

**OPTIMASI PERAMALAN METODE *BACKPROPAGATION*
MENGUNAKAN ALGORITME GENETIKA PADA JUMLAH
PENUMPANG KERETA API DI INDONESIA**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
M. Birky Auliya Akbar
NIM: 165150209111008



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

PENGESAHAN

OPTIMASI PERAMALAN METODE BACKPROPAGATION MENGGUNAKAN ALGORITME GENETIKA PADA JUMLAH PENUMPANG KERETA API DI INDONESIA

SKRIPSI

Untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh:

M.Birky Auliya Akbar
NIM: 165150209111008

Skripsi ini telah diuji dan dinyatakan lulus pada
27 Desember 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Indriati, S.T., M.Kom.
NIP: 19831013 201504 2 002



Dr. Eng. Ahmad Afif Supianto, S.Si, M.Kom.
NIK: 201201 820623 1 001



Mengetahui
Kasubidjurusan Teknik Informatika

Tri Astoto Karniawan, 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 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 disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam 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).



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 **“OPTIMASI PERAMALAN METODE BACKPROPAGATION MENGGUNAKAN ALGORITME GENETIKA PADA JUMLAH PENUMPANG KERETA API DI INDONESIA”**. 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 terselesainya skripsi ini, penulis mengucapkan rasa terima kasih kepada:

1. Indriati, S.T, M.Kom selaku dosen pembimbing pertama skripsi yang telah meluangkan waktu dan juga memberikan banyak pengarahan bagi penulis.
2. Dr. Eng. Ahmad Afif Supianto, S.Si., 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, 17 Desember 2018

Penulis

Birkyakbar11@gmail.com

ABSTRAK

Muhammd Birky Auliya Akbar, 2018. Optimasi Peramalan Metode Backpropagation Menggunakan Algoritme Genetika Pada Jumlah Penumpang Kereta Api Di Indonesia. Fakultas Ilmu Komputer, Universitas Brawijaya, Malang.

Dosen Pembimbing : Indriati, S.T, M.Kom. dan Dr. Eng. Ahmad Afif Supianto, S.Si., M.Kom.

Kereta api merupakan salah satu moda transportasi umum yang memiliki segmentasi peminat yang sangat tinggi, hal ini terjadi karena tingkat layanan perkeretaapian Indonesia memiliki nilai yang baik, yakni mencapai indeks 4.09 dari 5, pada tahun 2014, selain itu juga didukung dengan fakta yang diungkap harian tempo (www.bisnis.tempo.co) yang menyatakan bahwa pengguna kereta api dari tahun ke tahun terus mengalami peningkatan. Namun dengan peningkatan jumlah penumpang kereta api tanpa adanya prediksi akan berakibat buruk bagi perkeretaapian di Indonesia. Untuk itu dibutuhkan suatu metode peramalan dengan hasil yang dapat dipertanggungjawabkan, dengan menggunakan metode yang populer seperti jaringan saraf tiruan *backpropagation* dan dilakukan optimasi dalam penentuan inisialisasi bobot (w) dengan menggunakan variabel jumlah populasi sebesar 800, jumlah generasi sebesar 20, kombinasi nilai $Cr = 0.7$ dan $Mr = 0.3$, dengan variabel utama dari jaringan saraf tiruan *backpropagation* yang terdiri dari jumlah iterasi sebesar 100 dan nilai $alpha$ sebesar 0.9, juga dengan menggunakan dataset berupa data bulanan dari bulan januari 2006 sampai dengan bulan juni 2017 berupa data *timeseries*, dengan data latih 100 pola data awal dan data uji 10 pola data akhir. Sehingga menghasilkan tingkat akurasi berdasarkan nilai *error* (MSE) sebesar 0.065869861 dari hasil hibridisasi metode jaringan sarat tiruan *backpropagation* dengan menggunakan algoritme genetika, sedangkan jika tanpa menggunakan hibridisasi nilai *error* yang didapat sebesar 0.072517977.

Kata kunci: *Kereta Api Indonesia, Peramalan. Jaringan Syaraf Tiruan (JST), Backpropagation, Algoritme Genetika.*

ABSTRACT

Muhammd Birky Auliya Akbar, 2018. Optimisation of the Forecasting Method Backpropagation Using a Genetic Algorithm for The Number of Passenger Trains in Indonesia. Faculty of Informatics, Brawijaya University, Malang.

Supervisory Professor: Indriati, S.T, M.Kom. and Dr. Eng. Ahmad Afif Supianto, S.Si., M.Kom.

Train is one way of public transport that have very much interest in this segmentation, this happens due to the level of services railways Indonesia has good value, ie the index reached 4.09 of 5, in 2014, in addition it also supports the fact that the revealed tempo daily (www.bisnis.tempo.co) states that the train user from year to year continues to increase. However, with an increase in the number of passengers on board the train without the prediction would be bad for the railroad in Indonesia. After that, it needs a method of predicting the results that can be answered, using methods that are popular as backpropagation artificial neural networks and optimization when setting the initial balance (w) with using a variable population is 800, the amount of generation is 20, the combination of the value $C_r = 0.7$ and $M_r = 0.3$, with the main variable backpropagation artificial neural network, which consists of nuber iterations is 100 and the number of alpha is 0.9, also with use of data sets in the form of monthly data from January 2006 until June 2017 in time series data with data form, with 100 training pattern of the initial data and 10 pattern of test data of last data. So the result is the level of precision based on error value (MSE) results 0.065869861 from the results of the hybridization method backpropagation artificially neural networks using a genetic algorithm, while without using the hybridization error value is 0.072517977.

Keywords: *Indonesian's Railway, Forecasting, Artificial Neural Networks (ANN), Backpropagation, Genetic Algorithm.*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xii
DAFTAR <i>SOURCE CODE</i>	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN.....	5
2.1 Kajian Pustaka.....	5
2.2 PT KAI.....	8
2.3 Algoritme genetika.....	8
2.3.1 Struktur Algoritme Genetika.....	8
2.3.2 Representasi Kromosom.....	9
2.3.3 Inialisai.....	9
2.3.4 Reproduksi.....	9
2.3.5 Evaluasi.....	11
2.3.6 Seleksi.....	12
2.4 Prediksi Data <i>Time series</i>	12
2.5 Jaringan Saraf Tiruan.....	13
2.6 Jaringan Saraf Tiruan <i>Backpropagationn</i>	13
2.6.1 Arsitektur Jaringan Saraf Tiruan <i>Backpropagationn</i>	13
2.6.2 Pelatihan Jaringan Saraf Tiruan <i>Backpropagationn</i>	14

2.6.3	Tingkat Akurasi.....	19
2.6.4	<i>Normalization</i> dan <i>Denormalization</i> Data	19
2.6.5	Jumlah <i>Hidden layer</i>	20
BAB 3 METODOLOGI PENELITIAN		21
3.1	Tipe Penelitian	21
3.2	Rancang Bangun Penelitian	21
3.2.1	Metode Penelitian.....	21
3.2.2	Subjek Penelitian.....	21
3.2.3	Lokasi Penelitian.....	21
3.2.4	Metode Pengumpulan Data	22
3.2.5	Metode Analisis Data	22
3.2.6	Peralatan Pendukung	23
BAB 4 PERANCANGAN DAN IMPLEMENTASI		24
4.1	Deskripsi Umum Sistem.....	24
4.2	Rancang Bangun Sistem	25
4.3	Manualisasi Sistem	31
4.4	Perancangan Antarmuka	43
4.4.1	Perancangan Antarmuka Halaman Menu Olah Data.....	43
4.4.2	Perancangan Antarmuka Halaman Menu Pelatihan.....	45
4.4.3	Perancangan Antarmuka Halaman Menu Pengujian.....	47
4.5	Perancangan Pola Pengujian	48
4.5.1	Uji Coba Populasi.....	49
4.5.2	Uji Coba Generasi	50
4.5.3	Uji Coba Kombinasi <i>Cr</i> dan <i>Mr</i>	50
4.5.4	Uji Coba Jumlah Iterasi.....	51
4.5.5	Uji Coba Nilai <i>Alpha</i>	51
4.5.6	Uji Coba Perbandingan Hasil Prediksi ANN-BP dan Algoritme Genetka 52	
4.6	Lingkungan Pengembangan Sistem.....	52
4.6.1	Lingkungan Perangkat Keras (<i>Hardware</i>).....	53
4.6.2	Lingkungan Perangkat Lunak (<i>Software</i>)	53
4.7	Implementasi Algoritme	53
4.7.1	Implementasi Pengambilan Data	53

4.7.2	Implementasi <i>Normalization</i> Data	55
4.7.3	Implementasi Proses <i>Feedforward</i>	55
4.7.4	Implementasi Proses <i>Backpropagation</i>	56
4.7.5	Implementasi proses <i>Update</i> Bobot.....	57
4.7.6	Implementasi Proses <i>Crossover</i>	58
4.7.7	Implementasi Proses Mutasi.....	59
4.7.8	Implementasi Proses Evaluasi	60
4.7.9	Implementasi Proses Seleksi	61
4.8	Implementasi Antarmuka Sistem	62
4.8.1	Implementasi Antarmuka Olah Data	62
4.8.2	Implementasi Antarmuka Pelatihan	63
4.8.3	Implementasi Antarmuka Pengujian.....	64
BAB 5 PENGUJIAN DAN ANALISIS.....		65
5.1	Pengujian Jumlah Populasi	65
5.2	Pengujian Banyak Generasi	66
5.3	Pengujian Kombinasi <i>Cr</i> dan <i>Mr</i>	68
5.4	Pengujian Jumlah Iterasi.....	69
5.5	Pengujian Nilai <i>Alpha</i>	70
5.6	Pengujian Perbandingan Hasil Prediksi ANN-BP dan Algoritme Genetka 72	
BAB 6 PENUTUP		74
6.1	Kesimpulan	74
6.2	Saran	74
DAFTAR PUSTAKA.....		xiv

DAFTAR TABEL

Tabel 2. 1 Kajian Pustaka.....	6
Tabel 2. 2 Kajian Pustaka (Lanjutan)	7
Tabel 2. 3 Proses Evaluasi	11
Tabel 2. 4 Proses Seleksi	12
Tabel 4. 1 Normalization Data Latih.....	32
Tabel 4. 2 Normalization Data Uji	32
Tabel 4. 3 Pola Populasi yang Dibentuk	33
Tabel 4. 4 Penerapan Extended Intermediate Crossover	34
Tabel 4. 5. Penerapan Random Mutation.....	34
Tabel 4. 6 Evaluasi Individu	35
Tabel 4. 7 Seleksi Kromosom	36
Tabel 4. 8 Nilai Hidden Layer.....	37
Tabel 4. 9 Nilai Aktivasi Hidden layer.....	37
Tabel 4. 10 Nilai Output layer	37
Tabel 4. 11 Nilai Output layer (Lanjutan).....	38
Tabel 4. 12 Nilai Aktivasi Output layer.....	38
Tabel 4. 13 Selisih Nilai Kesalahan pada Output layer.....	38
Tabel 4. 14 Nilai Faktor Kesalahan Pada Output Layer ke Hidden Layer.....	39
Tabel 4. 15 Nilai Kesalahan Hidden Layer	39
Tabel 4. 16 Selisih nilai Kesalahan pada Hidden layer	40
Tabel 4. 17 Nilai Faktor Kesalahan pada Hidden layer ke Input layer	40
Tabel 4. 18 Nilai Faktor Kesalahan pada Hidden layer ke Input layer (lanjutan) .	41
Tabel 4. 19 Update Bobot Pada Hidden layer Menuju Output layer	41
Tabel 4. 20 Update Bobot pada Input layer Menuju ke Hidden layer	42
Tabel 4. 21 Hasil Perhitungan Menggunakan ANN-BP.....	42
Tabel 4. 22 Hasil Perhitungan Menggunakan ANN-BP (Lanjutan).....	43
Tabel 4. 23 Rancang Uji Coba Jumlah Populasi.....	49
Tabel 4. 24 Rancang Uji Coba Generasi	50
Tabel 4.25 Rancang Tabel kombinasi Cr dan Mr.....	50
Tabel 4. 26 Rancang Uji Coba Jumlah Iterasi	51
Tabel 4. 27 Rancang Uji Coba Alpha.....	52

Tabel 4. 28 Perbandingan Nilai Error MSE Menggunakan dan Tanpa Optimasi .. 52

Tabel 4. 29 Rancang Uji Coba Perbandingan Menggunakan dan Tanpa Optimasi 52

Tabel 5. 1 Pengujian Jumlah Populasi 65

Tabel 5. 2 Pengujian Banyak Generasi 67

Tabel 5. 3 Pengujian Kombinasi Cr dan Mr 68

Tabel 5. 4 Pengujian Jumlah Iterasi..... 69

Tabel 5. 5 Pengujian Nilai Alpha..... 71

Tabel 5. 6 Perbandingan Nilai Error MSE Menggunakan dan Tanpa Optimasi 72

Tabel 5. 7 Pengujian Coba perbandingan Hasil ANN-BP dan Algoritme Genetika 72



DAFTAR GAMBAR

Gambar 2. 1 Proses <i>Crossover</i>	10
Gambar 2. 2 Proses Mutasi	11
Gambar 2. 3 Arsitektur Jaringan Saraf Tiruan <i>Backpropagation</i>	14
Gambar 4. 1 Arsitektur Jaringan Saraf Tiruan <i>Backpropagation</i>	24
<i>Gambar 4. 2 Diagram Alir Crossover</i>	25
<i>Gambar 4. 3 Diagram Alir Mutasi</i>	27
Gambar 4. 4 Diagram Alir Evaluasi.....	27
<i>Gambar 4. 5 Diagram Alir Seleksi</i>	28
Gambar 4. 6 Diagram Alir Proses <i>Feedforward</i>	29
Gambar 4. 7 Diagram Alir Proses <i>Backpropagation</i>	30
Gambar 4. 8 Diagram Alir Proses <i>Weight Update</i>	31
Gambar 4. 9 Perancangan Antarmuka Halaman Submenu Data latih.....	44
Gambar 4. 10 Perancangan Antarmuka Halaman Submenu Data Uji	45
Gambar 4. 11 Perancangan Antar Muka Halaman Menu Pelatihan.....	46
Gambar 4. 12 Perancangan Antarmuka Halaman Menu Pengujian	47
Gambar 4. 13 Implementasi Antarmuka Submenu Data latih.....	62
Gambar 4. 14 Implementasi Antarmuka Submenu Data Uji	63
Gambar 4. 15 Implementasi Antarmuka Pelatihan	63
Gambar 4. 16 Implementasi Antarmuka Pengujian.....	64
Gambar 5. 1 Grafik Hasil Pengujian Jumlah Populasi	66
Gambar 5. 2 Grafik Hasil Pengujian Jumlah Generasi.....	67
Gambar 5. 3 Grafik Hasil Pengujian Kombinasi Cr & Mr	68
Gambar 5. 4 Grafik Hasil Pengujian Jumlah Iterasi	70
Gambar 5. 5 Grafik Hasil Pegujian Nilai Learning Rate	71
Gambar 5. 6 Grafik Perbandingan Nilai Peramalan ANN-BP dengan Optimasi GA pada ANN-BP.....	73

DAFTAR SOURCE CODE

Source Code 4. 1 Implementasi Pengambilan Data.....	55
Source Code 4. 2 Implementasi Normalization Data.....	55
Source Code 4. 3 Implementasi Proses Feedforward.....	56
Source Code 4. 4 Implementasi Proses Backpropagation.....	57
Source Code 4. 5 Implementasi Proses Update Bobot.....	58
Source Code 4. 6 Implementasi Proses Crossover.....	59
Source Code 4. 7 Implementasi Proses Mutasi.....	60
Source Code 4. 8 Implementasi Proses Evaluasi.....	61
Source Code 4. 9 Implementasi Proses Seleksi.....	62



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Kereta api merupakan suatu moda transportasi masif darat di Indonesia yang cukup populer, ditambah lagi moda transportasi ini tersebar di wilayah-wilayah besar yang ada di Indonesia, sehingga moda transportasi kereta api ini digunakan sebagai moda transportasi masif darat yang dapat diandalkan. Dan dengan didukung data statistik yang menunjukkan kepopuleran moda transportasi darat ini, yakni berdasarkan indeks keselamatan dan pelayanan dari skala 5, moda transportasi kereta api ini mendapat nilai 4.09 pada tahun 2014, berdasarkan laman resmi PT KAI (Persero Kereta Api Indonesia) www.kai.id. Selain itu juga berdasarkan fakta yang dikemukakan berita harian tempo (www.bisnis.tempo.co) pada tahun 2017 lalu, yang menyatakan bahwa pengguna kereta api dari tahun-ketahun terjadi peningkatan, bahkan pada tahun 2016 sudah mencapai 352 juta dan akan bertambah lagi. Hal itu dapat berakibat buruk jika tidak disiasati sebelumnya dengan cara meramalkan informasi tersebut, sehingga dapat berakibat menumpuknya penumpang tanpa terprediksi jumlahnya, ketidakpercayaan pengguna akan pelayanan dari PT KAI, dan yang secara otomatis akan memberikan citra buruk sehingga solusi kemacetan dengan penggunaan kereta api sebagai media solusi akan menimbulkan permasalahan baru. Sehingga harus ada suatu solusi untuk menekan permasalahan tersebut agar kereta api Indonesia dapat menjelma menjadi suatu moda transportasi masal yang difavoritkan oleh masyarakat Indonesia dan dapat lebih memenuhi ekspektasi dan harapan penggunanya.

Dengan penjabaran beberapa permasalahan yang telah dikemukakan sebelumnya, solusi yang terbaik untuk menangani permasalahan tersebut adalah dengan melakukan peramalan informasi yang dapat dipertanggung jawabkan secara matematis, dengan menggunakan metode yang unggul dalam hal peramalan suatu informasi, adapun beberapa metode yang terbilang unggul dan populer dalam melakukan peramalan informasi, diantaranya yakni metode *naive bayes*, sebagai metode peramalan informasi, metode ini pernah digunakan dalam penelitian yang dilakukan oleh Saleh (2015), untuk meramalkan penggunaan listrik rumah tangga, dengan pembagian berdasarkan tingkat penggunaannya yang mendapatkan hasil yang cukup akurat. Selain itu dengan menggunakan metode *Fuzzy*, peramalan informasi dapat dilakukan, dengan permasalahan yang diangkat yakni untuk meramalkan bencana banjir yang memakai model mamdani berdasarkan data yang fluktuatif, namun dengan penggabungan antara metode dan model ini mendapatkan hasil peramalan informasi dengan predikat baik dari hasil pengkajian yang dilakukan oleh Gani, Kolibu dan Tamuntuan (2016).

Dengan pilihan ragam metode yang dapat dipakai, terdapat suatu metode yang juga cukup baik dalam melakukan peramalan suatu informasi yakni metode Jaringan saraf tiruan *Backpropagation* atau *artificial neural network*

backpropagation (ANN-BP), seperti halnya yang telah diteliti oleh Yohanes, Mahmudy dan Rahmi (2015) dengan menerapkan teknik ANN-BP untuk meramalkan data tentang korelasi besar inflasi dengan ukuran upah minimum yang diberikan pada suatu wilayah. Tetapi teknik ANN-BP ini masih terbilang teknik yang pembangkitan solusi awal dilakukan secara *random*, sehingga dibutuhkan solusi untuk menstabilkan data *random* tersebut, dengan menggunakan optimasi nilai pembangkit akan lebih memperbesar peluang untuk mendapatkan informasi yang lebih akurat. Terdapat banyak pilihan model optimasi yang dapat digunakan, diantaranya *Evolution Strategies* (ES), *Ant Colony Optimization* (ACO), *Particle Swarm Optimization* (PSO), dan sebagainya, namun ada cabang keilmuan dari algoritme evolusi yang cukup populer yakni algoritme genetika/*genetic algorithm* (GA), dengan kelebihan pilihan variasi dalam penerapan variabel yang sesuai berdasarkan permasalahan yang diangkat (Mahmudy, 2015). Dan dengan membandingkan penelitian yang telah dilakukan Yohanes, Mahmudy dan Rahmi yang memiliki kesesuaian metode peramalan dan pola data yang digunakan serta korelasinya terhadap objek penelitian yang dilakukan.

Sehingga berdasarkan latar belakang yang ada, secara khusus peneliti akan membahas dalam suatu karya keilmuan dengan judul penelitian, "**Optimasi Peramalan Metode *Backpropagation* Menggunakan Algoritme Genetika Pada Jumlah Penumpang Kereta Api Di Indonesia**".

1.2 Rumusan Masalah

Dengan pembahasan latar belakang yang telah dijabarkan, secara khusus masalah yang dapat diangkat untuk ditentukan pemecahan masalahannya ialah :

1. Bagaimana pengaruh variabel pada algoritme genetika dan jaringan saraf tiruan *backpropagation* untuk menghasilkan informasi peramalan penumpang KAI?
2. Berapa besar tingkat keakurasian berdasarkan nilai *error* dari metode jaringan saraf tiruan *Backpropagation* dan algoritme genetika untuk menghasilkan informasi peramalan penumpang KAI?

1.3 Tujuan

Tujuan yang mendasari penelitian peramalan pengguna kereta api Indonesia yakni :

1. Mendapatkan variabel yang unggul untuk menghasilkan informasi peramalan penumpang KAI berdasarkan metode ANN-BP dan algoritme genetika.
2. Mengetahui nilai keakurasian berdasarkan nilai *error* yang dihasilkan dari metode jaringan saraf tiruan *backpropagation* dan algoritme genetika.

1.4 Manfaat

Manfaat dari penelitian peramalan penumpang KAI ialah :

1. Informasi hasil peramalan dapat digunakan sebagai acuan untuk peningkatan pelayanan terkait jumlah pengguna KAI.
2. Mengetahui pola data dan bagaimana untuk mengelola data tersebut dengan menggunakan variabel-variabel pendukung dari metode jaringan saraf tiruan *backpropagation* dan algoritme genetika
3. Mengetahi perubahan atau optimasi yang dapat dilakukan berdasarkan penelitian yang telah dilakukan, sehingga mendapatkan hasil yang lebih baik dari penggunaan metode dan langkah yang telah peneliti lakukan.

1.5 Batasan Masalah

Merujuk pada latar belakan dan solusi yang ingin dipecahkan, fokus bahasan yang ingin dibahas dalam penelitian ini menitik bertkan pada bahasan tertentu, dengan memberikan batasan masalah yang digunakan, yakni :

1. Data yang diperoleh merupakan data sekunder dari laman resmi Badan Pusat Statistik (BPS), yang terdiri dari data pengguna moda transportasi kereta api di Indonesia.
2. Jangka waktu data yang digunakan berada pada *range* tertentu, yakni antara bulan Januari 2006 sampai dengan bula juni 2017, yang disajikan dalam dataset bulanan.
3. Sistem mengolah data peramalan, dari data *time series*, dan tanpa memperhatikan *constraints* selain pengolahan data *time series* itu sendiri.
4. Peramalan informasi menggunakan metode ANN-BP dan algoritme genetika.
5. Data yang dihibridisasi ialah inialisasi data pada bobot (w).
6. Sistem yang dikembangkan merupakan sistem berbasis desktop
7. Metode ANN-BP yang digunakan merupakan metode ANN-BP *basic*.

1.6 Sistematika Pembahasan

Pada sistematika pembahasan, menyajikan gambaran ringkas prihal bahasan dari rangkaian kajian yang dilakukan secara terstruktur dalam tiap bab-babnya, yang disajikan secara runtut, sebagai berikut:

BAB 1 PENDAHULUAN

Berisi tentang bagian yang menjabarkan tentang dasar penulisan, yang berdasarkan pengalaman permasalahan, *urgency*-nya, macam solusi yang relevan, dan solusi yang digunakan, selain itu penjabaran rumusan masalah, tujuan penelitian, manfaat penelitian, dan juga batasan permasalahan.

BAB 2 LANDASAN KEPUSTAKAAN

Analisis pustaka yang berisi penelitian-penelitian serupa yang pernah dipublikasikan, dibahas pada bab 2 ini. Tidak hanya itu dalam landasan pustaka juga membahas teori-teori terkait permasalahan yang diangkat, sehingga akan memperkuat hipotesis peneliti dalam menjalankan penelitian ini.

BAB 3 METODOLOGI PENELITIAN

Pada bab 3, metodologi penelitian berisi tentang gambaran secara teknis alur penelitian yang dilakukan secara sistematis

BAB 4 PERANCANGAN DAN IMPLEMENTASI

Disini, dalam bab 4 perancangan dan implementasi menjabarkan mulai dari deskripsi umum sistem, selanjutnya merepresentasikannya dalam sebuah diagram alir terkait perencanaan operasi utama sistem, kemudian dilakukan proses perancangan antarmuka dan bagaimana sistem dapat berkomunikasi dengan pengguna, dan tahapan selanjutnya yakni dengan menentukan macam pengujian yang akan dilakukan, sehingga nantinya dapat dengan mudah untuk mengimplementasikan, dan tidak kalah penting juga saat proses implementasi, pengkodean sistem juga dijabarkan pada bab ini, dan juga penjabaran halaman antarmuka sistem.

BAB 5 PENGUJIAN DAN ANALISIS

Pada bab Pengujian dan Analisis, dijelaskan terkait penerapan variabel pembangun solusi pada sistem beserta detail hasil dari perhitungan tersebut, yang selanjutnya akan dilakukan analisa terhadap hasil dan teori yang mendukung hal tersebut.

BAB 6 PENUTUP

Pada bab 6, merangkum hasil penelitian berdasarkan permasalahan yang dijabarkan pada rumusan masalah pada bab 1, dan disertai dengan saran yang relevan terhadap penelitian yang dapat dilakukan kedepannya.

BAB 2

LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada analisis kepastakaan membahas tentang pustaka atau literatur yang sesuai dengan kajian yang dibahas. Dengan ketentuan bahwa literatur secara teknis hanya menggunakan literatur dengan jangka waktu maksimal 10 tahun dari penelitian ini, sehingga penarikan hipotesis dapat dilakukan berdasarkan penelitian yang terbaru. Dengan poin-poin rincian disajikan dalam **Tabel 2.1**.

Penelitian yang meneliti terkait populasi penumpang KAI sebelumnya sudah pernah ada yakni dengan meneliti jumlah penumpang kereta api pada daerah operasi IV Semarang dengan menggunakan ANN-BP, dengan data dari bulan Januari 2009 sampai dengan bulan September 2010, yang disajikan dalam bentuk data harian tanpa memperhitungkan *constraints* atau batasan yang tidak pasti, penelitian yang dilakukan oleh Luthfianto, Santoso, dan Christiyono (2011) menghasilkan nilai prediksi yang baik, yakni dengan menghasilkan nilai ketepatan yang besar, akan tetapi jika penggunaan data pada skala tertentu. Dengan hasil yang hampir sama penelitian yang dibuat oleh Nugraha dan Azhari (2014), yang melakukan kajian tentang peramalan inflasi, yang memakai dataset mulai tahun 2006 sampai dengan 2012, yang didapat dari data sekunder di Badan Pusat Statistika (BPS), menggunakan komposisi data latih lebih besar 40% dari data uji yang sebesar 30%, yang memanfaatkan algoritme *Particle Swarm Optimization* (PSO) sebagai media peramalan tinggi inflasi, sehingga mendapati nilai *Mean Square Error* (MSE) dengan rerata nilai $46.8085 \cdot 10^{-4}$ berdasarkan bentuk penggunaan jaringan saraf tiruannya. Berbeda halnya dengan penggunaan metode ANN-BP yang dilakukan hibridisasi dengan algoritme *evolution strategies* yang mendapatkan hasil tingkat keakurasian mencapai 96.30% dengan topik permasalahan memprediksi jumlah ketertarikan konsumen akan suatu objek, dengan dataset yang serupa, dengan penggunaan database yang berada di Irvine, University of California yang berisi 20 atribut pada kurang lebih 5000 klien dengan komposisi pembagian 1667 data latih dan 3333 data latih (Zeniarta & Luthfiarta, 2015). Masih dengan topik peramalan Davinsyah, Mahmudy, & Perdana (2017) melakukan penelitian terkait peramalan harga saham pada hari berikutnya, menggunakan dataset yang diperoleh dari bank BCA. Data sekunder berupa harga saham tiap harinya tersebut memiliki jangka waktu mulai dari 1 Januari 2013 sampai dengan 30 Desember 2016, dengan variabel *volum*, *open*, *close*, *high*, dan *low* sebagai variabel kontrol dalam metode jaringan saraf tiruan *backpropagation* dengan menggunakan hibridisasi algoritme genetika sebagai media optimasinya, didapati nilai akar dari *Mean Square Error* (RMSE) yang cukup rendah, dengan hasil yang bernilai $65.205 \cdot 10^{-3}$ serta $48.006 \cdot 10^{-3}$ dengan penerapan teknik ANN-BP.

Tabel 2. 1 Kajian Pustaka

No	Penulis	Bahasan	Solusi	Hasil
1.	(Luthfianto, Santoso, & Christiyono, 2011)	Pengguna moda transportasi kereta api daerah operasi iV	<p>Teknik ANN-BP</p> <p>Alur:</p> <ol style="list-style-type: none"> Membangkitkan sinyal atau bobot awal secara acak Menghitung nilai <i>hidden layer</i> berdasarkan sinyal dari unit <i>input</i> Melnghitung nilai unit <i>output layer</i> Melakukan perhitungan ukuran kesalahan yang terjadi pada unit <i>output</i> Menghitung nilai faktor kesalahan yang menuju ke unit <i>hidden layer</i> Menghitung nilai kesalahan pada unint <i>hidden layer</i> Menghitung selisih kesalahan pada hidden layer Menghitung faktor kesalahan yang menuju ke unit <i>input layer</i> Melakukan perbaikan sinyal atau bobot 	<p>Faktor yang mempengaruhi nilai keakurasian dari penelitian ini, berdasarkan pada jumlah unit <i>hidden layer</i> yang digunakan serta nilai alpha, atau besar jangkauan sistem untuk menentukan pola. Dengan hasil pengujian dilakukan dengan <i>hidden layer</i> berjumlah 10 unit yang menghasilkan tingkat keakurasian sebesar 85.34%. kemudian tingkat akurasi menurun sebesar 75.60% jika menggunakan unit <i>hidden layer</i> berjumlah 5. Dan jika menggunakan nilai alpha sebesar 0.01 akurasi bernilai 84.44% dan jika <i>alpha</i> bernilai 0.001 nilai akurasi yang didapat menjadi 0.001.</p>
2.	(Nugraha & Azhari, 2014)	Prediksi laju inflasi	<p><i>Particle Swarm Optimization</i> (PSO).</p> <p>Alur:</p> <ol style="list-style-type: none"> Proses penentuan nilai pada masing-masing parameter Melakukan <i>normalization</i> data latih dengan nilai anatar nol sampai dengan satu. Proses inialisasi sinyal atau bobot awal jaringan saraf tiruan <i>backpropagation</i> dengan nilai hasil hibridisasi dengan teknik <i>particle swarm optimization</i> (PSO) Melakukan pelatihan dengan melakukan modifikasi nilai bobot hingga mencapai nilai konvergen Melakukan pengujian terhadap hasil pelatihan 	<p>Hibridisasi nilai bobot yang digunakan untuk proses inialisasi bobot pada proses ANN-BP dengan menggunakan model kerangka 3-4-1. Didapati nilai MSE pada proses pelatihan sebesar 15566.4E dan pada pengujian 30775-3E. Namun jika menggunakan model kerangka 12-6-1, didapati nilai <i>error</i> pada pelatihan sebesar 967-4E dan pada pelatihan sebesar 62842-3E.</p>

Tabel 2. 2 Kajian Pustaka (Lanjutan)

No	Penulis	Objek	Metode	Hasil
3.	(Zeniarta & Luthfiarta, 2015)	Peramalan segmentasi pelanggan berdasarkan minatnya pada suatu objek.	<p>ANN-BP berbasis <i>evolution strategies</i></p> <p>Alur:</p> <ol style="list-style-type: none"> Penilaian paprameter populasi, jumlah data <i>training</i> dan data uji Menentukan nilai ke-<i>fitness</i>-an dari data <i>training</i>. Menentukan nilai konvergensi Setelah memenuhi nilai konvergensi pada proses <i>evolution strategies</i>, dilakukan seleksi data. Lakukan perhitungan mutasi untuk menghasilkan solusi Jika dalam pemrosesan data pada metode ANN-BP yang dihibridisasi dengan optimasi <i>evolution strategies</i>, mencapai titik konvergen maka hentikan proses. Dengan ANN-BP uji data latih. Uji kedua data pelatihan dengan data uji 	<p>Dengan menggunakan optimasi <i>evolution strategies</i> menghasilkan nilai <i>error</i> yang lebih kecil 0.8% daripada tanpa menggunakan optimasi, namun lama komputasi bertambah 5 jam 20 menit 54 detik daripada tanpa menggunakan optimasi.</p>
4.	(Davinsyah, Mahmudy, & Perdana, 2017)	Peramalan nilai saham dihasri berikutnya	<p>Multi <i>layer</i> pada jaringan saraf tiruan dan algoritme genetika</p> <p>Proses:</p> <ol style="list-style-type: none"> Melakukan inisialisasi data representasi kromosom Menentukan nilai solusi dengan kombinasi perhitungan parameter <i>Cr</i> dan <i>Mr</i> Dilakukan proses evaluasi dengan memberi label <i>fitness</i> Menyeleksi data yang tidak termasuk <i>fitness</i> Menggunakan data seleksi dari algoritme genetika digunakan sebagai nilai awal bobot Melakukan proses <i>feedforward</i> Menjalankan peoses <i>backpropagation</i> Melakukan perbaikan sinyal atau bobot Menghitung nilai akurasi dengan menggunakan RMSE Melakukan pengujian perbedaan penggunaan metode dalam meramal harga saham. 	<p>Hasil dari penerapan ANN-BP menghasilkan nilai lebih baik daripada dengan dilakukan hibridisasi dengan algoritme genetikaB, dengan hasil RMSE berselisih 17199-2E</p>

2.2 PT KAI

PT KAI adalah persero dibawah naungan pemerintah yang menyediakan moda transportasi umum yang masif (Suharyadi & S.K, 2009). Dalam laman resmi yang dimiliki PT KAI, mengungkapkan sejarah bahwa awal mula dibuatnya kereta api berasal dari pulau Jawa pada tahun 1864, khususnya di Semarang.

