

**OPTIMASI FUZZY TIME SERIES MENGGUNAKAN ALGORITME
PARTICLE SWARM OPTIMIZATION UNTUK PERAMALAN
PRODUK DOMESTIK BRUTO (PDB) INDONESIA**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Dloifur Rohman Alghifari
NIM: 135150201111048



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



PENGESAHAN



PERNYATAAN ORISINALITAS

PERNYATAAN ORISINALITAS

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

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

Malang, 2 Januari 2019

Dloifur Rohman Alghifari

NIM: 135150201111048



Malang, 2 Januari 2019

Berita

signature201@gmail.com

PRAKATA

Penulis senantiasa panjatkan puji dan syukur atas kehadiran Allah SWT yang telah melimpahkan rahmat serta hidayahnya, sehingga penulis dapat skripsi ini dengan judul **Optimasi Fuzzy Time Series Menggunakan Algoritme Particle Swarm Optimization Untuk Peramalan Produk Domestik Bruto (PDB) Indonesia**. Tidak lupa shalawat serta salam penulis panjatkan kepada jujungan kita nabi Muhammad SAW. Penulis mengucapkan terima kasih atas kepada semua pihak yang telah memberikan bantuan baik berupa bimbingan, semangat, saran dan kritik serta doa. Sehingga penulis ucapkan terima kasih sebanyak-banyaknya kepada:

1. Bapak Bayu Rahayudi, S.T, M.T, selaku pembimbing skripsi 1 atas bimbingan, waktu serta kritik dan saran yang telah diberikan kepada penulis.
2. Ibu Candra Dewi, S.Kom, M.Sc, selaku pembimbing skripsi 2 atas bimbingan, waktu serta kritik dan saran yang telah diberikan kepada penulis.
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, selaku dekan Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D, selaku ketua jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Agus Wahyu Widodo, S.T, M.Cs, selaku ketua Program Study Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
6. Kedua orang tua penulis bapak Asad, ibu Khotimah dan semua ketiga kakak kandung saya serta semua keluarga yang telah memberikan semangat, doa dan dukungan baik berupa materil maupun moril.
7. Teman-teman program study Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan semangat dan saran kepada penulis.
8. Dan semua pihak yang tidak dapat disebutkan satu persatu, yang telah terlibat langsung maupun tidak langsung dalam menyelesaikan skripsi ini.

Penulis menyadari bahwa dalam penulisan skripsi ini masih banyak kekurangan dan dapat dikatakan jauh dari kata sempurna, oleh karena itu penulis mengharapkan pembaca dapat memberikan saran dan kritik yang sifatnya membangun.

Malang, 2 Januari 2019

Penulis

alghifari2203@gmail.com

ABSTRAK

Dloifur Rohman Alghifari, Optimasi *Fuzzy Time Series* Menggunakan Algoritme *Particle Swarm Optimization* Untuk Peramalan Produk Domestik Bruto (PDB) Indonesia.

Pembimbing: Bayu Rahayudi, S.T, M.T. dan Candra Dewi, S.Kom, M.Sc.

Sebagai salah satu indikator masukan pada penyusunan rancangan program-program pembangunan. Peramalan Produk Domestik Bruto (PDB) ini diharapkan dapat memberikan gambaran mengenai pertumbuhan ekonomi dan kinerja perekonomian di Indonesia. Sumber data PDB biasanya berasal dari hasil survei maupun dari catatan administrasi dari berbagai Lembaga intansi. Terkadang terdapat permasahan sumber data kurang lengkap atau tidak tersedia saat perhitungan nilai PDB, sehingga diperlukan suatu cara untuk memperkirakan nilai PDB agar dapat menggambarkan keadaan sebenarnya. Untuk meningkatkan akurasi peramalan digunakan optimasi interval pada *fuzzy time series* menggunakan *particle swarm optimization* (PSO). Berdasarkan hasil parameter terbaik yang didapat, tingkat kesalahan peramalan yang dihasilkan menggunakan MAPE sebesar 2,48% dari 10 data uji. Hasil tersebut menunjukkan kemampuan peramalan yang baik dengan tingkat kesalahan yang rendah. Dilakukan perbandingan hasil peramalan untuk metode usulan sedikit lebih baik dari metode *fuzzy time series* dengan penentuan interval berbasis rata-rata dengan MAPE 2,66%. Tetapi tidak lebih baik dari metode regresi linier dengan MAPE 1,52%.

