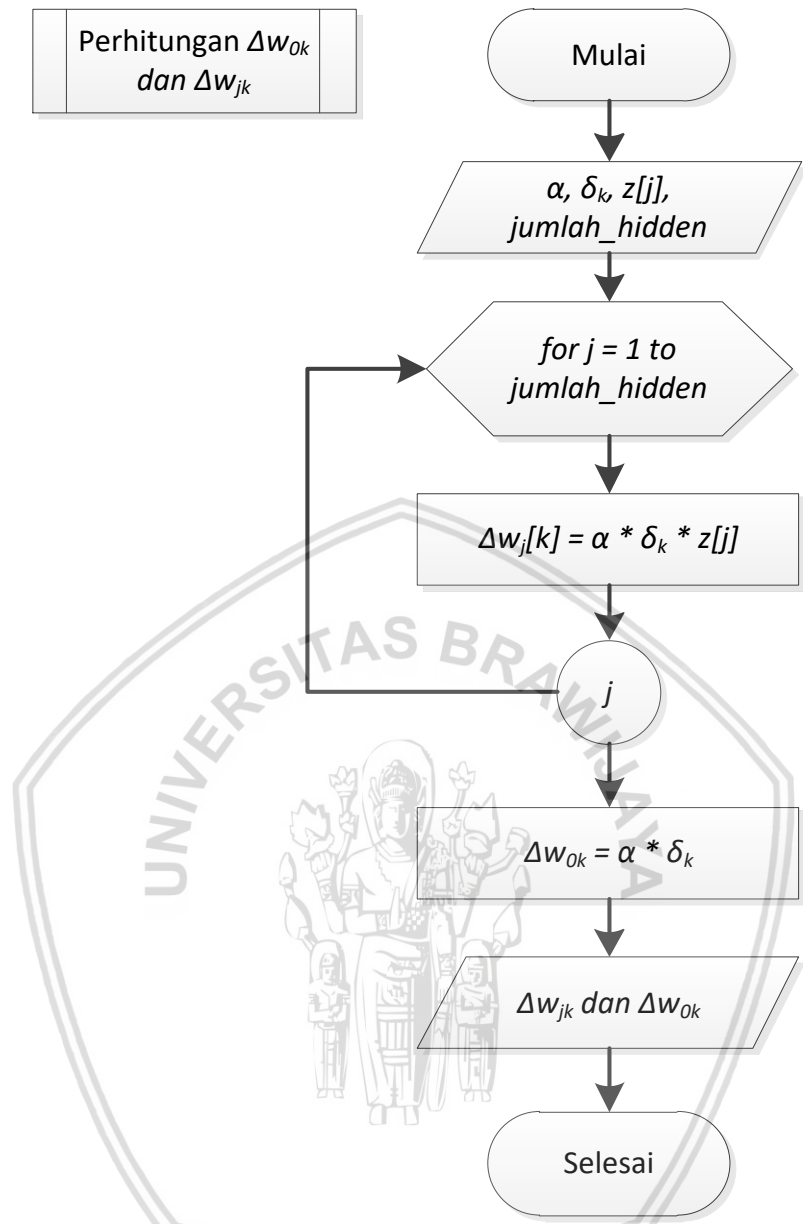
1. Sistem telah mempunyai nilai – nilai yang diantaranya diperoleh pada saat proses *feedforward*, diantaranya adalah $t, x, z_net_j, z_j, y_net_k$ dan y_k .
2. Perhitungan nilai δ_k digunakan untuk menghitung nilai *error* antara target dan *output*.
3. Sistem menghitung nilai Δw_{0k} dan Δw_{jk} yang nantinya digunakan untuk melakukan proses *update* bobot dan bias pada *hidden layer*.
4. Sistem menghitung nilai δ_net_j dan δ_j .
5. Sistem menghitung Δv_{0j} dan Δv_{ij} yang nantinya digunakan untuk melakukan proses *update* bobot dan bias pada *input layer*.
6. Proses ini menghasilkan nilai $\Delta v_{0j}, \Delta v_{ij}, \Delta w_{0k}$, dan Δw_{jk} .



Gambar 4.8 Proses Perhitungan δ_k

Gambar 4.7 menjelaskan mengenai proses perhitungan δ_k . Tahapan proses perhitungan δ_k adalah sebagai berikut:

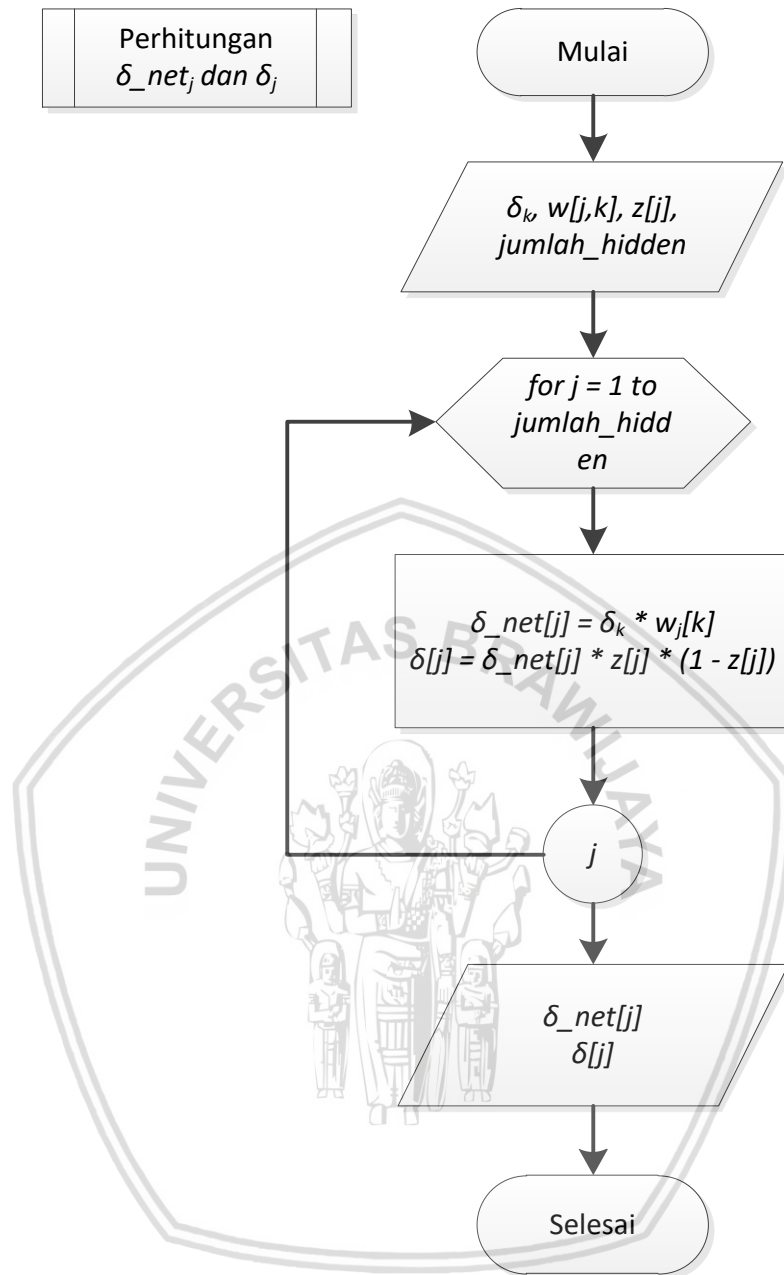
1. Sistem telah memiliki nilai t_k dari data pelatihan.
2. Sistem telah memiliki nilai y_k yang didapatkan dari proses perhitungan *feedforward*.
3. Sistem melakukan perhitungan δ_k .
4. Hasil akhir dari poses ini adalah menghasilkan *output* yatu nilai δ_k .



Gambar 4.9 Proses Perhitungan Δw_{jk} dan Δw_{0k}

Gambar 4.8 menggambarkan proses perhitungan Δw_{jk} dan Δw_{0k} . Tahapannya adalah sebagai berikut:

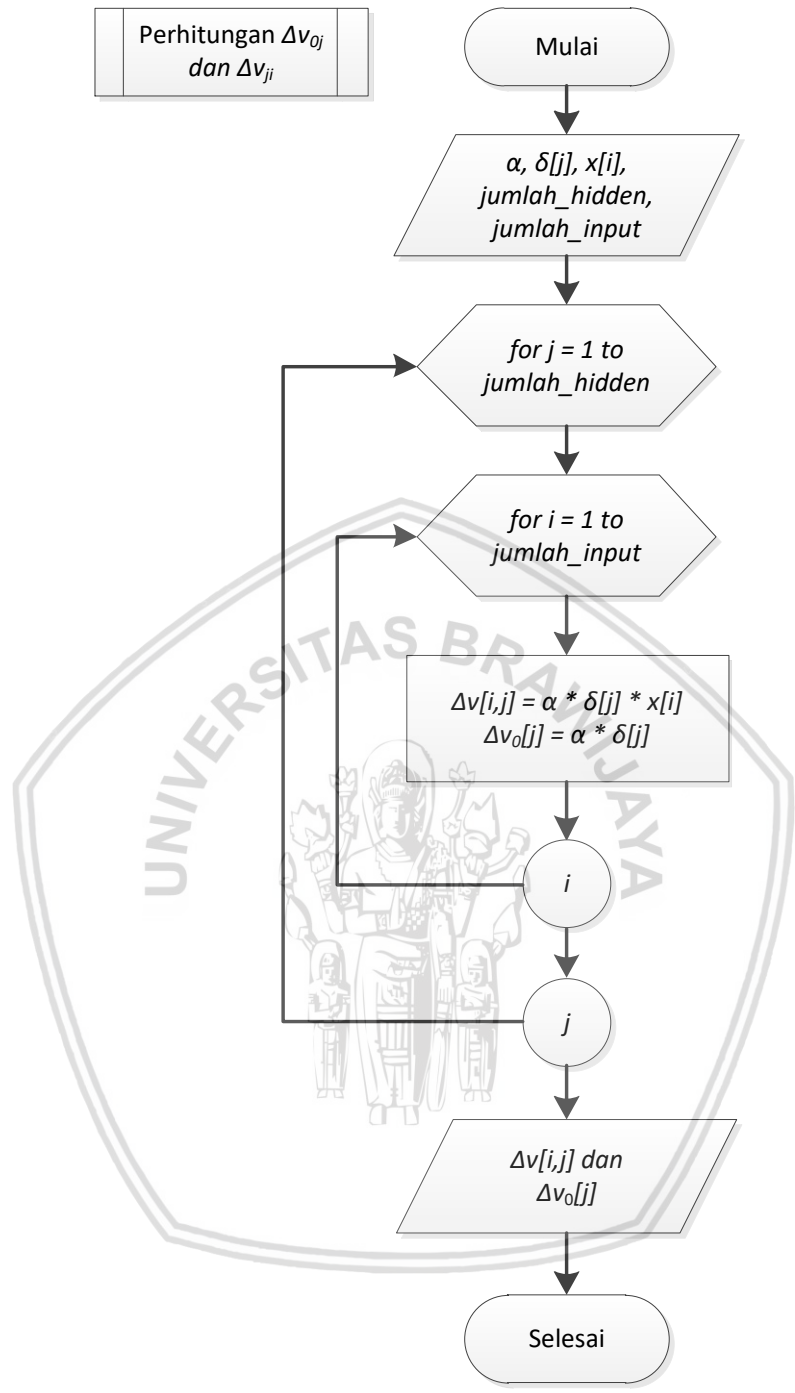
1. Sistem telah memiliki nilai $\delta_k, z[j]$ dari perhitungan *feedforward* yang telah dilakukan sebelumnya dan nilai α yang telah ditentukan.
2. Sistem melakukan perulangan untuk menghitung nilai Δw_{jk} yang nantinya digunakan untuk *update* bobot w .
3. Sistem melakukan perhitungan untuk mencari nilai Δw_{0k} yang nantinya digunakan untuk *update* bias pada w .
4. Hasil akhir dari proses ini adalah menghasilkan nilai Δw_{jk} dan Δw_{0k} .



Gambar 4.10 Proses Perhitungan δ_{net_j} dan δ_j

Gambar 4.9 menjelaskan mengenai proses perhitungan δ_{net_j} dan δ_j . Tahapannya adalah sebagai berikut:

1. Sistem telah memiliki nilai $\delta_k, w_j[k], z[j]$.
2. Sistem melakukan perhitungan nilai δ_{net_j} .
3. Selanjutnya melakukan perhitungan δ_j .
4. Hasil akhir dari proses ini adalah menghasilkan nilai δ_{net_j} dan δ_j .