2.3 Algoritme genetika

Mencari informasi dalam suatu populasi besar, merupakan konsep dasar algoritme genetika, seperti namanya algoritme genetika dalam proses kerjanya memiliki proses yang sama seperti proses evolusi makhluk hidup yang berkembang secara genetik dan proses seleksi alam, sehingga menghasilkan individu baru yang memiliki kondisi lebih baik yang dapat bertahan, dengan digambarkan dengan tingkat "*fitness*" makhluk hidup, sehingga individu yang memiliki tingkat *fitness* yang rendah akan terseleksi. Proses seleksi yang berkonsep seperti proses perkembangan makhluk hidup yakni dengan melakukan kawin silang, atau bisa disebut dengan *crossover* dan proses evolusi gen berdasarkan lingkungan atau dalam metode ini disebut dengan mutasi, sehingga tingkat variasi individu juga dapat terjaga dan memperluas atau memperbanyak kemungkinan untuk mendapatkan individu dengan tingkatan *fitness* yang lebih baik, (Arkeman, Seminar, & Gunawan, 2012). Solusi pencarian nilai *fitness* tersebut juga memiliki varian berdasarkan bentuk kasus yang bisa dipecahkan, dengan segala macam varian pencarian nilai optimal yang ditawarkan dalam penggunaan algoritme genetika membuat algoritme ini dapat dikatakan sebagai suatu algoritme yang fleksibel, sehingga dapat dipergunakan untuk melakukan optimasi atau dihibridisasi dengan metode lain, mengingat dalam penentuan baris solusi, algoritme genetika ini tidak berpaku pada informasi derivatif pada suatu metode, menurut Gen dan Cheng, dalam Mahmudy (2015). Dengan hanya menyesuaikan bentuk representasi kromosom sesuai dengan pola perhitungan yang digunakan dalam metode tertentu, algoritme dapat digunakan sebagai media hibridisasi pada metode lain. Selain itu keuntungan lain dalam penggunaan algoritme genetika sebagai media hibridisasi ini memungkinkan bekerja pada ruang pencarian yang luas serta kompleks, dan dapat menghasilkan kemungkinan lebih besar untuk mendapatkan optimum global.

2.3.1 Struktur Algoritme Genetika

Dalam penerapan algoritme genetika, setidaknya memiliki proses umum yang wajib dilakukan yakni dengan melakukan representasi kromosom dengan cara pengkodean (*encoding*) nilai input sebagai inisialisasi data, selanjutnya proses pencarian solusi dengan menghasilkan generasi baru dengan proses *crossover* dan mutasi, kemudian dilanjutkan dengan pemilihan solusi atau disebut dengan proses *decoding*, dengan cara evaluasi dan seleksi data, sehingga menghasilkan nilai solusi yang mendekati nilai optimum (Mahmudy, 2015).

2.3.2 Representasi Kromosom

Proses representasi kromosom perlu dilakukan dengan cermat, hal ini disebabkan pada proses ini sangat menentukan bentuk solusi yang akan diproses dan dihasilkan (Melin, Castillo Oscar. Prof., Ramirez, Kacprzyk, & Pedrycz, 2007). Ciri dari objek yang diproses harus dapat digambarkan dengan baik oleh kromosom yang dibentuk sehingga membentuk pola nilai data yang sama, sehingga dapat memperoleh hasil yang optimal dan hasilnya dapat dengan mudah untuk diolah oleh metode utamanya untuk menghasilkan solusi yang optimum pula. Secara sederhana untuk menentukan gambaran bentuk kromosom dapat digambarkan representasi kromosom biner yang merepresentasikan solusi dalam bentuk notasi angka 0 (nol) dan 1 (satu) sebagai representasi pola dari baris solusi atau kromosom tersebut, kemudian ada pula representasi kromosom *string* kromosom dengan merepreantasikan kromosom tersebut dengan nilai asli dari dataset yang telah ditentukan (*real-code*) sebagai baris solusi yang akan dimanipulasi, Representasi *chromosome indexing* dengan menggunakan cara pemberian label pada baris solusi sebagai bentuk manipulasi baris solusinya, dan metode lain sebagainya. Dengan ragam pilihan solusi dalam menentukan bentuk baris solusi yang dapat digunakan sehingga mempermudah dalam melakukan proses pengoptimasian data dengan merujuk pada representasi kromosom tertentu (Mahmudy, 2015)

2.3.3 Inisialisai

Secara tertulis proses inisialisasi merupakan tahapan awal dari perhitungan algoritme genetika, dengan cara memperhitungkan atau menentukan nilai awal dengan pola atau representasi kromosom tertentu sejumlah populasi yang diinginkan (Mahmudy, 2015), sehingga data tersebut yang disebut dengan nilai awal ini dapat digunakan sebagai *pop size* dalam proses algoritme genetika.

2.3.4 Reproduksi

Pada tahap reproduksi, generasi baru akan dihasilkan dari baris solusi awal atau induk saat proses inisialisasi berjalan, selanjutnya para induk yang terpilih akan menghasilkan anak atau *offspring*, sebagai solusi baru yang dihasilkan melalui proses *crossover* dan proses mutasi (Mahmudy, 2015). Jumlah solusi baru yang dapat dihasilkan dalam satu kali proses tergantung pada nilai *Cr* dan *Mr* yang telah ditentukan sebelumnya, pada tahapan ini juga dapat menunjukkan bagaimana data bereaksi terhadap pola *offspring* yang dapat dihasilkan. Contoh paling sederhana dari penerapan perhitungan algoritme genetika yang mulai digambarkan pada proses reproduksi ini yakni:

1. Crossover

Crossover atau kawin silang adalah suatu proses penentu solusi berdasarkan keidentikan pada induknya. Yang merujuk pada proses kawin

silang atau bentuk dasar reproduksi dari makhluk hidup, yang dihasilkan dari indukan yang terpilih secara acak. Proses *crossover* sendiri memiliki varian proses yang dapat digunakan, seperti cara *one-point crossover* dimana niali kromosom dimodifikasi atau dikawinsilangkan antar induk berdasarkan potongan gen dari masing -masing induknya, *two-point crossover* yang memiliki proses yang sama dengan *one-point crossover*, namun pada *two-point crossover* pemotongan kromosom menghasilkan tiga potong dan dilakukan kawin silang berdasarkan komposisi potongan tersebut, *partially-mapped crossover* bekerja dengan menyeleksi secara acak group gen kemudian dikawin silangkan, dan dilakukan perubahan jika terdapat gen ganda dengan aturan tertentu, *position-based crossover* dengan cara kerja memilih gen kromosom secara acak kemudian masukkan gen tersebut ke *parents* baru, dan dilakukan pengecekan jika terdapat kesamaan gen maka gen indukannya akan dibuang, *order crossover* dengan cara kerja pilih *group* gen secara acak kemudian pada *parents* kedua cari dan hapus berdasarkan gen yang terpilih pada *parents* satu diposisi manapun, *replace* gen yang terpilih diposisi gen yang terseleksi. Dan masih banyak lagi. Teknik optimasi yang digunakan sangat bergantung pada representasi data yang digunakan, sehingga kromosom ilegal tidak akan muncul, (Arkeman, Seminar, & gunawan, 2012) . seperti yang telah dijelaskan sebelumnya untuk penentuan nilai *crossover* sangat bergantung pada *Cr* yang ditentukan, sehingga dapat menentukan jumlah *offspring* tertentu, sebagai contoh nilai *Cr* = 0,4 maka akan dihasilkan jumlah *offspring* sebanyak 2, hasil tersebut berasal dari perkalian antara *Cr* dan populasi yang telah ditentukan, dalam hal ini jumlah populasi dimisalkan berjumlah 4, artinya hasil *offspring* berasal dari perhitungan $0,4 \times 4 = 1,6$ yang kemudian dibulatkan menjadi 2. Sehingga misalkan P_1 dan P_3 yang terpilih secara random sebagai induk penentu *offspring* dari proses *crossover* dan akan menghasilkan anak berupa C_1 dan C_2 yang dipresentasikan pada **Gambar 2.1** (Mahmudy, 2015) :

$$\begin{array}{l}
 P_1 \quad [0 \ 1 \ 1 \ 0] \\
 P_3 \quad [1 \ 0 \ 0 \ 0] \\
 \\
 C_1 \quad [0 \ 1 \ 0 \ 0] \\
 C_2 \quad [1 \ 0 \ 1 \ 0]
 \end{array}$$

Gambar 2. 1 Proses Crossover

2. Mutasi

Hal yang hampir sama juga terjadi pada proses mutasi, dimana pada proses ini juga memilih induk secara acak dan akan menghasilkan anak dari induk tersebut. Namun beda dengan proses *crossover*, pada

proses mutasi hasil reproduksi dihasilkan oleh sebuah induk saja, dan dengan satu induk tersebut hanya memilih gen tertentu yang akan dilakukan modifikasi atau mutasi sehingga melahirkan *offspring* baru dari proses perhitungan dengan cara dimutasi, dengan hasil yang disebut dengan mutan. Mutan ini dihasilkan dengan harapan dapat melahirkan solusi baru yang berbeda dengan yang lain, sehingga akan terjaga keragaman populasi yang ada, selain itu juga dapat memperbesar kemungkinan untuk mendapatkan variasi individu atau solusi baru yang diharapkan dapat mengungguli indukannya, (Arkeman, Seminar, & gunawan, 2012). Mutasi juga berperan dalam menggenerate jumlah *offspring*, dengan memanfaatkan variabel mereka (*Mr*) sebagai nilai penentu jumlah *offspring* yang dihasilkan, dengan cara perkalian antara populasi dengan *Mr* yang telah ditentukan, jadi perhitungan yang terjadi ialah $0,2 \times 4 = 0,8$ yang selanjutnya nilai tersebut diubah menjadi bilangan riil dengan hasil sama dengan 1, dan secara default jumlah solusi yang dihasilkan dengan menggunakan teknik *Mr* menghasilkan satu *offspring* dengan penjelasan manualisainya seperti pada **Gambar 2.2** (Mahmudy, 2015):



Gambar 2. 2 Proses Mutasi

2.3.5 Evaluasi

Dalam proses *decoding* untuk menentukan hasil dari proses perhitungan solusi yang didapat, dilakukanlah proses untuk mengevaluasi data tersebut dengan cara memberikan nilai pada masing-masing individu, baik individu induk dan juga individu anak, yakni dengan pemberian nilai *fitness* pada masing-masing individu yang merupakan nilai yang menunjukkan besaran tingkat kebugaran suatu individu, yang memungkinkan digunakan sebagai induk baru, secara manual ditunjukkan pada **Tabel 2.2** (Mahmudy, 2015):

Tabel 2. 3 Proses Evaluasi

	Chromosom	<i>x</i>	<i>f(x)</i>	<i>Fitness</i>
<i>P₁</i>	[0 1 1 0]	6	36	36
<i>P₂</i>	[1 1 0 1]	13	169	169
<i>P₃</i>	[1 0 0 0]	8	64	64
<i>P₄</i>	[1 1 1 1]	15	225	225
<i>C₁</i>	[0 1 0 0]	4	16	16
<i>C₂</i>	[1 0 1 0]	10	100	100
<i>C₃</i>	[1 1 1 0]	14	196	196

Dengan asumsi bahwa nilai *fitness* merupakan representasi dari fungsi *f(x)*.

2.3.6 Seleksi

Populasi yang telah memiliki nilai *fitness* akan dikelompokkan dengan metode tertentu kemudian dipilih sebagai kumpulan baris solusi yang baru, untuk menggantikan baris solusi induk yang sebelumnya dibangkitkan secara acak. Adapun macam kategori yang dapat diaplikasikan yakni dengan cara *elitism*, *roulette wheel*, dan *tournament*. Ketiga kategori tersebut memiliki pendekatan seleksi yang berbeda, misalnya *tournamen*, kategori ini berjalan dengan cara membandingkan dua buah populasi yang dipilih secara acak dan memilih populasi yang memiliki nilai *fitness tertinggi*, sedangkan *roulette wheel* bekerja dengan cara pemberian nilai pada populasi, artinya semakin besar nilai *fitness* pada suatu populasi maka peluang kemunculannya akan semakin besar, kategori lain adalah dengan cara *elitism*, kategori ini memiliki cara kerja yang sederhana, yakni dengan mengurutkan nilai terbaik dari *fitness* masing-masing populasi, kemudian populasi dengan nilai *fitness tertinggi* lah yang akan dipilih (Mahmudy, 2015), seperti yang ditunjukkan pada **Tabel 2.4**.

Tabel 2. 4 Proses Seleksi

	Chromosom	x	$f(x)$	<i>fitness</i>
P_4	[1 1 1 1]	15	225	225
C_3	[1 1 1 0]	14	196	196
P_2	[1 1 0 1]	13	169	169
C_2	[1 0 1 0]	10	100	100
P_3	[1 0 0 0]	8	64	64
P_1	[0 1 1 0]	6	36	36
C_1	[0 1 0 0]	4	16	16

2.4 Prediksi Data *Time series*

Peramalan atau juga disebut dengan prediksi merupakan suatu analisis perbandingan pada suatu kejadian mungkin akan terjadi (Watimena, 2008), dan data *time series* adalah sekumpulan data, atau dataset yang secara historis diurutkandata dalam suatu kesatuan data (Zulfikar & Budiantara, 2014). Sehingga prediksi data *time series* dapat dikatan sebagai suatu proses untuk menganalisis perbandingan data yang berada dalam kurun waktu tertentu yang memiliki urutan secara historis.

Data sendiri berdasarkan fungsi dan jenisnya dapat dibagi menjadi empat jenis, yakni (Ukhra, 2014) :

1. Pola *trend* adalah pola data yang memiliki sifat hasil pengukuran data yang dengan jangkan panjang, sehingga pola data ini sangat erat kaitannya pada pola data urutan waktu yang panjang dengan hasil yang bisa saja fluktuatif.
2. Pola siklis adalah Pola data yang dipengaruhi lingkungan politik dan perubahan ekonomi yang fluktuatif, yang terjadi dalam perubahan bisnis pada suatu wilayah.

3. Pola musiman adalah pola data yang terjadi pada kurun waktu atau dalam suatu periode tertentu.
4. Pola *irregular* adalah pola data yang bersifat fluktuatif dyang memiliki nilai acak sehingga sulit untuk ditebak.

2.5 Jaringan Saraf Tiruan

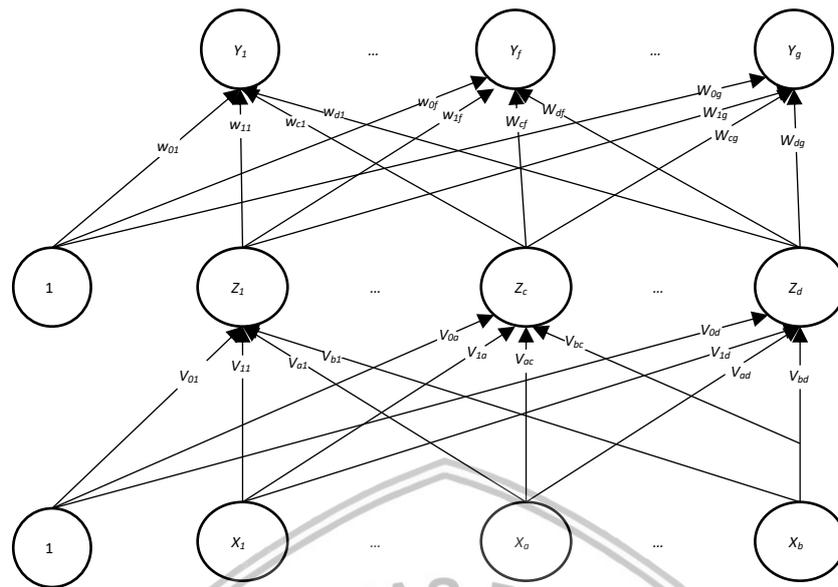
Jaringan Saraf Tiruan (JST) merupakan suatu metode pencarian informasi berbasis populasi yang memiliki cara kerja menghubungkan antar data atau yang selanjutnya disebut dengan *layer*, metode ini dibuat berdasarkan representasi dari jaringan saraf yang dimiliki oleh manusia yang memiliki tingkat kompleksitas yang tinggi, seperti halnya pada jaringan saraf pada otak. Dengan kapabilitas sebagai media penyimpanan, mengenal dan menirukan, dengan proses pembelajaran sebelumnya (Bustomi, Bisri, & Purwanti, 2014)

2.6 Jaringan Saraf Tiruan *Backpropagation*

Jaringan Saraf Tiruan *Backpropagation* ialah salah satu bentuk pengembangan dari jaringan saraf tiruan itu sendiri, dengan penambahan propagasi mundur untuk menyempurnakan metode peramalan ini agar dapat berjalan lebih akurat. Pada metode ANN-BP pada dasarnya mengelola informasi berdasarkan beberapa perulangan atau *epoch* dengan memakai beberapa data latih, yang selanjutnya akan digunakan untuk menguji hasilnya dengan menggunakan data uji, dengan tujuan utama yakni melakukan modifikasi pada bobot data sehingga menghasilkan informasi yang akurat dengan parameter nilai *error* MSE paling rendah, dalam prosesnya (Kusrini & Luthfi, 2009).

2.6.1 Arsitektur Jaringan Saraf Tiruan *Backpropagation*

Arsitektur Jaringan Saraf Tiruan *backpropagation* dibentuk dengan tiga bagian atau layer utama, yakni *input layer* yang terdiri dari beberapa unit, yang direpresentasikan dengan simbol X , sedangkan pada layer berikutnya terdiri dari sekumpulan unit hidden atau layer tambahan yang disimbolkan dengan Z , dan pada *output layer* terdiri dari unit hasil *generate* solusi yang disimbolkan dengan Y , yang digambarkan pada **Gambar 2.3**.



Gambar 2. 3 Arsitektur Jaringan Saraf Tiruan *Backpropagation*

Keterangan :

- v_{ij} : bobot atau nilai dari unit x_i menuju ke unit *hidden layer* z_j
- v_{0j} : nilai bobot dari unit bias *input* menuju ke unit *hidden layer* z_j
- w_{jk} : bobot atau nilai dari unit z_j ke unit *output layer* y_k
- w_{0k} : nilai bobot dari unit bias *hidden* ke unit *output layer* y_k

2.6.2 Pelatihan Jaringan Saraf Tiruan *Backpropagation*

Variabel penentu dalam proses jaringan saraf tiruan *backpropagation* pada umumnya adalah:

1. Inisialisasi data bobot

Nilai inisialisasi bobot diawal ditentukan secara acak berdasarkan populasi tertentu, dengan *range* yang sangat kecil, yakni mendekati nilai nol (0), untuk memudahkan dalam proses perhitungannya.

2. *Learning rate*

Nilai *learning rate* turut berpengaruh terhadap tingkat keefektifan, kecepatan, dan ketepatan dalam memproses data, hal ini dikarenakan nilai *learning rate* atau yang biasa disebut *alpha* ini menentukan jangkauan pencarian solusi pada ANN-BP, dengan nilai berada pada antara 0 sampai engan 9.

3. Penentuan jumlah *hidden layer*

Jumlah penentuan *hidden layer* yang tepat juga mempengaruhi hasil yang diperoleh, hal ini disebabkan segala proses utama dalam ANN-BP berada pada *hidden layer*, dan pada bagian paling mendasar agar penentuan bobot tidak terjadi *underfitting* atau *overfitting*.

Pelatihan menggunakan ANN-BP secara garis besar menurut Siang, dalam Hudiayawan (2015) terdiri atas tiga fase

1. Fase I : *Feedforward*

Fase I, untuk menghasilkan bobot awal (v_{ij}) dari *input* (x_i) menuju *hidden layer* (z_j) dan juga bobot (w_{jk}) dari *hidden* menuju *output layer* (y_k). Dengan kedua hasil tersebut selanjutnya diberikan fungsi aktivasi untuk menjaga keragaman hasil yang tidak terlalu jauh. Dan hal tersebut dilakukan berulang-ulang berdasarkan panjang data pada masing-masing bobot dan tujuan pencarian solusi yang diinginkan, sampai mencapai titik konvergen.

2. Fase II : *Backpropagation*

Fase II, dengan melakukan perbandingan nilai antara hasil sementara peramalan (y_k) dengan target yang ada, sehingga dapat ditentukan tingkat kesalahan yang terjadi, dan selanjutnya dapat dilakukan propagasi mundur, selanjutnya sinyal *error* tersebut diteruskan pada bagian selanjutnya, yakni dengan menentukan selisih nilai kesalahan bobot yang terjadi antara unit pada *output* menuju *hidden layer* (Δw_{kj}), selanjutnya dihitung nilai bobot kesalahan yang terjadi pada unit *output* menuju *hidden layer* dengan notasi (δ_{net_j}), dan selanjutnya δ_{net_j} akan dihitung dengan nilai kesalahan pada *hidden layer* dengan notasi (δ_j), sehingga pada tahap terakhir dapat ditentukan pula selisih nilai kesalahan dari unit *hidden layer* yang menuju ke *input layer* dengan notasi (Δv_{ki}). Pada keseluruhan tahapan tersebut dilakukan perhitungan berulang-ulang hingga mencapai titik konvergensi pada masing-masing proses.

3. Fase III : *Weight Update*

Fase III, tahapan modifikasi nilai bobot, baik itu bobot (v) maupun bobot (w) dilakukan dengan, menjumlahkan nilai bobot tersebut dengan masing-masing nilai faktor kesalahannya masing-masing, yakni (Δv_{ki}) dan (Δw_{kj}), sehingga dapat digunakan sebagai bobot baru, dan proses penentuan bobot tersebut secara keseluruhan juga dilakukan dengan iterasi tertentu sehingga mencapai nilai konvergensi.

Langkah metode ANN-BP dengan menggunakan fungsi *sigmoid* biner menurut Siang, dalam Hudiayawan (2015) ialah :

1. Langkah 0 : Membangkitkan nilai bobot secara acak dengan nilai yang mendekati nol (0).
2. Langkah 1 : Lakukan perulangan pada langkah 2 sampai dengan langkah 8 jika kondisi berhenti belum terpenuhi.
3. Langkah 2 : Untuk tiap data pelatihan lakukanlah langkah 3 sampai dengan 8.

Fase I : Feedforward

4. Langkah 3 : Setiap unit pada *hidden layer* menerima sinyal dari tiap unit-unit yang ada pada *input layer*.
5. Langkah 4 : Hasil sinyal dari unit *input layer* digunakan untuk menghasilkan nilai *hidden alyer* dengan menggunakan Persamaan 2.1.

$$z_net_j = v_{0j} + \sum_{i=1}^n x_i \cdot v_{ij} \tag{2.1}$$

Keterangan daftar notasi :

- z_net_j : Nilai *hidden layer*.
- v_{0j} : Nilai sinyal atau bobot bias dari *input layer*.
- x_i : dataset yang telah dibangkitkan
- v_{ij} : Nilai sinyal atau bobot dari unit *input layer* ke *hidden layer*.

Dengan menggunakan Persamaan 2.1 untuk menentukan nilai *hidden layer*, selanjutnya nilai *hidden layer* tersebut dibangkitkan dengan nilai silai sigmoid, dengan menggunakan Persamaan 2.2.

$$z_i = f (z_net_i) = \frac{1}{1+exp(-z_{net})} \tag{2.2}$$

- z_i : Hasil pengaktifan nilai *hidden layer*.
- z_net_i : Nilai hidden layer

Kemudian dari *hidden layer* sinyal dilanjutkan ke unit *output layer* dengan bantuan dari nilai *hidden layer*, sehingga dapat menghasilkan nilai *output layer* .

6. Langkah 5 : Tiap unit *output layer* dibangkitkan dengan menggunakan Persamaan 2.3, sepanjang data unit *output layer*.

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

Keterangan daftar notasi :

- y_net_k : Nilai unit *output layer*
- w_{0k} : Nilai sinyal atau bobot bias dari *hidden layer*.
- w_{jk} : Nilai sinyal atau bobot dari *hidden layer*.



z_j : Nilai aktivasi *hidden layer*.

Dengan menggunakan Persamaan 2.3 untuk menentukan nilai *output layer*, selanjutnya nilai *output layer* tersebut dibangkitkan dengan nilai silai sigmoid, yang menggunakan Persamaan 2.4.

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

Keterangan daftar notasi :

y_k : Hasil pengaktifan nilai unit *output layer*.

y_{net_k} : Nilai *output layer*.

Fase II : Backpropagation

7. Langkah 6

Menghitung nilai kesalahan berdasarkan perbandingan nilai peramalan dengan nilai aktual.

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

δ_k : Perbandingan nilai kesalahan antara ouput layer dengan nilai aktual.

t_k : Target atau nilai aktual.

y_k : Nilai unit *output layer*.

δ_k merupakan nilai kesalahan yang dihasilkan dari peramalan sementara pada proses *feedforward* yang digunakan untuk menentukan nilai faktor kesalahan pada sinyal atau bobot (w) dengan Persamaan 2.6.

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

Keterangan daftar notasi

Δw_{jk} : Niai fakto kesalahan bobot menuju ke *hidden layer*

α : Nilai jangkauan pelatihan (*learning rate*) berada antara 0 sampai 1.

δ_k : Nilai kesalahan dari *output layer*.

z_j : Nilai *hidden layer*.

j : indek pada *hidden layer*

k : indek pada *output layer*

8. Langkah 7



Menghitung nilai kesalahan pada *hidden layer* berdasarkan nilai kesalahan pada *output layer* (δ_k), dengan menggunakan persamaan 2.7.

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

Keterangan daftar notasi

δ_{net_j} : Nilai kesalahan pada *layer* tersembunyi atau *hidden layer*

δ_k : Nilai kesalahan pada *layer* keluaran atau *output layer*

w_{jk} : Nilai sinyal atau bobot dari *hidden layer* ke *output layer*.

Dengan menggunakan nilai kesalahan pada *hidden layer*, selanjutnya menghitung selisih nilai kesalahan tersebut dengan menggunakan Persamaan 2.8.

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

Keterangan daftar notasi

δ_j : Selisih nilai kesalahan pada *hidden layer*

Selanjutnya dilakukan perhitungan nilai faktor kesalahan dari *hidden layer* menuju ke *input layer*, dengan menggunakan Persamaan.

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

Keterangan daftar notasi :

Δv_{ij} : Nilai faktor kesalahan bobot menuju ke *input layer*

α : Nilai jangkauan pelatihan (*learning rate*) berada antara 0 sampai 1.

δ_i : Selisih nilai kesalahan pada *hidden layer*.

x_i : dataset yang telah dibangkitkan

j : indeks pada *hidden layer*.

i : indeks pada *input layer*

Fase III : Weight Update

9. Langkah 8

Menghitung nilai perbaikan sinyal atau bobot antara *hidden layer* dan *output layer*, dengan menggunakan Persamaan 2.10.

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (2.10)$$

Keterangan daftar notasi

$w_{jk}(\text{baru})$: Nilai perbaikan sinyal atau bobot antara *hidden layer* dan *output layer*.

$w_{jk}(\text{lama})$: Nilai sinyal atau bobot unit hidden layer ke output layer

Δw_{jk} : Nilai fakto kesalahan bobot menuju ke *hidden layer*

Menghitung nilai perbaikan sinyal atau bobot antara *input layer* dan *hidden layer*, dengan menggunakan Persamaan 2.11.

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

Keterangan daftar notasi :

v_{ij} (baru) : Nilai perbaikan sinyal atau bobot antara *input layer* dan *hidden layer*.

v_{ij} (lama) : Nilai sinyal atau bobot unit hidden layer ke output layer

Δv_{ij} : Nilai fakto kesalahan bobot menuju ke *input layer*

2.6.3 Tingkat Akurasi

Menentukan tingkat akurasi dapat dilakukan dengan menghitung nilai *error* atau seberapa kecil nilai kesalahan yang dihasilkan oleh sistem yang dibuat dalam tiap proses perulangan yang dilakukan sehingga dapat ditentukan nilai *error* berdasarkan nilai rata-rata *error* yang terjadi dalam tiap perulangan yang dilakukan. Nilai *error* (MSE) dikatakan baik jika nilai tersebut mendekati hasil 0 (nol) dan menjauhi nilai 1 (satu). Untuk mendapatkan nilai *error* (MSE) tersebut perhitungan yang digunakan berdasarkan Persamaan 2.12.

$$MSE = \frac{1}{n} \sum_{t=1}^n (t - y_t)^2 \quad (2.12)$$

Keterangan :

t = Nilai target

y_t = Nilai peramalan pada *output layer*.

n = Jumlah dataset

2.6.4 Normalization dan Denormalization Data

Proses *normalization* data digunakan untuk menyeragamkan data dengan cara menyederhanakan data tersebut dalam nilai yang sangat kecil sehingga mempermudah untuk dilakukan perhitungan dengan nilai antara 0 (nol) sampai dengan 1 (satu), dengan menggunakan Persamaan 2.13.

$$x' = \frac{0,8(x - \text{minvalue})}{(\text{maxvalue} - \text{minvalue})} + 0,1 \quad (2.13)$$

Sedangkan dengan proses yang berbeda, *denormalization* digunakan untuk membentuk kembali nilai hasil *normalization*, sehingga memiliki nilai data aktual

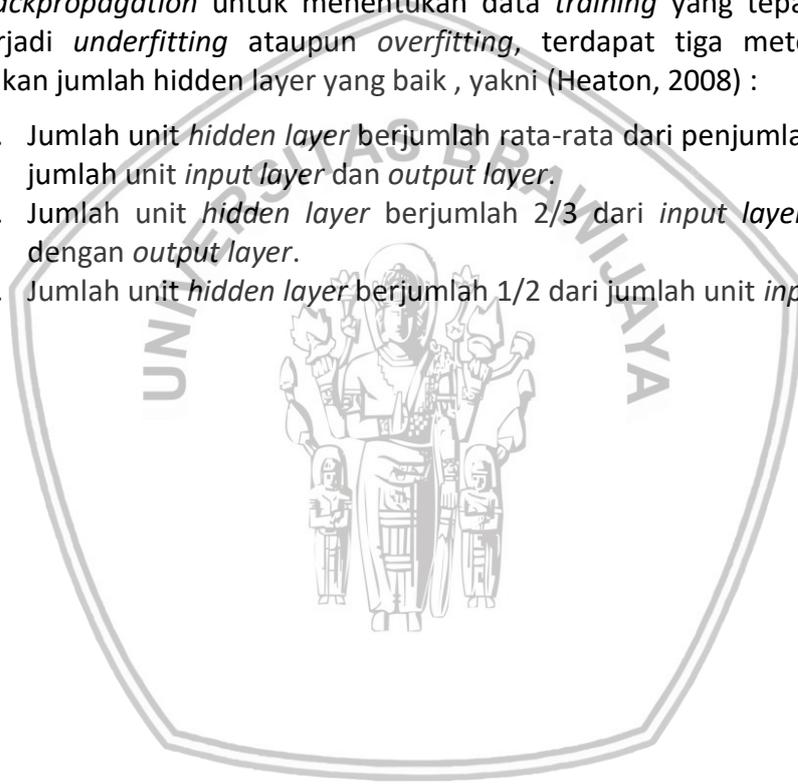
atau data mentah yang belum dilakukan proses samasekali, dengan menggunakan Persamaan 2.14.