Kata kunci: produk domestic bruto, peramalan, fuzzy time series, particle swarm optimization.



ABSTRACT

Dloifur Rohman Alghifari, Fuzzy Time Series Optimization Using Particle Swarm Optimization Algorithm for Forecasting Indonesia Gross Domestic Product (GDP).

Pembimbing: Bayu Rahayudi, S.T., M.T. dan Candra Dewi, S.Kom., M.Sc.

As one of the input indicators for development programs. This Gross Domestic Product (GDP) forecasting is expected to provide information about economic growth and performance in Indonesia. Data sources of GDP usually come from survey results or from administrative records from various institutions. Sometimes the source data is incomplete or not available when calculating GDP values, it must be determined how to calculate the GDP value so that it can be used to estimate GDP forecasting using fuzzy time series. To improve forecasting accuracy, we use fuzzy time series optimization intervals using particle swarm optimization (PSO). Based on the parameters obtained, the forecasting error rate generated using MAPE is 2.48% of the 10 test data. These results indicate good forecasting ability with a low error rate. The comparison of forecasting results for the proposed method is slightly better than the fuzzy time series method with the determination of the average interval based on MAPE 2.66%. But it is no better than the linear regression method with MAPE 1.52%

Keywords: gross domestic product, forecasting, fuzzy time series, particle swarm optimization.



DAFTAR ISI

PENGESAHANii
PERNYATAAN ORISINALITASiii
PRAKATA.....	.iv
ABSTRAK.....	.v
ABSTRACT.....	.vi
DAFTAR ISI.....	.vii
DAFTAR TABEL.....	.x
DAFTAR GAMBAR.....	.xii
DAFTAR LAMPIRANxiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Produk Domestik Bruto.....	8
2.3 Peramalan	9
2.4 Fuzzy Time Series.....	9
2.5 Particle Swarm Optimization	11
2.6 Nilai Evaluasi	12
BAB 3 METODOLOGI	13
3.1 Studi Pustaka.....	13
3.2 Pengumpulan Data	14
3.3 Perancangan Algoritme	14
3.4 Implementasi Algoritme	14
3.5 Pengujian Algoritme	14
3.6 Kesimpulan.....	19

BAB 4 PERANCANGAN.....	20
4.1 Formulasi Permasalahan.....	20
4.2 Siklus Model Fuzzy Time Series dengan Particle Swarm Optimization	21
4.2.1 Menentukan Himpunan Semesta	22
4.2.2 Menentukan Partikel Awal.....	23
4.2.3 Hitung <i>Fitness</i> Menggunakan FTS	23
4.2.4 Menentukan <i>pbest</i>	30
4.2.5 Menentukan <i>gbest</i>	31
4.3 Perhitungan Manual	31
4.3.1 Perhitungan Manual Optimasi <i>Fuzzy Time Series</i> Menggunakan <i>Particle Swarm Optimization</i>	32
4.3.2 Perhitungan Manual Nilai <i>Fitness</i> Menggunakan <i>Fuzzy Time Series</i>	38
4.4 Rancangan Antarmuka	46
4.4.1 Rancangan Antarmuka Halaman Home	46
4.4.2 Rancangan Antarmuka Halaman Optimasi	47
4.4.3 Rancangan Antarmuka Halaman Hasil Peramalan	48
BAB 5 IMPLEMENTASI	49
5.1 Implementasi Algoritme	49
5.1.1 Implemtasi Algoritme PSO	49
5.1.2 Implementasi Algoritme <i>Fuzzy Time Series</i> (FTS).....	54
5.2 Implementasi Antarmuka	60
5.2.1 Implementasi Antarmuka Halaman Home	60
5.2.2 Implementasi Antarmuka Halaman Optimasi.....	61
5.2.3 Implementasi Antarmuka Halaman Hasil Peramalan	62
BAB 6 PENGUJIAN DAN ANALISIS.....	63
6.1 Hasil dan Pengujian Jumlah Iterasi Maksimum	63
6.2 Hasil dan Analisis Pengujian Panjang Dimensi Partikel	64
6.3 Hasil dan Analisis Pengujian Jumlah Partikel	65
6.4 Hasil dan Analisis Pengujian Kombinasi Nilai variabel c1 dan c2	67
6.5 Hasil dan Analisis Pengujian Nilai Variabel Bobot Inersia.....	68
6.6 Pengujian Tingkat Kesalahan Algoritme	70