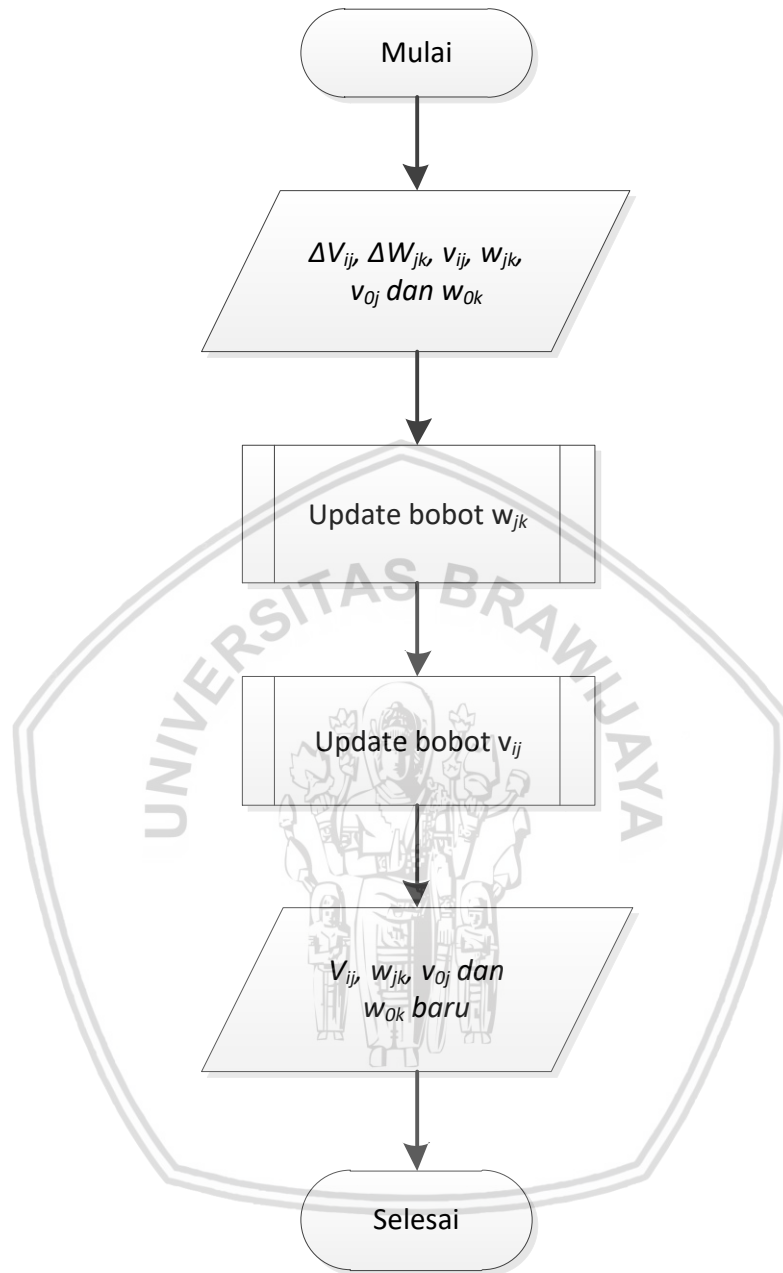
Gambar 4.11 Proses Perhitungan Δv_{ij} dan Δv_{0j}

Gambar 4.10 menjelaskan mengenai proses perhitungan Δv_{ij} dan Δv_{0j} . Tahapannya adalah sebagai berikut:

1. Sistem telah memiliki nilai α , δ_j , x .
2. Sistem melakukan perhitungan nilai Δv_{ij} dan Δv_{0j} .
3. Hasil akhir dari proses ini adalah mendapatkan nilai Δv_{ij} dan Δv_{0j} .



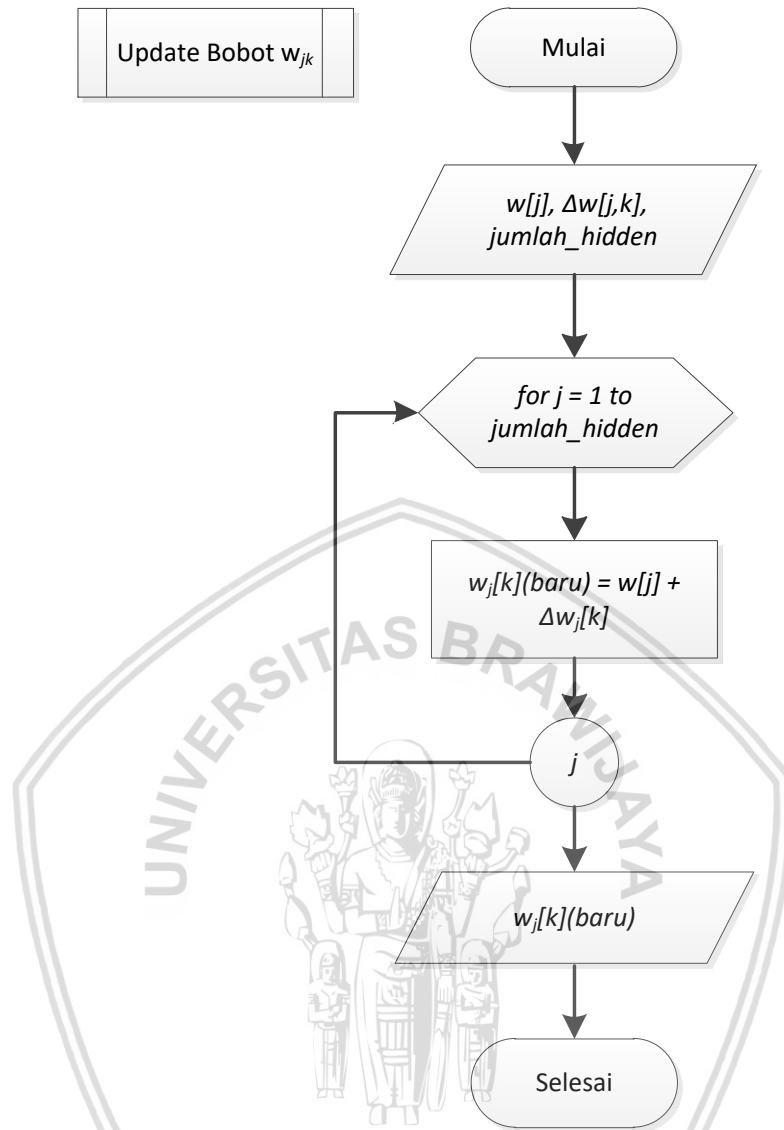
4.2.4 Proses *Update* Bobot dan Bias



Gambar 4.12 Proses *Update* Bobot dan Bias

Gambar 4.11 menjelaskan mengenai proses *update* bobot dan bias. Tahapannya adalah sebagai berikut:

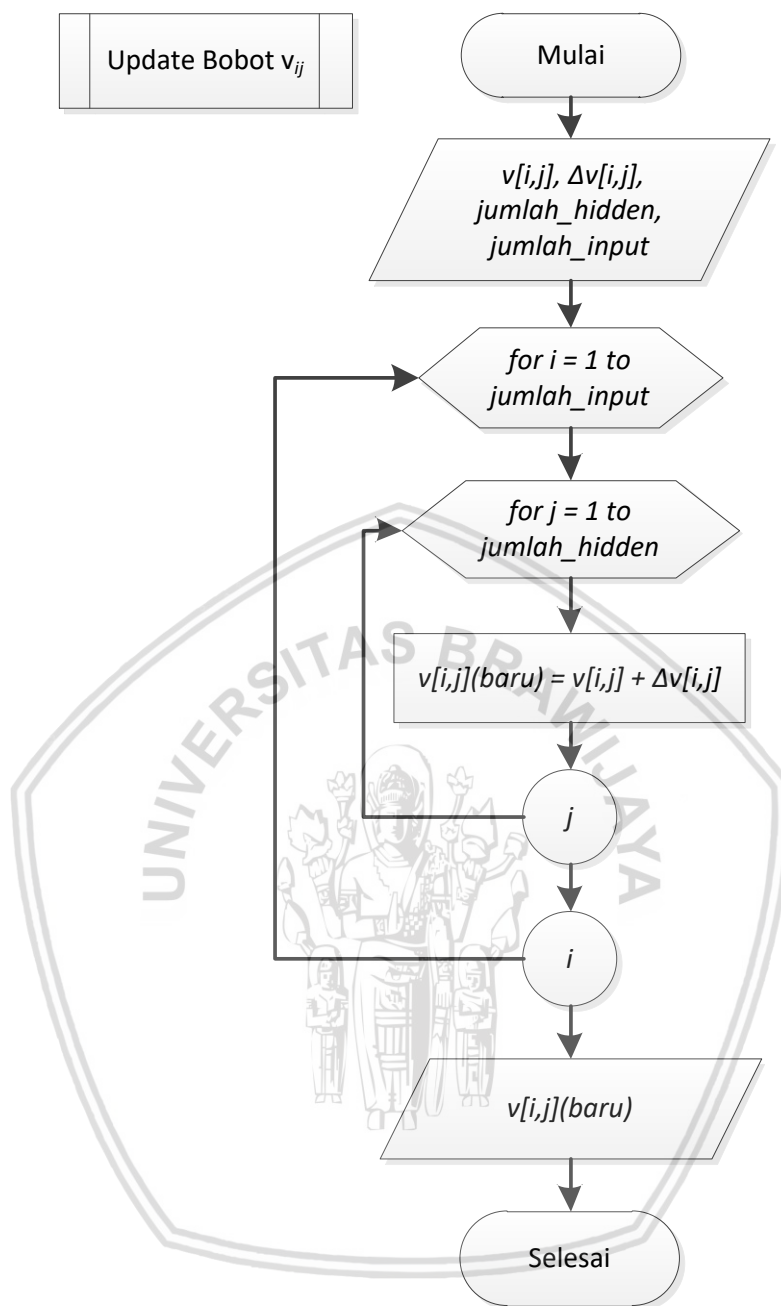
1. Sistem telah memiliki nilai yang dibutuhkan untuk *update* bobot dan bias yaitu Δv_{ij} , Δw_{jk} , v_{ij} , w_{jk} , v_{0j} dan w_{0k} .
2. Sistem melakukan *update* bobot w_{jk}
3. Sistem melakukan *update* bobot v_{ij}
4. Hasil akhir dari proses ini adalah menghasilkan bobot v dan w yang baru.



Gambar 4.13 Proses *Update* w_{jk}

Gambar 4.12 menjelaskan mengenai proses *update* bobot dan bias pada w . Tahapannya adalah sebagai berikut:

1. Sistem telah memiliki nilai yang dibutuhkan untuk *update* bobot dan bias yaitu nilai Δw_{jk} , w_{jk} .
2. Sistem melakukan *update* bobot berdasarkan nilai Δw_{jk} dan w_{jk} .
3. Hasil akhir dari proses ini adalah menghasilkan nilai bobot w dan bias terbaru.

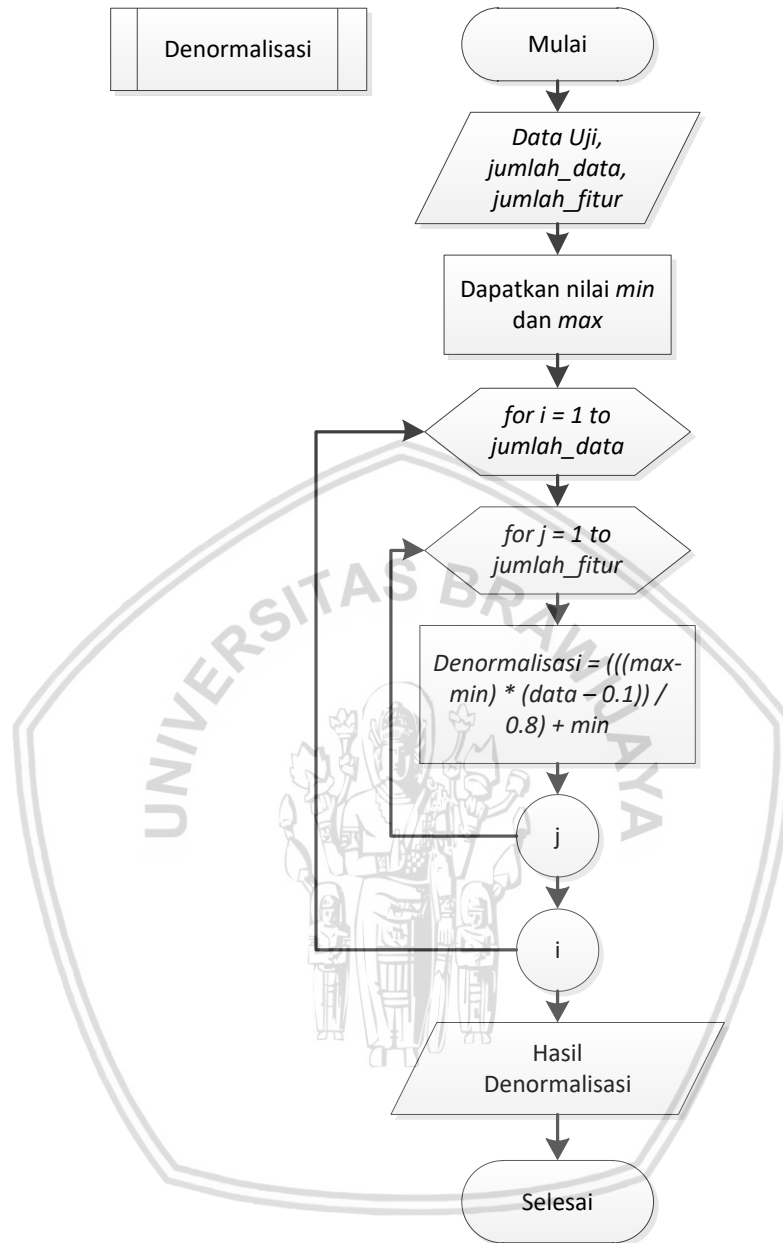


Gambar 4.14 Proses Update v_{ij}

Gambar 4.13 menjelaskan mengenai proses *update* bobot dan bias pada v . Tahapannya adalah sebagai berikut:

1. Sistem telah memiliki nilai yang dibutuhkan untuk *update* bobot dan bias yaitu nilai Δv_{ij} , v_{0j} .
2. Sistem melakukan *update* bobot berdasarkan nilai Δv_{ij} dan v_{ij} .
3. Hasil akhir dari proses ini adalah menghasilkan nilai bobot v dan bias terbaru.

4.2.5 Proses Denormalisasi



Gambar 4.15 Proses Denormalisasi

Gambar 4.14 menjelaskan mengenai proses denormalisasi. Tahapan denormalisasi ialah sebagai berikut:

1. Sistem mendapatkan data uji yang akan dilakukan proses denormalisasi.
2. Sistem mengambil nilai min dan max yang didapatkan dari proses normalisasi sebelumnya.
3. Sistem menghitung nilai denormalisasi pada tiap data.
4. Hasil dari proses ini adalah menghasilkan data denormalisasi atau data sebenarnya.