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

2.6.5 Jumlah *Hidden layer*

Keberadaan *hidden layer* merupakan bagian yang sangat penting dalam jaringan saraf tiruan *backpropagation* untuk menentukan bentuk arsitektur yang baik, meskipun begitu *hidden layer* tidak bersentuhan langsung dengan pengelolaan yang dapat dilakukan pengguna namun sangat berpengaruh terhadap hasil akhir dari *output* yang akan dihasilkan dari metode jaringan saraf tiruan *backpropagation* untuk menentukan data *training* yang tepat sehingga tidak terjadi *underfitting* ataupun *overfitting*, terdapat tiga metode dalam menentukan jumlah *hidden layer* yang baik, yakni (Heaton, 2008) :

1. Jumlah unit *hidden layer* berjumlah rata-rata dari penjumlahan antara jumlah unit *input layer* dan *output layer*.
2. Jumlah unit *hidden layer* berjumlah $\frac{2}{3}$ dari *input layer* ditambah dengan *output layer*.
3. Jumlah unit *hidden layer* berjumlah $\frac{1}{2}$ dari jumlah unit *input layer*.



BAB 3

METODOLOGI PENELITIAN

Dalam metodologi penelitian, menjelaskan teknik penerapan tiap metode atau atribut yang digunakan dan berjalan secara runtut berdasarkan data tingkat pengguna kereta api di Indonesia. Dengan menerapkan kaidah-kaidah yang diperlukan dalam pemrosesan sistem berdasarkan kasus yang diangkat sebagai topik pembahasan, yang meliputi tipe penelitian dan rancang bangun penelitian.

3.1 Tipe Penelitian

Pada penelitian peramalan penumpang kereta api ini, peneliti menggunakan tipe penelitian *non-implimentatif* deskriptif, dimana dalam penelitian ini lebih berfokus untuk penerapan dalam segi keilmuan pada kasus tertentu yang menghasilkan hasil analisis dari kasus yang diangkat untuk mencari solusi yang terbaik, dengan proses pengolahan data kuantitatif.

3.2 Rancang Bangun Penelitian

Dalam rancang bangun penelitian berfokus pada tahapan pengolahan sistem secara teknis, yang terdiri dari beberapa tahapan.

3.2.1 Metode Penelitian

Dalam metode penelitian secara umum ini, peneliti menggunakan metode eksperimental, sehingga untuk menyelesaikan permasalahan ini peneliti melakukan pengujian variabel-variabel tertentu untuk mengetahui permasalahannya, dan berdasarkan data permasalahan yang dihimpun maka dapat menentukan langkah yang dapat diambil dalam pemecahan permasalahan tersebut.

3.2.2 Subjek Penelitian

Penelitian yang dihimpun dari data sekunder ini diperoleh dari lembaga yang kredibel dalam menyajikan data statistik di Indonesia, hal ini sangat sesuai dengan data yang dibutuhkan oleh peneliti yang memerlukan data yang dijamin kebenarannya dan dikelola langsung oleh pemerintah sehingga memiliki korelasi dengan objek permasalahan yang juga dikelola oleh pemerintah.

3.2.3 Lokasi Penelitian

Pada penelitian ini dilakukan secara mandiri dan juga dengan memanfaatkan laboratorium komputasi cerdas Fakultas Ilmu Komputer Universitas Brawijaya untuk memproses penerapan penggunaan metode ANN-BP dan algoritme genetika.

3.2.4 Metode Pengumpulan Data

Data yang digunakan merupakan data dalam format .csv pada *file* excel berupa data *timeseries* dalam bentuk data bulanan, yang didapat dari BPS terkait jumlah penumpang kereta api di Indonesia, dengan rentang waktu januari 2006 sampai dengan juni 2017, kemudian diolah berdasarkan pola yang diinginkan guna mempermudah proses penelitian, dengan dilakukannya proses penentuan data latih dan data uji sebagai media peramalan tersebut.

Data latih ditentukan berdasarkan pola tertentu, pada penelitian ini menggunakan pola $x_1, x_2, x_3, x_4, x_5, x_6$, dan target (y) seperti yang digambarkan dalam **Tabel 4.1** sehingga pada x_1 akan menampung data awal mulai bulan januari 2006 dan seterusnya, x_6 akan menampung bulan ke-6 (data bulan juni 2006), sedangkan target (y) menampung bulan selanjutnya, yakni bulan juli 2006. Untuk pola kedua dimulai pada x_1 diisi dengan data pada bulan februari 2006 dan seterusnya, sampai nilai target (y) berisi data bulan agustus 2006, begitu pula pada pola-pola selanjutnya hingga membentuk panjang pola yang telah ditentukan peneliti, yakni peneliti menggunakan 100 pola pada data latih. Hal yang sama juga dilakukan pada data uji seperti yang digambarkan pada **Tabel 4.2**, namun yang membedakan data latih dan data uji yakni pada penggunaan data berada pada data akhir yang digunakan sebagai data uji, disini peneliti menggunakan 10 pola data uji. Sehingga data latih dan data uji tersebut akan dilakukan perhitungan sesuai fungsinya masing-masing. Dengan penggunaan data latih sebagai data untuk penentuan tren pada metode, dan tren tersebut akan digunakan sebagai acuan untuk melakukan pengujian peramalan dengan menggunakan data uji, sehingga hasil pengujian tersebut dapat digunakan sebagai rujukan untuk melakukan peramalan informasi.

3.2.5 Metode Analisis Data

Analisis dilakukan agar mengetahui pengaruh dari penggunaan variabel yang digunakan dan nilai dari variabel yang digunakan, sehingga didapati komposisi yang baik digunakan dalam melakukan peramalan pengguna kereta api di Indonesia dengan modifikasi terhadap variabel yang berpengaruh, yakni :

1. Dengan memodifikasi ukuran populasi untuk menentukan luas area pencarian solusi.
2. Melakukan modifikasi pada jumlah generasi untuk memperbesar kemungkinan solusi yang dihasilkan.
3. Melakukan modifikasi pada komposisi Cr dan Mr untuk mengetahui model yang dapat menghasilkan solusi yang baik.
4. Modifikasi pada iterasi yang juga merupakan cara untuk memperluas area pencarian solusi, namun yang bekerja pada metode ANN-BP.
5. Modifikasi pada jangkauan pencarian solusi dengan menentukan nilai alpha.

Selain kelima model yang dilakukan analisa untuk menentukan variabel bagi metode peramalan ANN-BP dan algoritme genetika, adapun variabel lain yang turut mempengaruhi penilaian peramalan metode ANN-BP dan algoritme genetika, yakni dengan melakukan pengujian untuk membandingkan hasil antara metode *backpeopagation* yang telah dilakukan optimasi nilai bobot (w) dengan menggunakan algoritme genetika dan hasil dari penggunaan metode ANN-BP saja.

3.2.6 Peralatan Pendukung

Dalam melakukan penerapan sistem, peneliti dibantu dengan peralatan pendukung yang terdiri dari :

1. Kebutuhan perangkat lunak:
 - a. Windows 10 dengan sistem operasi 64 bit.
 - b. NetBeans sebagai IDE (bahasa pemrograman Java)
2. Kebutuhan perangkat keras:
 - a. Laptop dengan kinerja prosesor sebesar 2.71 GHz
 - b. Akses memori sebesar 4 GB
 - c. Ruang penyimpanan sebesar 500 GB



BAB 4

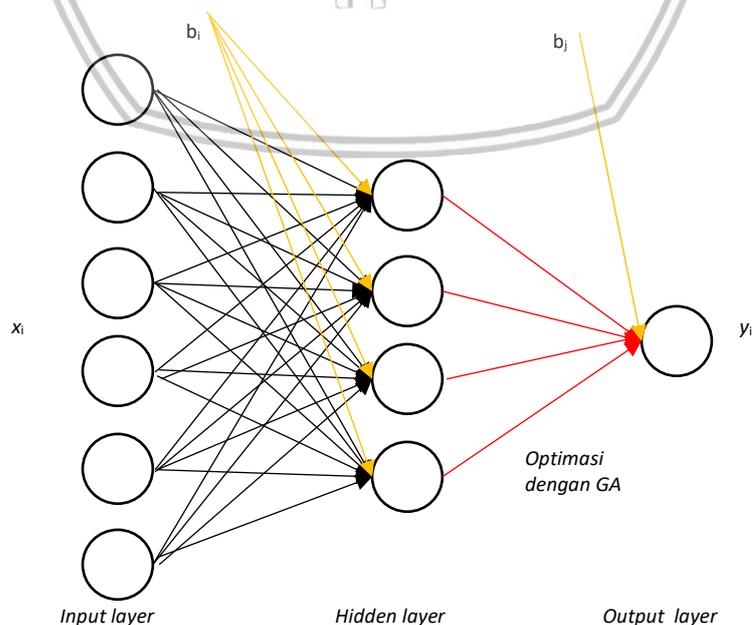
PERANCANGAN DAN IMPLEMENTASI

Pembahasan pada bab perancangan dan implementasi sistem, merupakan tahapan penerapan metode dan algoritme baik dari segi perancangan dan dilanjutkan dengan proses implementasinya berdasarkan teori yang telah dikemukakan pada bab-bab sebelumnya, dengan penjelasan berupa deskripsi terkait sistem yang dibuat, diagram alir, bentuk manualisasi, perancangan sistem dan hasil keluaran yang diharapkan beserta dengan implementasinya.

4.1 Deskripsi Umum Sistem

Sistem peramalan jumlah penumpang kereta api Indonesia ini digunakan untuk mendapatkan suatu olahan informasi dari peramalan jumlah penumpang kereta api Indonesia berdasarkan pada tahun-tahun sebelumnya dalam bentuk data perbulan, diurutkan secara historis dan berbentuk data *time series* dari bulan januari 2006 sampai dengan bulan juli 2017. Dengan pemrosesan yang dibentuk dari arsitektur ANN-BP berupa data unit *layer input* sejumlah enam dan ditambah satu unit bias, data pada *hidden layer* sejumlah empat unit dan ditambah satu unit bias dengan ketentuan jumlah *hidden layer* berasal dari jumlah rata-rata antar *input layer* dengan *output layer* (Heaton, 2008), sedangkan pada *layer output* berjumlah satu.

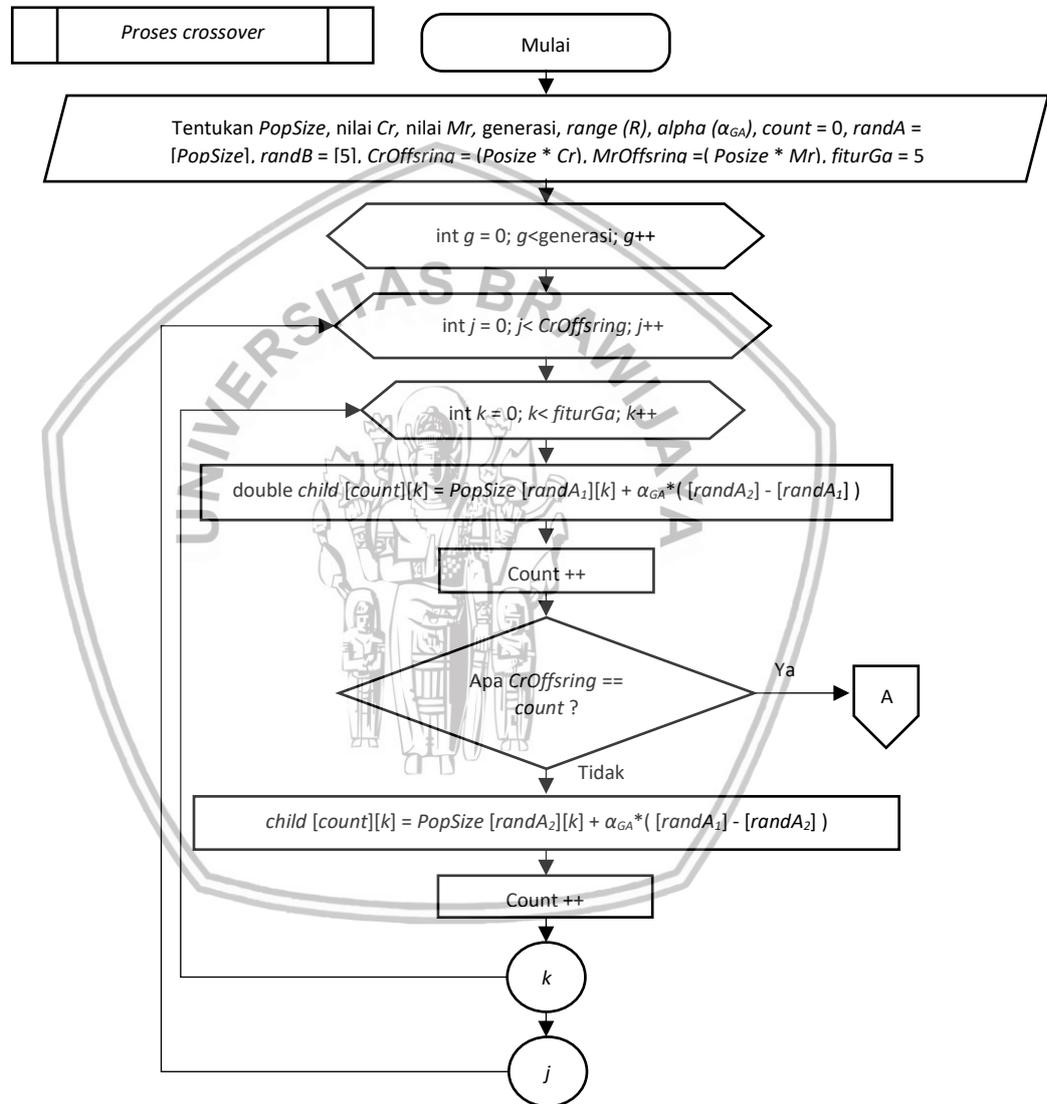
Optimasi yang dilakukan oleh algoritme genetika berada tepat antara *hidden layer* dan *input layer*, berupa bobot (w), seperti halnya yang juga pernah dilakukan oleh Wiharto, Palgunadi, & Nugroho (2013), dengan penggunaan bobot (w) sebagai media hibridisasi untuk mengurangi jangka waktu eksekusi. Dengan arsitektur ANN-BP dan algoritme genetika dijabarkan pada **Gambar 4.1**.



Gambar 4. 1 Arsitektur Jaringan Saraf Tiruan *Backpropagation*

4.2 Rancang Bangun Sistem

Pada perancangan bangun sistem ini menjelaskan secara sistematis proses pencarian solusi dengan bentuk diagram alir, dengan berfokus pada cara penentuan variabel-variabel utama dalam peramalan jumlah penumpang kereta api Indonesia dengan menggunakan ANN-BP dan algoritme genetika, sehingga memudahkan dalam melakukan pelacakan proses dalam representasi dari sistem yang dibuat, dengan digambarkan pada serangkaian diagram alir sebagai berikut :

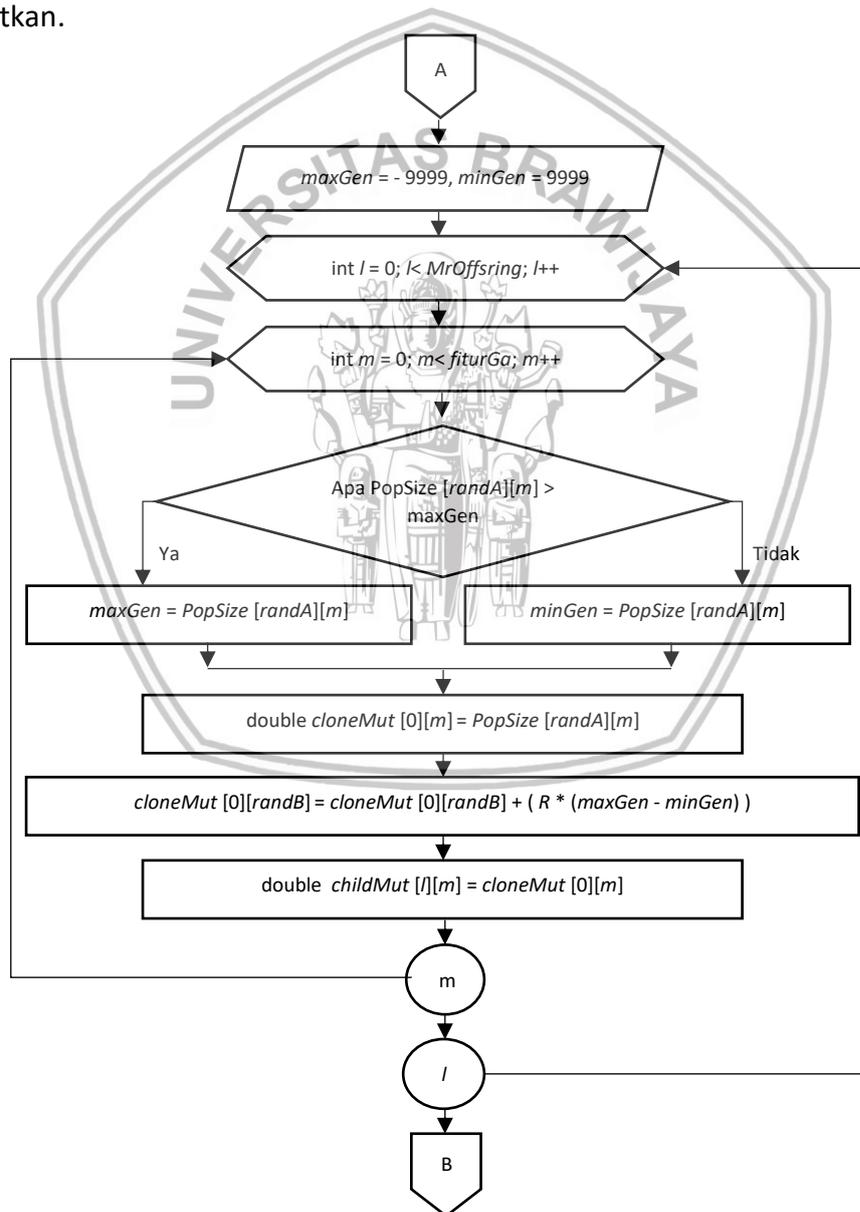


Gambar 4. 2 Diagram Alir Crossover

Pada **Gambar 4.2** merupakan bagian dari proses dalam menentukan solusi yang terjadi dalam proses *crossover*, dimana pada tahap awal dilakukan inisialisasi nilai variabel-variabel penentu. Selanjutnya dalam proses *crossover* sistem akan menentukan jumlah *offspring* yang dapat dihasilkan dalam proses *crossover*,

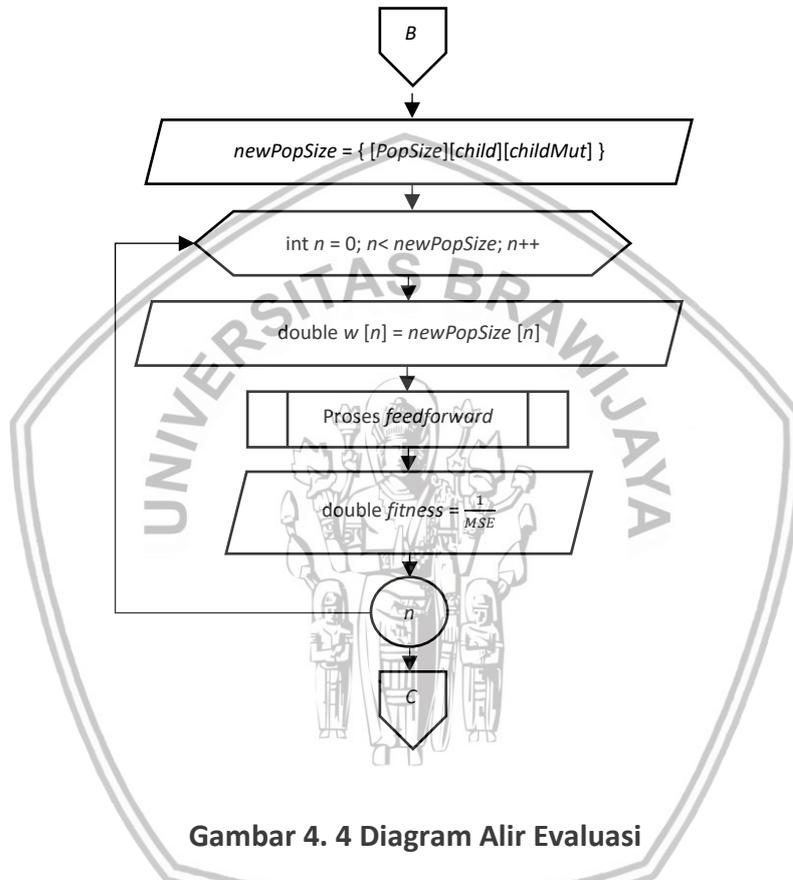
dengan melakukan perkalian jumlah *pop size* dan nilai *Cr*, selanjutnya sistem akan menghasilkan *offspring* baru dan dilakukan pengecekan, apakah dalam satu perulangan tersebut menghasilkan dua atau satu *offspring*, kemudian proses dilanjutkan dengan melakukan perulangan berdasarkan indeks *k* sehingga menghasilkan *offspring* yang lain. Sampai pada kondisi berhenti, yakni telah mencapai indeks *j*.

Pada **Gambar 4.2** juga menunjukkan bahwa dilakukan pengecekan iterasi dalam proses algoritme genetika, sehingga jika setiap perulangan yang dilakukan kembali pada proses *crossover*, maka sistem akan menanyakan apakah iterasi sudah terpenuhi, jika telah mencapai indeks *g* maka proses akan dilanjutkan pada proses ANN-BP, jika tidak maka proses regenerasi algoritme akan kembali dilanjutkan.



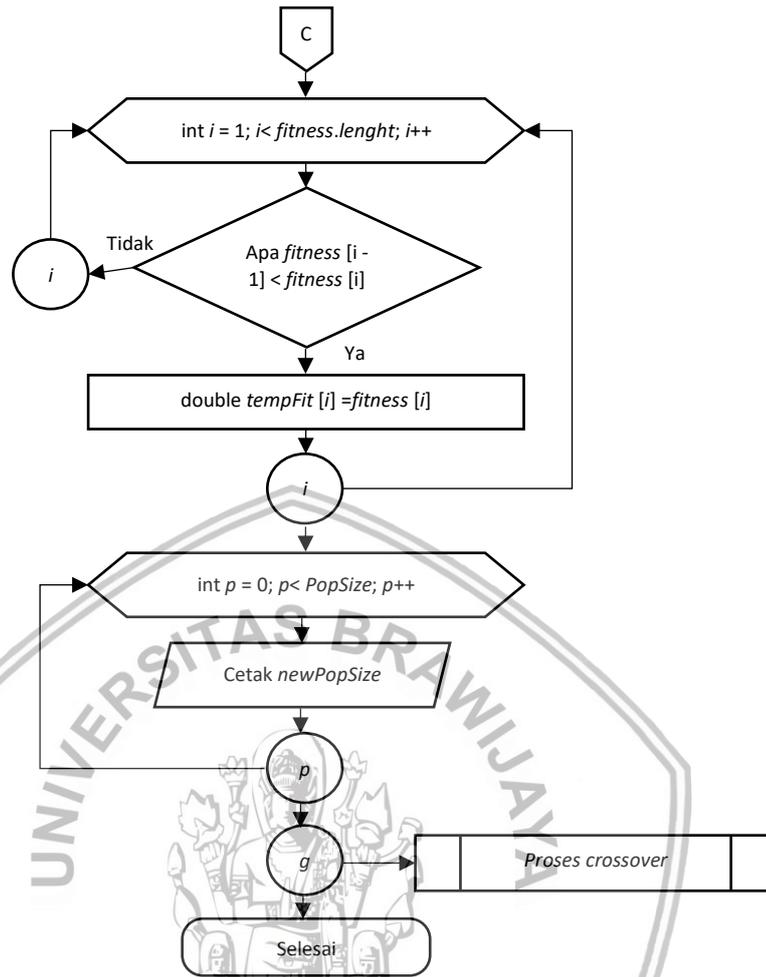
Gambar 4. 3 Diagram Alir Mutasi

Selanjutnya pada **Gambar 4.3**, menunjukkan proses penentuan *offspring* yang dilakukan berdasarkan *Mr* atau *Mutation rate*, dengan menentukan jumlah *offspring* berdasarkan perkalian antara *Pop size* dan *Mr*, selanjutnya dipilih baris solusi secara acak dan unit dari baris solusi secara acak pula, yang selanjutnya pada unit solusi dilakukan perhitungan penentuan *offspring* yakni *cloneMut* dan begitu pula seterusnya dilakukan perulangan sampai mencapai nilai indeks *l*.



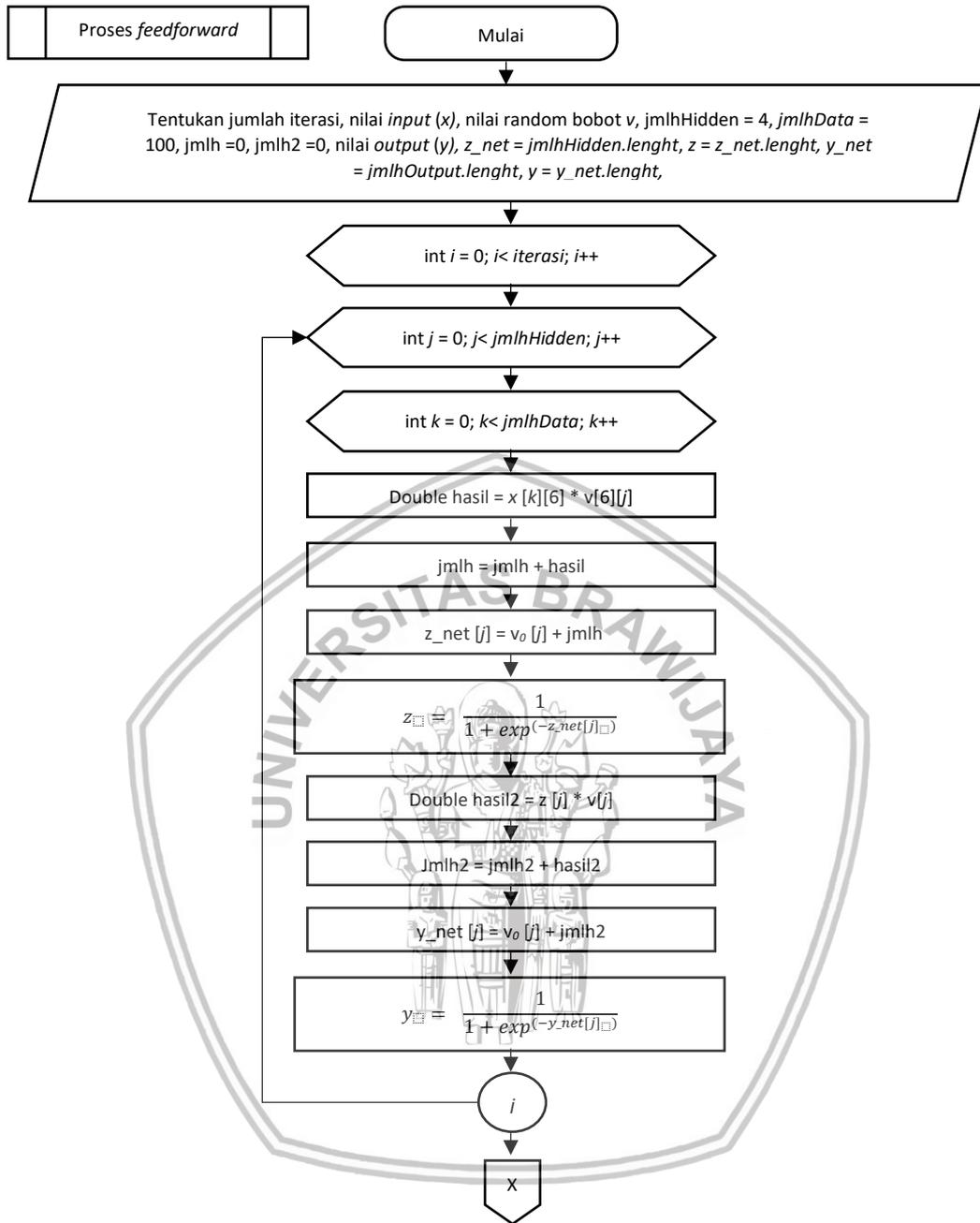
Gambar 4. 4 Diagram Alir Evaluasi

Setelah dilakukan proses reproduksi, tahap selanjutnya yakni dilakukan proses evaluasi, seperti pada **Gambar 4.4**. Pada gambar tersebut menyatakan bahwa semua populasi, baik induk maupun *offspring* digabungkan sehingga dapat dilakukan perhitungan nilai *fitness* pada tiap individu yang ada dalam satu iterasi.



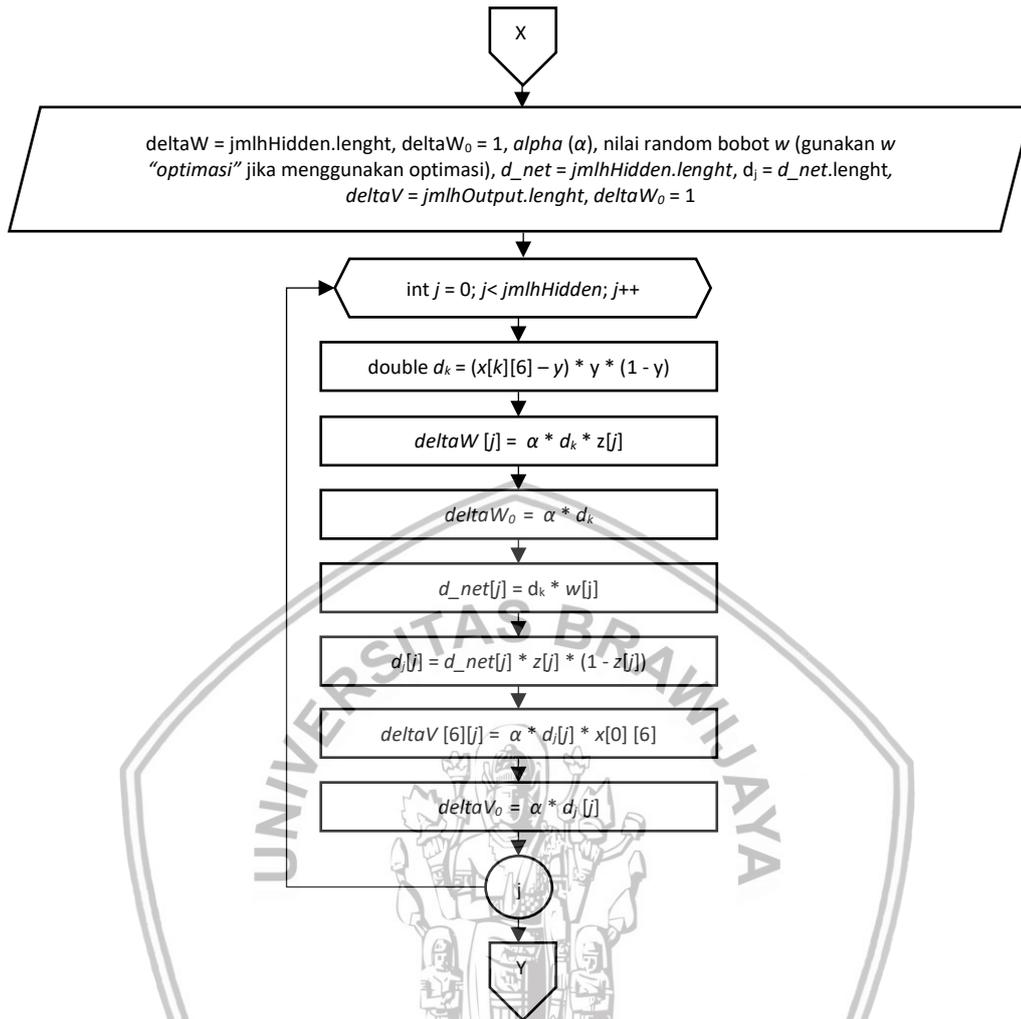
Gambar 4. 5 Diagram Alir Seleksi

Selanjutnya, setelah ditentukan nilai *fitness* pada semua populasi, maka tahapan berikutnya adalah melakukan seleksi pada populasi yang memiliki nilai *fitness* terbaik sejumlah populasi induk sebelumnya, seperti pada **Gambar 4.5** menunjukkan bahwa tiap populasi yang baru tersebut diurutkan secara *descending* kemudian diambil sejumlah populasi induk sebelumnya, dan diproses ke iterasi selanjutnya untuk menyeleksi generasi baru lagi, sampai mencapai nilai konvergensi. Pada tahapan ini juga menunjukkan langkah lanjutan yang diproses pada proses *crossover*, dengan mengembalikan indeks perulangan sejumlah iterasi berdasarkan generasi yang telah ditentukan pada proses *crossover*.