BAB 7 KESIMPULAN.....	73
7.1 Kesimpulan.....	73
7.2 Saran	73
DAFTAR PUSTAKA.....	74
LAMPIRAN A DATA PDB	76



DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian Terdahulu dengan usulan	6
Tabel 3.1 Perancangan Pengujian Jumlah Iterasi Maksimum	15
Tabel 3.2 Perancangan Pengujian Panjang Dimensi Partikel.....	15
Tabel 3.3 Percangan pengujian banyak partikel	16
Tabel 3.4 Perancangan Pengujian Kombinasi Nilai c1 dan c2.....	17
Tabel 3.5 Perancangan Pengujian Bobot Inersia	18
Tabel 3.6 Perancangan Pengujian Tingkat Kesalahan Agoritme.....	18
Tabel 3.7 Perancangan Pengujian Perbandingan MAPE.....	19
Tabel 4.1 Tabel data aktual total PDB Indonesia menurut lapangan usaha.....	20
Tabel 4.2 Data Latih	32
Tabel 4.3 Inisialisasi Kecepatan Awal	33
Tabel 4.4 Inisialisasi Partikel Awal.....	34
Tabel 4.5 Hasil Hitung <i>Fitness</i> Iterasi 0	34
Tabel 4.6 <i>Pbest</i> pada Iterasi 0	35
Tabel 4.7 <i>Gbest</i> pada Iterasi 0.....	35
Tabel 4.8 <i>Update</i> Kecepatan Iterasi 1	36
Tabel 4.9 <i>Update</i> Posisi Partikel iterasi 1.....	37
Tabel 4.10 <i>Pbest</i> Baru pada Iterasi 1.....	37
Tabel 4.11 <i>Gbest</i> pada Iterasi 1.....	38
Tabel 4.12 Sub-Himpunan Partike ke-1 ilterasi ke-0	39
Tabel 4.13 Himpunan <i>Fuzzy</i>	39
Tabel 4.14 Fuzzifikasi.....	40
Tabel 4.15 Hasil FLR.....	41
Tabel 4.16 Hasil FLRG	42
Tabel 4.17 Hasil Nilai Tengah	43
Tabel 4.18 Hasil Defuzzifikasi	43
Tabel 4.19 Hasil Peramalan.....	44
Tabel 4.20 Hasil Perhitungan Error	45
Tabel 6.1 Hasil Pengujian Banyak Iterasi.....	63
Tabel 6.2 Hasil Pengujian Panjang Dimensi Partikel	64