4.3 Perhitungan Manual Algoritme *Backpropagation*

Tabel 4.2 merupakan data latih yang digunakan untuk proses pelatihan sekaligus digunakan sebagai data uji. Banyaknya data yang digunakan sebanyak 10 data. Setiap data memiliki 6 *input* dan 1 target yang didapatkan dari pembentukan pola dari laporan permintaan semen di Indonesia. *Input* x1 merupakan banyaknya permintaan semen pada 6 bulan sebelum target. *Input* x2 merupakan banyaknya permintaan semen pada 5 bulan sebelum target. Begitu seterusnya hingga *input* x6 yang merupakan banyaknya permintaan semen pada 1 bulan sebelum target. Target adalah hasil yang diharapkan berdasarkan pola yang dihasilkan dari proses pelatihan data.

Detail perhitungan yang akan dilakukan dalam penelitian ini adalah sebagai berikut:

Jumlah data latih	: 10
Jumlah data uji	: 10
Iterasi	: 2
Jumlah <i>hidden layer</i>	: 4
<i>Learning rate</i>	: 0,9
Fungsi aktivasi	: sigmoid biner

Keseluruhan data latih dan data uji dapat dilihat pada Tabel 4.2.

Tabel 4.2 Data Latih dan Data Uji

<i>Data</i>	<i>x1</i>	<i>x2</i>	<i>x3</i>	<i>x4</i>	<i>x5</i>	<i>x6</i>	<i>target</i>
1	5633478	6062829	5737245	5503160	5172287	4554653	5027788
2	6062829	5737245	5503160	5172287	4554653	5027788	5033066
3	5737245	5503160	5172287	4554653	5027788	5033066	5475102
4	5503160	5172287	4554653	5027788	5033066	5475102	3731358
5	5172287	4554653	5027788	5033066	5475102	3731358	5638695
6	4554653	5027788	5033066	5475102	3731358	5638695	6495831
7	5027788	5033066	5475102	3731358	5638695	6495831	6303276
8	5033066	5475102	3731358	5638695	6495831	6303276	6754075
9	5475102	3731358	5638695	6495831	6303276	6754075	6365491
10	3731358	5638695	6495831	6303276	6754075	6365491	5798323

4.3.1 Normalisasi Data

Penelitian ini menggunakan normalisasi data terlebih dahulu untuk mendapatkan nilai kecil dan menghindari nilai 0. Proses normalisasi ini menggunakan metode *min-max* sehingga menghasilkan nilai terkecil sebesar 0,1

dan nilai terbesar 0,9. Nilai maksimum dan minimum didapat dari keseluruhan data latih.

Nilai maksimum : 6754075

Nilai minimum : 3731358

Dibawah ini merupakan rumus normalisasi seperti pada Persamaan 2.13.

$$x' = \frac{0,8(x - \text{min value})}{(\text{max value} - \text{min value})} + 0,1$$

$$x' = \frac{0,8(5633478 - 3731358)}{(6754075 - 3731358)} + 0,1 = 0,60342$$

x' merupakan hasil normalisasi, x adalah nilai awal data sebelum dinormalisasi, minvalue adalah nilai terkecil dari keseluruhan data latih dan maxvalue adalah nilai terbesar dari keseluruhan data latih. Dibawah ini adalah tabel data latih dan data uji setelah dilakukan normalisasi pada Tabel 4.3.

Tabel 4.3 Hasil Normalisasi Data Latih dan Data Uji

<i>Data</i>	<i>x1</i>	<i>x2</i>	<i>x3</i>	<i>x4</i>	<i>x5</i>	<i>x6</i>	<i>target</i>
1	0,60342	0,717053	0,630883	0,56893	0,48136	0,317895	0,443116
2	0,717053	0,630883	0,56893	0,48136	0,317895	0,443116	0,444513
3	0,630883	0,56893	0,48136	0,317895	0,443116	0,444513	0,561504
4	0,56893	0,48136	0,317895	0,443116	0,444513	0,561504	0,1
5	0,48136	0,317895	0,443116	0,444513	0,561504	0,1	0,604801
6	0,317895	0,443116	0,444513	0,561504	0,1	0,604801	0,831652
7	0,443116	0,444513	0,561504	0,1	0,604801	0,831652	0,78069
8	0,444513	0,561504	0,1	0,604801	0,831652	0,78069	0,9
9	0,561504	0,1	0,604801	0,831652	0,78069	0,9	0,797156
10	0,1	0,604801	0,831652	0,78069	0,9	0,797156	0,647048

4.3.2 Inisialisasi Bobot

Dalam penelitian ini inisialisasi bobot dan bias dilakukan secara acak dengan interval antara -1 sampai 1. Proses pembobotan dilakukan hanya pada iterasi pertama saja, sedangkan pada iterasi selanjutnya bobot akan diperbarui setelah dilakukan proses *backpropagation*. Inisialisasi bobot v dan w akan disajikan pada Tabel 4.4 dan Tabel 4.5.



Tabel 4.4 Inisialisasi Bobot dan Bias v

v	z1	z2	z3	z4
x1	0,313207	0,589315	-0,03124	0,23807
x2	0,272615	0,523063	0,411931	0,561262
x3	0,018999	0,766697	0,719645	0,318909
x4	-0,14878	-0,03624	0,048037	0,729079
x5	0,563229	0,533547	0,739758	0,249745
x6	-0,72138	-0,24635	0,59024	0,722783
bias	0,198291	0,580634	0,548087	0,747117

Tabel 4.5 Inisialisasi Bobot dan Bias w

w	y
z1	0,690769
z2	0,659942
z3	0,798908
z4	0,375527
bias	0,389938

4.3.3 Proses Feedforward

Setelah data *training* dan data uji telah tersedia dan nilai bobot telah selesai diinisialisasi, langkah selanjutnya dalam proses *feedforward* adalah menghitung nilai z_{net_j} . Perhitungan nilai z_{net_j} dilakukan sesuai dengan Persamaan 2.1. Nilai z_{net_j} didapat dari jumlah perkalian antara *input* x_i dengan bobot v pada masing – masing unit *hidden layer* ditambah dengan bias.

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

$$z_{net_j} = 0,198291348 + (0,603419936 * 0,313207271) + (0,717053068 * 0,272615346) + (0,630883176 * 0,018999196) + (0,568929642 * -0,148781525) + (0,481359949 * 0,563229184) + (0,317895357 * -0,721375585) = 0,551900606$$



Hasil perhitungan z_{net_j} untuk data ke-1 iterasi ke-1 ditunjukkan pada Tabel 4.6.

Tabel 4.6 Nilai z_{net_j}

Iterasi	Data	Nilai z_{net_j}			
		z_{net1}	z_{net2}	z_{net3}	z_{net4}
1	1	0.551901	1.952894	1.849676	2.259202

Setelah mendapatkan nilai z_{net_j} seperti pada Tabel 4.6, langkah berikutnya adalah membangkitkan nilai z_{net_j} dengan menggunakan fungsi aktivasi sigmoid guna memudahkan *hidden layer* untuk menemukan hasil pembelajaran yang direpresentasikan sebagai (z_j), (Yamanashi, Umeda, & Yoshikawa, 2013). Fungsi aktivasi yang digunakan seperti pada Persamaan 2.2.

$$z_j = f(z_{net_j}) = \frac{1}{1 + \exp(-z_{net_j})}$$

$$z_j = \frac{1}{1 + \exp(-0,551901)} = 0,634576434$$

Hasil perhitungan z_j untuk data ke-1 iterasi ke-1 ditunjukkan pada Tabel 4.7.

Tabel 4.7 Nilai z_j

Iterasi	Data	Nilai z_j			
		$z1$	$z2$	$z3$	$z4$
1	1	0.634576	0.875762	0.864089	0.905441

Setelah mendapatkan nilai z_j yang berada pada *hidden layer*, langkah selanjutnya adalah menghitung nilai y_{net_k} . Dalam penelitian ini hanya terdapat 1 unit *output* sehingga menghasilkan 1 y_{net_k} . Perhitungan y_{net_k} dilakukan berdasarkan Persamaan 2.3.

$$y_{net_k} = w_{0k} + \sum_{j=1}^p z_j \cdot w_{jk}$$

$$y_{net_k} = 0,389938746 + (0,634576434 * 0,690768683) + (0,875761882 * 0,659942041) + (0,864089034 * 0,798908143) + (0,905441363 * 0,375526681) = 2,436581513$$

Hasil perhitungan y_{net_k} untuk data ke-1 iterasi ke-1 ditunjukkan pada Tabel 4.8.

Tabel 4.8 Nilai y_{net_k}

Iterasi	Data	Nilai y_{net_k}
1	1	y_{net1}
		2.436581513

Setelah diketahui nilai y_{net_k} , langkah selanjutnya adalah mencari nilai y_k . Menghitung nilai y_k dilakukan dengan menggunakan fungsi aktivasi seperti pada persamaan 2.4.

$$y_k = f(y_{net_k}) = \frac{1}{1 + \exp(-y_{net_k})}$$

$$y_k = \frac{1}{1 + \exp(-2,436581513)} = 0,919574629$$

Hasil perhitungan y_k untuk data ke-1 iterasi ke-1 ditunjukkan pada Tabel 4.9.

Tabel 4.9 Nilai y_k

Iterasi	Data	Nilai y_k
1	1	$y1$
		0.919575

4.3.4 Proses *Backpropagation*

Proses *backpropagation* dilakukan untuk melakukan perhitungan *error* antara target dan keluaran proses *feedforward*. Hal ini dilakukan untuk melakukan koreksi bobot sehingga didapatkan bobot baru dengan tingkat *error* yang relatif rendah.

Langkah pertama yang dilakukan pada proses *backpropagation* adalah menghitung faktor *error* pada *output layer* atau disebut nilai δ_k . Menghitung nilai δ_k dengan menggunakan rumus pada Persamaan 2.5.

$$\delta_k = (t_k - y_k)f'(y_{net_k}) = (t_k - y_k)y_k(1 - y_k)$$

$$\delta_k = (0,443116474 - 0,919574629) * 0,919574629 * (1 - 0,919574629)$$

$$= -0,035237478$$

Hasil perhitungan δ_k untuk data ke-1 iterasi ke-1 ditunjukkan pada Tabel 4.10.

Tabel 4.10 Nilai δ_k

Iterasi	Data	Nilai δ_k
1	1	δ_1
		-0.035237

Setelah nilai δ_k diketahui, langkah selanjutnya yang perlu dilakukan adalah mencari nilai Δw_{jk} . Δw_{jk} digunakan untuk melakukan *update* bobot pada w . Untuk menghitung Δw_{jk} penulis menggunakan *learning rate* atau α sebesar 0,9. Menghitung nilai Δw_{jk} dengan menggunakan Persamaan 2.6. Nilai Δw_{jk} dicari dengan mengalikan nilai α , δ_k dan z_j .

$$\Delta w_{jk} = \alpha \delta_k z_j$$

$$\Delta w_{jk} = 0,9 * -0,035237478 * 0,634576434 = -0,020124786$$

Hasil perhitungan Δw_{jk} untuk data ke-1 iterasi ke-1 ditunjukkan pada Tabel 4.11.

Tabel 4.11 Nilai Δw_{jk}

Iterasi	Data	Nilai Δw_{jk}				
		Δw_{01}	Δw_{11}	Δw_{21}	Δw_{31}	Δw_{41}
1	1	-0.03171	-0.02012	-0.02777	-0.0274	-0.02871

Setelah diketahui nilai Δw_{jk} maka langkah selanjutnya adalah mencari nilai δ_{net} dari proses perkalian antara δ_k dan Δw_{jk} seperti yang ditunjukkan pada Persamaan 2.7.

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{jk}$$

$$\delta_{net_j} = -0,035237478 * -0,020124786 = 0,000709147$$

Hasil perhitungan δ_{net_j} untuk data ke-1 iterasi ke-1 ditunjukkan pada Tabel 4.12.

Tabel 4.12 Nilai δ_{net_j}

Iterasi	Data	δ_{net_j}			
		δ_{net1}	δ_{net2}	δ_{net3}	δ_{net4}
1	1	0.000709	0.000979	0.000966	0.001012

Langkah selanjutnya adalah mencari nilai δ_j . Nilai δ_j didapatkan dengan mengalikan δ_{net_j} , z_j dan hasil pengurangan antara 1 dikurangi z_j . Menghitung nilai δ_j menggunakan rumus seperti pada Persamaan 2.8.



$$\delta_j = \delta_{net_j} \cdot f'(z_{net_j}) = \delta_{net_j} \cdot z_j(1 - z_j)$$

$$\delta_j = 0,000709147 * 0,634576434 * (1 - 0,634576434) = 0,000164443$$

Hasil perhitungan δ_j untuk data ke-1 iterasi ke-1 ditunjukkan pada Tabel 4.13.

Tabel 4.13 Nilai δ_j

Iterasi	Data	δ_j			
		δ_1	δ_2	δ_3	δ_4
1	1	0.000164	0.000106	0.000113	8.66E-05

Setelah dilakukan perhitungan nilai δ_j , langkah berikutnya adalah mencari nilai Δv_{ij} yang nantinya digunakan untuk melakukan *update* bobot pada v . Mencari nilai Δv_{ij} dengan mengalikan antara α , δ_j dan x_i seperti ditunjukkan pada Persamaan 2.9.

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

$$\Delta v_{11} = 0,9 * 0,000164443 * 0,603419936 = 8,93056E - 05$$

Hasil perhitungan nilai Δv_{ij} dan Δv_{oj} untuk data ke-1 iterasi ke-1 ditunjukkan pada Tabel 4.14.