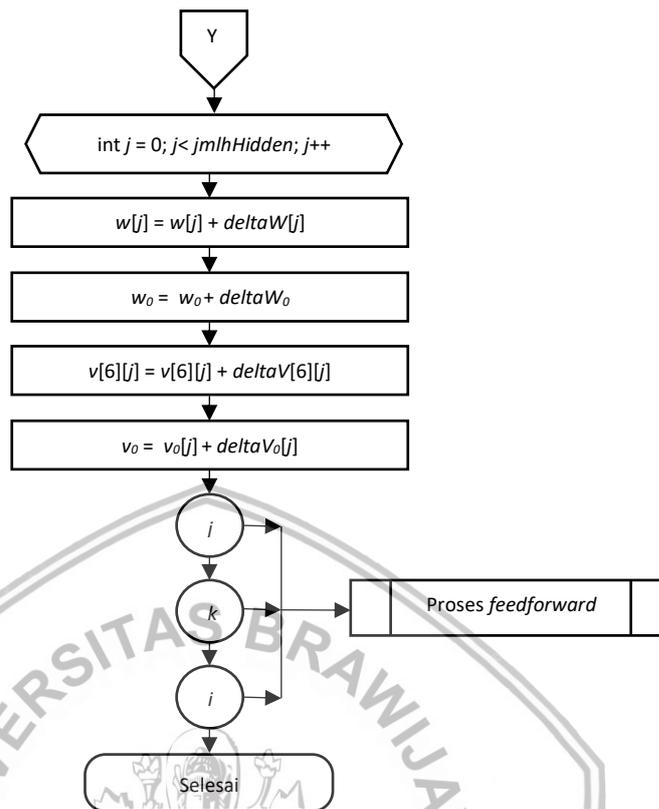
Gambar 4. 6 Diagram Alir Proses Feedforward

Pada **Gambar 4.6** yang merupakan bagian awal dari fase jaringan saraf tiruan *backpropagation* menjelaskan, bahwa pada fase ini menghasilkan dua hasil dari empat persamaan, yakni nilai *hidden layer* (z) yang dihasilkan dari penentuan nilai z_net dan diaktifkan dengan menggunakan fungsi aktivasi seperti yang ditunjukkan pada Persamaan 2.1 juga 2.2, dan pada fase ini juga menghasilkan nilai *output layer* (y), hasil dari aktivasi dari nilai y_net seperti yang ditunjukkan dari Persamaan 2.3 dan 2.4. Yang masing-masing hasil tersebut dilakukan perulangan sebanyak panjang data yang mereka miliki.



Gambar 4. 7 Diagram Alir Proses *Backpropagation*

Pada fase kedua dalam proses ANN-BP seperti yang terlihat pada **Gambar 4.7** menunjukkan proses utama yakni proses propagasi mundur, dengan tujuan untuk membawa informasi perbaikan dari proses *feedforward* yang telah dilakukan sebelumnya. Pada tahap ini menghasilkan informasi data selisih kesalahan yang terjadi pada *output layer* dengan notasi (d_k), yang selanjutnya dilakukan pencarian nilai faktor kesalahan yang terjadi pada bobot dari *output* menuju *hidden layer* ($\text{delta } w_{kj}$) sepanjang data bobot (w), selanjutnya dilakukan pencarian nilai kesalahan yang terjadi pada *hidden layer* dengan notadi (d_j) sepanjang data *hidden layer*, kemudian dilakukan juga perhitungan nilai faktor kesalahan yang terjadi pada bobot dari *hidden* menuju *input layer* ($\text{delta } v_{ji}$) sepanjang data bobot input layer (v).



Gambar 4. 8 Diagram Alir Proses Weight Update

Pada proses terakhir dari fase ANN-BP, seperti yang ditunjukkan pada **Gambar 4.8** yakni dilakukan proses pembaruan data, berdasarkan nilai kesalahan yang telah diketahui sebelumnya, lebih tepatnya pembaruan data terjadi pada nilai bobot, baik itu bobot v maupun bobot w , dengan cara menjumlahkan bobot sebelumnya dengan masing-masing nilai faktor kesalahan kedua bobot tersebut.

Setelah keseluruhan proses dijalankan dengan baik, maka tahap selanjutnya yakni dengan melakukan proses *denormalization*, yang bertujuan agar pengguna dapat mengetahui bentuk data yang dihasilkan dari proses peramalan tersebut, dan dapat digunakan sebagai media untuk menentukan tingkatan akurasi dari proses peramalan tersebut. Proses *denormalization* melibatkan dataset yang belum dilakukan *normalization* dan hasil peramalan, dengan bentuk perhitungan seperti yang dijabarkan pada Persamaan 2.14.

4.3 Manualisasi Sistem

Dalam manualisasi sistem menjelaskan bagaimana metode ANN-BP dan algoritme genetika diproses secara manual, dengan melakukan perhitungan manual juga, sehingga memudahkan peneliti untuk mengetahui cara kerja sistem yang akan dibangun. Dengan mengacu pada diagram alir yang telah dibuat sebelumnya, hal pertama yang perlu diperhatikan ialah inialisasi data dengan menentukan data latih yang telah dilakukan *normalization* data dengan menggunakan Persamaan 2.13, seperti yang dijabarkan pada **Table 4.1**, data uji

yang didapat dari sebagian kecil dari dataset yang juga telah dilakukan pemrosesan *normalization* juga dengan menggunakan Persamaan 2.13, yang direpresentasikan pada **Tabel 4.2**. Selain kedua data tersebut, penentuan jumlah populasi juga ditentukan dengan panjang data sesuai dengan jumlah sinyal atau bobot (w).

Tabel 4. 1 Normalization Data Latih

No	x1	x2	x3	x4	x5	x6	Y
1	0.129442	0.132972	0.180374	0.166492	0.189319	0.176569	0.218727
2	0.132972	0.180374	0.166492	0.189319	0.176569	0.218727	0.178351
3	0.180374	0.166492	0.189319	0.176569	0.218727	0.178351	0.184555
4	0.166492	0.189319	0.176569	0.218727	0.178351	0.184555	0.213825
5	0.189319	0.176569	0.218727	0.178351	0.184555	0.213825	0.191239
6	0.176569	0.218727	0.178351	0.184555	0.213825	0.191239	0.190656
7	0.218727	0.178351	0.184555	0.213825	0.191239	0.190656	0.202515
8	0.178351	0.184555	0.213825	0.191239	0.190656	0.202515	0.1
9	0.184555	0.213825	0.191239	0.190656	0.202515	0.1	0.18363
10	0.213825	0.191239	0.190656	0.202515	0.1	0.18363	0.21811
...
90	0.41270	0.41973	0.49357	0.67655	0.62546	0.64504	0.69451
91	0.41973	0.49357	0.67655	0.62546	0.64504	0.69451	0.65629
92	0.49357	0.67655	0.62546	0.64504	0.69451	0.65629	0.74940
93	0.67655	0.62546	0.64504	0.69451	0.65629	0.74940	0.72920
94	0.62546	0.64504	0.69451	0.65629	0.74940	0.72920	0.66120
95	0.64504	0.69451	0.65629	0.74940	0.72920	0.66120	0.83760
96	0.69451	0.65629	0.74940	0.72920	0.66120	0.83760	0.77992
97	0.65629	0.74940	0.72920	0.66120	0.83760	0.77992	0.84704
98	0.74940	0.72920	0.66120	0.83760	0.77992	0.84704	0.90000
99	0.72920	0.66120	0.83760	0.77992	0.84704	0.90000	0.81671
100	0.66120	0.83760	0.77992	0.84704	0.90000	0.81671	0.86016

Tabel 4. 2 Normalization Data Uji

No.	x1	x2	x3	x4	x5	x6	y
1	0.22549	0.48588	0.30861	0.27095	0.35786	0.34960	0.43536
2	0.48588	0.30861	0.27095	0.35786	0.34960	0.43536	0.36958
3	0.30861	0.27095	0.35786	0.34960	0.43536	0.36958	0.65201
4	0.27095	0.35786	0.34960	0.43536	0.36958	0.65201	0.51412
5	0.35786	0.34960	0.43536	0.36958	0.65201	0.51412	0.1
6	0.34960	0.43536	0.36958	0.65201	0.51412	0.1	0.65431
7	0.43536	0.36958	0.65201	0.51412	0.1	0.65431	0.57761
8	0.36958	0.65201	0.51412	0.1	0.65431	0.57761	0.83513
9	0.65201	0.51412	0.1	0.65431	0.57761	0.83513	0.48817
10	0.51412	0.1	0.65431	0.57761	0.83513	0.48817	0.9

Selain memproses nilai dari dataset yang digunakan sebagai data latih dan data uji, penentuan populasi juga merupakan bagian terpenting dalam proses inialisasi, khususnya yang berjalan pada algoritme genetika. Dengan jumlah gen sebanyak lima diterapkan pada penelitian ini, dan sudah termasuk unit bias, kemudian untuk populasinya sendiri berjumlah sepuluh. Nilai gen pada tiap



kromosom memiliki nilai mendekati nol (0), lebih tepatnya antara -1 sampai dengan 1, sehingga memudahkan dalam melakukan perhitungan dan sesuai dengan pola data pada bobot (w) yang akan dihibridisasi dengan menggunakan algoritme genetika. Dengan hasil perhitungan manual seperti pada **Tabel 4.3**.

Tabel 4. 3 Pola Populasi yang Dibentuk

P	Bias				
	$x1$	$x2$	$x3$	$x4$	$x5$
$P1$	-0.32348	-0.01441	-0.39758	0.83157	-0.77692
$P2$	-0.41981	-0.29217	0.06727	0.82298	-0.79821
$P3$	0.47474	-0.80182	0.80705	0.00026	0.87883
$P4$	0.84861	-0.85226	0.69380	0.05919	-0.56273
$P5$	-0.09971	-0.22120	-0.61472	0.70646	0.54296
$P6$	0.92294	-0.09116	0.36379	0.33867	-0.48881
$P7$	-0.52064	0.13111	0.18729	-0.65601	0.58907
$P8$	0.98031	-0.79643	0.05427	-0.69008	0.17815
$P9$	-0.20112	0.62366	-0.36016	-0.49203	-0.06812
$P10$	0.52171	-0.06609	-0.57873	-0.16142	-0.05831

Tahapan selanjutnya setelah dilakukan proses inialisasi, dilakukan proses manual reproduksi atau pencarian solusi berdasarkan nilai populasi dan *crossover rate* ataupun *mutation rate*. Dengan membandingkan dengan beberapa penelitian yang sudah ada, bentuk representasi kromosom *real-code* ini, metode kawin silang atau *crossover* yang tepat, yakni dengan menggunakan *extended intermediate crossover* dan dengan menggunakan *random mutation* sebagai proses penentuan solusi melalui proses mutasi (Mahmudy, 2015), dengan pendekatan yang serupa juga dilakukan oleh Rahardiani, Mahmudy, & Indriati (2018) dalam penelitian mereka yang menggunakan algoritme genetika sebagai algoritma hibridisasi pada ANN-BP itu sendiri dengan hasil yang memuaskan. Proses perhitungan *extended intermediate crossover*, dapat dilakukan dengan Persamaan 4.1 dan 4.2.

$$C_1 = P_1 + \alpha(P_2 - P_1) \tag{4.1}$$

$$C_2 = P_2 + \alpha(P_1 - P_2) \tag{4.2}$$

Keterangan :

- C_i = panjang solusi *offspring* atau kromosom anak
- P_i = panjang solusi induk
- α = Angka acak sangat kecil antara [-0,25 , 1,25]

Seperti yang terlihat pada Persamaan 4.5 dan 4.6, proses penentuan solusi dari *extended intermediate crossover* juga menggunakan dua induk yang dipilih secara acak untuk melakukan proses *crossover*. Disini dimisalkan kromosom yang



terpilih secara random adalah $P3$ dan $P7$ sebagai *parent*, dan nilai alpha yang digunakan adalah sebesar 0,343222.

Tabel 4. 4 Penerapan Extended Intermediate Crossover

Individu	$x1$	$x2$	$x3$	$x4$	$x5$
$P3$	0.262285	0.759386	0.216506	0.587569	0.243935
$P7$	0.568418	0.763613	0.310457	0.608659	0.735254

$C1$	0.367356	0.760837	0.248752	0.594808	0.412567
$C2$	0.463347	0.762162	0.278211	0.601421	0.566623

Tahap selanjutnya dari serangkaian proses reproduksi yakni melakukan proses mutasi, lebih tepatnya *random mutation*. Dengan memilih kromosom induk secara acak, dan memilih gen didalam kromosom tersebut secara acak pula, gen tersebut yang digunakan sebagai media mutasi atau pengkodean nilai gen, dengan cara mengurangi atau menambah nilai pada gen tersebut (Mahmudy, 2015), dengan penerapan Persamaan 4.3.

$$x_i = x_i(t-1) + r(max_i - min_i) \tag{4.3}$$

Keterangan :

x_i = Nilai solusi atau gen baru

$x_i(t-1)$ = Nilai solusi atau gen lama

r = jarak data dengan nilai acak sangat kecil antara [-0,1 , 0,1]

max_i = Nilai gen paling tinggi dari x_i

min_i = Nilai gen paling rendah dari x_i

Dalam proses *random mutation* ini, menghasilkan nilai maksimal sebesar 0.9, dan nilai minimal sebesar 0.2, kemudian untuk jarak sebesar 0.05, maka proses reproduksinya menghasilkan nilai seperti yang dijelaskan pada **Tabel 4.5**.

Tabel 4. 5. Penerapan *Random Mutation*

Individu	$x1$	$x2$	$x3$	$x4$	$x5$
$P4$	93.86-E2	57.27-E2	22.64-E2	23.08-E2	43.57-E2

$C3$	93.86-E2	60.83-E2	22.64-E2	23.08-E2	43.57-E2
------	----------	----------	----------	----------	----------

BAB 3

METODOLOGI PENELITIAN

Dalam metodologi penelitian, menjelaskan teknik penerapan tiap metode atau atribut yang digunakan dan berjalan secara runtut berdasarkan data tingkat pengguna kereta api di Indonesia. Dengan menerapkan kaidah-kaidah yang diperlukan dalam pemrosesan sistem berdasarkan kasus yang diangkat sebagai topik pembahasan, yang meliputi tipe penelitian dan rancang bangun penelitian.

3.1 Tipe Penelitian

Pada penelitian peramalan penumpang kereta api ini, peneliti menggunakan tipe penelitian *non-implimentatif* deskriptif, dimana dalam penelitian ini lebih berfokus untuk penerapan dalam segi keilmuan pada kasus tertentu yang menghasilkan hasil analisis dari kasus yang diangkat untuk mencari solusi yang terbaik, dengan proses pengolahan data kuantitatif.

3.2 Rancang Bangun Penelitian

Dalam rancang bangun penelitian berfokus pada tahapan pengolahan sistem secara teknis, yang terdiri dari beberapa tahapan.

3.2.1 Metode Penelitian

Dalam metode penelitian secara umum ini, peneliti menggunakan metode eksperimental, sehingga untuk menyelesaikan permasalahan ini peneliti melakukan pengujian variabel-variabel tertentu untuk mengetahui permasalahannya, dan berdasarkan data permasalahan yang dihimpun maka dapat menentukan langkah yang dapat diambil dalam pemecahan permasalahan tersebut.

3.2.2 Subjek Penelitian

Penelitian yang dihimpun dari data sekunder ini diperoleh dari lembaga yang kredibel dalam menyajikan data statistik di Indonesia, hal ini sangat sesuai dengan data yang dibutuhkan oleh peneliti yang memerlukan data yang dijamin kebenarannya dan dikelola langsung oleh pemerintah sehingga memiliki korelasi dengan objek permasalahan yang juga dikelola oleh pemerintah.

3.2.3 Lokasi Penelitian

Pada penelitian ini dilakukan secara mandiri dan juga dengan memanfaatkan laboratorium komputasi cerdas Fakultas Ilmu Komputer Universitas Brawijaya untuk memproses penerapan penggunaan metode ANN-BP dan algoritme genetika.

3.2.4 Metode Pengumpulan Data

Data yang digunakan merupakan data dalam format .csv pada *file* excel berupa data *timeseries* dalam bentuk data bulanan, yang didapat dari BPS terkait jumlah penumpang kereta api di Indonesia, dengan rentang waktu januari 2006 sampai dengan juni 2017, kemudian diolah berdasarkan pola yang diinginkan guna mempermudah proses penelitian, dengan dilakukannya proses penentuan data latih dan data uji sebagai media peramalan tersebut.

Data latih ditentukan berdasarkan pola tertentu, pada penelitian ini menggunakan pola $x_1, x_2, x_3, x_4, x_5, x_6$, dan target (y) seperti yang digambarkan dalam **Tabel 4.1** sehingga pada x_1 akan menampung data awal mulai bulan januari 2006 dan seterusnya, x_6 akan menampung bulan ke-6 (data bulan juni 2006), sedangkan target (y) menampung bulan selanjutnya, yakni bulan juli 2006. Untuk pola kedua dimulai pada x_1 diisi dengan data pada bulan februari 2006 dan seterusnya, sampai nilai target (y) berisi data bulan agustus 2006, begitu pula pada pola-pola selanjutnya hingga membentuk panjang pola yang telah ditentukan peneliti, yakni peneliti menggunakan 100 pola pada data latih. Hal yang sama juga dilakukan pada data uji seperti yang digambarkan pada **Tabel 4.2**, namun yang membedakan data latih dan data uji yakni pada penggunaan data berada pada data akhir yang digunakan sebagai data uji, disini peneliti menggunakan 10 pola data uji. Sehingga data latih dan data uji tersebut akan dilakukan perhitungan sesuai fungsinya masing-masing. Dengan penggunaan data latih sebagai data untuk penentuan tren pada metode, dan tren tersebut akan digunakan sebagai acuan untuk melakukan pengujian peramalan dengan menggunakan data uji, sehingga hasil pengujian tersebut dapat digunakan sebagai rujukan untuk melakukan peramalan informasi.

3.2.5 Metode Analisis Data

Analisis dilakukan agar mengetahui pengaruh dari penggunaan variabel yang digunakan dan nilai dari variabel yang digunakan, sehingga didapati komposisi yang baik digunakan dalam melakukan peramalan pengguna kereta api di Indonesia dengan modifikasi terhadap variabel yang berpengaruh, yakni :

1. Dengan memodifikasi ukuran populasi untuk menentukan luas area pencarian solusi.
2. Melakukan modifikasi pada jumlah generasi untuk memperbesar kemungkinan solusi yang dihasilkan.
3. Melakukan modifikasi pada komposisi Cr dan Mr untuk mengetahui model yang dapat menghasilkan solusi yang baik.
4. Modifikasi pada iterasi yang juga merupakan cara untuk memperluas area pencarian solusi, namun yang bekerja pada metode ANN-BP.
5. Modifikasi pada jangkauan pencarian solusi dengan menentukan nilai alpha.

Selain kelima model yang dilakukan analisa untuk menentukan variabel bagi metode peramalan ANN-BP dan algoritme genetika, adapun variabel lain yang turut mempengaruhi penilaian peramalan metode ANN-BP dan algoritme genetika, yakni dengan melakukan pengujian untuk membandingkan hasil antara metode *backpeopagation* yang telah dilakukan optimasi nilai bobot (w) dengan menggunakan algoritme genetika dan hasil dari penggunaan metode ANN-BP saja.

3.2.6 Peralatan Pendukung

Dalam melakukan penerapan sistem, peneliti dibantu dengan peralatan pendukung yang terdiri dari :

1. Kebutuhan perangkat lunak:
 - a. Windows 10 dengan sistem operasi 64 bit.
 - b. NetBeans sebagai IDE (bahasa pemrograman Java)
2. Kebutuhan perangkat keras:
 - a. Laptop dengan kinerja prosesor sebesar 2.71 GHz
 - b. Akses memori sebesar 4 GB
 - c. Ruang penyimpanan sebesar 500 GB



BAB 4

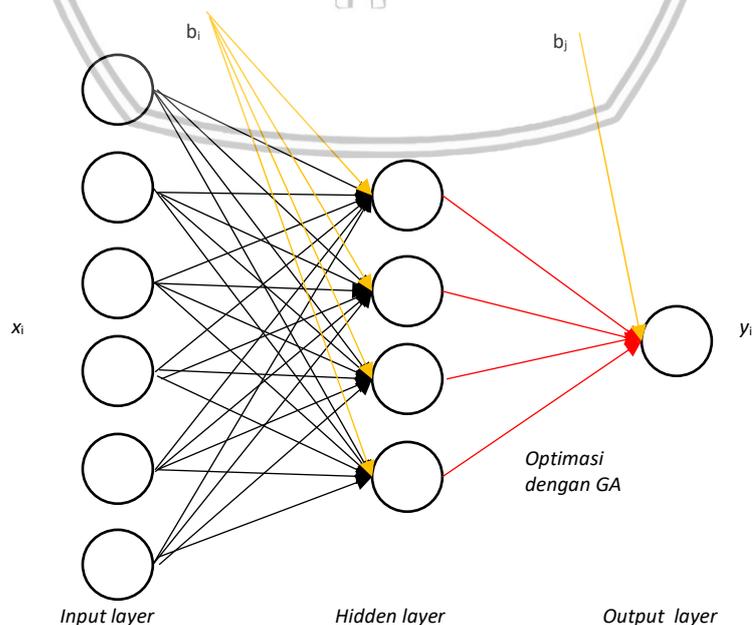
PERANCANGAN DAN IMPLEMENTASI

Pembahasan pada bab perancangan dan implementasi sistem, merupakan tahapan penerapan metode dan algoritme baik dari segi perancangan dan dilanjutkan dengan proses implementasinya berdasarkan teori yang telah dikemukakan pada bab-bab sebelumnya, dengan penjelasan berupa deskripsi terkait sistem yang dibuat, diagram alir, bentuk manualisasi, perancangan sistem dan hasil keluaran yang diharapkan beserta dengan implementasinya.

4.1 Deskripsi Umum Sistem

Sistem peramalan jumlah penumpang kereta api Indonesia ini digunakan untuk mendapatkan suatu olahan informasi dari peramalan jumlah penumpang kereta api Indonesia berdasarkan pada tahun-tahun sebelumnya dalam bentuk data perbulan, diurutkan secara historis dan berbentuk data *time series* dari bulan januari 2006 sampai dengan bulan juli 2017. Dengan pemrosesan yang dibentuk dari arsitektur ANN-BP berupa data unit *layer input* sejumlah enam dan ditambah satu unit bias, data pada *hidden layer* sejumlah empat unit dan ditambah satu unit bias dengan ketentuan jumlah *hidden layer* berasal dari jumlah rata-rata antar *input layer* dengan *output layer* (Heaton, 2008), sedangkan pada *layer output* berjumlah satu.

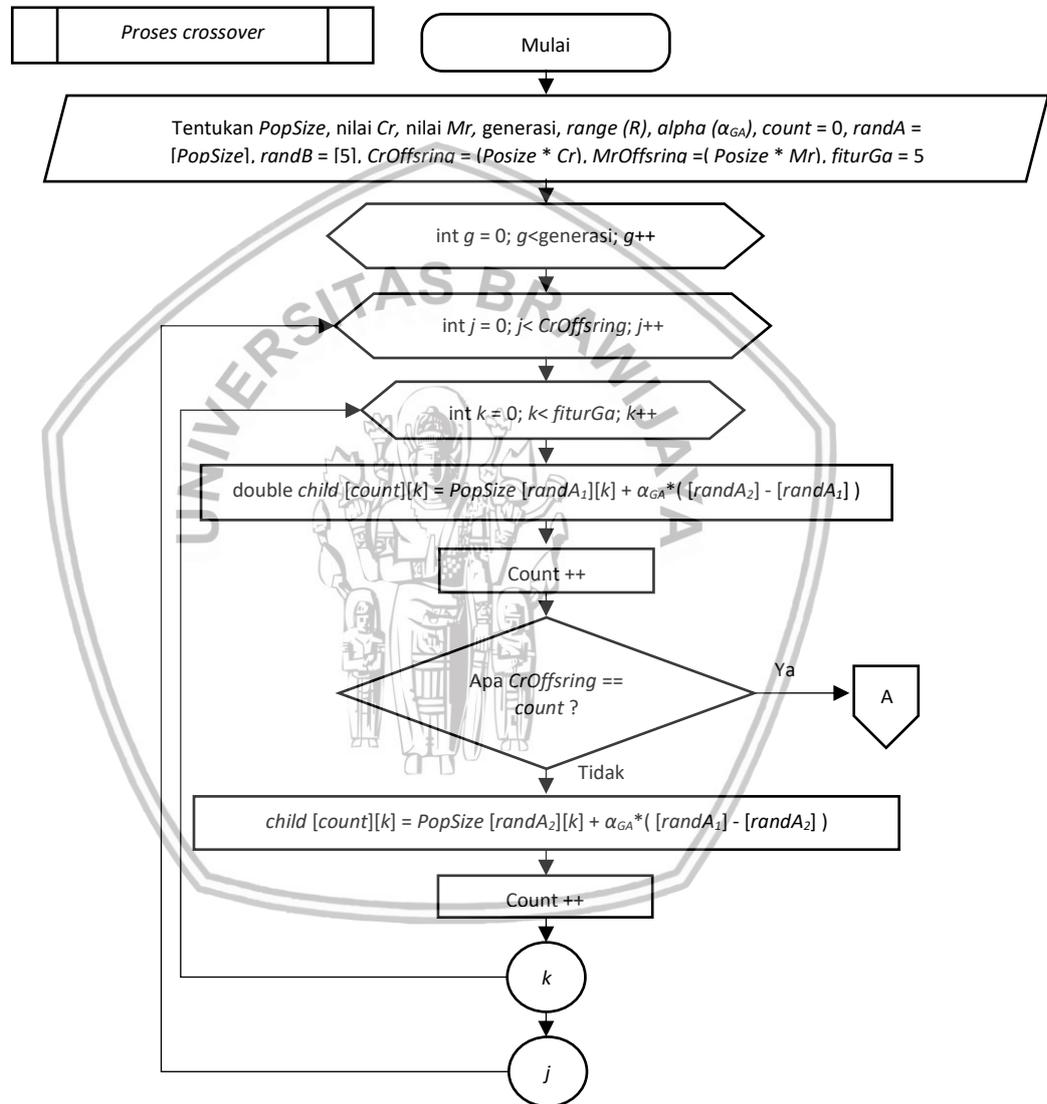
Optimasi yang dilakukan oleh algoritme genetika berada tepat antara *hidden layer* dan *input layer*, berupa bobot (w), seperti halnya yang juga pernah dilakukan oleh Wiharto, Palgunadi, & Nugroho (2013), dengan penggunaan bobot (w) sebagai media hibridisasi untuk mengurangi jangka waktu eksekusi. Dengan arsitektur ANN-BP dan algoritme genetika dijabarkan pada **Gambar 4.1**.



Gambar 4. 1 Arsitektur Jaringan Saraf Tiruan *Backpropagation*

4.2 Rancang Bangun Sistem

Pada perancangan bangun sistem ini menjelaskan secara sistematis proses pencarian solusi dengan bentuk diagram alir, dengan berfokus pada cara penentuan variabel-variabel utama dalam peramalan jumlah penumpang kereta api Indonesia dengan menggunakan ANN-BP dan algoritme genetika, sehingga memudahkan dalam melakukan pelacakan proses dalam representasi dari sistem yang dibuat, dengan digambarkan pada serangkaian diagram alir sebagai berikut :

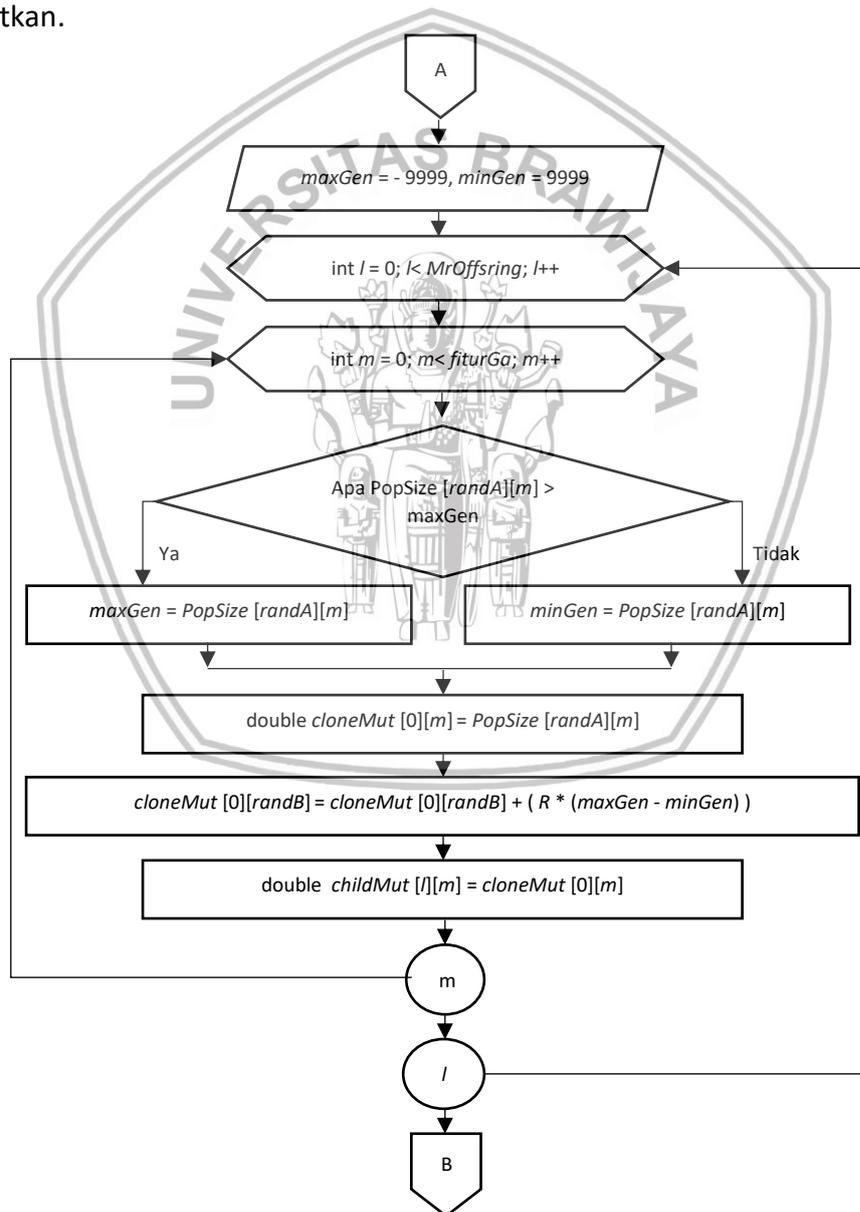


Gambar 4. 2 Diagram Alir Crossover

Pada **Gambar 4.2** merupakan bagian dari proses dalam menentukan solusi yang terjadi dalam proses *crossover*, dimana pada tahap awal dilakukan inisialisasi nilai variabel-variabel penentu. Selanjutnya dalam proses *crossover* sistem akan menentukan jumlah *offspring* yang dapat dihasilkan dalam proses *crossover*,

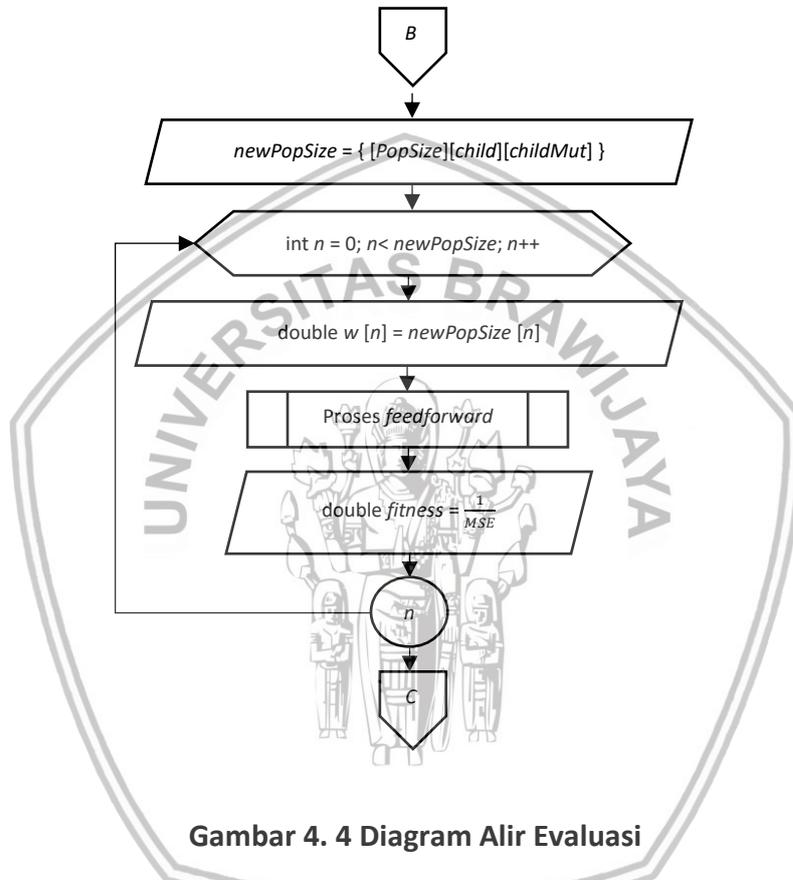
dengan melakukan perkalian jumlah *pop size* dan nilai *Cr*, selanjutnya sistem akan menghasilkan *offspring* baru dan dilakukan pengecekan, apakah dalam satu perulangan tersebut menghasilkan dua atau satu *offspring*, kemudian proses dilanjutkan dengan melakukan perulangan berdasarkan indeks *k* sehingga menghasilkan *offspring* yang lain. Sampai pada kondisi berhenti, yakni telah mencapai indeks *j*.