Tabel 6.3 Hasil Pengujian Jumlah Partikel.....	66
Tabel 6.4 Hasil Pengujian Kombinasi Nilai Variabel c1 dan c2.....	67
Tabel 6.5 Hasil Pengujian Nilai Variabel Bobot Inersia	69
Tabel 6.6 Hasil Pengujian Tingkat Kesalahan Sistem	70
Tabel 6.7 Hasil Peramalan <i>Fuzzy Time Series</i> dengan penentuan interval berbasis rata-rata dan metode <i>Regresi Linier</i>	72
Tabel 6.8 Perbandingan Nilai MAPE dengan 10 data uji.....	72



DAFTAR GAMBAR

Gambar 3.1 Diagram Alir Tahapan Penelitian.....	13
Gambar 4.1 Diagram Alir Optimasi <i>Fuzzy Time Series</i> menggunakan <i>Particle Swarm Optimization</i>	21
Gambar 4.2 Diagram Alir Menentukan Himpunan Semesta	22
Gambar 4.3 Diagram Alir Menentukan Partikel awal	23
Gambar 4.4 Diagram Alir Hitung <i>Fitness</i> menggunakan FTS	24
Gambar 4.5 Diagram Alir Fuzzifikasi	25
Gambar 4.6 Diagram Menentukan <i>Fuzzy Logic Relationship</i> (FLR) ..	26
Gambar 4.7 Diagram Alir Menentukan <i>Fuzzy Logic Relationship Group</i> (FLRG) ..	27
Gambar 4.8 Diagram Alir Proses Defuzzifikasi.....	28
Gambar 4.9 Diagram Alir Proses Peramalan.....	29
Gambar 4.10 Diagram Alir Menentukan <i>Pbest</i>	30
Gambar 4.11 Diagram Air Menentukan <i>Gbest</i>	31
Gambar 4.12 Pembangkitan Partikel	33
Gambar 4.13 Fungsi Keanggoaan <i>Fuzzy</i>	40
Gambar 4.14 Perancangan Antarmuka Halaman Home.....	46
Gambar 4.15 Rancangan Antarmuka Halaman Optimasi	47
Gambar 4.16 Rancangan Antarmuka Halaman Hasil Peramalan	48
Gambar 5. 1 Implementasi Antarmuka Halaman Home.....	61
Gambar 5.2 Implementasi Antarmuka Halaman Optimasi.....	61
Gambar 5.3 Implementasi Antarmuka Halaman Hasil Peramalan	62
Gambar 6.1 Grafik Hasil Pengujian Jumlah Iterasi Terhadap Nilai Rata-rata <i>fitness</i>	64
Gambar 6.2 Grafik Hasil Pengujian Panjang Dimensi Partikel Terhadap Nilai Rata-rata <i>fitness</i>	65
Gambar 6.3 Grafik Hasil Pengujian Jumlah Partikel Terhadap Nilai Rata-rata <i>fitness</i>	66
Gambar 6.4 Grafik Hasil Pengujian Kombinasi Nilai Variabel c1 dan c2 Terhadap Rata-rata <i>fitness</i>	68
Gambar 6.5 Grafik Hasil Pengujian Nilai Bobot inersia (w).....	69
Gambar 6.6 Grafik Perbandingan Data uji dengan Hasil Peramalan	71

DAFTAR LAMPIRAN

LAMPIRAN A DATA PDB 76



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Perkembangan perekonomian di Indonesia yang terus bergulir menuntut pemerintah dan seluruh stakeholder (pelaku ekonomi) untuk peka terhadap fenomena yang terus berubah dari waktu ke waktu. Tanpa mengikuti pergerakan tersebut sangat mustahil untuk dapat menangkap fenomena yang ada beserta keterkaitan unsur-unsur dan strukur di dalamnya. Sedangkan peningkatan kesejahteraan masyarakat sendiri sebagai tujuan dari dilaksanakannya pembangunan ekonomi menuntut para pembuat kebijakan (pemerintah) peka terhadap kondisi ekonomi yang terjadi. Perencanaan pembangunan memerlukan berbagai macam indikator ekonomi (Badan Pusat Statistika, 2018). Dengan indikator ekonomi yang representatif, strategi pembangunan baik itu jangka menengah maupun jangka panjang akan lebih terarah.