Tabel 4.14 Nilai Δv_{ij} dan Δv_{oj}

Iterasi	Data	Nilai Δv_{ij}				
		v	z_1	z_2	z_3	z_4
Iterasi 1	Data 1	x1	8.93056E-05	5.78284E-05	6.15865E-05	4.70475E-05
		x2	0.000106123	6.87184E-05	7.31842E-05	5.59072E-05
		x3	9.33701E-05	6.04603E-05	6.43895E-05	4.91887E-05
		x4	8.42011E-05	5.45231E-05	5.80664E-05	4.43583E-05
		x5	7.12408E-05	4.61309E-05	4.91288E-05	3.75307E-05
		x6	4.70482E-05	3.04653E-05	3.24452E-05	2.47857E-05
		bias	0.000147999	9.58344E-05	0.000102062	7.7968E-05

4.3.5 Proses *Update* Bobot

Proses *update* bertujuan untuk memperbaiki bobot awal dengan menggunakan perbedaan *error* seperti yang telah didapatkan saat proses *backpropagation*.

Langkah pertama yaitu melakukan *update* bobot dan bias pada w . Mencari nilai w baru dengan menjumlahkan nilai w lama dengan Δw_{jk} seperti ditunjukkan pada Persamaan 2.10.

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

$$w_{jk}(\text{baru}) = 0,690768683 + (-0,020124786) = 0,670643897$$

Hasil perhitungan bobot dan bias w (baru) untuk data ke-2 iterasi ke-1 ditunjukkan pada Tabel 4.15.

Tabel 4.15 Nilai Bobot dan Bias w (baru)

Iterasi	Data	Nilai w (baru)				
		$w01$	$w11$	$w12$	$w13$	$w14$
1	2	0.358225	0.670644	0.632168	0.771505	0.346812

Setelah melakukan *update* bobot pada w , selanjutnya adalah melakukan *update* bobot pada v . Mencari nilai v baru dengan menjumlahkan nilai v lama dengan Δv_{ij} seperti ditunjukkan pada Persamaan 2.11.

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

$$v_{ij}(\text{baru}) = 0,313207271 + 8,93056E - 05 = 0,313296577$$

Hasil perhitungan bobot dan bias v (baru) untuk data ke-2 iterasi ke-1 ditunjukkan pada Tabel 4.16.

Tabel 4.16 Nilai Bobot dan Bias v (baru)

Iterasi	Data	Nilai v (baru)				
		v	$z1$	$z2$	$z3$	$z4$
1	2	$x1$	0,313297	0,589372	-0,03118	0,238117
		$x2$	0,272721	0,523131	0,412005	0,561318
		$x3$	0,019093	0,766758	0,719709	0,318958
		$x4$	-0,1487	-0,03619	0,048095	0,729123
		$x5$	0,5633	0,533594	0,739807	0,249782
		$x6$	-0,72133	-0,24632	0,590272	0,722808
		bias	0,198439	0,58073	0,548189	0,747195

4.3.6 Proses Perhitungan *Mean Square Error (MSE)*

Setelah dilakukan proses perhitungan menggunakan algoritme *backpropagation* hingga menghasilkan nilai v_{ij} dan w_{jk} baru, maka dapat dilakukan perhitungan *error* pada tiap data yang dilatih. Nilai *error* dihitung menggunakan rumus sebagai berikut:

$$\text{Error} = t_k - y_k$$

Dimana t_k adalah target pada data ke-k dan y_k adalah hasil perhitungan *feedforward* pada data ke-k. Hasil perhitungan *error* pada masing – masing data ditunjukkan

Dalam perhitungan manual ini semua proses yang telah dijelaskan diatas diulang sebanyak jumlah data kemudian diulang kembali sesuai dengan banyaknya iterasi yang telah ditentukan. Hasil perhitungan *error* pada iterasi ke-1 ditunjukkan pada Tabel 4.17.

Tabel 4.17 Nilai Error

	Data	Error
Iterasi 1	1	0.476458
	2	0.461713
	3	0.332949
	4	0.779541
	5	0.248473
	6	-0.00671
	7	0.059722
	8	-0.06303
	9	0.042865
	10	0.198333

Dari data tersebut kemudian dihitung MSE-nya dengan mengkuadratkan setiap *error* pada data dalam satu iterasi. Kemudian diambil rata – ratanya dengan menjumlahkan semua *error* dan dibagi banyaknya data. Dibawah ini adalah contoh perhitungan MSE berdasarkan pada Persamaan 2.12.

$$MSE = \frac{\sum(t - y_k)^2}{n}$$

$$MSE = \frac{(0.476458)^2 + (0.461713)^2 + (0.332949)^2 + (0.779541)^2 + (0.248473)^2 + (-0.00671)^2 + (0.059722)^2 + (-0.06303)^2 + (0.042865)^2 + (0.198333)^2}{10}$$

$$= 0.126923$$

Pada Tabel 4.18 menunjukkan hasil perhitungan MSE pada iterasi ke-1.

Tabel 4.18 Nilai MSE

Iterasi	MSE
1	0.126923



Dalam Perhitungan MSE yang perlu diperhatikan adalah nilai MSE yang terus menurun. Hal tersebut menunjukkan bahwa proses pelatihan berjalan dengan baik.

4.3.7 Hasil Perhitungan Algoritme *Backpropagation* Iterasi 2

Proses perhitungan algoritme *backpropagation* seperti yang telah dijelaskan pada subbab 4.3.3 sampai 4.3.6 selanjutnya diulang sebanyak iterasi yang telah ditentukan. Dalam kasus ini penulis menggunakan iterasi sebanyak 2. Dibawah ini menyajikan hasil perhitungan algoritme *backpropagation* pada iterasi ke-2. Data yang ditampilkan adalah data ke-10 yang mana merupakan data terakhir dari proses pelatihan.

Hasil perhitungan z_{netj} pada iterasi ke-2 data ke-10 ditunjukkan pada Tabel 4.19

Tabel 4.19 Nilai z_{netj}

Iterasi	Data	Nilai z_{netj}			
		z_{net1}	z_{net2}	z_{net3}	z_{net4}
2	10	0,238278	1,859134	2,576013	2,753088

Hasil perhitungan z_j pada iterasi ke-2 data ke-10 ditunjukkan pada Tabel 4.20.

Tabel 4.20 Nilai z_j

Iterasi	Data	Nilai z_j			
		$z1$	$z2$	$z3$	$z4$
2	10	0,559289257	0,865195923	0,92930177	0,940088

Hasil perhitungan y_{netk} pada iterasi ke-2 data ke-10 ditunjukkan pada Tabel 4.21.

Tabel 4.21 y_{netk}

Iterasi	Data	Nilai y_{netk}
2	10	y_{net1}
		1,104856399

Hasil perhitungan y_k pada iterasi ke-2 data ke-10 ditunjukkan pada Tabel 4.22.



Tabel 4.22 Nilai y_k

Iterasi	Data	Nilai y_k
2	10	y_1
		0,751168942

Hasil perhitungan δ_k pada iterasi ke-2 data ke-10 ditunjukkan pada Tabel 4.23.

Tabel 4.23 Nilai δ_k

Iterasi	Data	Nilai δ_k
2	10	δ_1
		-0,019461635

Hasil perhitungan Δw_{jk} pada iterasi ke-2 data ke-10 ditunjukkan pada Tabel 4.24.

Tabel 4.24 Nilai Δw_{jk}

Iterasi	Data	Nilai Δw_{jk}				
		Δw_{jk_0}	Δw_{jk_1}	Δw_{jk_2}	Δw_{jk_3}	Δw_{jk_4}
2	10	-0,01752	-0,0098	-0,01515	-0,01628	-0,01647

Hasil perhitungan δ_{net_j} pada iterasi ke-2 data ke-10 ditunjukkan pada Tabel 4.25.

Tabel 4.25 Nilai δ_{net_j}

Iterasi	Data	δ_{net_j}			
		$\delta_{net_j_1}$	$\delta_{net_j_2}$	$\delta_{net_j_3}$	$\delta_{net_j_4}$
2	10	0,000191	0,000295	0,000317	0,00032

Hasil perhitungan δ_j pada iterasi ke-2 data ke-10 ditunjukkan pada Tabel 4.26.

Tabel 4.26 Nilai δ_j

Iterasi	Data	δ_j			
		δ_{j_1}	δ_{j_2}	δ_{j_3}	δ_{j_4}
2	10	4,69924E-05	3,4398E-05	2,08E-05	1,80491E-05

Hasil perhitungan Δv_{ij} dan Δv_{oj} pada iterasi ke-2 data ke-10 ditunjukkan pada Tabel 4.27.



Tabel 4.27 Nilai Δv_{ij} dan Δv_{0j}

		Nilai Δv_{ij}			
v		$\Delta z1$	$\Delta z2$	$\Delta z3$	$\Delta z4$
$x1$		4,22932E-06	3,09582E-06	1,87312E-06	1,62442E-06
$x2$		2,55789E-05	1,87235E-05	1,13286E-05	9,82448E-06
$x3$		3,51732E-05	2,57465E-05	1,55779E-05	1,35095E-05
$x4$		3,30179E-05	2,41688E-05	1,46233E-05	1,26817E-05
$x5$		3,80639E-05	2,78624E-05	1,68581E-05	1,46198E-05
$x6$		3,37143E-05	2,46785E-05	1,49317E-05	1,29491E-05
bias		4,22932E-05	3,09582E-05	1,87312E-05	1,62442E-05

Hasil perhitungan Bobot dan Bias w (baru) pada iterasi ke-2 data ke-10 ditunjukkan pada Tabel 4.28.

Tabel 4.28 Nilai Bobot dan Bias w (baru)

Nilai w (baru)				
$w01$	$w11$	$w12$	$w13$	$w14$
0,028031	0,467426	0,351386	0,494818	0,054831

Hasil perhitungan Bobot dan Bias v (baru) pada iterasi ke-2 data ke-10 ditunjukkan pada Tabel 4.29.

Tabel 4.29 Nilai Bobot dan Bias v (baru)

Bobot dan Bias v (baru)				
v	$z1$	$z2$	$z3$	$z4$
$x1$	0,315673236	0,591320159	-0,02931392	0,239547798
$x2$	0,274864538	0,524883311	0,413683947	0,562600802
$x3$	0,020703089	0,768058074	0,720972051	0,31992765
$x4$	-0,1468021	-0,0346151	0,04958987	0,730262785
$x5$	0,565125322	0,535101221	0,741229816	0,250877076
$x6$	-0,71919947	-0,24452174	0,591956832	0,724091317
bias	0,202563318	0,584148976	0,551446523	0,749691115

4.3.8 Proses Pengujian

Proses pengujian memerlukan bobot v dan w terakhir dari proses pelatihan. Tabel 4.30 dan 4.31 menyajikan bobot v dan w terbaru yang nantinya



akan dilakukan proses *feedforward* saja untuk melakukan pengujian data. Proses dilakukannya pengujian adalah sebagai berikut:

Tabel 4.30 Bobot v baru

Nilai Δv_{ij}				
v	$z1$	$z2$	$z3$	$z4$
$x1$	0,315673236	0,591320159	-0,02931392	0,239547798
$x2$	0,274864538	0,524883311	0,413683947	0,562600802
$x3$	0,020703089	0,768058074	0,720972051	0,31992765
$x4$	-0,1468021	-0,0346151	0,04958987	0,730262785
$x5$	0,565125322	0,535101221	0,741229816	0,250877076
$x6$	-0,71919947	-0,24452174	0,591956832	0,724091317
bias	0,202563318	0,584148976	0,551446523	0,749691115

Tabel 4.31 Bobot w baru

Nilai w (baru)				
bias	$w11$	$w12$	$w13$	$w14$
0,028031	0,467426	0,351386	0,494818	0,054831

Langkah pertama adalah menghitung nilai z_{netj} seperti pada Persamaan (2.1) berikut ini:

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

$$\begin{aligned} z_{netj} &= 0,202563318 + (0,603419936 * 0,315673236) + (0,717053068 \\ &\quad * 0,274864538) + (0,630883176 * 0,020703089) \\ &\quad + (0,568929642 * -0,1468021) + (0,481359949 \\ &\quad * 0,565125322) + (0,317895357 * -0,71919947) \\ &= 0,563078992 \end{aligned}$$

Hasil perhitungan z_{netj} untuk data uji ke-1 sampai ke-10 ditunjukkan pada Tabel 4.32.

Tabel 4.32 Nilai z_{netj}

Data Uji	z_{net1}	z_{net2}	z_{net3}	z_{net4}
1	0,563079	1,962038	1,858432	2,265906
2	0,404401	1,82136	1,723405	2,21054
3	0,452115	1,742952	1,722705	2,040082
4	0,303372	1,50261	1,64694	2,100184
5	0,631212	1,636606	1,485761	1,723502
6	-0,02698	1,23231	1,505903	2,090416
7	0,205233	1,627569	2,072731	2,112512
8	0,319022	1,451716	1,951371	2,419666
9	0,091643	1,602079	2,165073	2,588815
10	0,238278	1,859134	2,576013	2,753088

Setelah mendapatkan nilai z_{netj} seperti pada Tabel 4.32, langkah berikutnya adalah mencari nilai z_j dengan menggunakan fungsi aktivasi sigmoid seperti pada Persamaan (2.2).

$$z_j = f(z_{netj}) = \frac{1}{1 + \exp(-z_{netj})}$$

$$z_j = \frac{1}{1 + \exp(-0,563079)} = 0,637165$$

Hasil perhitungan z_j untuk data uji ke-1 sampai ke-10 ditunjukkan pada Tabel 4.33.