Pada **Gambar 4.2** juga menunjukkan bahwa dilakukan pengecekan iterasi dalam proses algoritme genetika, sehingga jika setiap perulangan yang dilakukan kembali pada proses *crossover*, maka sistem akan menanyakan apakah iterasi sudah terpenuhi, jika telah mencapai indeks *g* maka proses akan dilanjutkan pada proses ANN-BP, jika tidak maka proses regenerasi algoritme akan kembali dilanjutkan.



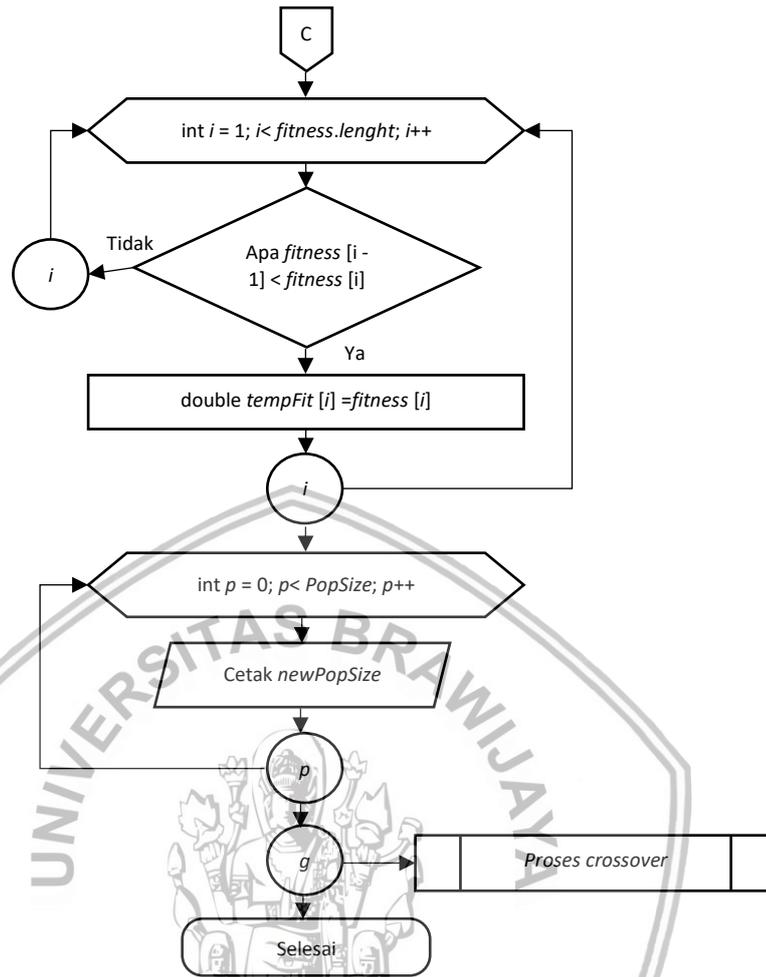
Gambar 4. 3 Diagram Alir Mutasi

Selanjutnya pada **Gambar 4.3**, menunjukkan proses penentuan *offspring* yang dilakukan berdasarkan *Mr* atau *Mutation rate*, dengan menentukan jumlah *offspring* berdasarkan perkalian antara *Pop size* dan *Mr*, selanjutnya dipilih baris solusi secara acak dan unit dari baris solusi secara acak pula, yang selanjutnya pada unit solusi dilakukan perhitungan penentuan *offspring* yakni *cloneMut* dan begitu pula seterusnya dilakukan perulangan sampai mencapai nilai indeks *l*.



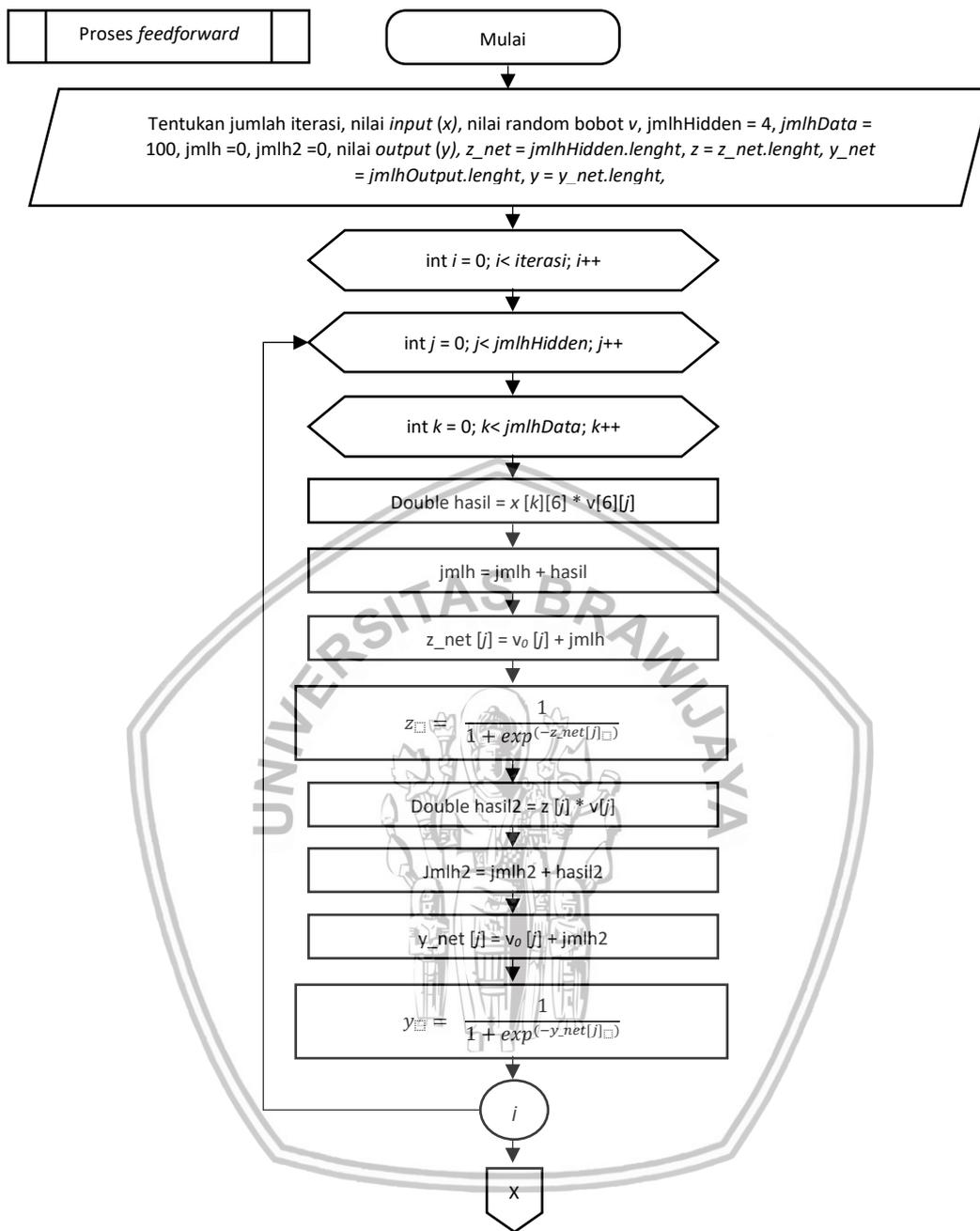
Gambar 4. 4 Diagram Alir Evaluasi

Setelah dilakukan proses reproduksi, tahap selanjutnya yakni dilakukan proses evaluasi, seperti pada **Gambar 4.4**. Pada gambar tersebut menyatakan bahwa semua populasi, baik induk maupun *offspring* digabungkan sehingga dapat dilakukan perhitungan nilai *fitness* pada tiap individu yang ada dalam satu iterasi.



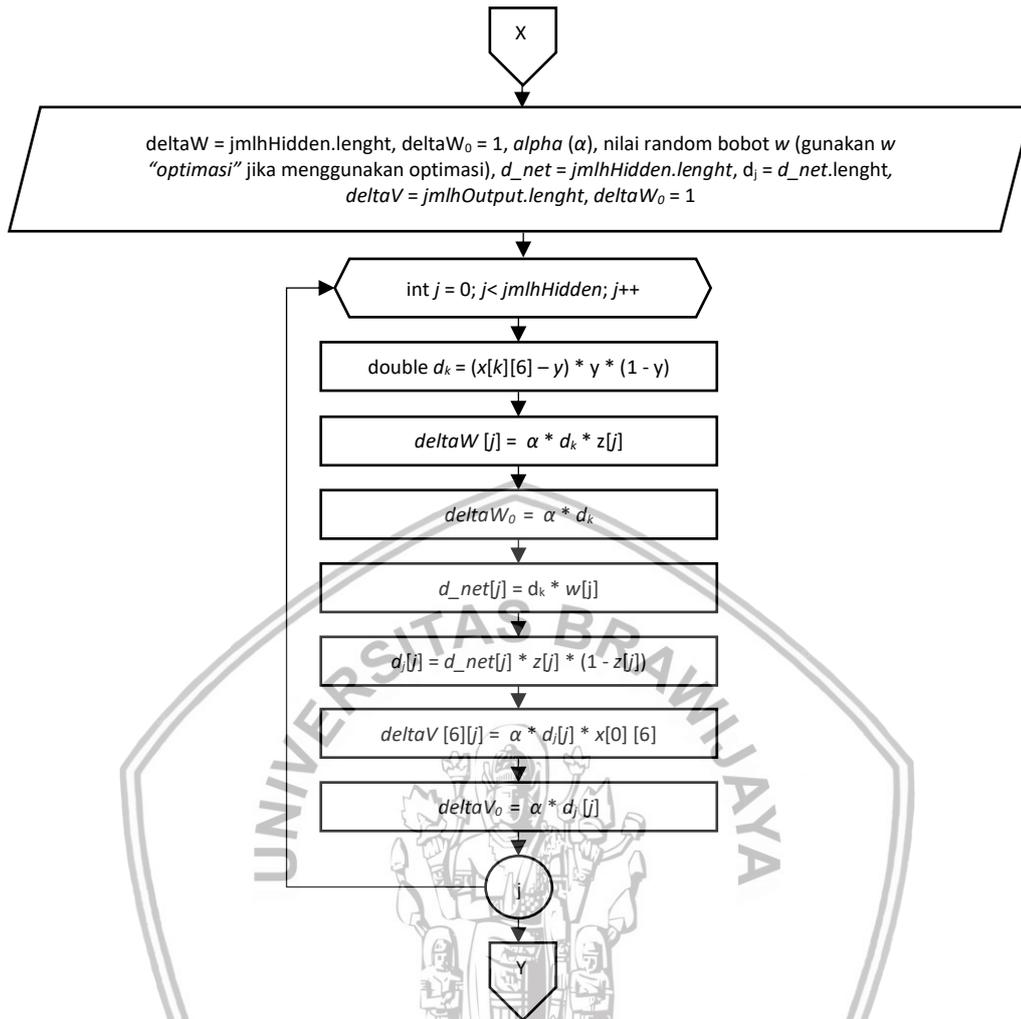
Gambar 4. 5 Diagram Alir Seleksi

Selanjutnya, setelah ditentukan nilai *fitness* pada semua populasi, maka tahapan berikutnya adalah melakukan seleksi pada populasi yang memiliki nilai *fitness* terbaik sejumlah populasi induk sebelumnya, seperti pada **Gambar 4.5** menunjukkan bahwa tiap populasi yang baru tersebut diurutkan secara *descending* kemudian diambil sejumlah populasi induk sebelumnya, dan diproses ke iterasi selanjutnya untuk menyeleksi generasi baru lagi, sampai mencapai nilai konvergensi. Pada tahapan ini juga menunjukkan langkah lanjutan yang diproses pada proses *crossover*, dengan mengembalikan indeks perulangan sejumlah iterasi berdasarkan generasi yang telah ditentukan pada proses *crossover*.



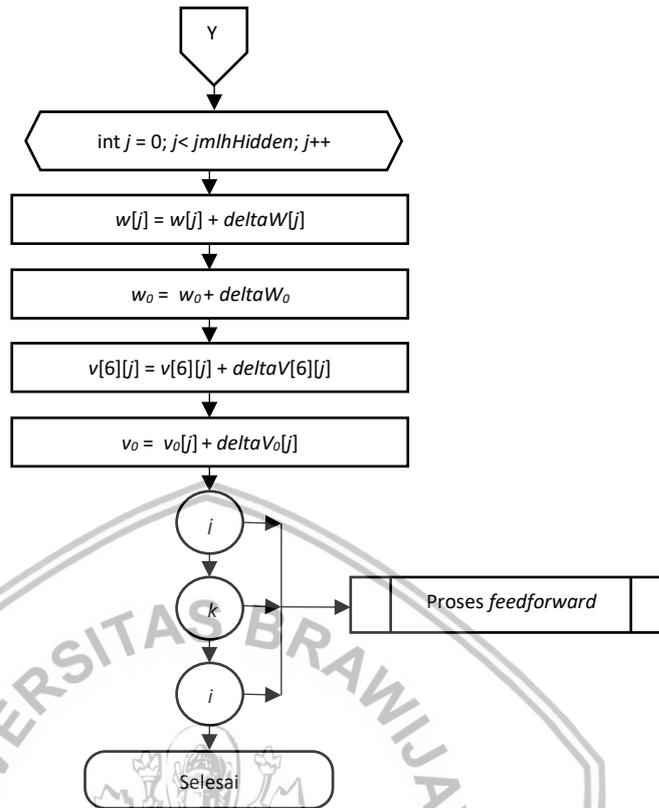
Gambar 4. 6 Diagram Alir Proses Feedforward

Pada **Gambar 4.6** yang merupakan bagian awal dari fase jaringan saraf tiruan *backpropagation* menjelaskan, bahwa pada fase ini menghasilkan dua hasil dari empat persamaan, yakni nilai *hidden layer* (z) yang dihasilkan dari penentuan nilai z_net dan diaktifkan dengan menggunakan fungsi aktivasi seperti yang ditunjukkan pada Persamaan 2.1 juga 2.2, dan pada fase ini juga menghasilkan nilai *output layer* (y), hasil dari aktivasi dari nilai y_net seperti yang ditunjukkan dari Persamaan 2.3 dan 2.4. Yang masing-masing hasil tersebut dilakukan perulangan sebanyak panjang data yang mereka miliki.



Gambar 4. 7 Diagram Alir Proses *Backpropagation*

Pada fase kedua dalam proses ANN-BP seperti yang terlihat pada **Gambar 4.7** menunjukkan proses utama yakni proses propagasi mundur, dengan tujuan untuk membawa informasi perbaikan dari proses *feedforward* yang telah dilakukan sebelumnya. Pada tahap ini menghasilkan informasi data selisih kesalahan yang terjadi pada *output layer* dengan notasi (d_k), yang selanjutnya dilakukan pencarian nilai faktor kesalahan yang terjadi pada bobot dari *output* menuju *hidden layer* ($\text{delta } w_{kj}$) sepanjang data bobot (w), selanjutnya dilakukan pencarian nilai kesalahan yang terjadi pada hidden layer dengan notadi (d_j) sepanjang data *hidden layer*, kemudian dilakukan juga perhitungan nilai faktor kesalahan yang terjadi pada bobot dari *hidden* menuju *input layer* ($\text{delta } v_{ji}$) sepanjang data bobot input layer (v).



Gambar 4. 8 Diagram Alir Proses Weight Update

Pada proses terakhir dari fase ANN-BP, seperti yang ditunjukkan pada **Gambar 4.8** yakni dilakukan proses pembaruan data, berdasarkan nilai kesalahan yang telah diketahui sebelumnya, lebih tepatnya pembaruan data terjadi pada nilai bobot, baik itu bobot v maupun bobot w , dengan cara menjumlahkan bobot sebelumnya dengan masing-masing nilai faktor kesalahan kedua bobot tersebut.

Setelah keseluruhan proses dijalankan dengan baik, maka tahap selanjutnya yakni dengan melakukan proses *denormalization*, yang bertujuan agar pengguna dapat mengetahui bentuk data yang dihasilkan dari proses peramalan tersebut, dan dapat digunakan sebagai media untuk menentukan tingkatan akurasi dari proses peramalan tersebut. Proses *denormalization* melibatkan dataset yang belum dilakukan *normalization* dan hasil peramalan, dengan bentuk perhitungan seperti yang dijabarkan pada Persamaan 2.14.

4.3 Manualisasi Sistem

Dalam manualisasi sistem menjelaskan bagaimana metode ANN-BP dan algoritme genetika diproses secara manual, dengan melakukan perhitungan manual juga, sehingga memudahkan peneliti untuk mengetahui cara kerja sistem yang akan dibangun. Dengan mengacu pada diagram alir yang telah dibuat sebelumnya, hal pertama yang perlu diperhatikan ialah inialisasi data dengan menentukan data latih yang telah dilakukan *normalization* data dengan menggunakan Persamaan 2.13, seperti yang dijabarkan pada **Table 4.1**, data uji

yang didapat dari sebagian kecil dari dataset yang juga telah dilakukan pemrosesan *normalization* juga dengan menggunakan Persamaan 2.13, yang direpresentasikan pada **Tabel 4.2**. Selain kedua data tersebut, penentuan jumlah populasi juga ditentukan dengan panjang data sesuai dengan jumlah sinyal atau bobot (w).

Tabel 4. 1 Normalization Data Latih

No	x1	x2	x3	x4	x5	x6	Y
1	0.129442	0.132972	0.180374	0.166492	0.189319	0.176569	0.218727
2	0.132972	0.180374	0.166492	0.189319	0.176569	0.218727	0.178351
3	0.180374	0.166492	0.189319	0.176569	0.218727	0.178351	0.184555
4	0.166492	0.189319	0.176569	0.218727	0.178351	0.184555	0.213825
5	0.189319	0.176569	0.218727	0.178351	0.184555	0.213825	0.191239
6	0.176569	0.218727	0.178351	0.184555	0.213825	0.191239	0.190656
7	0.218727	0.178351	0.184555	0.213825	0.191239	0.190656	0.202515
8	0.178351	0.184555	0.213825	0.191239	0.190656	0.202515	0.1
9	0.184555	0.213825	0.191239	0.190656	0.202515	0.1	0.18363
10	0.213825	0.191239	0.190656	0.202515	0.1	0.18363	0.21811
...
90	0.41270	0.41973	0.49357	0.67655	0.62546	0.64504	0.69451
91	0.41973	0.49357	0.67655	0.62546	0.64504	0.69451	0.65629
92	0.49357	0.67655	0.62546	0.64504	0.69451	0.65629	0.74940
93	0.67655	0.62546	0.64504	0.69451	0.65629	0.74940	0.72920
94	0.62546	0.64504	0.69451	0.65629	0.74940	0.72920	0.66120
95	0.64504	0.69451	0.65629	0.74940	0.72920	0.66120	0.83760
96	0.69451	0.65629	0.74940	0.72920	0.66120	0.83760	0.77992
97	0.65629	0.74940	0.72920	0.66120	0.83760	0.77992	0.84704
98	0.74940	0.72920	0.66120	0.83760	0.77992	0.84704	0.90000
99	0.72920	0.66120	0.83760	0.77992	0.84704	0.90000	0.81671
100	0.66120	0.83760	0.77992	0.84704	0.90000	0.81671	0.86016

Tabel 4. 2 Normalization Data Uji

No.	x1	x2	x3	x4	x5	x6	y
1	0.22549	0.48588	0.30861	0.27095	0.35786	0.34960	0.43536
2	0.48588	0.30861	0.27095	0.35786	0.34960	0.43536	0.36958
3	0.30861	0.27095	0.35786	0.34960	0.43536	0.36958	0.65201
4	0.27095	0.35786	0.34960	0.43536	0.36958	0.65201	0.51412
5	0.35786	0.34960	0.43536	0.36958	0.65201	0.51412	0.1
6	0.34960	0.43536	0.36958	0.65201	0.51412	0.1	0.65431
7	0.43536	0.36958	0.65201	0.51412	0.1	0.65431	0.57761
8	0.36958	0.65201	0.51412	0.1	0.65431	0.57761	0.83513
9	0.65201	0.51412	0.1	0.65431	0.57761	0.83513	0.48817
10	0.51412	0.1	0.65431	0.57761	0.83513	0.48817	0.9

Selain memproses nilai dari dataset yang digunakan sebagai data latih dan data uji, penentuan populasi juga merupakan bagian terpenting dalam proses inialisasi, khususnya yang berjalan pada algoritme genetika. Dengan jumlah gen sebanyak lima diterapkan pada penelitian ini, dan sudah termasuk unit bias, kemudian untuk populasinya sendiri berjumlah sepuluh. Nilai gen pada tiap



kromosom memiliki nilai mendekati nol (0), lebih tepatnya antara -1 sampai dengan 1, sehingga memudahkan dalam melakukan perhitungan dan sesuai dengan pola data pada bobot (w) yang akan dihibridisasi dengan menggunakan algoritme genetika. Dengan hasil perhitungan manual seperti pada **Tabel 4.3**.

Tabel 4. 3 Pola Populasi yang Dibentuk

P	Bias				
	$x1$	$x2$	$x3$	$x4$	$x5$
$P1$	-0.32348	-0.01441	-0.39758	0.83157	-0.77692
$P2$	-0.41981	-0.29217	0.06727	0.82298	-0.79821
$P3$	0.47474	-0.80182	0.80705	0.00026	0.87883
$P4$	0.84861	-0.85226	0.69380	0.05919	-0.56273
$P5$	-0.09971	-0.22120	-0.61472	0.70646	0.54296
$P6$	0.92294	-0.09116	0.36379	0.33867	-0.48881
$P7$	-0.52064	0.13111	0.18729	-0.65601	0.58907
$P8$	0.98031	-0.79643	0.05427	-0.69008	0.17815
$P9$	-0.20112	0.62366	-0.36016	-0.49203	-0.06812
$P10$	0.52171	-0.06609	-0.57873	-0.16142	-0.05831

Tahapan selanjutnya setelah dilakukan proses inialisasi, dilakukan proses manual reproduksi atau pencarian solusi berdasarkan nilai populasi dan *crossover rate* ataupun *mutation rate*. Dengan membandingkan dengan beberapa penelitian yang sudah ada, bentuk representasi kromosom *real-code* ini, metode kawin silang atau *crossover* yang tepat, yakni dengan menggunakan *extended intermediate crossover* dan dengan menggunakan *random mutation* sebagai proses penentuan solusi melalui proses mutasi (Mahmudy, 2015), dengan pendekatan yang serupa juga dilakukan oleh Rahardiani, Mahmudy, & Indriati (2018) dalam penelitian mereka yang menggunakan algoritme genetika sebagai algoritma hibridisasi pada ANN-BP itu sendiri dengan hasil yang memuaskan. Proses perhitungan *extended intermediate crossover*, dapat dilakukan dengan Persamaan 4.1 dan 4.2.

$$C_1 = P_1 + \alpha(P_2 - P_1) \tag{4.1}$$

$$C_2 = P_2 + \alpha(P_1 - P_2) \tag{4.2}$$

Keterangan :

- C_i = panjang solusi *offspring* atau kromosom anak
- P_i = panjang solusi induk
- α = Angka acak sangat kecil antara [-0,25 , 1,25]

Seperti yang terlihat pada Persamaan 4.5 dan 4.6, proses penentuan solusi dari *extended intermediate crossover* juga menggunakan dua induk yang dipilih secara acak untuk melakukan proses *crossover*. Disini dimisalkan kromosom yang



terpilih secara random adalah *P3* dan *P7* sebagai *parent*, dan nilai alpha yang digunakan adalah sebesar 0,343222.

Tabel 4. 4 Penerapan Extended Intermediate Crossover

Individu	<i>x1</i>	<i>x2</i>	<i>x3</i>	<i>x4</i>	<i>x5</i>
<i>P3</i>	0.262285	0.759386	0.216506	0.587569	0.243935
<i>P7</i>	0.568418	0.763613	0.310457	0.608659	0.735254

<i>C1</i>	0.367356	0.760837	0.248752	0.594808	0.412567
<i>C2</i>	0.463347	0.762162	0.278211	0.601421	0.566623

Tahap selanjutnya dari serangkaian proses reproduksi yakni melakukan proses mutasi, lebih tepatnya *random mutation*. Dengan memilih kromosom induk secara acak, dan memilih gen didalam kromosom tersebut secara acak pula, gen tersebut yang digunakan sebagai media mutasi atau pengkodean nilai gen, dengan cara mengurangi atau menambah nilai pada gen tersebut (Mahmudy, 2015), dengan penerapan Persamaan 4.3.

$$x_i = x_i(t-1) + r(max_i - min_i) \tag{4.3}$$

Keterangan :

x_i = Nilai solusi atau gen baru

$x_i(t-1)$ = Nilai solusi atau gen lama

r = jarak data dengan nilai acak sangat kecil antara [-0,1 , 0,1]

max_i = Nilai gen paling tinggi dari x_i

min_i = Nilai gen paling rendah dari x_i

Dalam proses *random mutation* ini, menghasilkan nilai maksimal sebesar 0.9, dan nilai minimal sebesar 0.2, kemudian untuk jarak sebesar 0.05, maka proses reproduksinya menghasilkan nilai seperti yang dijelaskan pada **Tabel 4.5**.

Tabel 4. 5. Penerapan *Random Mutation*

Individu	<i>x1</i>	<i>x2</i>	<i>x3</i>	<i>x4</i>	<i>x5</i>
<i>P4</i>	93.86-E2	57.27-E2	22.64-E2	23.08-E2	43.57-E2

<i>C3</i>	93.86-E2	60.83-E2	22.64-E2	23.08-E2	43.57-E2
-----------	----------	----------	----------	----------	----------



Selanjutnya setelah mendapatkan ragam solusi baru dalam bentuk keomosom baru dari proses reproduksi, hasil solusi baru tersebut digabungkan dengan kromosom induk yang telah dibangkitkan sebelumnya dan masing-masing dilabeli dengan nilai *fitness* dalam proses evaluasi ini, dengan hasil yang digambarkan pada **Tabel 4.6**. Dengan proses penentuan nilai *fitness* seperti yang digambarkan pada Persamaan 4.4.

$$fitness = \frac{1}{MSE} \tag{4.4}$$

Pemilihan nilai MSE sebagai faktor penentu, dikarenakan nilai MSE merupakan suatu *constrain* dalam metode yang dihibridisasi oleh algoritme genetika. Selain itu beberapa penelitian yang serupa juga menggunakan nilai MSE sebagai faktor penentu dalam menghasilkan nilai *fitness*, diantaranya yakni kajian ilmiah oleh Rahardiani, Mahmudy, & Indriati (2018). Dengan nilai *error* yang berasal dari proses *feedforward* dengan memakai hasil pemilihan kromosom teraik sebagai representasi dari nilai bobot yang akan digunakan dalam peramalan pada metode ANN-BP. Dengan hasil perhitungan seperti terlihat pada **Tabel 4.6**.

Tabel 4. 6 Evaluasi Individu

Individu	x1	x2	x3	x4	x5	fitness
P1	-0.32348	-0.01441	-0.39758	0.83157	-0.77692	807.1186
P2	-0.41981	-0.29217	0.06727	-0.82298	-0.79821	2132.657
P3	0.47474	-0.80182	0.80705	0.00026	0.87883	69.16185
P4	0.84861	-0.85226	0.69380	0.05919	-0.56273	853.92
P5	-0.09971	-0.22120	-0.61472	0.70646	0.54296	386.9753
P6	0.92294	-0.09116	0.36379	0.33867	-0.48881	319.2992
P7	-0.52064	0.13111	0.18729	-0.65601	0.58907	1306.452
P8	0.98031	-0.79643	0.05427	-0.69008	0.17815	1776.798
P9	-0.20112	0.62366	-0.36016	-0.49203	-0.06812	1993.429
P10	0.52171	-0.06609	-0.57873	-0.16142	-0.05831	19662.09
C1	0.65402	-0.51756	0.62975	0.13563	0.33178	103.937
C2	0.74366	-0.37542	0.54110	0.20331	0.05825	138.0095
C3	-0.41981	-0.29217	0.14833	0.82298	-0.79821	3443.217

Setelah pelabelan dengan menggunakan nilai *fitnes* pada masing-masing kromosom telah dilakukan, tahap selanjutnya yakni dengan nilai *fitness* tersebut dilakukan pemilihan kromosom dengan cara *elitism*, yakni dengan cara menyeleksi kromosom dengan nilai *fitness* tertinggi sesuai dengan jumlah populasi awal akan dipilih dan membuang kromosom yang tidak fit. Berdasarkan jenis penentuan solusi yang telah dilakukan sebelumnya yakni pada proses mutasi, menjadi dasar peneliti untuk memilih menggunakan media *elitism*, sebagai media untuk melakukan evaluasi data. Dengan fakta bahwa kedua media ini, yakni proses *random mutation* dan *elitism* memiliki pola pencarian solusi yang saling keterbalikan, sehingga dengan perbedaan itu diharapkan hasilnya akan saling mendukung satu sama lain untuk mendapatkan solusi yang baik dan memperluas



kemungkinannya. Dengan detail representasi hail seleksi ditampilkan pada **Tabel 4.7**.

Tabel 4. 7 Seleksi Kromosom

<i>P</i>	<i>P</i> (baru)	<i>x1</i>	<i>x2</i>	<i>x3</i>	<i>x4</i>	<i>x5</i>	<i>Fitness</i>
P10	P1	0.52171	-0.06609	-0.57873	-0.16142	-0.05831	19662.09
C3	P2	-0.41981	-0.29217	0.14833	0.82298	-0.79821	3443.217
P2	P3	-0.41981	-0.29217	0.06727	0.82298	-0.79821	2132.657
P9	P4	-0.20112	0.62366	-0.36016	-0.49203	-0.06812	1993.429
P8	P5	0.98031	-0.79643	0.05427	-0.69008	0.17815	1776.798
P7	P6	-0.52064	0.13111	0.18729	-0.65601	0.58907	1306.452
P4	P7	0.84861	-0.85226	0.69380	0.05919	-0.56273	853.92
P1	P8	-0.32348	-0.01441	-0.39758	0.83157	-0.77692	807.1186
P5	P9	-0.09971	-0.22120	-0.61472	0.70646	0.54296	386.9753
P6	P10	0.92294	-0.09116	0.36379	0.33867	-0.48881	319.2992
C2	P11	0.74366	-0.37542	0.54110	0.20331	0.05825	138.0095
C1	P12	0.65402	-0.51756	0.62975	0.13563	0.33178	103.937
P3	P13	0.47474	-0.80182	0.80705	0.00026	0.87883	69.16185

Setelah keseluruhan kromosom ditentukan nilai terbaik berdasarkan nilai *fitness*-nya, maka diakhir pencarian akan ditentukan hasil akhir berupa hanya satu solusi yang terpilih dan digunakan sebagai inisialisasi bobot (*w*) pada proses peramalan ANN-BP.

Pada proses manualisasi yang dilakukan ini merupakan gambaran pemrosesan perhitungan algoritme secara manual dengan satu kali perulangan, sehingga solusi yang dihasilkan pada proses manualisasi ini merupakan hanya gambaran dari cara kerja algoritme dalam memproses data. Dengan hasil penentuan sementara bahwa individu pada *P10* merupakan individu yang terpilih sebagai nilai inisialisasi bobot awal (*w*) dalam ANN-BP. Sehingga pada tahapan selanjutnya yakni melakukan proses manualisasi pada ANN-BP untuk melakukan peramalan jumlah penumpang kereta api Indonesia.