Produk Domestik Bruto (PDB) merupakan salah satu dari beberapa indikator atau alat ukur ekonomi yang biasanya digunakan sebagai bahan evaluasi dan perencanaan ekonomi makro dalam suatu periode tertentu. Melalui PDB nantinya akan dapat menggambarkan tentang kinerja perekonomian di suatu daerah pada periode tertentu, dimana dapat bermanfaat bagi para ahli yang bergerak di bidang perencanaan, pengambilan keputusan baik bersifat jangka pendek maupun panjang dalam membuat keputusan atau kebijakan. Meskipun PDB yang besar tidak menjamin seluruh penduduk di suatu wilayah mendapat kebahagiaan atau kesejahteraan, tetapi setidaknya dapat digunakan sebagai resep kebahagiaan terbaik yang ditawarkan oleh para ahli *makroekonomi* (Dalimunthe, 2017). PDB sendiri merupakan nilai tambah (*value added*) secara total dalam suatu negara tertentu yang diperoleh dari hasil seluruh unit ekonomi (Badan Pusat Statistika, 2018). Ada 2 macam PDB yaitu berdasarkan harga berlaku dimana perhitungannya menggunakan harga pada tahun yang bersangkutan dan harga konstan yang perhitungannya menggunakan harga pada tahun tertentu. PDB juga dihitung menggunakan 3 pendekatan yaitu melalui pendekatan produksi atau lapangan usaha, pendekatan pendapatan dan pendekatan pengeluaran (Badan Pusat Statistika, 2018).

Sebagai salah satu indikator masukan pada penyusunan rancangan program-program pembangunan. Peramalan Produk Domestik Bruto ini diharapkan dapat memberikan gambaran mengenai pertumbuhan ekonomi dan kinerja perekonomian di Indonesia. Jika melihat pentingnya peramalan (*forecasting*) yang menjadi dasar bagi perencanaan jangka pendek maupun panjang untuk para pembuat kebijakan (*Decision Making*). Maka dirasa penting melakukan peramalan PDB Kedepannya, agar dapat melakukan hal-hal yang bersifat *Preventif* demi terwujudnya sistem kinerja perekonomian yang lebih baik. Umumnya data PDB disajikan dalam runtun waktu/ *time series* triwulanan maupun tahunan. Sumber data PDB biasanya berasal dari hasil survei maupun dari catatan administrasi dari berbagai Lembaga intansi. Terkadang terdapat permasahan sumber data kurang

lengkap atau tidak tersedia saat perhitungan nilai PDB, sehingga diperlukan suatu cara untuk memperkirakan nilai PDB agar dapat menggambarkan keadaan sebenarnya.

Untuk memprediksi PDB bisa menggunakan beberapa metode peramalan. Metode peramalan yang dapat digunakan untuk peramalan data runtun waktu/*time series* salah satunya yaitu *Fuzzy Time series* (FTS) (Prasojo, 2018). Teori fuzzy memiliki kelebihan dapat melakukan perkiraan pada data yang tidak lengkap atau kurang spesifik, umumnya dapat memperoleh hasil yang lebih baik dengan lebih mudah dan cepat (Cai, et al., 2013). *Fuzzy time series* telah diterapkan dalam penelitian sebelumnya terkait peramalan data Time series menggunakan algoritme *fuzzy time series*, penelitian yang dilakukan Efendi, et al (2015) untuk peramalan kebutuhan beban listrik di Malaysia menghasilkan tingkat kesalahan dengan menggunakan MAPE sebesar 1,63% untuk 10 data uji. Dalam penelitian peramalan menggunakan metode *Fuzzy Time Series* yang lainnya, yang dilakukan oleh Wanayasa (2012). Untuk peramalan produk domestik regional bruto (PDRB) provinsi Bali. Dalam penelitian tersebut dilakukan perbandingan nilai *Average Forecasting Error Rate* (AFER) antara metode *Holt-Winter Aditif* dengan *Fuzzy Time Series*, menghasilkan peramalan menggunakan metode *Fuzzy Time Series* memiliki nilai AFER yang lebih kecil yaitu 0,64%.