Tabel 4.33 Nilai z_j

Data Uji	$z1$	$z2$	$z3$	$z4$
1	0,637165	0,876753	0,865114	0,906014
2	0,599745	0,860729	0,848567	0,901192
3	0,611142	0,851062	0,848477	0,884942
4	0,575267	0,817963	0,838477	0,890921
5	0,652764	0,837073	0,815441	0,848579
6	0,493257	0,774223	0,818453	0,889968
7	0,551129	0,835836	0,888224	0,892113
8	0,579086	0,810262	0,875596	0,918315
9	0,522895	0,832309	0,897069	0,930138
10	0,559289	0,865196	0,929302	0,940088



Setelah mendapatkan nilai z_j yang berada pada *hidden layer*, langkah selanjutnya adalah menghitung nilai y_{net_k} . Dalam penelitian ini hanya terdapat 1 unit *output* sehingga menghasilkan 1 y_{net_k} . Perhitungan y_{net_k} dilakukan berdasarkan Persamaan (2.3).

$$y_{net_k} = w_{0k} + \sum_{j=1}^p z_j \cdot w_{jk}$$

$$\begin{aligned} y_{net_k} &= 0,028031 + (0,655078 * 0,467426) + (0,883708 * 0,351386) \\ &\quad + (0,871966 * 0,494818) + (0,909828 * 0,054831) \\ &= 1,111689 \end{aligned}$$

Hasil perhitungan y_{net_k} untuk data uji ke-1 sampai ke-10 ditunjukkan pada Tabel 4.34.

Tabel 4.34 Nilai y_{net_k}

Data Uji	y_{net}
1	1,111689
2	1,080115
3	1,08111
4	1,04809
5	1,077309
6	0,984426
7	1,067769
8	1,067039
9	1,059794
10	1,104856

Setelah diketahui nilai y_{net_k} , langkah selanjutnya adalah mencari nilai y_k . Menghitung nilai y_k dilakukan dengan menggunakan fungsi aktivasi seperti pada persamaan (2.4).

$$y_k = f(y_{net_k}) = \frac{1}{1 + \exp(-y_{net_k})}$$

$$y_k = \frac{1}{1 + \exp(-1,111689)} = 0,752443854$$

Hasil perhitungan y_k untuk data uji ke-1 sampai ke-10 ditunjukkan pada Tabel 4.35.



Tabel 4.35 Nilai y_k

Data Uji	y
1	0,752443854
2	0,746515742
3	0,746703956
4	0,740407987
5	0,745984429
6	0,727985604
7	0,744172447
8	0,74403332
9	0,742651098
10	0,751168942

Selanjutnya dilakukan denormalisasi data. Denormalisasi bertujuan untuk mengembalikan nilai y_k menjadi nilai yang sebenarnya. Proses denormalisasi dilakukan sesuai dengan Persamaan (2.14) seperti dibawah ini:

$$x = \frac{(\max\ value - \min\ value)(x' - 0,1)}{0,8} + \min\ value$$

$$x = \frac{(6754075 - 3731358)(0,752443854 - 0,1)}{0,8} + 3731358 = 6196549$$

Hasil denormalisasi y_k untuk data uji ke-1 sampai ke-10 ditunjukkan pada Tabel 4.36.

Tabel 4.36 Hasil Prediksi

Data Uji	Aktual	Prediksi	Simpangan	Absolute
1	5027788	6196549	1168761	1168761
2	5033066	6174151	1141085	1141085
3	5475102	6174862	699760	699760
4	3731358	6151073	2419715	2419715
5	5638695	6172143	533448	533448
6	6495831	6104136	-391695	391695
7	6303276	6165297	-137979	137979
8	6754075	6164771	-589304	589304
9	6365491	6159548	-205943	205943
10	5798323	6191732	393409	393409
Mean Absolute Deviation (MAD)				768109,9

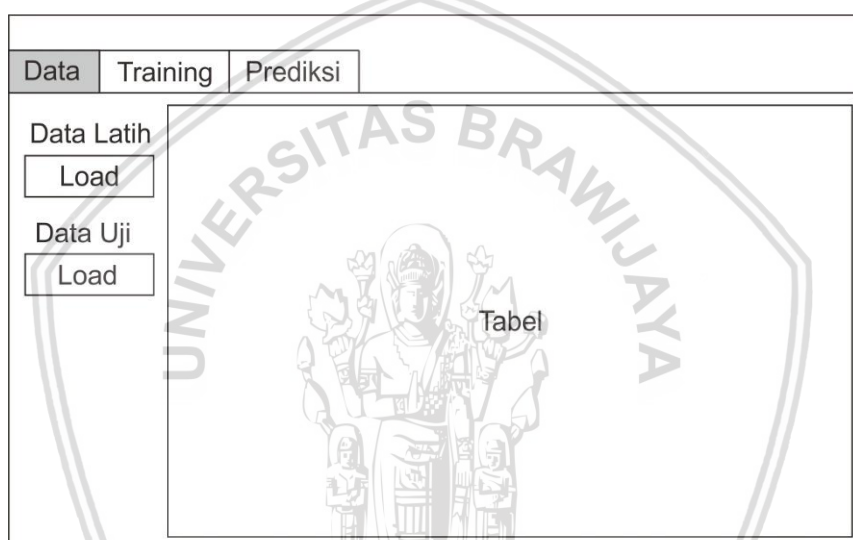


Dari Tabel 4.36 diatas, proses pengujian menghasilkan MSE sebesar 0,070146116 dan *Mean Absolute Deviation* (MAD) sebesar 768109,9 sehingga tingkat kesalahan prediksi mencapai 768109,9 ton semen. hasil prediksi tersebut kurang memuaskan sehingga memerlukan proses pelatihan lebih banyak lagi untuk mendapatkan bobot yang lebih baik.

4.4 Perancangan Antarmuka

Perancangan antarmuka menggambarkan bentuk dari program prediksi permintaan semen di Indonesia yang akan dibuat. Rancangan antarmuka dari program ini terdiri atas tiga tab yaitu tab load data, tab proses pelatihan dan tab prediksi.

4.4.1 Perancangan Halaman *Load Data*



Gambar 4.16 Perancangan Halaman *Load Data*

Keterangan:

1. Tab Data berisi halaman untuk melakukan *load* data baik data latih maupun data uji.
2. *Button load* digunakan untuk memilih data yang akan dimasukkan.
3. Data akan telah di-*load* akan ditampilkan pada tabel.

4.4.2 Perancangan Halaman *Training*

Data	Training	Prediksi
Iterasi	<input type="text"/>	Tabel
Learning Rate	<input type="text"/>	
Interval Bobot	<input type="text"/>	
PROSES		

Gambar 4.17 Perancangan Halaman *Training*

Keterangan:

1. Tab *Training* berisi halaman untuk melakukan proses pelatihan algoritme *backpropagation*.
2. Terdapat 3 *textbox* yaitu jumlah iterasi, interval bobot awal dan *alpha*.
3. Tombol proses digunakan untuk memulai proses pelatihan.

4.4.3 Perancangan Halaman *Prediksi*

Data	Training	Prediksi
		Tabel

Gambar 4.18 Perancangan Halaman *Prediksi*

Keterangan:

1. Tab *Prediksi* berisi halaman untuk menampilkan hasil dari proses prediksi pada tabel.



4.5 Perancangan Pengujian Sistem

Tahap perancangan pengujian sistem digunakan untuk melihat keandalan sistem dalam melakukan pelatihan sampai menghasilkan MSE terkecil. Termasuk sistem yang minim terjadi *error*. Pengujian sistem direncanakan akan menggunakan beberapa parameter pengujian seperti berikut:

1. Pengujian *learning rate (alpha)*.
2. Pengujian iterasi maksimum.
3. Pengujian jumlah data latih.
4. Pengujian hasil prediksi.

4.5.1 Pengujian Konvergensi

Pengujian ini dilakukan untuk mengetahui berapa banyak iterasi yang dibutuhkan untuk mencapai MSE terkecil. Pengujian ini dilakukan sebanyak 10 kali percobaan kemudian dilihat pada iterasi keberapa MSE mengalami konvergensi yaitu situasi dimana MSE berada pada titik yang rendah dan tidak mengalami perubahan yang signifikan pada iterasi selanjutnya. Perancangan pengujian konvergensi ditunjukkan pada Tabel 4.38.

Tabel 4.37 Pengujian Iterasi Maksimum

Iterasi	MSE Percobaan ke- <i>i</i>									
	1	2	3	4	5	6	7	8	9	10
20										
40										
60										
80										
100										
120										
140										
160										
180										
200										
220										
240										
260										
280										
300										

4.5.2 Pengujian *Learning rate*

Pengujian ini dimaksudkan untuk mencari nilai *learning rate* terbaik. Dalam implementasinya, *learning rate* digunakan untuk menentukan seberapa cepat proses pembelajaran berjalan hingga mencapai nilai MSE minimum. Pengujian *learning rate* ini dilakukan sebanyak 10 kali percobaan kemudian masing – masing pengujian diambil rata – ratanya. Perancangan pengujian *learning rate* ditunjukkan pada Tabel 4.37.

Tabel 4.38 Pengujian *Learning rate*

<i>Alpha</i>	MSE Percobaan ke-i										Rata-rata MSE
	1	2	3	4	5	6	7	8	9	10	
0,1											
0,2											
0,3											
0,4											
0,5											
0,6											
0,7											
0,8											
0,9											

4.5.3 Pengujian Interval Bobot Awal

Percobaan ini dimaksudkan untuk mengetahui interval bobot awal yang optimal. Dalam pengujian ini akan dilakukan pengujian interval bobot awal mulai dari -1 sampai 1, -0.9 sampai 0.9, -0.8 sampai 0.8 dan seterusnya hingga interval -0.5 sampai 0.5. Percobaan dilakukan sebanyak 10 kali dan diambil rata – ratanya. Perancangan pengujian jumlah data latih ditunjukkan pada Tabel 4.39.

Tabel 4.39 Pengujian Interval Bobot Awal

<i>Range Bobot</i>	MSE Percobaan ke-i										Rata-rata MSE
	1	2	3	4	5	6	7	8	9	10	
-1,0 s/d 1,0											
-0,9 s/d 0,9											
-0,8 s/d 0,8											
-0,7 s/d 0,7											
-0,6 s/d 0,6											
-0,5 s/d 0,5											



4.5.4 Pengujian Hasil Prediksi

Percobaan hasil prediksi dilakukan dengan menggunakan proses *feedforward* setelah mendapatkan bobot terakhir dari proses pelatihan. Pengujian ini akan membandingkan data prediksi dengan data asli pada 10 bulan terakhir tahun 2017. Perancangan pengujian hasil prediksi ditunjukkan pada Tabel 4.40.

Tabel 4.40 Pengujian Hasil Prediksi

Data Uji	Aktual	Prediksi	Error	Selisih	Absolute
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
MSE				MAD	

4.6 Pengambilan Kesimpulan

Proses pengambilan kesimpulan dilakukan jika semua proses penelitian telah selesai dilakukan. Proses ini menjelaskan mengenai kesimpulan hasil penelitian dan saran yang nantinya diharapkan dapat memberi masukan untuk pengembangan sistem di kemudian hari.

BAB 5 IMPLEMENTASI

Bab implementasi membahas mengenai penerapan algoritme *backpropagation* untuk memprediksi jumlah permintaan semen di Indonesia sesuai dengan perancangan yang telah dibahas pada bab sebelumnya.

5.1 Implementasi Program

Dalam subbab implementasi program ini akan menjelaskan tentang pembuatan program berdasarkan perancangan yang telah dijelaskan pada bab sebelumnya mengenai prediksi permintaan semen di Indonesia menggunakan algoritme *backpropagation*. Pembuatan program ini menggunakan bahasa pemrograman java.

5.1.1 Proses *Generate* Bobot

Proses *generate* bobot merupakan proses yang digunakan untuk menghasilkan bobot v dan bobot w secara acak dengan interval antara -1 sampai dengan 1. Proses *generate* bobot disajikan pada Kode Program 5.1.

```
1 public void generateBobot() {
2     Random random = new Random();
3     String a = combo_range.getSelectedItem().toString();
4
5     double min = 0;
6     double max = 0;
7
8     if (a == "-1,5 s/d 1,5") {
9         min = -1.5;
10        max = 1.5;
11    }
12    if (a == "-1,4 s/d 1,4") {
13        min = -1.4;
14        max = 1.4;
15    }
16    if (a == "-1,3 s/d 1,3") {
17        min = -1.3;
18        max = 1.3;
19    }
20    if (a == "-1,2 s/d 1,2") {
21        min = -1.2;
22        max = 1.2;
23    }
24    if (a == "-1,1 s/d 1,1") {
25        min = -1.1;
26        max = 1.1;
27    }
28    if (a == "-1 s/d 1") {
29        min = -1;
30        max = 1;
31    }
32    if (a == "-0,9 s/d 0,9") {
33        min = -0.9;
34        max = 0.9;
35    }
}
```

```

36     if (a == "-0,8 s/d 0,8") {
37         min = -0.8;
38         max = 0.8;
39     }
40     if (a == "-0,7 s/d 0,7") {
41         min = -0.7;
42         max = 0.7;
43     }
44     if (a == "-0,6 s/d 0,6") {
45         min = -0.6;
46         max = 0.6;
47     }
48     if (a == "-0,5 s/d 0,5") {
49         min = -0.5;
50         max = 0.5;
51     }
52
53     for (int i = 0; i < jumlah_input; i++) {
54         for (int j = 0; j < jumlah_hidden; j++) {
55             double ve0 = Math.random() * (max-min) + min;
56             v0[j] = ve0;
57             v0_temp[j] = ve0;
58             double ve = Math.random() * (max-min) + min;
59             v[i][j] = ve;
60             v_temp[i][j] = ve;
61         }
62     }
63
64     for (int i = 0; i < jumlah_output; i++) {
65         double we0 = Math.random() * (max-min) + min;
66         w0[i] = we0;
67         w0_temp[i] = we0;
68         for (int j = 0; j < jumlah_hidden; j++) {
69             double we = Math.random() * (max-min) + min;
70             w[j][i] = we;
71             w_temp[j][i] = we;
72         }
73     }
74 }

```

Kode Program 5.1 Proses Generate Bobot

Penjelasan Kode Program 5.1 :

1. Baris 4 sampai 11 adalah proses mendapatkan nilai *random*. Dapatkan nilai *random* dengan interval -1 sampai 1 kemudian dimasukkan pada *array v* dan *v0*.
2. Baris 13 sampai 20 merupakan perulangan untuk mendapatkan nilai *random*. Dapatkan nilai *random* dengan interval -1 sampai 1 kemudian dimasukkan pada *array w* dan *w0*.