Seperti yang telah dijelaskan pada **Gambar 4.6**, tahapan awal dari peramalan dengan menggunakan metode ANN-BP ini adalah tahap *feedforward*, dimana secara garis besar proses ini mencakup proses inisialisasi bobot secara acak, namun bedanya pada penentuan bobot (*w*) akan menggunakan nilai yang telah dihasilkan oleh algoritme genetika yang telah menyesuaikan representasi kromosom bobot (*w*) dari variabel yang ada pada proses ANN-BP. Dengan tahap awal *feedforward* yakni menentukan nilai hidden layer, yang menghasilkan nilai dengan penjabaran pada **Tabel 4.8**.

Tabel 4. 8 Nilai Hidden Layer

<i>z_{net}</i>	1	2	3	4
1	0.09640	-0.72800	1.19956	0.51967
2	-0.06344	-0.64767	0.92554	0.24075
3	0.03656	-0.45725	0.97168	0.27892
4	-0.10475	-0.66558	0.93655	0.25389
5	0.07517	-0.46026	1.12656	0.44421
6	0.58512	-0.53874	0.87154	0.16617
7	0.10562	-0.46524	0.71518	0.02192
8	0.09799	-0.65980	1.62145	0.93470
9	-0.14620	-1.14578	0.91981	0.24801
10	0.25202	0.00558	0.80336	0.12775

Selanjutnya setelah nilai *hidden layer* telah ditentukan, maka dilakukan proses aktivasi data, agar pola data seragam dengan nilai yang diinginkan, yakni berhilai antara nol (0) dan satu (1). Dengan hasil aktivasi nilai *hidden layer* seperti yang digambarkan pada **Tabel 4.9**.

Tabel 4. 9 Nilai Aktivasi *Hidden layer*

<i>z_j</i>	1	2	3	4
1	0.52408	0.32563	0.76845	0.62707
2	0.48414	0.34351	0.71617	0.55990
3	0.50914	0.38764	0.72545	0.56928
4	0.47384	0.33949	0.71840	0.56313
5	0.51878	0.38692	0.75520	0.60926
6	0.64224	0.36848	0.70507	0.54145
7	0.52638	0.38574	0.67154	0.50548
8	0.52448	0.34078	0.83500	0.71803
9	0.46352	0.24126	0.71500	0.56169
10	0.56267	0.50139	0.69069	0.53190

Tahap selanjutnya dari proses propagasi maju ialah menentukan nilai *output* dari ANN-BP, dengan memanfaatkan perkalian nilai *hidden layer* dan bobot hasil hibridisasi (*w*) sehingga menghasilkan data peramalan dengan hasil yang direpresentasikan oleh **Tabel 4.10**.

Tabel 4. 10 Nilai Output layer

<i>y_{net}</i>	<i>k</i>
1	0.097232619
2	0.012536363
3	-0.076140279
4	0.035197412
5	0.029854312
6	-0.190192853

Tabel 4. 11 Nilai Output layer (Lanjutan)

7	-0.110733299
8	0.037974548
9	0.21103263
10	0.028619168

Selanjutnya, pada tahapan ini juga memerlukan fungsi aktivasi sehingga data dari proses hibridisasi yang mungkin memiliki pola yang berbeda dapat dikontrol dan terjaga keseragaman data, dengan hasil ditunjukkan pada **Tabel 4.12**.

Tabel 4. 12 Nilai Aktivasi Output layer

Y_k	k
1	0.524289022
2	0.50313405
3	0.480974121
4	0.508798445
5	0.507463024
6	0.452594602
7	0.472344928
8	0.509492496
9	0.552563228
10	0.507154304

Memasuki tahap selanjutnya yakni pada fase propagasi mundur, hal pertama yang ditentukan ialah selisih nilai kesalahan dari hasil peramalan dengan data aktualnya, yang menghasilkan perbandingan seperti pada **Tabel 2.13**.

Tabel 4. 13 Selisih Nilai Kesalahan pada Output layer

δ_k	k
1	-0.022179379
2	-0.0333884
3	0.042696854
4	0.001330402
5	-0.101843062
6	0.0499744
7	0.026236244
8	0.081380541
9	-0.01591928
10	0.098191317

Selanjutnya hasil dari perbandingan nilai kesalahan tersebut digunakan untuk menentukan faktor kesalahan yang terjadi pada bobot (w_{jk}), yang memperoleh nilai sebagaimana yang dijabarkan pada **Tabel 4.14**.

Tabel 4. 14 Nilai Faktor Kesalahan Pada *Output Layer* ke *Hidden Layer*

Δw_{kj}	0	1	2	3	4
1	-0.01996	-0.01046	-0.00650	-0.01534	-0.01252
2	-0.03005	-0.01455	-0.01032	-0.02152	-0.01682
3	0.03843	0.01956	0.01490	0.02788	0.02188
4	0.00120	0.00057	0.00041	0.00086	0.00067
5	-0.09166	-0.04755	-0.03547	-0.06922	-0.05584
6	0.04498	0.02889	0.01657	0.03171	0.02435
7	0.02361	0.01243	0.00911	0.01586	0.01194
8	0.07324	0.03841	0.02496	0.06116	0.05259
9	-0.01433	-0.00664	-0.00346	-0.01024	-0.00805

Selain dimanfaatkan untuk menentukan nilai faktor kesalahan pada *hidden layer*, nilai kesalahan pada *output layer* juga digunakan untuk menentukan nilai kesalahan bobot (w_{kj}) sesuai dengan jumlah data pada bobot tersebut, yang menghasilkan nilai pada **Tabel 4.15**.

Tabel 4. 15 Nilai Kesalahan *Hidden Layer*

δ_{net_j}	1	2	3	4
1	-0.00182	0.01306	0.00962	-0.01595
2	-0.00239	0.01988	0.01499	-0.02359
3	0.00243	-0.02587	-0.02008	0.02945
4	0.00010	-0.00079	-0.00059	0.00095
5	-0.00786	0.06014	0.04498	-0.07254
6	0.00148	-0.03129	-0.02553	0.03280
7	0.00153	-0.01599	-0.01257	0.01786
8	0.00577	-0.04886	-0.03771	0.05637
9	-0.00174	0.00916	0.00640	-0.01186

Selanjutnya selisih kesalahan juga diperhitungkan pada proses selanjutnya dari propagasi mundur, dengan notasi (δ_j) selisih kesalahan ini akan digunakan sebagai bahan perhitungan dalam menentukan faktor kesalahan pada bobot (v_{ji}), dengan hasil direpresentasikan dalam **Table 4.16**.

Tabel 4. 16 Selisih nilai Kesalahan pada *Hidden layer*

δ_j	1	2	3	4
1	-0.00045	0.00287	0.00171	-0.00373
2	-0.00060	0.00448	0.00305	-0.00581
3	0.00061	-0.00614	-0.00400	0.00722
4	0.00003	-0.00018	-0.00012	0.00023
5	-0.00196	0.01427	0.00832	-0.01727
6	0.00034	-0.00728	-0.00531	0.00814
7	0.00038	-0.00379	-0.00277	0.00446
8	0.00144	-0.01098	-0.00519	0.01141
9	-0.00043	0.00168	0.00130	-0.00292
10	0.00248	-0.01421	-0.00865	0.01802

Selanjutnya, seperti myang telah dijelaskan sebelumnya, dengan menggunakan nilai selisih kesalahan pada *hidden layer*, nilai faktor kesalahan pada *Hidden layer* ke *Input layer* dapat ditentukan, dengan hasil yang telah dijabarkan pada **Tabel 4.17**.

Tabel 4. 17 Nilai Faktor Kesalahan pada *Hidden layer* ke *Input layer*

ΔV_{ji}	v01	...	v14	v02	...	v51	...	v54	v61	...	v64
1	-4.1E-04	...	-7.6E-04	2.6E-03	...	-1.5E-04	...	-1.2E-03	-1.4E-04	...	-1.2E-03
2	-5.4E-04	...	-2.5E-03	4.0E-03	...	-1.9E-04	...	-1.8E-03	-2.3E-04	...	-2.3E-03
3	5.5E-04	...	2.0E-03	-5.5E-03	...	2.4E-04	...	2.8E-03	2.0E-04	...	2.4E-03
4	2.3E-05	...	5.7E-05	-1.6E-04	...	8.4E-06	...	7.7E-05	1.5E-05	...	1.4E-04
5	-1.8E-03	...	-5.6E-03	1.3E-02	...	-1.2E-03	...	-1.0E-02	-9.1E-04	...	-8.0E-03
6	3.1E-04	...	2.6E-03	-6.6E-03	...	1.6E-04	...	3.8E-03	3.1E-05	...	7.3E-04
7	3.4E-04	...	1.7E-03	-3.4E-03	...	3.4E-05	...	4.0E-04	2.3E-04	...	2.6E-03

Tabel 4. 18 Nilai Faktor Kesalahan pada Hidden layer ke Input layer (lanjutan)

8	1.3E-03	...	3.8E-03	-9.9E-03	...	8.5E-04	...	6.7E-03	7.5E-04	...	5.9E-03
9	-3.9E-04	...	-1.7E-03	1.5E-03	...	-2.3E-04	...	-1.5E-03	-3.3E-04	...	-2.2E-03
10	2.2E-03	...	8.3E-03	-1.3E-02	...	1.9E-03	...	1.4E-02	1.1E-03	...	7.9E-03

Selanjutnya memasuki pada fase ketiga, yakni proses update bobot. Dengan menggunakan beberapa hasil yang telah didapatkan sebelumnya pembaruan bobot dapat dilakukan, lebih tepatnya hasil dari nilai faktor kesalahan pada bobot antara *output layer* dan *hidden layer* (w_{kj}) dijumlahkan dengan nilai bobot sebelumnya, sehingga menghasilkan hasil yang dijabarkan pada **Tabel 4.18**.

Tabel 4. 19 Update Bobot Pada Hidden layer Menuju Output layer

W_{kj}	$W01$	$W11$	$W12$	$W13$	$W14$
1	0.10835	0.07157	-0.59554	-0.44888	0.70653
2	0.07830	0.05703	-0.60586	-0.47041	0.68970
3	0.11673	0.07659	-0.59096	-0.44253	0.71158
4	0.11793	0.07716	-0.59056	-0.44167	0.71225
5	0.02627	0.02961	-0.62602	-0.51089	0.65641
6	0.07125	0.05849	-0.60945	-0.47918	0.68076
7	0.09486	0.07092	-0.60034	-0.46332	0.69270
8	0.16810	0.10934	-0.57538	-0.40216	0.74529
9	0.15378	0.10270	-0.57884	-0.41241	0.73724

Selain bobot w_{kj} yang dilakukan pembaruan bobot, bobot yang ada anantara *hidden layer* dan *input layer* (v_{ji}) juga dilakukan pembaruan data, sehingga dapat menghasilkan solusi yang diharapkan. Dengan hasil seperti yang dijabarkan pada **Tabel 4.19**.

Tabel 4. 20 Update Bobot pada Input layer Menuju ke Hidden layer

ΔV_{ji}	v01	...	v14	v02	...	v44	v51	...	v54	v61	...	v64
1	-3.8E-01	...	-1.4E-01	-4.8E-01	...	-9.0E-01	1.9E-01	...	5.1E-01	-8.4E-01	...	-1.7E-02
2	-3.8E-01	...	-1.4E-01	-4.8E-01	...	-9.0E-01	1.9E-01	...	5.1E-01	-8.4E-01	...	-1.9E-02
3	-3.8E-01	...	-1.4E-01	-4.8E-01	...	-9.0E-01	1.9E-01	...	5.1E-01	-8.4E-01	...	-1.7E-02
4	-3.8E-01	...	-1.4E-01	-4.8E-01	...	-9.0E-01	1.9E-01	...	5.1E-01	-8.4E-01	...	-1.7E-02
5	-3.9E-01	...	-1.5E-01	-4.7E-01	...	-9.1E-01	1.9E-01	...	5.0E-01	-8.4E-01	...	-2.5E-02
6	-3.9E-01	...	-1.4E-01	-4.8E-01	...	-9.0E-01	1.9E-01	...	5.1E-01	-8.4E-01	...	-2.4E-02
7	-3.9E-01	...	-1.4E-01	-4.8E-01	...	-9.0E-01	1.9E-01	...	5.1E-01	-8.4E-01	...	-2.1E-02
8	-3.8E-01	...	-1.4E-01	-4.9E-01	...	-9.0E-01	1.9E-01	...	5.1E-01	-8.4E-01	...	-1.5E-02
9	-3.8E-01	...	-1.4E-01	-4.9E-01	...	-9.0E-01	1.9E-01	...	5.1E-01	-8.4E-01	...	-1.7E-02

42

Dengan hasil keseluruhan dari berbagai bentuk manuali yang telah dibuat, secara garis besar manualisasi menghasilkan hasil peramalan dengan dilengkapi nilai MSE sebagai nilai acuan tingkat keakuratan perhitungan tersebut dilakukan. Dengan detail seperti yang dijabarkan pada Tabel 4.21.

Tabel 4. 21 Hasil Perhitungan Menggunakan ANN-BP

p	x1	...	x6	t	v01	...	v64	z1	...	z4	w01	w11	w12	w13	w14	y'	δ_k	MSE
p1	0.23	...	0.35	0.44	-0.38	...	-0.02	0.52	...	0.63	0.13	0.08	-0.59	-0.43	0.72	0.52	-0.02	0.053734
p2	0.49	...	0.44	0.37	-0.38	...	-0.02	0.48	...	0.56	0.11	0.07	-0.60	-0.45	0.71	0.50	-0.03	
p3	0.31	...	0.37	0.65	-0.38	...	-0.02	0.51	...	0.57	0.08	0.06	-0.61	-0.47	0.69	0.48	0.04	
p4	0.27	...	0.65	0.51	-0.38	...	-0.02	0.47	...	0.56	0.12	0.08	-0.59	-0.44	0.71	0.51	0.00	

Tabel 4. 22 Hasil Perhitungan Menggunakan ANN-BP (Lanjutan)

<i>p</i>	<i>x1</i>	...	<i>x6</i>	<i>t</i>	<i>v01</i>	...	<i>v64</i>	<i>z1</i>	...	<i>z4</i>	<i>w01</i>	<i>w11</i>	<i>w12</i>	<i>w13</i>	<i>w14</i>	<i>y'</i>	δk	<i>MSE</i>
<i>p5</i>	0.36	...	0.51	0.10	-0.38	...	-0.02	0.52	...	0.61	0.12	0.08	-0.59	-0.44	0.71	0.51	-0.10	0.053734
<i>p6</i>	0.35	...	0.10	0.65	-0.39	...	-0.02	0.64	...	0.54	0.03	0.03	-0.63	-0.51	0.66	0.45	0.05	
<i>p7</i>	0.44	...	0.65	0.58	-0.39	...	-0.02	0.53	...	0.51	0.07	0.06	-0.61	-0.48	0.68	0.47	0.03	
<i>p8</i>	0.37	...	0.58	0.84	-0.39	...	-0.02	0.52	...	0.72	0.09	0.07	-0.60	-0.46	0.69	0.51	0.08	
<i>p9</i>	0.65	...	0.84	0.49	-0.38	...	-0.02	0.46	...	0.56	0.17	0.11	-0.58	-0.40	0.75	0.55	-0.02	
<i>p10</i>	0.51	...	0.49	0.90	-0.38	...	-0.02	0.56	...	0.53	0.15	0.10	-0.58	-0.41	0.74	0.51	0.10	

43

Proses akhir dari penentuan nilai peramalan ialah dengan diuji apakah penggunaan variabel telah sesuai dengan permasalahan yang diangkat sehingga menghasilkan informasi peramalan yang baik, salah satunya yakni penentuan nilai MSE yang telah dibahas sebelumnya, dan juga secara manual dapat dilakukan dengan membandingkan nilai aktual dengan hasil prediksi yang telah dilakukan dan dinormalisasi, dengan menggunakan Persamaan 2.15.

4.4 Perancangan Antarmuka

Dalam perancangan antar muka, menjabarkan bentuk sistem berkomunikasi dengan penggunanya, dengan diberikan fitur-fitur yang dapat digunakan oleh pengguna untuk menggunakan sistem berdasarkan fungsinya masing-masing, serta menampilkan sisi hasil dari proses yang dilakukan oleh sistem dengan kondisi yang dapat ditentukan oleh pengguna.

4.4.1 Perancangan Antarmuka Halaman Menu Olah Data

Halaman awal dari sistem ini menjabarkan dan menampilkan dataset pelatihan dan hasil *normalization*nya. Halaman awal ini dikelola oleh menu "Olah Data", dengan salah satu sub menu yang dapat ditampilkan yaitu halaman "Data Latih". Untuk dapat menampilkan data

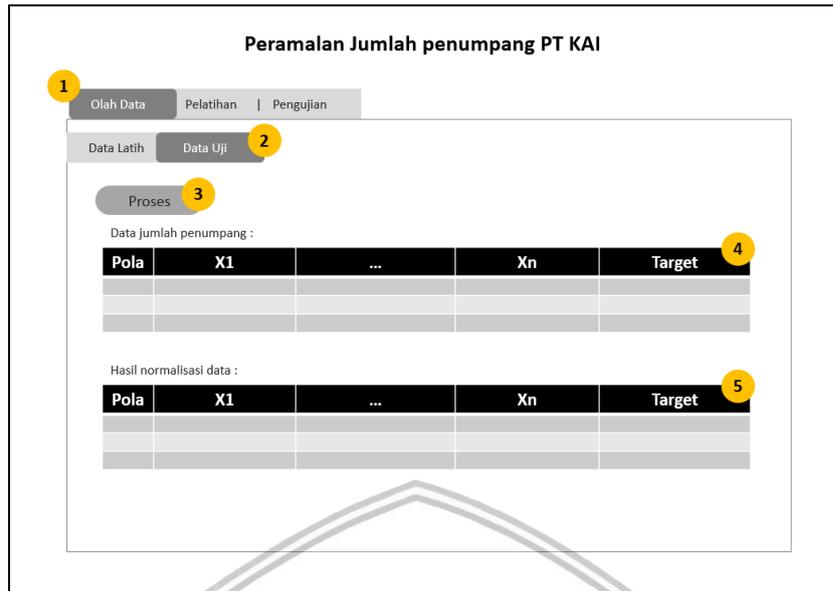
latih dan *normalization*nya, pengguna hanya perlu menekan tombol “proses”. Dengan tampilan perancangan sistem digambarkan pada **Gambar 4.9**. Sama halnya dengan **Gambar 4.9**, pada **Gambar 4.10** juga digunakan untuk menampilkan data dan menghasilkan *normalization* data dari data yang ditampilkan tersebut dengan cara yang sama pula, yakni dengan menekan tombol “proses”, namun data yang dihasilkan pada **Gambar 4.10** merupakan data uji.



Gambar 4. 9 Perancangan Antarmuka Halaman Submenu Data latih

Dengan perancangan seperti pada **Gambar 4.9** ada tiga atribut aktif dan dua atribut aktif, yang dapat dikelola pada submenu data latih, dengan penjelasan masing-masing atribut adalah :

1. Menu utama olah data atau yang disebut dengan *tabbed pane* digunakan untuk menampilkan salah satu halaman submenu data latih.
2. Submenu data latih atau yang disebut dengan *tabbed pane* dimanfaatkan untuk halaman yang terfokus untuk mengolah perintah-perintah yang dinaunginya.
3. Tombol proses atau yang disebut *button* digunakan sebagai pemicu sistem untuk mengolah perintah dari pengguna untuk menampilkan data latih.
4. Tabel data pelatihan yang dalam bentuk *table*, memiliki fungsi untuk menampung dan menampilkan data latih dalam bentuk data mentah, yang diseleksi dengan jumlah yang telah ditentukan oleh pengguna dalam bentuk file .CSV.
5. Tabel data hasil *normalization* dalam bentuk *table*, yang fungsi untuk menampilkan hasil *normalization* dari data mentah pelatihan yang telah dipilih, sehingga data ini yang akan dipergunakan untuk proses pelatihan sistem.



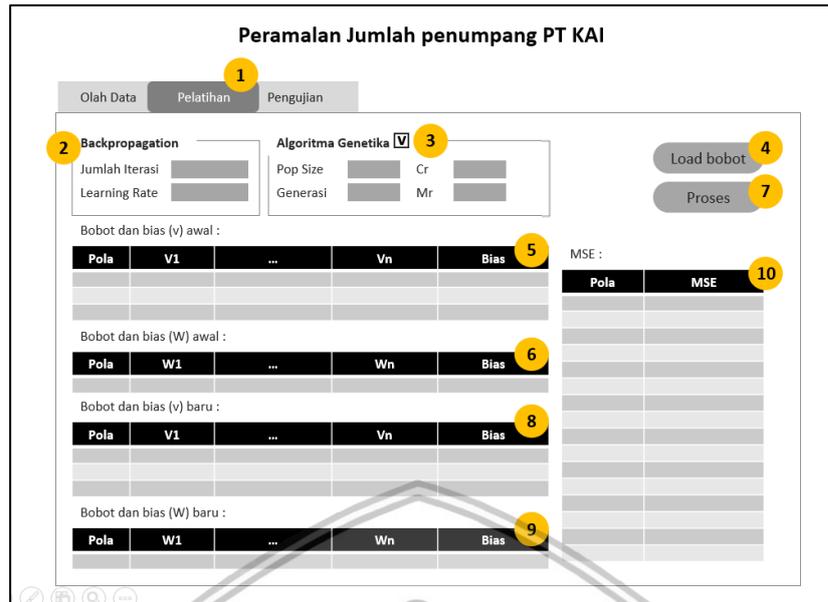
Gambar 4. 10 Perancangan Antarmuka Halaman Submenu Data Uji

Perancangan halaman submenu data uji memiliki komposisi yang serupa dengan sub menu data latih, namun fungsi yang dimiliki berbeda satu sama lain, dengan penjabaran seperti yang terlihat pada **Gambar 4.10**, atribut yang ada pada submenu data uji memiliki fungsi :

1. Menu utama olah data atau yang disebut dengan *tabbed pane* digunakan untuk menampilkan salah satu halaman submenu data uji.
2. Submenu data uji atau yang disebut dengan *tabbed pane* dimanfaatkan untuk halaman yang terfokus untuk mengolah perintah-perintah yang dinaunginya.
3. Tombol proses atau yang disebut *button* digunakan sebagai pemicu sistem untuk mengolah perintah dari pengguna untuk menampilkan data uji.
4. Tabel data pelatihan yang dalam bentuk *table*, memiliki fungsi untuk menampung dan menampilkan data uji dalam bentuk data mentah, yang diseleksi dengan jumlah yang telah ditentukan oleh pengguna dalam bentuk file .CSV.
5. Tabel data hasil *normalization* dalam bentuk *table*, yang fungsi untuk menampilkan hasil *normalization* dari data mentah uji yang telah dipilih, sehingga data ini yang akan dipergunakan untuk proses pengujian sistem.

4.4.2 Perancangan Antarmuka Halaman Menu Pelatihan

Pada halaman pelatihan, merupakan halaman pengelolaan atau halaman kontrol, dimana pengguna dapat menentukan variabel penentu peramalan, baik itu dengan ANN-BP saja maupun variabel dengan hibridisasi menggunakan algoritma genetika. Dengan perancangan antarmuka digambarkan pada **Gambar 4.11**.



Gambar 4. 11 Perancangan Antar Muka Halaman Menu Pelatihan

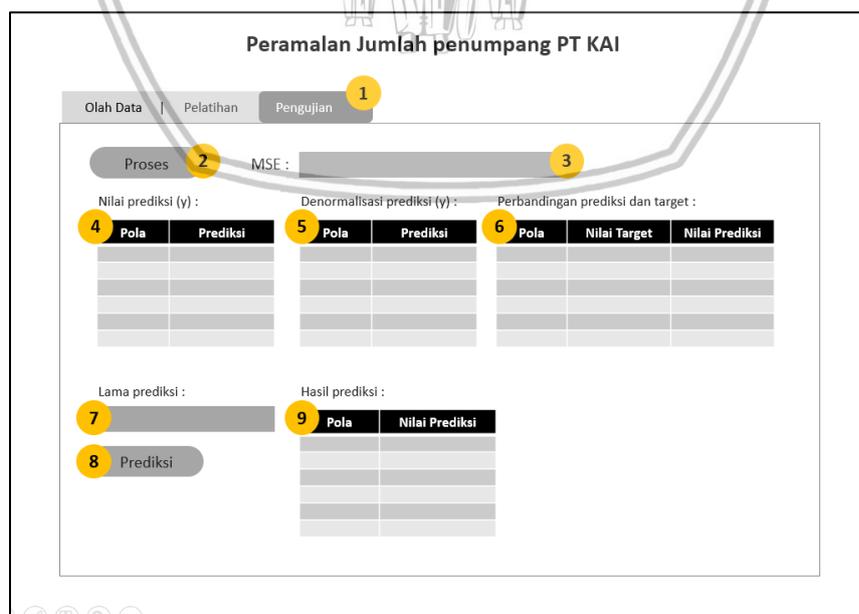
Pada perancangam antarmuka pelatihan ini menampilkan banyak atribut aktif yang dapat dikelola oleh pengguna, diantaranya yakni :

1. Menu pelatihan atau yang disebut dengan *tabbed pane* digunakan untuk mengelompokkan atribut yang hanya dikelola pada menu tersebut.
2. Panel *backpropagation* atau yang disebut dengan *panel* yang digunakan sebagai pengelompokan atribut yang dikelola oleh metode ANN-BP saja, sehingga pengguna mengetahui atribut mana saja yang mempengaruhi kinerja dari metode ANN-BP tersebut. Dengan atribut yang terdiri dari jumlah *iterasi* dan nilai *learning rate* yang bisa digunakan oleh pengguna untuk melakukan modifikasi algoritme pada sistem dengan cara mengisi *field* dalam bentuk *text field* yang disediakan. Dengan *field* iterasi menampung banyaknya perulangan perhitungan ANN-BP yang dapat dilakukan, sedangkan dibagian *text field alpha*, digunakan untuk mengatur panjang jangkauan sistem untuk menemukan solusi, jangkauan tersebut dibatasi dengan nilai antara 0 (nol) sampai dengan 1 (satu).
3. Panel berikutnya digunakan untuk mengelompokkan variabel yang bekerja dibawah aturan algoritme genetika, yang terdiri dari *pop size* atau jumlah populasi, *generasi*, *Cr*, dan *Mr*, yang memiliki fungsi yang sama yakni mencatat nilai masukan atas masing-masing variabel yang ada, pada panel algoritme genetika juga diberikan *radio button* yang dapat digunakan untuk mengaktifkan atau menonaktifkan panel algoritme genetika.
4. Tombol load bobot atau yang disebut dengan *button* digunakan sebagai pemicu eksekusi atau *generate* bobot awal berdasarkan

- 5. Tabel bobot awal (v) dalam bentuk *table* yang memiliki fungsi untuk menampilkan hasil *generate* bobot berdasarkan syarat tertentu, dengan jumlah yang sudah ditentukan oleh *backend* sistem.
- 6. Tabel bobot awal (w) dalam bentuk *table* yang memiliki fungsi untuk menampilkan hasil *generate* bobot berdasarkan syarat tertentu, apakah bobot tersebut dihasilkan dari *generate* secara acak atau hasil dari hibridisasi algoritma genetika dengan jumlah yang sudah ditentukan oleh *backend* sistem.
- 7. Tombol proses atau yang disebut *button* digunakan sebagai pemicu eksekusi solusi hasil perhitungan ANN-BP dan algoritme genetika dengan menggunakan data latih
- 8. Tabel *update* bobot berisi nilai sinyal atau bobot (v) yang dipergunakan untuk menghasilkan solusi pada peramalan informasi.
- 9. Tabel *update* bobot berisi nilai sinyal atau bobot (w) yang dipergunakan untuk menghasilkan solusi pada peramalan informasi.
- 10. Tabel MSE digunakan untuk menampilkan nilai *error* (MSE) berdasarkan tiap perulangan yang dilakukan oleh sistem.

4.4.3 Perancangan Antarmuka Halaman Menu Pengujian

Pada perancang sistem halaman pengujian ditampilkan perintah-perintah yang memungkinkan pengguna untuk melakukan tindakan akhir dari hasil pelatihan yang telah dilakukan di halaman sebelumnya, dengan menggunakan beberapa atribut pengelola pengujian data ini sesuai yang dijabarkan pada **Gambar 4.12**.



Gambar 4. 12 Perancangan Antarmuka Halaman Menu Pengujian

Pada **Gambar 4.12**, menggambarkan atribut pengujian, sebagai atribut yang menanggapi proses pelatihan yang telah dilakukan, dengan penjelasan tiap atribut yakni :

1. Menu pelatihan atau yang disebut dengan *tabbed pane* digunakan untuk mengelompokkan atribut yang hanya dikelola pada menu pengujian data.
2. Tombol proses atau yang disebut *button* digunakan sebagai pemicu eksekusi solusi untuk menghasilkan nilai MSE pada proses pengujian.
3. *Text Field* MSE menampilkan hasil nilai kesalahan (MSE) yang dihasilkan dari proses pengujian dengan menggunakan data uji.
4. Tabel hasil prediksi yang berupa *table* digunakan untuk menampilkan data hasil prediksi berdasarkan pengujian yang dilakukan atas pola yang dibentuk dari proses pelatihan.
5. Tabel *dinormalization* hasil prediksi yang berupa *table*, memiliki fungsi untuk menampilkan hasil *dinormalization* atau mengembalikan data perhitungan sebelumnya yang dalam bentuk *normalization* data, sehingga dapat dilakukan proses lain untuk pencarian nilai error.
6. Tabel perbandingan hasil prediksi yang berbentuk *table*, menampilkan hasil prediksi dalam bentuk bilangan asli, sehingga dapat dengan mudah untuk dibandingkan hasilnya dengan data awal atau target.
7. *Text Field* digunakan untuk menampung besaran atau jumlah prediksi yang ingin dilakukan. Dengan asumsi tiap angka mewakili data tiap bulan.
8. Tombol proses disajikan dalam bentuk *button* berfungsi untuk memproses perintah masukan dari jumlah prediksi yang di masukkan pengguna dalam *text field* lama prediksi.
9. Tabel Hasil prediksi yang disajikan dalam bentuk *table* berfungsi untuk menampilkan hasil prediksi yang telah dihitung berdasarkan pola yang telah terbentuk.

4.5 Perancangan Pola Pengujian

Untuk mendapatkan hasil keakurasian dari penerapan metode jaringan saraf tiruan *backpropagation* dan algoritme genetika dalam memprediksi jumlah penumpang kereta api Indonesia, maka dibutuhkan nilai variabel-variabel pendukung sehingga memperbesar kemungkinan untuk mendapatkan solusi yang optimal. Variabel-variabel tersebut dapat dibangkitkan dengan melakukan skema pengujian sistem. Dengan penerapan beberapa parameter pendukung maka akan menghasilkan variabel yang baik tersebut, dengan melakukan pengujian pada variabel berikut :

1. Uji coba populasi atau *pop size*, untuk mengetahui jumlah populasi yang unggul dalam melakukan pengujian sistem.
2. Uji coba generasi dilakukan agar memperluas area pencarian sehingga memungkinkan untuk menghasilkan solusi yang optimal.
3. Uji coba komposisi pada nilai *Cr* dan *Mr*, sehingga mengetahui jenis regenerasi mana yang berpeluang lebih besar untuk menghasilkan solusi yang optimal.

4. Uji coba jumlah iterasi, dilakukan pada tahapan pemrosesan metode ANN-BP, yang bertujuan untuk menentukan area solusi.
5. Uji coba nilai α , bertujuan agar mengetahui komposisi jangkauan pencarian yang tepat terhadap pencarian solusi dalam peramalan jumlah penumpang kereta api Indonesia.
6. Uji coba perbandingan akurasi, dengan membandingkan nilai $error$, baik dalam bentuk MSE ataupun dalam bentuk *median absolute deviation* (MAD), sehingga dapat menentukan solusi yang lebih baik.

4.5.1 Uji Coba Populasi

Jumlah populasi sangat menentukan tingkat keberhasilan algoritme untuk mendapatkan solusi yang baik, pasalnya jika populasi yang digunakan besar maka akan memperbesar pula ditemukannya solusi yang lebih baik. Namun penggunaan populasi yang terlalu besar akan berakibat pada waktu sistem untuk melakukan komputasi dan memungkinkan terjadinya hasil yang *overfitting*. Untuk itu komposisi jumlah populasi sangat penting ditentukan, dalam hal ini pengujian yang akan dilakukan menggunakan jumlah populasi pengujian sebanyak 50 dan keipatannya hingga mencapai jumlah 800, dengan dilakukan 10 kali percobaan sehingga akan didapatkan solusi yang akurat. Dengan rancangan pengujian populasi digambarkan pada **Tabel 4.23**.