5.1.2 Proses Pelatihan

Proses pelatihan merupakan inti dari program dimana melakukan pelatihan terhadap data. Proses pelatihan mencakup proses *feedforward*, *backpropagation* dan *update* bobot. Proses pelatihan akan diperlihatkan pada Kode Program 5.2.

```

1 private void btn_prosesActionPerformed
2 (java.awt.event.ActionEvent evt) {
3     DefaultTableModel Modeltbl_v_baru = (DefaultTableModel)
4     tbl_v_baru.getModel();
5     DefaultTableModel Modeltbl_w_baru = (DefaultTableModel)
6     tbl_w_baru.getModel();
7     DefaultTableModel Modeltbl_v_baru2 = (DefaultTableModel)
8     tbl_v_baru2.getModel();
9     DefaultTableModel Modeltbl_w_baru2 = (DefaultTableModel)
10    tbl_w_baru2.getModel();
11    DefaultTableModel Modeltbl_mse = (DefaultTableModel)
12    tbl_mse.getModel();
13    Modeltbl_mse.setRowCount(0);
14
15    int iterasi = Integer.parseInt(txt_iterasi.getText());
16    double alpha = Double.parseDouble(txt_alpha.getText());
17
18    for (int s = 0; s < iterasi; s++) {
19        double mse = 0;
20        for (int h = 0; h < jumlah_data; h++) {
21            for (int j = 0; j < jumlah_hidden; j++) {
22                double jml = 0;
23                for (int i = 0; i < jumlah_input; i++) {
24                    double hitung = x[h][i] * v[i][j];
25                    jml = jml + hitung;
26                }
27                z_net[j] = v0[j] + jml;
28                z[j] = 1 / (1 + (double) Math.exp(-
29                z_net[j]));
30            }
31            for (int k = 0; k < jumlah_output; k++) {
32                double jml = 0;
33                for (int j = 0; j < jumlah_hidden; j++) {
34                    double hitung = z[j] * w[j][k];
35                    jml = jml + hitung;
36                }
37                y_net[k] = w0[k] + jml;
38                y[k] = 1 / (1 + (double) Math.exp(-
39                y_net[k]));
40            }
41            mse = mse + Math.pow((x[h][6] - y[k]), 2);
42        }
43    }
44
45    for (int k = 0; k < jumlah_output; k++) {
46        dk[k] = (x[h][6] - y[k]) * y[k] * (1 -
47        y[k]);
48        for (int j = 0; j < jumlah_hidden; j++) {
49            delta_w[j][k] = alpha * dk[k] * z[j];
50            delta_w0[k] = alpha * dk[k];
51        }
52    }
53
54    for (int j = 0; j < jumlah_hidden; j++) {
55        for (int k = 0; k < jumlah_output; k++) {
56            d_netj[j] = (dk[k] * delta_w[j][k]);
57        }
58        dj[j] = d_netj[j] * (z[j]) * (1 - z[j]);
59        for (int i = 0; i < jumlah_input; i++) {

```

```

52         delta_v[i][j] = alpha * dj[j] * x[0][i];
53     }
54     delta_v0[j] = alpha * dj[j];
55 }
56
57 for (int j = 0; j < jumlah_hidden; j++) {
58     for (int k = 0; k < jumlah_output; k++) {
59         w[j][k] = w[j][k] + delta_w[j][k];
60     }
61 }
62
63 for (int k = 0; k < jumlah_output; k++) {
64     w0[k] = w0[k] + delta_w0[k];
65 }
66
67 for (int i = 0; i < jumlah_input; i++) {
68     for (int j = 0; j < jumlah_hidden; j++) {
69         v[i][j] = v[i][j] + delta_v[i][j];
70     }
71 }
72
73 for (int j = 0; j < jumlah_hidden; j++) {
74     v0[j] = v0[j] + delta_v0[j];
75 }
76 }
77 mse = mse / jumlah_data;
78
79 Modeltbl_mse.addRow(new Object[]{(s + 1), mse});
80
81 Modeltbl_w_baru.setRowCount(0);
82 Modeltbl_w_baru2.setRowCount(0);
83 Modeltbl_v_baru.setRowCount(0);
84 Modeltbl_v_baru2.setRowCount(0);
85
86 for (int i = 0; i < jumlah_output; i++) {
87     Modeltbl_w_baru.addRow(new Object[]{w[0][i],
88     w[1][i], w[2][i], w[3][i], w0[i]});
89     Modeltbl_w_baru2.addRow(new Object[]{w[0][i],
90     w[1][i], w[2][i], w[3][i], w0[i]});
91 }
92 for (int i = 0; i < jumlah_input; i++) {
93     Modeltbl_v_baru.addRow(new Object[]{v[i][0],
94     v[i][1], v[i][2], v[i][3], v0[i]});
95     Modeltbl_v_baru2.addRow(new Object[]{v[i][0],
96     v[i][1], v[i][2], v[i][3], v0[i]});
97 }
98 }
99 }

```

Kode Program 5.2 Proses Pelatihan

Penjelasan Kode Program 5.2 :

1. Baris 9 sampai 10 mendapatkan nilai banyak iterasi dan nilai *alpha* dari *textbox*.
2. Baris 14 sampai 23 merupakan perulangan untuk mendapatkan nilai *z_{net}* dan *Z_j*.

3. Baris 25 sampai 35 merupakan perulangan untuk mendapatkan nilai y_{net} dan y_k dan menghitung nilai MSE.
4. Baris 37 sampai 43 merupakan perulangan untuk mendapatkan nilai δ_k sekaligus untuk mendapatkan nilai Δw dan Δw_0 .
5. Baris 45 sampai 55 perulangan untuk mendapatkan nilai δ_{net_j} dan nilai δ_j . Setelah itu menghitung nilai Δv dan Δv_0 .
6. Baris 57 sampai 76 merupakan perhitungan perubahan bobot w dan w_0 , bobot v dan v_0 .
7. Baris 57 sampai 76 merupakan perhitungan untuk mencari nilai MSE dari tiap iterasi.

5.1.3 Proses Pengujian

Proses pengujian merupakan proses untuk mengetahui tingkat kebenaran dari proses pelatihan berdasarkan data uji. Proses pengujian dapat dilihat pada Kode Program 5.3.

```

1 private void btn_prosesActionPerformed
  (java.awt.event.ActionEvent evt) {
2     DefaultTableModel Modeltbl_y = (DefaultTableModel)
  tbl_y.getModel();
3     DefaultTableModel Modeltbl_y_denorm =
  (DefaultTableModel) tbl_y_denorm.getModel();
4     DefaultTableModel Modeltbl_hasil = (DefaultTableModel)
  tbl_hasil.getModel();
5     SemenGUI main = new SemenGUI();
6
7     double mse = 0;
8     for (int h = 0; h < jumlah_uji; h++) {
9         for (int j = 0; j < jumlah_hidden; j++) {
10            double jml = 0;
11            for (int i = 0; i < jumlah_input; i++) {
12                double hitung = x_uji[h][i] * v[i][j];
13                jml = jml + hitung;
14            }
15            z_net_uji[j] = v0[j] + jml;
16            z_uji[j] = 1 / (1 + (double) Math.exp(-
  z_net_uji[j]));
17        }
18
19        for (int k = 0; k < jumlah_output; k++) {
20            double jml = 0;
21            for (int j = 0; j < jumlah_hidden; j++) {
22                double hitung = z_uji[j] * w[j][k];
23                jml = jml + hitung;
24            }
25            y_net_uji[k] = w0[k] + jml;
26            y_prediksi[h][k] = 1 / (1 + (double) Math.exp(-
  y_net_uji[k]));
27
28            mse = mse + Math.pow((x[h][6] - y[k]), 2);
29        }
  }

```

```

30     }
31
32     double mse_akhir = mse / jumlah_uji;
33     label_mse.setText(String.valueOf(mse_akhir));
34
35     Modeltbl_y.setRowCount(0);
36     for (int i = 0; i < jumlah_uji; i++) {
37         Modeltbl_y.addRow(new Object[]{(i + 1),
38             y_prediksi[i][0]});
39     }
40
41     main.denormalisasi(x_uji_asli, y_prediksi,
42         y_prediksi_denorm);
43
44     Modeltbl_y_denorm.setRowCount(0);
45     for (int i = 0; i < jumlah_uji; i++) {
46         Modeltbl_y_denorm.addRow(new Object[]{(i + 1),
47             y_prediksi_denorm[i][0]});
48     }
49
50     Modeltbl_hasil.setRowCount(0);
51     for (int i = 0; i < jumlah_uji; i++) {
52         for (int j = 0; j < jumlah_output; j++) {
53             Modeltbl_hasil.addRow(new Object[]{(i + 1),
54                 x_uji_asli[i][6], y_prediksi_denorm[i][0]});
55         }
56     }
57
58     double mad = 0;
59     for (int i = 0; i < jumlah_uji; i++) {
60         mad += Math.abs(x_uji_asli[i][6] -
61             y_prediksi_denorm[i][0]);
62     }
63
64     double aveg = mad / jumlah_uji;
65     label_mad.setText(String.valueOf(aveg));
66 }

```

Kode Program 5.3 Proses Pengujian

Penjelasan Kode Program 5.3 :

1. Baris 9 sampai 17 adalah perulangan untuk mendapatkan nilai z_{net} dan z_j .
2. Baris 19 sampai 26 merupakan perulangan untuk mendapatkan nilai y_{net} dan y_k .
3. Baris 32 menghitung nilai MSE dari keseluruhan data.
4. Baris 53 sampai 57 menghitung nilai kesalahan prediksi berdasarkan data asli dengan data prediksi. Menghitung nilai *Mean Absolute Deviation* (MAD).

5.2 Implementasi *Interface*

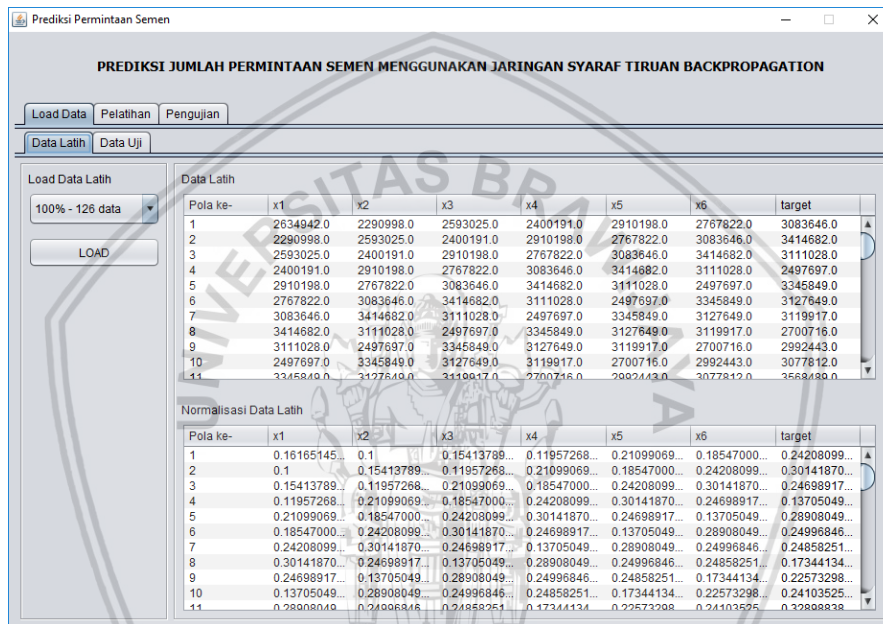
Pada subbab ini menjelaskan mengenai implementasi halaman antarmuka dari sistem untuk prediksi jumlah permintaan semen di Indonesia menggunakan algoritme *backpropagation*.

1. Implementasi Halaman *Load Data*

Pada halaman ini menampilkan tabel untuk melakukan *load* data latih dan data uji yang nantinya digunakan pada proses pelatihan dan pengujian. Langkah prosesnya adalah sebagai berikut:

1. Masuk ke tab Data Latih. Pilih banyak data yang akan diambil kemudian tekan *button LOAD* maka akan muncul data latih sekaligus data latih yang telah dinormalisasi.
2. Masuk ke tab Data Uji. Tekan tombol *LOAD* maka akan muncul data uji sekaligus data uji yang telah dinormalisasi.

Halaman *Load Data* diperlihatkan pada Gambar 5.1.



Gambar 5.1 Implementasi Halaman *Load Data*

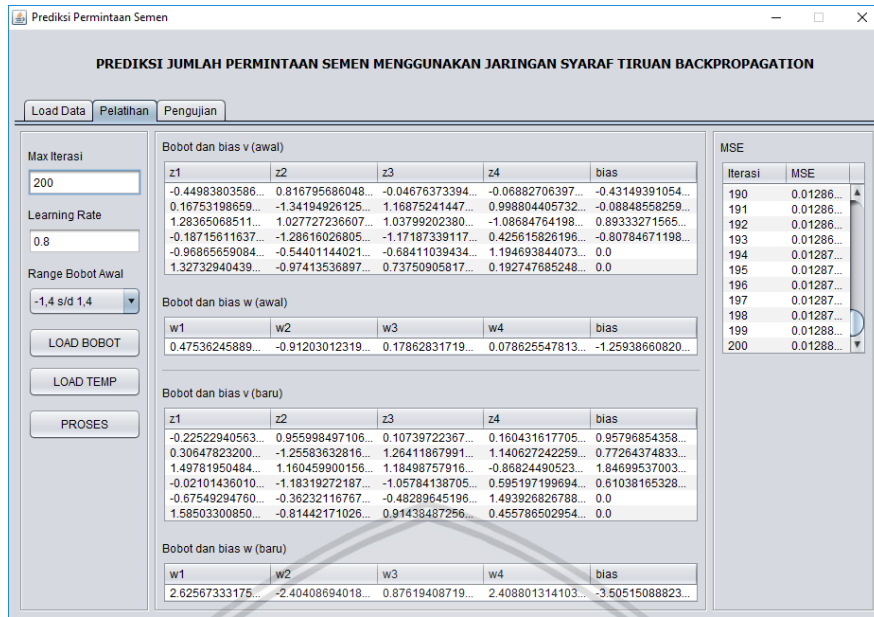
2. Implementasi Halaman Pelatihan

Halaman pelatihan merupakan halaman dimana proses pelatihan dilakukan. Proses pelatihan akan menghasilkan bobot terakhir sesuai dengan banyaknya iterasi yang telah ditentukan. Langkah prosesnya adalah sebagai berikut:

1. Masukkan banyaknya iterasi yang diinginkan.
2. Masukkan nilai *learning rate*.
3. Tentukan interval bobot yang diinginkan.
4. Klik *LOAD* bobot untuk menghasilkan bobot awal pelatihan.
5. Klik *PROSES* untuk melakukan proses pelatihan.

Halaman Pelatihan diperlihatkan pada Gambar 5.2.

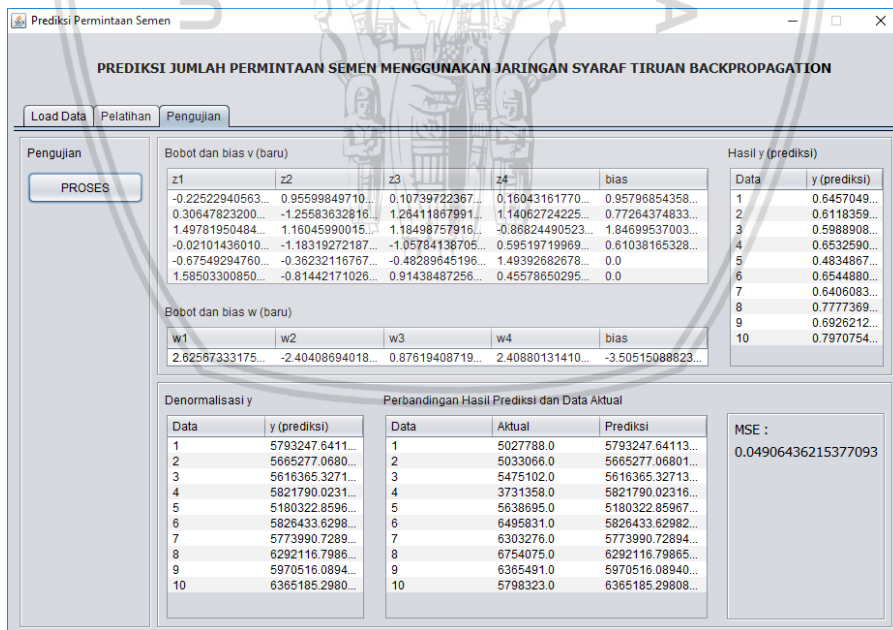




Gambar 5.2 Implementasi Halaman Pelatihan

3. Implementasi Halaman Pengujian

Halaman pengujian berisi tentang proses pengujian dimana hasil akhir dari proses ini adalah menampilkan perbedaan data prediksi dan data asli beserta MSE yang dihasilkan. Halaman Pengujian diperlihatkan pada Gambar 5.3.



Gambar 5.3 Implementasi Halaman Pengujian



BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai hasil pengujian dan analisis dari prediksi jumlah permintaan semen di Indonesia menggunakan Jaringan Syaraf Tiruan *Backpropagation*.

6.1 Pengujian

Dalam pengujian ini menggunakan data dari laporan penjualan semen PT. Semen Indonesia pada tahun 2007 sampai 2017. Data tersebut kemudian dibuat pola sehingga menghasilkan 126 data latih. Pengujian ini dilakukan sebanyak 4 model pengujian seperti yang telah dibahas pada bab perancangan, yaitu:

1. Pengujian konvergensi.
2. Pengujian *learning rate*.
3. Pengujian interval bobot awal.
4. Pengujian hasil prediksi.

6.2 Hasil Pengujian

6.2.1 Pengujian Konvergensi

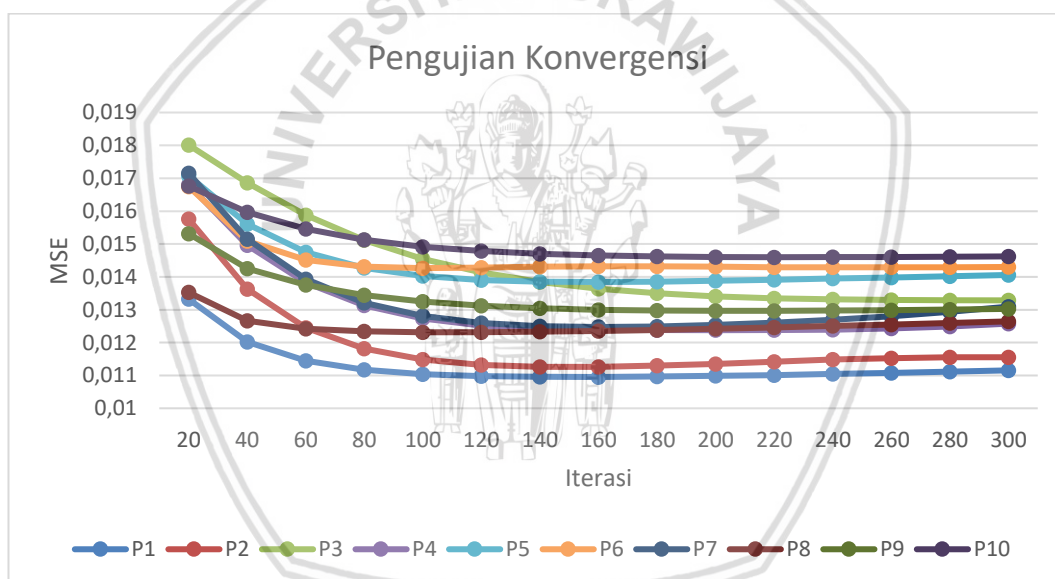
Dalam pengujian ini akan mencari berapa banyak iterasi yang dibutuhkan hingga menghasilkan MSE terkecil. Pengujian ini menggunakan 126 data latih dengan maksimum terasi sebesar 300, *learning rate* sebesar 0.9 dan bobot awal dengan interval -1 sampai 1. Pengujian akan dilakukan sebanyak 10 kali percobaan kemudian dilihat pada iterasi keberapa MSE mencapai konvergen, yaitu situasi dimana nilai MSE sedikit atau tidak menjadi lebih baik secara signifikan jika dilanjutkan ke iterasi selanjutnya. Hasil pengujian konvergensi ditunjukkan pada Tabel 6.1 dan Gambar 6.1.

Tabel 6.1 Pengujian Konvergensi

Iterasi	MSE percobaan ke- <i>i</i>					
	1	2	3	...	9	10
20	0,013323	0,01576	0,018012	...	0,015314	0,016764
40	0,012018	0,013631	0,016864	...	0,014253	0,015969
60	0,011444	0,012453	0,015879	...	0,013755	0,015457
80	0,011173	0,011817	0,015115	...	0,013448	0,015127
100	0,011042	0,011485	0,014547	...	0,013248	0,014917
120	0,01098	0,011323	0,014134	...	0,013122	0,014784
140	0,010955	0,011262	0,013841	...	0,013045	0,014701

Tabel 6.1 Pengujian Konvergensi (lanjutan)

Iterasi	MSE percobaan ke- <i>i</i>					
	1	2	3	...	9	10
160	0,010953	0,011261	0,013637	...	0,013	0,01465
180	0,010965	0,011297	0,013498	...	0,012977	0,01462
200	0,010986	0,011354	0,013407	...	0,012967	0,014604
220	0,011013	0,011419	0,013348	...	0,012967	0,014598
240	0,011044	0,011481	0,013312	...	0,012975	0,014598
260	0,011079	0,011529	0,013293	...	0,012988	0,014603
280	0,011115	0,011554	0,013284	...	0,013004	0,014611
300	0,011154	0,011552	0,013284	...	0,013024	0,014622



Gambar 6.1 Grafik Pengujian Konvergensi

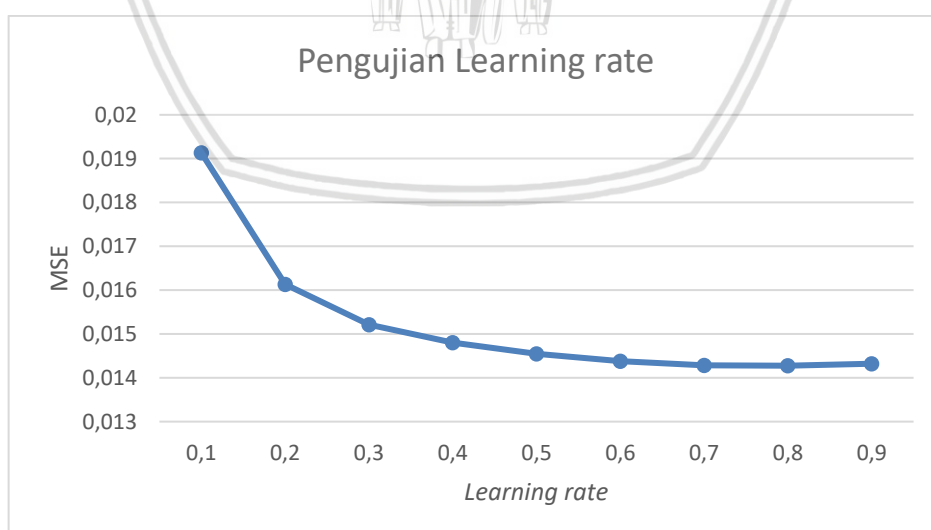
Berdasarkan hasil pengujian pada Tabel 6.1 dan Gambar 6.1, nilai MSE dipengaruhi oleh banyaknya iterasi yang dilakukan. Dalam pengujian ini terdapat berbagai macam grafik perubahan MSE pada tiap-tiap percobaan. Percobaan 1 memperlihatkan penurunan MSE yang stabil dan mencapai konvergen pada iterasi ke 120. Percobaan 2 mencapai MSE terbaik pada iterasi ke 160 namun mengalami kenaikan pada iterasi selanjutnya lalu mencapai konvergen pada iterasi ke 280. Gambar 6.1 memperlihatkan bahwa nilai MSE mungkin akan naik pada iterasi tertentu meskipun kenaikan tersebut tidak terlalu signifikan. Hal ini disebabkan oleh nilai random dari bobot awal perhitungan *backpropagation* sehingga menghasilkan pola yang berbeda – beda pada tiap percobaan. Berdasarkan data yang dihasilkan dapat disimpulkan bahwa iterasi terbaik berada pada iterasi ke 200 dengan mempertimbangkan bentuk pola dari semua percobaan.

6.2.2 Pengujian *Learning rate*

Dalam pengujian ini akan mencari nilai MSE terbaik dari nilai *learning rate* yang digunakan pada proses *backpropagation*. Pengujian ini menggunakan parameter awal yaitu jumlah data sebanyak 126 data dan jumlah iterasi dari pengujian sebelumnya yaitu sebanyak 200 iterasi dan bobot awal dengan *range* antara -1 sampai 1. Pengujian dilakukan sebanyak 10 kali dengan bobot awal yang berubah - ubah pada tiap percobaan. Hasil pengujian *learning rate* disajikan pada Tabel 6.2 dan Gambar 6.2.

Tabel 6.2 Pengujian *Learning rate*

<i>Learning rate</i>	MSE Percobaan ke- <i>i</i>					Rata rata
	1	2	3	...	10	
0,1	0,028622	0,020939	0,01346	...	0,016482	0,019132
0,2	0,023338	0,019293	0,011788	...	0,01508	0,01613
0,3	0,020128	0,018902	0,011691	...	0,014809	0,015216
0,4	0,018132	0,01851	0,011798	...	0,014619	0,014806
0,5	0,016833	0,017848	0,011967	...	0,014381	0,014551
0,6	0,015938	0,0171	0,012178	...	0,014116	0,014379
0,7	0,015339	0,016488	0,012433	...	0,013909	0,014285
0,8	0,015004	0,01605	0,012732	...	0,013812	0,014276
0,9	0,014851	0,015744	0,013063	...	0,013823	0,01432



Gambar 6.2 Grafik Pengujian *Learning rate*



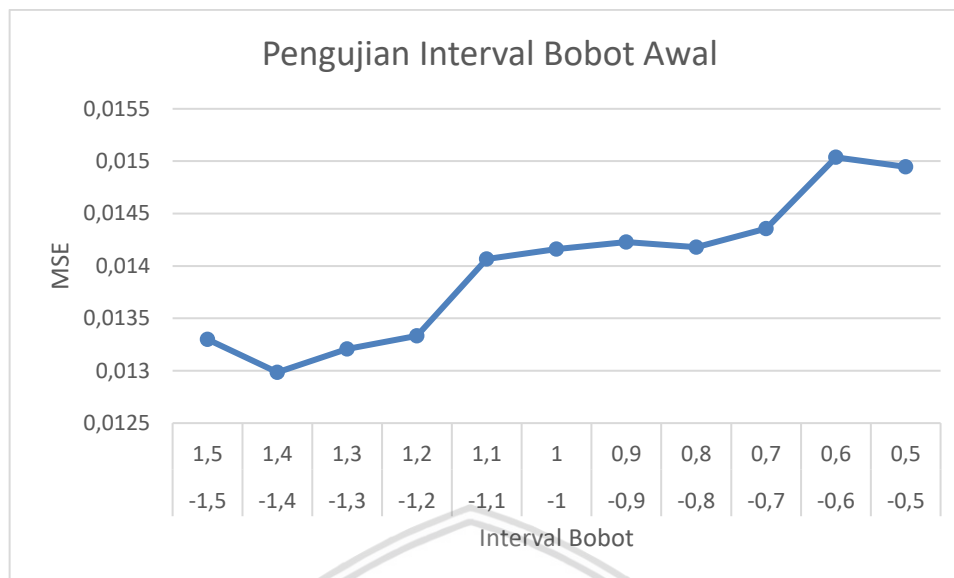
Berdasarkan hasil pengujian pada Tabel 6.2 dan Gambar 6.2, nilai MSE dipengaruhi oleh besarnya *learning rate*. Pada *learning rate* yang kecil proses pembelajaran berjalan lambat sehingga menghasilkan MSE yang besar. Saat *learning rate* ditambah maka proses pembelajaran akan berjalan lebih cepat sehingga menghasilkan MSE yang lebih baik. Terbukti pada Gambar 6.2 bahwa nilai MSE terus turun seiring bertambahnya nilai *learning rate*. Pada *learning rate* sebesar 0,1 menghasilkan MSE sebesar 0,019132 dan pada *learning rate* 0,8 menghasilkan MSE terbaik yaitu sebesar 0,014276. Dapat disimpulkan bahwa pada pengujian ini semakin besar *learning rate* yang digunakan maka MSE yang dihasilkan akan semakin baik.