Tabel 4. 23 Rancang Uji Coba Jumlah Populasi

Pop size	Percobaan ke- n			Rata-rata MSE
	1	...	10	
50		...		
100		...		
150		...		
200		...		
250		...		
300		...		
350		...		
400		...		
450		...		
500		...		
550		...		
600		...		
650		...		
700		...		
750		...		
800		...		

4.5.2 Uji Coba Generasi

Sama halnya dengan populasi, generasi juga mempengaruhi besar kecilnya area pencarian solusi, bahkan generasi juga sebagai media untuk menggandakan area pencarian tersebut, sehingga memungkinkan generasi ini sangat mempengaruhi hasil dari nilai optimum, untuk itu pengujian generasi dilakukan dengan nilai awal berjumlah 2 dan keipatannya hingga mencapai jumlah generasi 20, dan dilakukan dalam 10 kali percobaan. Sehingga rancangan tabel pengujian generasi dapat digambarkan seperti pada **Tabel 4.24**.

Tabel 4. 24 Rancang Uji Coba Generasi

Generasi	Percobaan ke- n			Rata-rata MSE
	1	...	10	
2		...		
4		...		
6		...		
8		...		
10		...		
12		...		
14		...		
16		...		
18		...		
20		...		

4.5.3 Uji Coba Kombinasi C_r dan M_r

Kombinasi nilai C_r dan M_r , memiliki peran langsung atas dibangkitkannya solusi baru, sehingga solusi yang dihasilkan dengan metode C_r ataukah M_r yang mendominasi untuk menghasilkan *offsring* yang lebih fit, dengan cara kerja pengujian dilakukan dengan pemberian nilai 0.1 sampai mencapai nilai 0.9, yang dicocokkan secara terbalik, seperti yang direpresentasikan pada **Tabel 4.25**.

Tabel 4.25 Rancang Tabel kombinasi C_r dan M_r

C_r	M_r	Percobaan ke- n			Rata-rata MSE
		1	...	10	
0.1	0.9		...		
0.2	0.8		...		
0.3	0.7		...		
0.4	0.6		...		
0.5	0.5		...		
0.6	0.4		...		
0.7	0.3		...		
0.8	0.2		...		
0.9	0.1		...		



4.5.4 Uji Coba Jumlah Iterasi

Pengujian iterasi merupakan pengujian yang dilakukan pada proses metode jaringan saraf tiruan *backpropagation*, memiliki tingkat *urgency* yang sama dengan penentuan jumlah populasi, penentuan jumlah iterasi ini mempengaruhi terhadap besar kecilnya area pencarian solusi yang dilakukan, serta tingkat keefektifan sistem untuk mendapatkan solusi hal sama juga terjadi pada pencarian iterasi jika area pencarian terlalu besar maka akan mempengaruhi lama komputasi sistem untuk menjalankan sistem tersebut. Untuk perlu ditentukan komposisi yang baik dalam menentukan nilai iterasi yang digunakan, dengan melakukan percobaan terhadap 10 iterasi, 25 dan kelipatannya hingga mencapai jumlah 350 iterasi, dan dilakukan dalam 10 kali percobaan, sehingga representasi pengujiannya dapat digambarkan pada **Tabel 4.26**.

Tabel 4. 26 Rancang Uji Coba Jumlah Iterasi

Iterasi	Percobaan ke- <i>n</i>		
	1	...	10
10		...	
25		...	
50		...	
75		...	
100		...	
125		...	
150		...	
175		...	
200		...	
225		...	
250		...	
275		...	
300		...	
325		...	
350		...	

4.5.5 Uji Coba Nilai *Alpha*

Selanjutnya pengujian pada nilai *alpha* atau *learning rate*. Pengujian pada nilai *alpha* bertujuan untuk mengetahui pola jangkauan pelatihan yang baik diterapkan pada pola data yang digunakan, jadi jika pola data cukup stabil nilai jangkauan pencarian solusi dapat menggunakan nilai *alpha* yang besar, namun jika nilai *alpha* data berbeda jauh antar data, maka diperlukan jangkauan yang lebih teliti, yakni dengan menggunakan nilai *alpha* yang kecil, nilai dari *alpha* sendiri berada antara 0.1 sampai dengan 0.9. Berdasarkan alasan yang dikemukakan sebelumnya maka pengujian dilakukan 10 kali pada masing-masing nilai *alpha*, dengan representasi pengujian seperti yang dijabarkan pada **Tabel 4.27**.

Tabel 4. 27 Rancang Uji Coba Alpha

Alpha	Percobaan ke- <i>n</i>			Rata-rata MSE
	1	...	10	
0.1		...		
0.2		...		
0.3		...		
0.4		...		
0.5		...		
0.6		...		
0.7		...		
0.8		...		
0.9		...		

4.5.6 Uji Coba Perbandingan Hasil Prediksi ANN-BP dan Algoritme Genetka

Pengujian selanjutnya adalah pengujian keakurasian penerapan metode jaringan saraf tiruan *backpropagation* dan algoritma genetika, dalam bentuk nilai *error* yang dihasilkan, dengan cara menentukan nilai MSE dan MAD dari pengujian metode ANN-BP dan tanpa melakukan hibridisasi nilai bobot awal (*w*) dengan menggunakan algoritme genetika. Dengan skema pengujian nilai MSE yang ditampilkan pada **Tabel 4.28** dan skema pengujian MAD pada **Tabel 4.29**.

Tabel 4. 28 Perbandingan Nilai Error MSE Menggunakan dan Tanpa Optimasi

Pelatihan	Percobaan ke- <i>n</i>			Rata-rata MSE
	1	...	10	
ANN-BP		...		
Algoritme genetika		...		

Tabel 4. 29 Rancang Uji Coba Perbandingan Menggunakan dan Tanpa Optimasi

Data Aktual	ANN-BP	ANN-BP + GA	Selisih ANN-BP	Selisih ANN-BP + GA
			MAD	

4.6 Lingkungan Pengembangan Sistem

Dalam lingkungan pengembangan sistem dibutuhkan lingkungan *hardware* dan *software* untuk mendukung pengolahan dan pengujian sistem.



4.6.1 Lingkungan Perangkat Keras (*Hardware*)

Instrumen yang digunakan peneliti untuk melakukan penelitian dan pengujian sistem didukung oleh laptop acer aspire E5-478G, yang memiliki spesifikasi:

- CPU Intel Core i5 dengan kecepatan kinerja prosesor sebesar 2.5 GHz.
- GPU intel NVIDIA GeForce 940MX.
- Kapasitas laju operasi 4 GB DDR 4.
- Kapasitas penyimpanan data 500 GB.

4.6.2 Lingkungan Perangkat Lunak (*Software*)

Sedangkan perangkat lunak dalam melengkapi proses kebutuhan perangkat lunak yang peneliti gunakan ialah :

- Windows 10 dengan sistem operasi 64 bit.
- NetBeans sebagai IDE (bahasa pemrograman Java)
- Tipe sistem: 64-bit.

4.7 Implementasi Algoritme

Implementasi algoritme ialah bagian dimana sitem ini dikembangkan berdasarkan gambaran manualisasi dan diagram alir yang dibuat sebelumnya dengan langkah awal dimulai dengan melakukan inialisasi data yang berasal dari data dalam format CSV, dan data tersebut dilakukan *normalization*, kemudian dilakukan pembangkitan nilai awal secara random berdasarkan pola data yang digunakan, untuk pemrosesan data menggunakan metode ANN-BP dengan hibridisasi bobot awal dengan menggunakan algoritme genetika, dilakukan terlebih dahulu nilai bobot (w) yang dioptimasi pada penelitian ini, dengan menerapkan atribut pengolahan optimasi algoritme genetika, sehingga hasilnya dapat digunakan sebagai rujukan unruk nilai bobot awal (w) dan selanjutnya dilakukan pemrosesan pada jaringan saraf tiruan *backpropagation*, bagi data yang tidak dilakukan hibridisasi, maka data tersebut dapat langsung diproses pada jaringan saraf tiruan *backpeopagation* untuk menghasilkan nilai peramalan jumlah penumpang kereta api Indonesia.

4.7.1 Implementasi Pengambilan Data

Pada proses pengambilan data yang berupa data dengan format CSV dengan bentuk data saja didalam file tersebut, sehingga memudahkan sistem untuk melakukan penyeleksian data oleh sistem berdasarkan urutan data tersebut, dengan implementasinya ditunjukkan pada **Source Code 4.1**.

```
1 public class PenumpangKai {
2
3 String path100 =
4 "D:/[Lesson]/[Practice]/Java/penumpangKAI/data/100Data.csv";
5
6 void loadDataset(){
7     String InputLines ="";
8     int rows = 0;
9     try{
10 Scanner inpt = new Scanner(new BufferedReader(new
11     FileReader(path100)));
12         while (inpt.hasNextLine ()){
13             InputLines = inpt.nextLine();
14             String [] InArrays = InputLines.split(",");
15             for(int m=0; m<InArrays.length; m++){
16                 algen1.xAwal1[rows][m] =
17                 Double.parseDouble(InArrays[m]);
18                 GUIKai.xAwal1[rows][m] =
19                 Double.parseDouble(InArrays[m]);
20             }
21             rows++;
22         }
23     }catch(FileNotFoundException ex){
24         Logger.getLogger(GUI.class.getName()).log(Level.SEVERE,
25         null, ex);
26     }
27 }
28 void loadDataUjil(){
29     String InputLines = "";
30     int rows = 0;
31     try{
32         Scanner input = new Scanner(new
33         BufferedReader(new FileReader(path50 )));
34         while (input.hasNextLines ()){
35             InputLines = input.nextLine();
36             String[] InArrays = InputLines.split(",");
37             for (int m = 0; m < InArrays.length; m++) {
38                 xUjiAwal1[rows][m] =
39                 Double.parseDouble(InArrays[m]);
```

```

35         }
36         rows++;
37     }
38     } catch (FileNotFoundException ex) {
39     Logger.getLogger(GUI.class.getName()).log(Level.SEVERE,
40     null, ex);
41     }
42     }}

```

Source Code 4. 1 Implementasi Pengambilan Data

4.7.2 Implementasi *Normalization* Data

Untuk mempermudah perhitungan maka dilakukanlah *normalization* data, yang di implementasikan pda kedua dataset, yakni data *traing* dan data *testing*, dengan penerapan algoritme yang ditunjukkan pada *line 6* pada algoritme yang ditampilkan pada **Source Code 4.2**.

```

1 public void normData1(double1[][] data1,
2 double1[][] target){
3     double min1 = min1(data1);
4     double max1 = maks1(data1);
5
6     for (int m=0; m<jmlhData1; m++){
7         for(int n=0; n<fitur1; n++){
8             target[m][n] = ((.8 * (data1[m][n] - min)) /
9             (max1 - min1)) + .1;
10        }
11    }
12 }

```

Source Code 4. 2 Implementasi *Normalization* Data

4.7.3 Implementasi Proses *Feedforward*

Implementasi proses *feedforward*, pada proses ini diterapkan algoritme dalam melakukan inialisasi data pada metode ANN-BP. Yakni dengan memberikan nilai pada tiap unit *input layer*, *hidden layer*, dan *output layer*, dengan penerapan algoritma yang ditunjukkan *source code* pada *line 4* sampai dengan *line 13*. Kemudian untuk mengimplementasikan algoritme dalam menentukan nilai *output layer* dijelaskan pada *line 15* sampai dengan *line 27*. Dengan penjelasan lebih lengkap ditunjukkan oleh **Source Code 4.3**.

```

1  for(int m=0; m<jmlhData1 ; m++){
2      for(int n=0; n<jmlhHidden1 ; n++){
3          double jmlh1 = 0;
4          for(int i=0; i<jmlhInput1 ; i++){
5              double hasil1 = x1[m][i] * v1[i][n];
6              jmlh1 = jmlh1 + hasil1
7                  }
8          z_net1[n] = v01[n] + jmlh1;
9          Sistem.out.println("z_net-" + (n+1) + " = " +
10 z_net1[n]);
11         z1[n] = 1/ (1+(double) Math.exp(-z_net1[n]));
12         Sistem.out.println("z-" + (n+1) + " = " + z1[n]);
13     }
14     Sistem.out.println("");
15     for(int j=0; j<jmlhOutput1 ; j++){
16         double jmlh1 = 0;
17         for(int l=0; l<jmlhHidden1 ; l++){
18             double hasil1 = z1[j] * w1[j][l];
19             jmlh1 = jmlh1 + hasil1;
20         }
21         y_net1[l] = w01[l] + jmlh1;
22         Sistem.out.println("y_net-" + (l + 1) + " = " +
23 y_net1[l]);
24         Sistem.out.println("");
25         Y1[l] = 1 / (1 + (double) Math.exp(-y_net1[l]));
26         Sistem.out.println("y1-" + (l + 1) + " = " +
27 y1[l]);
28         mse1 = mse1 + Math.pow((y1[l] - x1[j][6]), 2);
29     }

```

Source Code 4. 3 Implementasi Proses Feedforward

4.7.4 Implementasi Proses Backpropagation

Implementasi dari algoritme backpropagation digunakan untuk menerapkan algoritme pencarian faktor kesalahan berdasarkan proses inialisasi sebelumnya yakni mulai dari penentuan nilai dari δ_k yang ditunjukkan oleh *line 2*, Δw_{jk} dan Δw_{0jk} yang ditunjukkan oleh *line 4* dua line berikutnya, δ_{net_j} yang ditunjukkan oleh *line 12* hingga tiga baris depannya, δ_j yang ditunjukkan oleh *line 18*, Δv_{ij} juga pada Δv_{0j} yang ditunjukkan oleh *line 20* hingga empat line selanjutnya. Dengan proses lengkapnya tercantum pada **Source Code 4.4**.

```

1 for(int i=0; i<jmlhOutput1 ; i++){
2     dk1[i] = (x1[1][6] - y1[i]) * y1[i] * (1 - y1[i]);
3     Sistem.out.println("dk-" + (i + 1) + " = " + dk1[i]);
4     for(int m=0; <jmlhHidden1; m++){
5         deltaW1[m][i] = alfa1 * dk1[i] * z1[m];
6         deltaW01[i] = alfa1 * dk1[i];
7         Sistem.out.println("\ndelta w" + (m + 1) + (i + 1)
+ " = " + deltaW1[m][i]);}
8     Sistem.out.println("delta w0" + (i + 1) + " = " +
deltaW01[i]); }
9     Sistem.out.println("");
10    for(int m=0; m<jmlhHidden1; m++){
11        for(int m=0; m<jmlhOutput1; m++){
12            d_netj1[i] = (dk1[i] * deltaW1[m][i]); }
13            Sistem.out.println("d_net-" + (m + 1) + " = " +
d_netj1[m]);
14            Sistem.out.println("");
15            dj1[m] = d_netj1[m] * (z1[m]) * (1 - z1[m]);
16            Sistem.out.println("dj-" + (m + 1) + " = " +
dj1[m]);
17            for(int h=0; h<jmlhInput1; h++){
18                deltaV1[h][m] = alfa1 * dj1[m] * x1[0][1];
19                Sistem.out.println("\ndelta v" + (h + 1) +
(m + 1) + " = " + deltaV1[h][m]);
20            }
21            deltaV01[m] = alfa1 * dj1[m];
22            Sistem.out.println("delta v0" + (m + 1) + " =
" + deltaV01[m]); }

```

Source Code 4. 4 Implementasi Proses Backpropagation

4.7.5 Implementasi peoses Update Bobot

Penerapan algoritme untuk memperbaiki nilai bobot sehingga dapat digunakan untuk melakukan pencarian solusi dengan proses yang ditunjukkan oleh *line 3* dan juga *line 8*, sedangkan pada boboht (*v*) diimplementasikan pada *source code* pada *line 14* dan *line 19*. Dengan proses lengkapnya ditampilkan pada **Source Code 4.5**.

```

1 for(int m=0; m<jmlhHidden1; m++){
2     for(int n=0; n<jmlhOutput1; n++){
3         w1[m][n] = w1[m][n] + deltaW1[m][n];

```

```

4      Sistem.out.println("w" + (m + 1) + (n + 1) + " = " +
w1[m][n]);
5      }
6  }
7  for(int n=0; n<jmlhOutput1; n++){
8      w01[n] = w01[n] + deltaW01[n];
9      Sistem.out.println("w0" + (n + 1) + " = " + w01[n]);
10 }
11 Sistem.out.println("");
12 for(int l1=0; l1<jmlhInput1; l1++){
13     for(int k1=0; k1<jmlhHidden1; k1++){
14         v1[l1][k1] = v1[l1][k1] + deltaV1[l1][k1];
15         Sistem.out.println("v" + (l1 + 1) + (k1 + 1) + " =
" + v1[l1][k1]);
16     }
17 }
18 for(int k1=0; k1<jmlhHidden1; k1++){
19     v01[k1] = v01[k1] + deltaV01[k1];
20     Sistem.out.println("v0" + (k1 + 1) + " = " +
v01[k]);
21 }
22 }

```

Source Code 4. 5 Implementasi Proses Update Bobot

4.7.6 Implementasi Proses Crossover

Penerapan algoritme untuk proses *crossover*, lebih tepatnya *extended intermediate crossover* yang di representasikan dalam kode sistem yang ditunjukkan pada *line 15* sampai dengan *line 26*. direalisasikan dalam bentuk algoritme pada *source code* yang terletak pada baris 15 sampai dengan baris 28. Dengan proses lengkapnya ditampilkan pada **Source Code 4.6**.

```

1  int count1 = 0;
2      Sistem.out.println( "offspring Cr = " +
CrOffspring1);
3  for (int h1 = 0; h1 < ceer_int1; h1++) {
4      int acakB1 = b.nextInt1(individu1);
5      int acakC1 = b.nextInt1(individu1);
6      if (acakB1 == acakC1) {
7          if (acakC1 >= (individu1 - 1)) {
8              acakC1 = acakC1 - 1;

```

```

9         } else {
10             acakC1 = acakC1 + 1;
11         }
12     }
13     Sistem.out.println( "\nindividu ke " + (acakB + 1) + "
dan " + (acakC1 + 1) + " terpilih dalam proses Crossover
:" );
14     Sistem.out.println("Anak ke-" + (count1 + 1));
15     for (int k1 = 0; k1 < fiturGA1; k1++) {
16         child1[count1][k1] = xGA1[acakB1][k1] +
(alfaGA1 * (xGA1[acakC1][k1] - xGA1[acakB1][k1]));
17         Sistem.out.println(child[count1][k1]);
18     }
19     Sistem.out.println("");
20     Count1++;
21     if (CrOffspring1 == count1) {
22         break;
23     } else {
24         Sistem.out.println("Anak ke-" + (count1 +
1));
25         for (int k1 = 0; k1 < fiturGA1; k1++) {
26             child1[count1][k1] = xGA1[acakC1][k1] +
(alfaGA1 * (xGA1[acakB1][k1] - xGA1[acakC1][k1]));
27             Sistem.out.println(child1[count1][k1]);
28         }
29         Count1++;
30     }
31 }

```

Source Code 4. 6 Implementasi Proses Crossover

4.7.7 Implementasi Proses Mutasi

Penerapan algoritme dalam melakukan perhitungan nilai mutasi yang digunakan untuk menghasilkan solusi baru ditunjukkan pada *line* 14, Dengan proses lengkapnya ditampilkan pada **Source Code 4.7**.

```

1     for (int i1 = 0; i1 < offspringMr1; i1++) {
2         int acakD1 = b.nextInt(individu1);
3         int acakE1 = b.nextInt(fiturGA1);
4         Sistem.out.println("\nPopulasi yang terpilih : " +
(acakD1 + 1) + " dengan gen " + (acakE1 + 1));
5         for (int d1 = 0; d1 < fiturGA1; d1++) {

```

```

6         for (int u1 = 0; u < fiturGA1; u1++) {
7             if (xGA1[acakD1][u1] > maxGen11) {
8                 maxGen11 = xGA1[acakD1][u1];
9             } else if (xGA1[acakD1][u1] < minGen11) {
10                minGen11 = xGA1[acakD1][u1];
11            }
12        }
13        for (int k1 = 0; k1 < fiturGA1; k1++) {
14            cloneMut1[0][k1] = xGA1[acakD1][k1];
15        }
16        cloneMut1[0][acakE1] = cloneMut1[0][acakE1] + (R1 *
17        (maxGen11 - minGen11));
18        Sistem.out.println("krom : " + cloneMut1[0][d1]);
19        childGen1[i1][d1] = cloneMut1[0][d1];
20    }
21 }

```

Source Code 4.7 Implementasi Proses Mutasi

4.7.8 Implementasi Proses Evaluasi

Implementasian algoritme untuk perhitungan proses evaluasi agar mendapatkan nilai *fitness* dari masing-masing populasi yang terbentuk dari inialisasi dan proses reproduksi ditunjukkan pada *line 11* yang telah memiliki nilai MSE sebelumnya. Dengan proses lengkapnya ditampilkan pada **Source Code 4.8**.

```

1     for (int k1 = 0; k1 < popSizeBarul; k1++) {
2         for (int i1 = 0; i1 < jmlhOutput1; i1++) {
3             for (int j1 = 0; j1 < jmlhHidden1; j1++) {
4                 fitness_sort1[k1][j1] =
5                 himpunanBarul[k1][j1];
6             }
7             fitness_sort1[k1][4] = himpunanBarul[k1][4];
8         }
9         double jmlh1 = 0;
10        for (int k1 = 0; k1 < popSizeBarul; k1++) {
11            double fitness1 = 1/mse1;
12            Sistem.out.println("Fitness Indv ke "+(k1+1)+" =
13            "+fitness1);
14            fitness_sort1[k1][5] = fitness1;
15        }

```

```

15         Sistem.out.println("\nRobot + fitness belum sort
sort");
16         for (int i1 = 0; i1 < popSizebaru1; i1++) {
17             for (int j1 = 0; j1 < 6; j1++) {
18                 Sistem.out.println(fitness_sort1[i1][j1)+"\t\t");
19             }
20         }

```

Source Code 4. 8 Implementasi Proses Evaluasi

4.7.9 Implementasi Proses Seleksi

Penerapan algoritme pada proses seleksi dengan tujuan untuk menentukan nilai *fitness* terbaik sepanjang ukuran populasi induk, sehingga dapat menggantikan induk sebelumnya untuk melakukan proses reproduksi atau untuk mencari solusi baru lagi dengan proses utama ditunjukkan pada *line* 1 sampai dengan *line* 13. Dengan proses lengkapnya ditampilkan pada **Source Code 4.9**.

```

1  boolean selesai = true;
2      while (selesai) {
3          selesai = false;
4          for (int i = 1; i < this.fitness.length; i++) {
5              if (this.fitness[i - 1] < this.fitness[i]) {
6                  temp = this.fitness[i - 1];
7                  index = tempIndex[i - 1];
8                  this.fitness[i - 1] = this.fitness[i];
9                  tempIndex[i - 1] = tempIndex[i];
10                 this.fitness[i] = temp;
11                 tempIndex[i] = index;
12             }
13             selesai = true;
14         }
15     }
16 }
17 for (int i = 0; i < tempIndex.length; i++) {
18     System.out.print((tempIndex[i] + 1) + " | ");
19 }
20 for (int i = 0; i < himpunanBaru.length; i++) {
21     for (int j = 0; j < this.fiturGA; j++) {
22         tempHimpunanIndividu[i][j] = himpunanBaru[tempIndex[i]][j];
23         System.out.print(tempHimpunanIndividu[i][j] + " | ");
24     }

```

```

25     System.out.println("");
26     }
27     for (int i = 0; i < individu; i++) {
28         for (int j = 0; j < fiturGA; j++) {
29             pGA[i][j] = tempHimpunanIndividu[i][j];
30             System.out.print(pGA[i][j] + " | ");
31         }
32         System.out.println("");
33     }
    
```

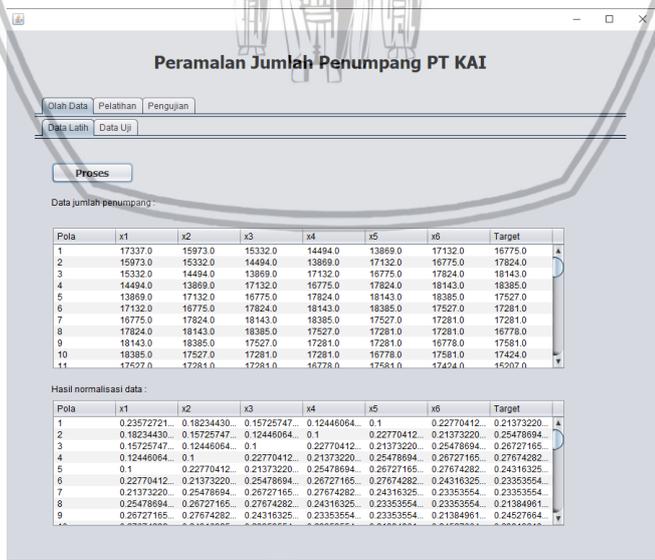
Source Code 4. 9 Implementasi Proses Seleksi

4.8 Implementasi Antarmuka Sistem

Seperti yang telah ditunjukkan pada perancangan antarmuka sebelumnya, untuk implementasi antarmuka juga menghasilkan tiga halaman utama dengan representasi seperti pada fase di jaringan saraf tiruan *backpropagation*, yang meliputi

4.8.1 Implementasi Antarmuka Olah Data

Halaman utama olah data menampilkan data dari file CSV berupa data asli atau data mentah dan data olah berupa hasil *normalization* data asli, dengan cara menekan tombol proses yang disediakan, untuk pengolah data *training* ditampilkan pada submenu data latih, seperti yang ditunjukkan pada **Gambar 4.13**.



Gambar 4. 13 Implementasi Antarmuka Submenu Data latih

Sedangkan submenu kedua dalam halaman utama olah data akan mengelola data uji, yang nantinya digunakan untuk melakukan proses pengujian



data setelah pola penentu telah dipilih, dengan tampilan seperti pada **Gambar 4.14**.

Peramalan Jumlah Penumpang PT KAI

Olah Data | Pelatihan | Pengujuan

Data Lath | Data Uji

Proses

Data jumlah penumpang:

Pola	x1	x2	x3	x4	x5	x6	Target
1	16104.0	14900.0	14504.0	15826.0	16000.0	16113.0	17301.0
2	14900.0	14594.0	15826.0	16000.0	16113.0	17301.0	20245.0
3	14594.0	15826.0	16000.0	16113.0	17301.0	20245.0	19423.0
4	15826.0	16000.0	16113.0	17301.0	20245.0	19423.0	19738.0
5	16000.0	16113.0	17301.0	20245.0	19738.0	20534.0	20534.0
6	16113.0	17301.0	20245.0	19423.0	19738.0	20534.0	19919.0
7	17301.0	20245.0	19423.0	19738.0	20534.0	19919.0	21417.0
8	20245.0	19423.0	19738.0	20534.0	19919.0	21417.0	21092.0
9	19423.0	19738.0	20534.0	19919.0	21417.0	21092.0	19998.0
10	19738.0	20534.0	19919.0	21417.0	21092.0	19998.0	22838.0

Hasil normalisasi data:

Pola	x1	x2	x3	x4	x5	x6	Target
1	0.16127003	0.11241631	0.1	0.14998985	0.16170911	0.16163522	0.20983972
2	0.11241631	0.1	0.14998985	0.15705011	0.16163522	0.20983972	0.32929600
3	0.1	0.14998985	0.15705011	0.16163522	0.20983972	0.32929600	0.29594238
4	0.14998985	0.15705011	0.16163522	0.20983972	0.32929600	0.29594238	0.30872387
5	0.15705011	0.16163522	0.20983972	0.32929600	0.29594238	0.30872387	0.34102251
6	0.16163522	0.20983972	0.32929600	0.29594238	0.30872387	0.34102251	0.31606816
7	0.20983972	0.32929600	0.29594238	0.30872387	0.34102251	0.31606816	0.37685128
8	0.32929600	0.29594238	0.30872387	0.34102251	0.31606816	0.37685128	0.36366402

Gambar 4. 14 Implementasi Antarmuka Submenu Data Uji

4.8.2 Implementasi Antarmuka Pelatihan

Untuk implementasi halaman pelatihan ini memproses data latih yang telah dilakukan proses *normalization* sebelumnya, selanjutnya data tersebut dioleh oleh atribut-atribut pada masing-masing metode agar didapatkan nilai yang memiliki nilai yang baik dalam menentukan pola peramalan. Dengan detail dan macam atribut seperti yang ditampilkan pada **Gambar 4.15**.

Peramalan Jumlah Penumpang PT KAI

Olah Data | Pelatihan | Pengujuan

*Backpropagation | *Algoritma Genetika (Guaikan Optimasi)

Jumlah Iterasi: 2 | Alpha: 0.9 | Pop Size: Cr | Generasi: Mr

Load Bobot | Proses

Bobot dan bias (v) awal:

v1	v2	v3	v4	bias (v0)
0.8210012568	0.8213869125	-0.459531969	-0.779350714	0.6282918001
0.5343790479	0.7808799426	-0.797917995	0.9408949128	-0.540841269
-0.781134636	0.8693466447	0.1712075452	-0.563308870	-0.319637071
0.2698949378	-0.132517935	-0.386738186	-0.661036576	-0.075600956

MSE:

Iterasi	MSE

Bobot dan bias (w) awal:

w1	w2	w3	w4	bias (w0)
-0.279038165	0.1000609895	0.4307744157	0.8861218681	0.3831159866

Bobot dan bias (v) baru:

v1	v2	v3	v4	bias (v0)
0.8210012568	0.8213869125	-0.459531969	-0.779350714	0.6282918001
0.5343790479	0.7808799426	-0.797917995	0.9408949128	-0.540841269
-0.781134636	0.8693466447	0.1712075452	-0.563308870	-0.319637071
0.2698949378	-0.132517935	-0.386738186	-0.661036576	-0.075600956
0.0830181002	0.5508119699	-0.544800362	0.5076510034	0.0

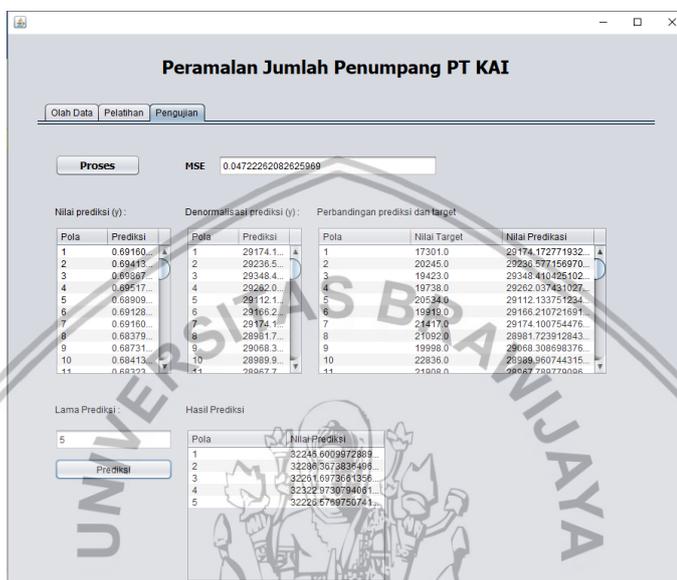
Bobot dan bias (w) baru:

w1	w2	w3	w4	bias (w0)
-0.279038165	0.1000609895	0.4307744157	0.8861218681	0.3831159866

Gambar 4. 15 Implementasi Antarmuka Pelatihan

4.8.3 Implementasi Antarmuka Pengujian

Implementasi halaman pengujian akan mengolah data uji yang telah dilakukan *normalization* pada menu olah data, dan dengan menggunakan komposisi atribut yang sudah dilakukan pelatihan pada halaman pelatihan, selanjutnya dengan pola tersebut digunakan dalam ngolah data uji untuk mendapatkan nilai peramalan. Dengan representasi hasil yang ditampilkan pada **Gambar 4.6**.



Gambar 4. 16 Implementasi Antarmuka Pengujian



BAB 5 PENGUJIAN DAN ANALISIS

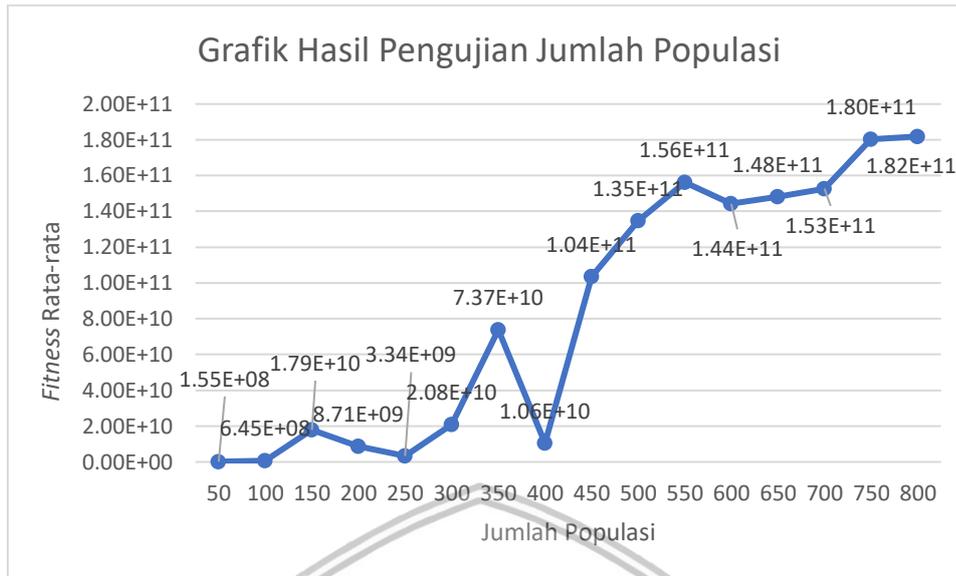
Prediksi jumlah penumpang kereta api pada PT KAI, dengan menggunakan metode *backpropagation* dan dioptimasi dengan menggunakan algoritme genetika akan dibahas pada pembahasan bab pengujian dan analisis ini. Sehingga dapat menentukan tingkat atau besaran variabel yang tepat dan dapat menghasilkan informasi yang optimal dalam penerapan sistem peramalan yang dilakukan.

5.1 Pengujian Jumlah Populasi

Untuk memilih nilai populasi yang tepat pengujian dibantu oleh beberapa parameter yang ditetapkan, antara lain generasi sebanyak 10 generasi, dengan nilai $Cr = 0.2$ dan $Mr = 0.2$, dan dengan nilai $alpha$ sebesar 0.9, yang dilakukan pengujian populasi sebanyak 50 dan kelipatannya, sampai mencapai nilai 800. Yang memanfaatkan perulangan pengujian sebanyak 10 kali. Dengan hasil pengujian akan ditunjukkan pada **Tabel 5.1** dan **Gambar 5.1**.

Tabel 5. 1 Pengujian Jumlah Populasi

Pop size	Percobaan ke- <i>n</i>			Fitness Rata-rata
	1	...	10	
50	6.92E+06	...	5.11E+08	1.55E+08
100	5.45E+08	...	8.03E+07	6.45E+08
150	3.71E+08	...	1.31E+11	1.79E+10
200	9.04E+08	...	2.07E+08	8.71E+09
250	2.09E+08	...	2.30E+10	3.34E+09
300	1.65E+10	...	2.03E+09	2.08E+10
350	6.69E+10	...	1.90E+11	7.37E+10
400	9.43E+09	...	4.07E+10	1.06E+10
450	1.74E+10	...	3.40E+09	1.04E+11
500	9.45E+10	...	4.13E+11	1.35E+11
550	1.18E+10	...	8.39E+08	1.56E+11
600	3.94E+10	...	2.51E+10	1.44E+11
650	4.78E+10	...	5.30E+10	1.48E+11
700	4.92E+10	...	5.27E+10	1.53E+11
750	1.02E+10	...	5.13E+11	1.80E+11
800	2.16E+11	...	5.88E+10	1.82E+11



Gambar 5. 1 Grafik Hasil Pengujian Jumlah Populasi

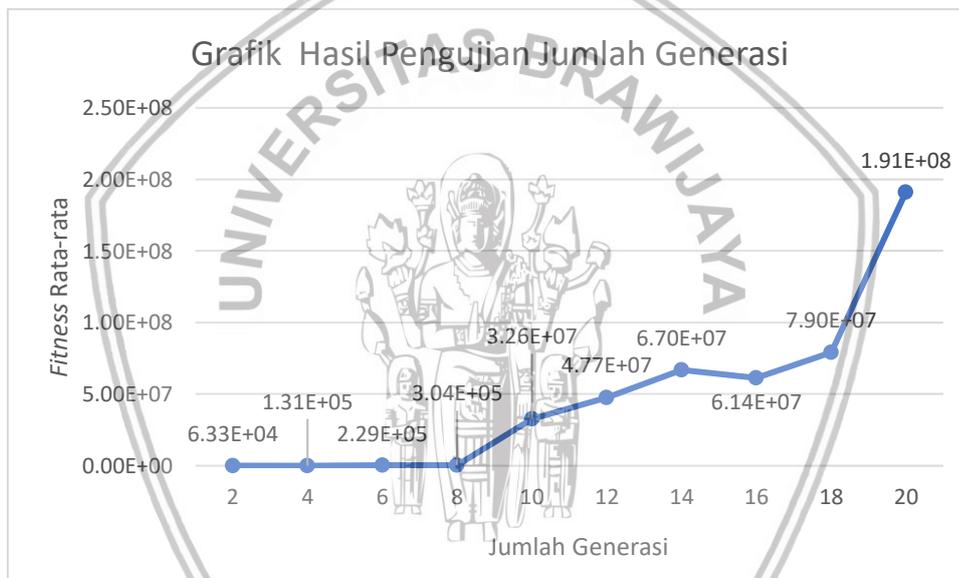
Hasil yang didapat dari pengujian yang dilakukan untuk menentukan jumlah populasi yang baik dalam melakukan prediksi pada studi kasus peramalan jumlah pengguna kereta api di Indonesia, menghasilkan bahwa semakin banyak populasi yang digunakan maka semakin besar peluang untuk menghasilkan nilai *fitness* yang tinggi, terbukti dalam penelitian ini, nilai *fitness* yang tertinggi berada pada jumlah populasi yang berada pada titik 800, namun seperti yang ditunjukkan pula pada **Gambar 5.1** bahwa pada titik 400 memiliki nilai *fitness* tidak lebih baik daripada nilai *fitness* pada titik 300 dan 350, hal ini mungkin saja terjadi disebabkan proses pencarian dengan menggunakan populasi sejumlah 400 tidak atau belum menemukan nilai terbaiknya dan juga dapat disebabkan variabel dalam pengujian kurang memungkinkan hal tersebut terjadi, sehingga hal itu sangat mungkin terjadi, selain itu didukung juga pada data yang ditampilkan pada **Tabel 5.1** bahwa terkadang pada percobaan tertentu memang mendapatkan hasil yang optimal, namun tidak menjamin tren tersebut akan sering muncul pada percobaan-percobaan selanjutnya, sehingga mempengaruhi nilai rata-rata dari pengujian yang dilakukan.

5.2 Pengujian Banyak Generasi

Pada pengujian banyak generasi memanfaatkan parameter nilai $Cr = 0.2$ dan $Mr = 0.2$, dan dengan nilai α sebesar 0.9, dan dengan jumlah populasi sebanyak 10 data, dengan pengujian menggunakan nilai generasi mulai dari 2 dan kelipatannya, sampai menapai 20, yang dilakukan sebanyak 10 kali percobaan. Dengan hasil yang direpresentasikan seperti pada **Tabel 5.2** dan **Gambar 5.2**.

Tabel 5. 2 Pengujian Banyak Generasi

Generasi	Percobaan ke- <i>n</i>			Fitness Rata-rata
	1	...	10	
2	1766.354725	...	177677.1411	6.33E+04
4	68812.12972	...	46502.60283	1.31E+05
6	8436.171225	...	607.885809	2.29E+05
8	1323824.35	...	31963.55622	3.04E+05
10	5533242.519	...	90.97696481	3.26E+07
12	72636.51051	...	1538.13202	4.77E+07
14	2.84E+06	...	641382629.5	6.70E+07
16	8.63E+07	...	1.42E+08	6.14E+07
18	4898.486626	...	110514.5193	7.90E+07
20	17289.21137	...	120436.5335	1.91E+08



Gambar 5. 2 Grafik Hasil Pengujian Jumlah Generasi

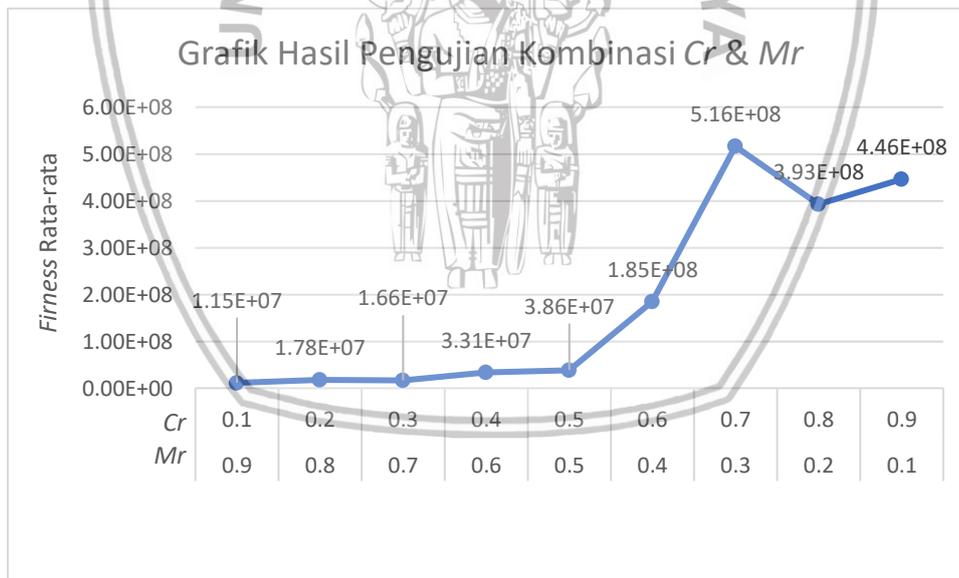
Pada **Gambar 5.2**, menunjukkan pergerakan pola nilai *fitness* hampir berbanding lurus dengan jumlah generasi yang digunakan, bahkan pada generasi ke-20 yang menghasilkan nilai *fitness* tertinggi memungkinkan untuk mempertahankan tren tersebut. Dengan hasil tersebut, menunjukkan jumlah generasi sangat mempengaruhi pada besar *fitness* yang dihasilkan, namun kasus yang serupa dari kasus yang terjadi pada proses pengujian populasi juga terjadi pada proses pengujian jumlah generasi, yakni seperti yang terlihat pada **Tabel 5.2** menunjukkan bahwa tidak semua percobaan akan absolut berbanding lurus, namun hanya menghasilkan tren rata-rata yang berbanding lurus.

5.3 Pengujian Kombinasi Cr dan Mr

Kombinasi Cr dan Mr dilakukan pengujian secara terbalik antara *ascending* dan *descending* dengan *range* nilai antara (0.1,0.9), dengan menggunakan parameter lain berupa nilai *alpha* sebesar 0.9, jumlah populasi sebesar 10, dan dengan 10 generasi serta dilakukan pengujian sebanyak 10 kali. Dengan perolehan nilai pengujian seperti pada **Tabel 5.3** dan **Gambar 5.3**.

Tabel 5. 3 Pengujian Kombinasi Cr dan Mr

Cr	Mr	Percobaan ke- n			Fitness Rata-rata
		1	...	10	
0.1	0.9	3.02E+07	...	488617.7203	1.15E+07
0.2	0.8	1.08E+08	...	1004.76105	1.78E+07
0.3	0.7	21664.78747	...	975576.9616	1.66E+07
0.4	0.6	2.67E+08	...	1.17E+07	3.31E+07
0.5	0.5	1.63E+07	...	69808.91291	3.86E+07
0.6	0.4	8.25E+08	...	1.65E+08	1.85E+08
0.7	0.3	2.48E+09	...	509.8165708	5.16E+08
0.8	0.2	2174.820026	...	5.72E+07	3.93E+08
0.9	0.1	6.63E+08	...	85044.22661	4.46E+08



Gambar 5. 3 Grafik Hasil Pengujian Kombinasi Cr & Mr

Bedasarkan perolehan yang dihasilkan pada **Tabel 5.3** meskipun pada percobaan tertentu terkadang nilai penentu yang baik dihasilkan pada titik dimana nilai Mr berjumlah lebih besar daripada nilai Cr , namun tren tersebut tidak selalu terjadi, dan bahkan pada penelitian dengan studi kasus peramalan jumlah penumpang kereta api Indonesia ini, menghasilkan perolehan yang berbeda. Dengan hasil yang direpresentasikan pada **Gambar 5.3** tren yang menghasilkan nilai *fitness* terbaik muncul saat nilai Cr lebih besar daripada nilai Mr , lebih

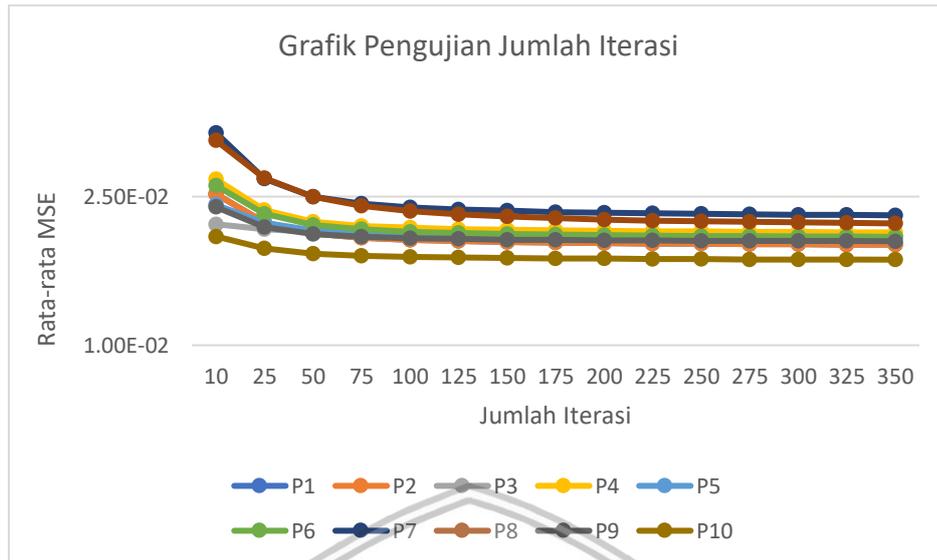
tepatnya pada titik nilai $Cr = 0.7$ dan nilai $Mr = 0.3$, tidak hanya itu, pada **Gambar 5.3** juga menunjukkan tren nilai *fitness* jauh membaik mulai dari nilai Cr itu lebih besar daripada nilai Mr , meskipun pada nilai $Cr = 0.8$ dan $Mr = 0.2$ sempat terjadi penurunan nilai, namun pola *fitness* yang naik tidak berubah, berdasarkan hal tersebut menunjukkan bahwa nilai Cr ini mempengaruhi dalam menghasilkan nilai *fitness* dalam perhitungan algoritme genetika ini.

5.4 Pengujian Jumlah Iterasi

Epoch atau yang biasa disebut dengan iterasi diuji dengan menggunakan parameter jumlah data latih sebesar 100 dan 10 data uji, dengan menggunakan nilai *alpha* sebesar 0.9, dengan iterasi yang diujikan bernilai awal 10 dan selanjutnya dengan 25 kelipatannya, sampai dengan 350 iterasi, dan dilakukan sebanyak 10 kali percobaan untuk mengetahui nilai rata-rata MSE-nya. Dengan perolehan nilai pengujian seperti pada **Tabel 5.4** dan **Gambar 5.4**.

Tabel 5. 4 Pengujian Jumlah Iterasi

Iterasi	Percobaan ke- <i>n</i>		
	1	...	10
10	0.025251504	...	0.020979449
25	0.022393472	...	0.019747488
50	0.021272167	...	0.01923104
75	0.020859896	...	0.019027238
100	0.02064425	...	0.018915052
125	0.020511525	...	0.018843213
150	0.02042163	...	0.018793044
175	0.020356757	...	0.018755965
200	0.02030777	...	0.018727431
225	0.020269493	...	0.018704795
250	0.020238773	...	0.018686403
275	0.020213582	...	0.018671168
300	0.020192553	...	0.018658345
325	0.020174738	...	0.018647406
350	0.020159452	...	0.018637966



Gambar 5. 4 Grafik Hasil Pengujian Jumlah Iterasi

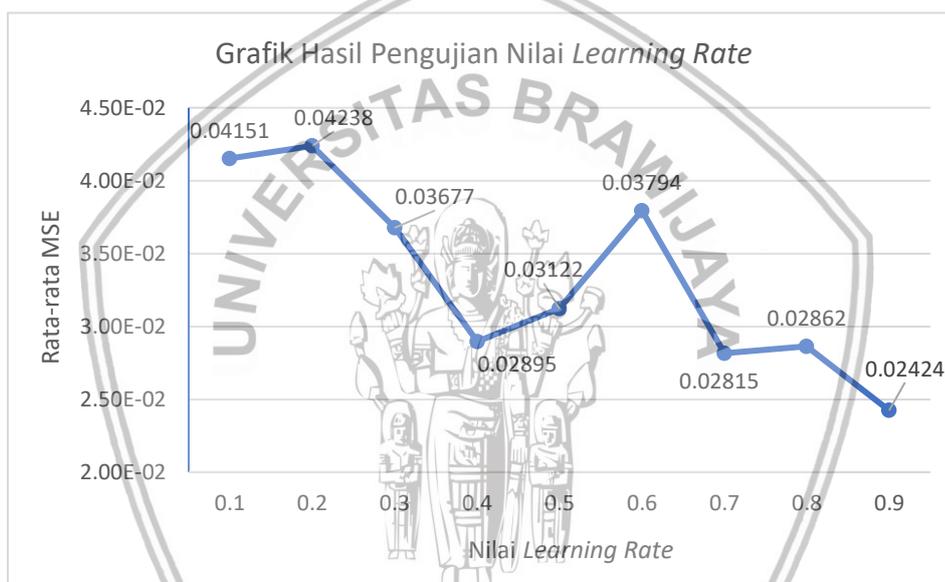
Berdasarkan uji coba besar jumlah perulangan yang seperti pada **Tabel 5.4** dan **Gambar 5.4** menghasilkan hasil yang stabil, terlihat pada proses peramalan informasi menggunakan 10 percobaan, memiliki tren yang hampir sama, seperti yang ditunjukkan pada percobaan *p1* sampai *p6* dan *p9*, *p10*, menghasilkan penurunan nilai MSE yang baik terjadi pada iterasi ke-75 sedangkan pada percobaan *p7* dan percobaan *p8*, menghasilkan penurunan nilai MSE yang membaik berada pada iterasi ke-100, dan berangsur-angsur pada semua percobaan menghasilkan nilai yang sama, sehingga pada iterasi ke-100 ini dapat dikatakan sebagai titik balik dari pencarian jumlah iterasi yang baik untuk menghasilkan solusi yang baik dalam peramalan suatu informasi. Berdasarkan hasil pengujian yang dilakukan dapat ditarik suatu kesimpulan bahwa semakin besar suatu iterasi pada pencarian solusi dalam metode ANN-BP maka hasil kesalahan yang terjadi akan semakin mengecil, namun pada titik tertentu penurunan kesalahan tersebut akan mulai stabil dan tidak terjadi perubahan yang signifikan sehingga dapat digunakan sebagai patokan yang paling efektif dalam menghasilkan suatu solusi.

5.5 Pengujian Nilai Alpha

Dengan menggunakan parameter yang sama dengan pengujian pada pengujian jumlah iterasi, yakni dengan menggunakan data latih sebanyak 100 dan data uji sebanyak 10, dengan iterasi sebanyak 10 kali, dan dengan parameter nilai *alpha* yang diuji memiliki rentang nilai antara (0.1,0.9), yang dilakukan sebanyak 10 kali percobaan. Dengan hasil pengujian akan ditunjukkan pada **Tabel 5.5** dan **Gambar 5.5**.

Tabel 5. 5 Pengujina Nilai *Alpha*

<i>Alpha</i>	Percobaan ke- <i>n</i>			Rata-rata MSE
	1	...	10	
0.1	0.0468266	...	0.0367718	0.0415057
0.2	0.0472402	...	0.0382427	0.0423816
0.3	0.0430346	...	0.0321924	0.0367656
0.4	0.0399704	...	0.0238269	0.0289548
0.5	0.0453919	...	0.0237496	0.0312248
0.6	0.0470315	...	0.0326475	0.0379417
0.7	0.0484848	...	0.0219497	0.0281518
0.8	0.0453497	...	0.0236434	0.0286190
0.9	0.0359332	...	0.0201550	0.0242369



Gambar 5. 5 Grafik Hasil Pegujian Nilai Learning Rate

Berdasarkan **Tabel 5.5** menunjukkan hasil dari penggunaan nilai *alpha* yang baik dalam peramalan memiliki tren yang cukup stabil anatar percobaan ke-1 sampai dengan percobaan ke-10, sehingga menghasilkan data yang bahkan sudah dapat disimpulkan saat percobaan ke-1, sehingga meskipun dilakukan pengujian 10 kali tren rata-rata nilai dari MSE yang dihasilkan dari pengaruh nilai *learning rate* tidak terlalu mempengaruhi. Dan berdasarkan hasil pengujian seperti yang terlihat pada **Gambar 5.5**, hasil MSE terbaik dihasilkan pada *learning rate* dititik 0.9, berdasarkan dari hasil yang didapat, dapat diambil suatu kesimpulan bahwa penentuan nilai *learning rate* hanya bergantung pada pola data yang digunakan, hal ini dikarenakan perubahan nilai antar percobaan tidak berubah terlalu signifikan yang biasanya harus distabilkan dengan variabel-variabel lain sehingga pola dapat terlihat lebih teratur dan stabil.



5.6 Pengujian Perbandingan Hasil Prediksi ANN-BP dan Algoritme Genetka

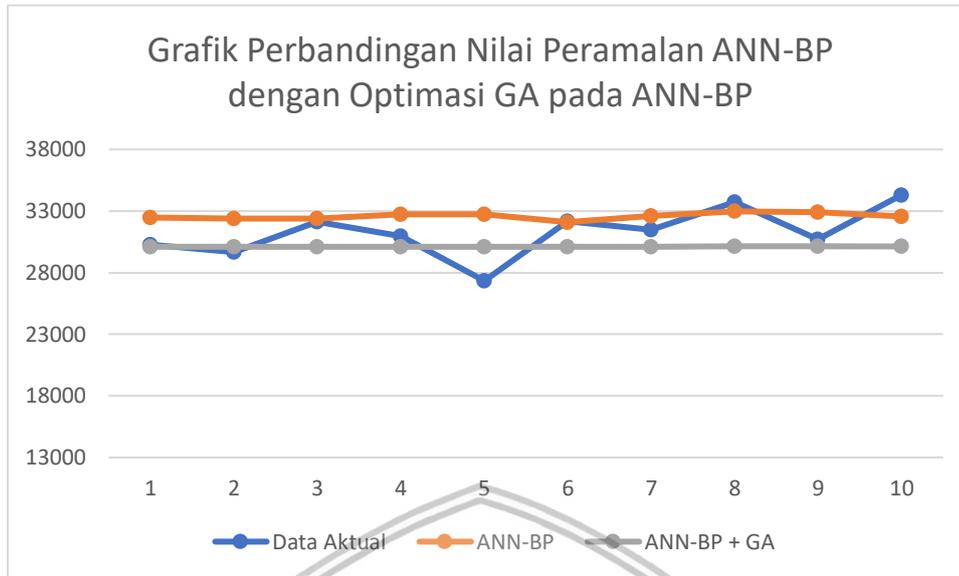
Dengan menggunakan berbagai parameter yang dilakukan pengujian, untuk menentukan metode yang paling sesuai berdasarkan permasalahan yang peramalan jumlah penumpang kereta api PT.KAI. Perbandingan ini berupa perbandingan metode peramalan dengan menggunakan metode ANN-BP atau dengan metode ANN-BP yang dioptimasi dengan algoritme genetika. Dengan hasil pengujian seperti pada **Tabel 5.6**, **Tabel 5.7** dan **Gambar 5.6**.

Tabel 5. 6 Perbandingan Nilai Error MSE Menggunakan dan Tanpa Optimasi

Pelatihan	Percobaan ke- <i>n</i>			Rata-rata MSE
	1	...	10	
ANN-BP	0.078340546	...	0.069969206	0.072517977
ANN-BP + GA	0.065392717	...	0.069618904	0.065869861

Tabel 5. 7 Pengujian Coba perbandingan Hasil ANN-BP dan Algoritme Genetika

Data Aktual	ANN-BP	ANN-BP + GA	Selisih ANN-BP	Selisih ANN-BP + GA
30263.000	32491.091	30103.059	2228.091	159.941
29690.000	32399.609	30104.747	2709.609	414.747
32150.000	32408.789	30105.675	258.789	2044.325
30949.000	32745.815	30115.785	1796.815	833.215
27342.000	32714.818	30122.101	5372.818	2780.101
32170.000	32102.314	30113.026	67.686	2056.974
31502.000	32626.204	30117.512	1124.204	1384.488
33745.000	32974.523	30124.213	770.477	3620.787
30723.000	32914.038	30130.029	2191.038	592.971
34310.000	32579.800	30130.132	1730.200	4179.868
		MAD	1824.973	1806.742



Gambar 5. 6 Grafik Perbandingan Nilai Peramalan ANN-BP dengan Optimasi GA pada ANN-BP

Berdasarkan hasil yang diperoleh dari perbandingan peramalan informasi tanpa menggunakan optimasi bobot (w) menghasilkan nilai kesalahan sebesar 0.072517977, sedangkan jika bobot (w) dilakukan proses optimasi dengan menggunakan algoritme genetika dapat mencapai nilai MSE sebesar 0.065869861, seperti pada hasil yang dijabarkan pada **Tabel 5.6**, selain itu juga dari hasil perbandingan penggunaan metode tersebut menghasilkan nilai *Median absolute deviation* (MAD) pada metode ANN-BP dengan optimasi bobot (w) menggunakan algoritme genetika menghasilkan nilai sebesar 1806.74 yang nilainya lebih kecil dari pada tanpa menggunakan optimasi, yang juga telah dijabarkan pada **Tabel 5.7**. Dan berdasarkan **Gambar 5.6** menunjukkan bahwa pola dari data permasalahan merupakan jenis pola *trend*, yang mengartikan bahwa pola terdiri dari data dengan pola yang memiliki jangkauan waktu yang panjang sehingga terbentuklah pola *trend* yang dihasilkan dari perbandingan nilai aktual dan pola hasil prediksi.

BAB 6 PENUTUP

6.1 Kesimpulan

Hasil kesimpulan yang didapat dari penelitian peramalan jumlah penumpang kereta api PT.KAI, dengan menggunakan metode jaringan saraf tiruan *backpropagation* serta menggunakan optimasi menggunakan algoritme genetika, menghasilkan kesimpulan pokok sebagai berikut:

1. Dengan menggunakan beberapa parameter penentu tingkat keakurasian hasil pengoperasian sistem dengan menggunakan metode ANN-BP dan dengan menggunakan algoritme genetika, berupa penentuan nilai iterasi pada metode ANN-BP dengan hasil nilai *iterasi* yang optimum berada pada nilai 100 dan nilai *learning rate* yang optimum dengan nilai 0.9, Sedangkan parameter penentu dari proses algoritme genetika hasil pencarian solusi yang optimum dengan memakai parameter banyak populasi yang digunakan yakni sebanyak 800, jumlah generasi sebanyak 20, dan dengan kombinasi nilai $Cr = 0.7$ dan $Mr = 0.3$.
2. Dengan menggunakan beberapa parameter yang telah diuji, dengan menghasilkan nilai yang optimal, menunjukkan bahwa metode ANN-BP dan menggunakan optimasi bobot (w_i) dengan menggunakan algoritme genetika ini dapat digunakan dalam peramalan jumlah penumpang kereta api di PT.KAI, hal ini terbukti dengan hasil kesalahan (MSE) yang dihasilkan dari proses peramalan dengan menggunakan metode ANN-BP dan optimasi bobot menggunakan algoritme genetika ini menghasilkan nilai 0.065869861, sekaligus membuktikan dengan hasil nilai MSE yang dihasilkan menunjukkan bahwa metode peramalan hanya dengan menggunakan metode ANN-BP saja tidak lebih baik dengan nilai MSE sebesar 0.072517977 yang dibandingkan dengan ANN-BP yang bobotnya (w) telah dioptimasi dengan menggunakan algoritme genetika.

6.2 Saran

Penelitian peramalan dengan menggunakan metode ANN-BP dan optimasi nilai bobot (w) menggunakan algoritme genetika ini menghasilkan hasil yang cukup baik, namun tingkat keakurasian dari penelitian ini dapat lebih dioptimasi, dari berbagai pendekatan lain yang berbeda, yang dapat digunakan sebagai saran untuk penelitian yang serupa selanjutnya, diantaranya dengan penambahan variabel yang lebih beragam, sehingga memperkaya kemungkinan-kemungkinan yang dapat terjadi dan dapat dicari pola solusinya, selain itu dapat juga dengan melakukan metode lain yang juga terbilang unggul dalam hal peramalan suatu informasi.

DAFTAR PUSTAKA

- Arkeman, Y., Seminar, K. B., & gunawan, H. (2012). *Algoritma Genetika, Teori dan Aplikasinya untuk Bisnis dan Industri*. Bogor: IPB Perss.
- 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*, 19-23.
- Davinsyah, G., Mahmudy, W. F., & Perdana, R. S. (2017). Pelatihan Multi-Layer Neural Network Menggunakan Algoritma Genetika untuk Memprediksi Harga Saham Esok Hari (T+1). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2302-2308.
- Gani, E., Kolibu, H. S., & Tamuntuan, G. H. (2016). Pemanfaatan Logika Fuzzy Untuk Sistem Prediksi Banjir. *JURNAL MIPA UNSRAT*, 81-84 .
- Heaton, J. (2008). *Introduction to Neural Networks with Java*. Missouri: Heaton Research, Inc.
- 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.
- Luthfianto, R., Santoso, I., & Christiyono, Y. (2011). Peeamalan Jumlah Penumpang Kereta Api Dengan Jaringan Saraf Tituan Metode Perambatan Balik (Backptopagation) . *Reserarch gate*, 1-9.
- Mahmudy, W. F. (2015). *Dasar-Dasar Algoritma Evolusi*. Malang: Program Teknologi Informasi dan Ilmu Komputer (PTIHK) Universitas Brawijaya.
- Melin, P. P., Castillo Oscar. Prof., D., Ramirez, E. G., Kacprzyk, J. P., & Pedrycz, W. P. (2007). *Analysis and Design of Intelligent Systems Using Soft Computing Techniques*. Condesa: Springer.
- Nugraha, H. G., & Azhari. (2014). Pelatihan Bobot Jaringan Syaraf Tiruan Menggunakan Particle Swarm Optimization untuk Peramalan Tingkat Inflasi. *Journal of Mathematics and Natural Sciences*, 292-302.
- Rahardiani, N. O., Mahmudy, W. F., & Indriati. (2018). Optimasi Bobot Multi-Layer Perceptron Menggunakan Algoritma Genetika Untuk Klasifikasi Tingkat Resiko Penyakit Stroke. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2352-2360.

- Saleh, A. (2015). Implementasi Metode Klasifikasi Naïve Bayes. *Citec Journal*, 207.
- Suharyadi, & S.K, P. (2009). *Statistika : Untuk ekonomi dan Kauangan Modern, Edisi 2*. Jakarta: Salemba Empat.
- Ukhra, A. U. (2014). Pemodelan dan Peramalan Data Deret Waktu dengan Metode Seasonal ARIMA. *Jurnal Matematika UNAND*, 59-67.
- Watimena, R. A. (2008). *Filsafat Dan Sains (Sebuah Pengantar)*. Jakarta: Grasindo.
- Wiharto, Palgunadi, Y., & Nugroho, M. A. (2013). Analisis Penggunaan Algoritma Genetika Untuk Perbaikan Jaringan Syaraf Tiruan Radial Basis Function. *Seminar Nasional Teknologi Informasi dan Komunikasi 2013 (SENTIKA 2013)* (hal. 181-188). Surakarta: Riset Group Ilmu Rekayasa dan Komputasi FMIPA Universitas Sebelas Maret.
- Yamanashi, Y., Umeda, K., & Yoshikawa, N. (2013). Pseudo Sigmoid Function Generator for a Superconductive Neural Network. *IEEE Transactions on Applied Superconductivity*.
- Yohannes, E., Mahmudy, W. F., & Rahmi, A. (2015). Penentuan Upah Minimum Kota Berdasarkan Tingkat Inflasi Menggunakan Backpropagation Neural Network (BPNN). *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 34-40.
- Zeniarja, J., & Luthfiarta, A. (2015). Prediksi Churn dan Segmentasi Pelanggan Menggunakan Backpropagation Neural Network Berbasis Evolution Strategies. *Techno.Com*, 49-54.
- Zulfikar, & Budiantara, I. N. (2014). *Manajemen Dengan Pendekatan Komputasi Statistika*. Yogyakarta: Deepublish.