6.2.3 Pengujian Interval Bobot Awal

Dalam pengujian ini akan mencari nilai MSE terbaik dari besarnya interval bobot awal yang digunakan pada proses pelatihan. Pengujian ini dilakukan dengan menggunakan parameter yang dihasilkan dari pengujian sebelumnya yaitu nilai *learning rate* sebesar 0,8 dan iterasi sebanyak 200. Pengujian dilakukan sebanyak 10 kali percobaan dengan bobot awal yang berbeda - beda untuk setiap interval bobot awal yang diuji. Hasil pengujian interval bobot awal ditunjukkan pada Tabel 6.3 dan Gambar 6.3.

Tabel 6.3 Pengujian Interval Bobot Awal

Interval Bobot	MSE Percobaan ke- <i>i</i>					Rata rata
	1	2	3	...	10	
-1,5 s/d 1,5	0,012528	0,013106	0,012021	...	0,013065	0,013301
-1,4 s/d 1,4	0,015376	0,012799	0,012006	...	0,011562	0,012985
-1,3 s/d 1,3	0,013948	0,011663	0,012417	...	0,012939	0,013209
-1,2 s/d 1,2	0,015611	0,012276	0,013697	...	0,012682	0,013332
-1,1 s/d 1,1	0,014659	0,014062	0,012226	...	0,014457	0,014065
-1 s/d 1	0,012677	0,012903	0,015186	...	0,013145	0,01416
-0,9 s/d 0,9	0,013334	0,016041	0,015132	...	0,01269	0,014229
-0,8 s/d 0,8	0,013804	0,014275	0,01434	...	0,014226	0,014178
-0,7 s/d 0,7	0,014841	0,015969	0,012867	...	0,01408	0,014357
-0,6 s/d 0,6	0,014773	0,016774	0,014384	...	0,015111	0,015037
-0,5 s/d 0,5	0,015663	0,015018	0,014275	...	0,013572	0,014946



Gambar 6.3 Grafik Pengujian Interval Bobot Awal

Berdasarkan hasil pengujian pada Tabel 6.3 dan Gambar 6.3, nilai MSE dipengaruhi oleh interval bobot awal yang digunakan pada proses pelatihan. Pada interval bobot yang sempit, nilai MSE yang dihasilkan dalam pengujian ini relatif tinggi dan terus menurun jika interval bobot awal yang digunakan lebih lebar. Telihat pada interval -1,4 sampai dengan 1,4 menghasilkan rata-rata MSE sebesar 0,012985 dan pada interval -0.6 sampai dengan 0.6 menghasilkan rata-rata MSE sebesar 0,015037. Dapat disimpulkan bahwa dalam pengujian ini nilai interval terbaik berada pada interval -1,4 sampai dengan 1,4. Hal ini disebabkan karena tingkat keacakan bobot awal yang lebar memungkinkan untuk mendapatkan bobot akhir yang lebih cepat.

6.2.4 Pengujian Hasil Prediksi

Pengujian hasil prediksi dilakukan setelah mendapatkan nilai bobot baru berdasarkan parameter terbaik yang didapatkan pada pengujian sebelumnya. Data uji didapatkan dari 10 data terakhir dari semua pola yang terbentuk pada data permintaan semen di Indonesia. Data aktual dan data prediksi akan dibandingkan sehingga dapat diketahui *error* yang dihasilkan. Berdasarkan selisih permintaan semen juga dapat diketahui besarnya *Mean Absolute Deviation* (MAD). Hasil pengujian hasil prediksi ditunjukkan pada Tabel 6.3 dan Gambar 6.3.

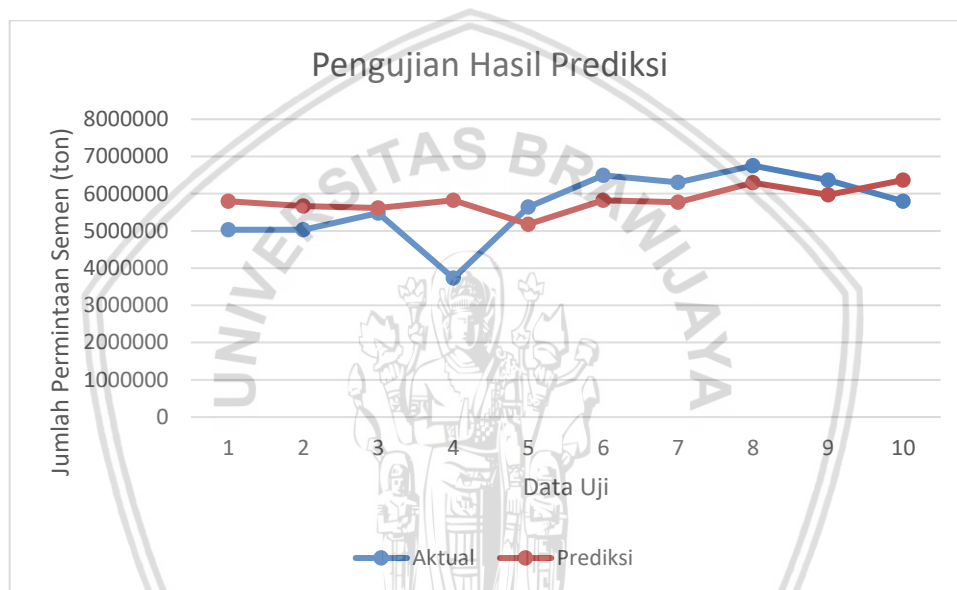
Tabel 6.4 Pengujian Hasil Prediksi

Data Uji	Aktual	Prediksi	Error	Selisih	Absolute
1	5027788	5793247	0,041042	-765459	765459
2	5033066	5665277	0,027997	-632211	632211
3	5475102	5616365	0,001398	-141263	141263
4	3731358	5821790	0,306096	-2090432	2090432



Tabel 6.4 Pengujian Hasil Prediksi (lanjutan)

5	5638695	5180322	0,014717	458373	458373
6	6495831	5826433	0,031387	669398	669398
7	6303276	5773990	0,019623	529286	529286
8	6754075	6292116	0,014948	461959	461959
9	6365491	5970516	0,010928	394975	394975
10	5798323	6365185	0,022508	-566862	566862
MSE			0,049064	MAD	671021,8



Gambar 6.4 Grafik Pengujian Hasil Prediksi

Berdasarkan hasil pengujian pada Tabel 6.4 dan Gambar 6.4, hasil prediksi yang dihasilkan cukup baik meskipun terdapat 1 data yang jauh melenceng dari data asli. Terbukti pada data uji ke-4 memiliki tingkat *error* sebesar 0,306096 dengan selisih data sebanyak 2090432, dimana perbedaan tersebut cukup besar. Namun rata-rata *error* dari semua data cukup baik dengan menghasilkan MSE sebesar 0,049064 dengan *Mean Absolute Deviation* (MAD) sebesar 671021,8.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan tentang Prediksi Jumlah Permintaan Semen di Indonesia menggunakan Jaringan Syaraf Tiruan *Backpropagation*, maka dapat disimpulkan bahwa:

1. Algoritme Jaringan Syaraf Tiruan *Backpropagation* dapat digunakan untuk melakukan prediksi jumlah permintaan semen di Indonesia dengan baik. Untuk mengimplementasikan algoritme *backpropagation*, langkah pertama yang dilakukan adalah mengumpulkan data permintaan semen di Indonesia yang berasal dari laporan penjualan bulanan PT. Semen Indonesia. Data tersebut kemudian dibuat pola sehingga menghasilkan data latih. Langkah selanjutnya adalah menginisialisasi bobot awal sesuai dengan interval bobot yang telah ditentukan, kemudian masuk pada proses *feedforward* untuk mendapatkan nilai y . Setelah itu masuk proses *backpropagation* dengan laju pelatihan yang ditentukan untuk *update* bobot berdasarkan tingkat *error* yang dihasilkan. Proses akan diulang sebanyak iterasi yang telah ditentukan. Bobot akhir dari proses pelatihan kemudian dapat digunakan untuk melakukan prediksi.
2. Untuk mendapatkan nilai MSE terbaik, pada penelitian ini menggunakan parameter-parameter yang telah didapatkan pada proses pelatihan, antara lain menggunakan data latih sebanyak 126 data, iterasi sebanyak 200, *learning rate* sebesar 0,8 dan inisialisasi bobot awal dengan interval antara -1,4 sampai dengan 1,4. Berdasarkan parameter-parameter tersebut, diperoleh nilai MSE prediksi sebesar 0,049064.

7.2 Saran

Beberapa saran tambahan yang dapat dilakukan untuk penelitian dengan menggunakan topik yang seragam yaitu dapat menambahkan beberapa parameter pada data latih seperti tingkat pembangunan daerah dan parameter yang dapat mempengaruhi harga semen. Selanjutnya dapat dilakukan optimasi untuk mengoptimalkan hasil prediksi seperti menggunakan algoritme genetika atau algoritme optimasi yang lain.



DAFTAR PUSTAKA

- Adwandha, D. P., Ratnawati, D. E. & Adikara, P. P., 2017. Prediksi Jumlah Pengangguran Terbuka di Indonesia menggunakan Metode Genetic-Based Backpropagation. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 1, pp. 341-351.
- Bustomi, M. A., Bisri, H. & Purwanti, E., 2014. Desain Perangkat Lunak Berbasis Jaringan Syaraf Tiruan Backpropagation untuk Klasifikasi Citra Rontgen Paru-paru. *Jurnal Fisika dan Aplikasinya*, pp. 19-23.
- Cynthia, E. P., 2017. Jaringan Syaraf Tiruan Algoritma Backpropagation Dalam memprediksi Ketersediaan Komoditi Pangan Provinsi Riau. *Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIK) 9*.
- Diputra, M. I., Dewi, C. & Wihandika, R. C., 2018. Prediksi Tingkat Keuntungan Usaha Peternakan Itik Alabio Petelur menggunakan Jaringan Syaraf Tiruan Backpropagation (Kasus di Kabupaten Hulu Sungai Utara Kalimantan Selatan). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 785-792.
- Fausett, L. V., 1994. *Fundamentals of Neural Networks : Architectures, Algorithms, and Applications*. New Jersey: Prentice Hall.
- Haviluddin & Alfred, R., 2015. A Genetic-Based Backpropagation Neural Network for Forecasting in Time-Series Data. *International Conference on Science in Information Technology (ICSITech)*, pp. 158-163.
- Hudiyawan, A. R., 2015. *Prediksi FOREX (Foreigen Exchange) Menggunakan Jaringan Syaraf Tiruan Al-Alaoi Backpropagation*. Malang: Teknik Informatika Universitas Brawijaya Malang.
- Kusrini & Luthfi, E. T., 2009. *Algoritma Data Mining*. Yogyakarta: ANDI.
- Mahfud, M. & Zakir, S., 2018. *Industri Kimia Indonesia*. 1 penyunt. Sleman: CV Budi Utama.
- Rachman, A. S., Cholissodin, I. & Fauzi, M. A., 2018. Peramalan Produksi Gula Menggunakan Metode Jaringan Syaraf Tiruan Backpropagation Pada PG Candi Baru Sidoarjo. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 1683-1689.
- Ridhani, M. N., Putri, R. R. M. & Sutopo, 2017. Peramalan Dosis Pupuk Berdasarkan Karakteristik dan Lingkungan Tanaman Jeruk Siam Menggunakan Metode Backpropagation. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 1, pp. 1214-1223.
- Sakinah, N. P., Cholissodin, I. & Widodo, A. W., 2018. Prediksi Jumlah Permintaan Koran Menggunakan Metode Jaringan Syaraf Tiruan Backpropagation. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 2612-2618.

- Ukhra, A. U., 2014. Pemodelan dan Peramalan Data Deret Waktu dengan Metode Seasonal ARIMA. *Jurnal Matematika UNAND*, pp. 59-67.
- Watimena, R. A., 2008. *Filsafat Dan Sains (Sebuah Pengantar)*. Jakarta: Grasindo.
- Yamanashi, Y., Umeda, K. & Yoshikawa, N., 2013. Pseudo Sigmoid Function Generator for a Superconductive Neural Network. *IEEE Transactions on Applied Superconductivity*.
- Zulfikar & Budiantara, I. N., 2014. *Manajemen Dengan Pendekatan Komputasi Statistika*. Yogyakarta: Deepublish.