Menurut Elfajar, et al (2017), salah satu indikator dalam metode *Fuzzy Time Series* yang mempengaruhi tingkat akurasi yaitu panjang interval dari partisi pada semesta pembicara (*universe of discourse*), karena sangat berpengaruh dalam pembentukan langkah selanjutnya yaitu pembentukan *fuzzy relationship* yang tentunya akan memberikan hasil prediksi. Dalam penelitian ini agar menghasilkan nilai akurasi yang lebih baik, dapat dilakukan optimasi Panjang interval *Fuzzy Time Series*.

Salah satu algoritme optimasi yaitu *Paricle Swarm Optimization* (PSO), PSO merupakan sebuah Teknik optimasi yang tergolong dalam salah satu perkembangan teknik *heuristik*. PSO pertama kali dikemukakan oleh Kennedy dan Eberhart yang cara kerja dari PSO tersebut didasarkan pada perilaku individu (partikel atau gen) dari sekelompok burung, yang optimasi ruang pencariannya berdasarkan populasi. Berdasarkan penelitian Khoslah, et al (2012) yang melakukan empat kali percobaan terhadap data *Mackey-Glass Time Series* dengan menggunakan data *Times Series* yang tidak berpola. Percobaan tersebut menghasilkan bahwa algoritme *evolusioner* yang dapat mengoptimalkan nilai parameter agar dapat menghasilkan nilai akurasi dan konvergensi lebih baik yaitu algoritme *Particle Swarm Optimization* (PSO).

Dalam Penilitian lainnya yang dilakukan oleh Qiu, et al (2015) dengan menggunakan metode *Generalized Fuzzy Time Series Forecasting Model* dengan *Particle Swarm Optimization* terhadap data pendaftaran *University of Allabama* dan data *Shanghai Exchange Composite Index*. Dalam penelitian tersebut menghasilkan *Root Mean Square Error* (RMSE) sebesar 2,59, *Mean Absolute Error* (MAE) sebesar 0,64 dan *Mean Absolute Percentage Error* (MAPE) sebesar 0.0004.

Nilai *error* tersebut lebih baik jika dibandingkan dengan *Fuzzy Time Series* konvesional yang memiliki nilai sebesar 3,23.

Berdasarkan penjelasan di atas penelitian ini akan melakukan peramalan pengeluaran Produk Domestik Bruto (PDB) di Indonesia menggunakan algoritme *Fuzzy Time Series* (FTS), kemudian nantinya akan dilakukan pengoptimasian pada Panjang interval pada *fuzzy time series* menggunakan algoritme *Particle Swarm Optimization* (PSO). Peramalan akan dilakukan terhadap data Produk Domestik Bruto (PDB) menurut lapangan usaha di Indonesia pada tahun 2000-2014. Dan diharapkan nantinya hasil dari penelitian ini nanti dapat dijadikan informasi untuk penelitian yang lainnya.

1.2 Rumusan Masalah

Berdasarkan Latar belakang yang telah dijelaskan diatas sebelumnya dapat diambil suatu rumusan masalah sebagai berikut.

1. Berapa nilai parameter *Particle Swarm Optimization* yang sesuai untuk menghasilkan tingkat Kesalahan terendah?
2. Bagaimana tingkat kesalahan yang didapatkan dari hasil peramalan Produk Domestik Bruto (PDB) menurut lapangan usaha menggunakan algoritme *Fuzzy Time Series* dengan *Particle Swarm Optimization* (FTSPSO)?

1.3 Tujuan

Dilakukannya skripsi ini dengan maksud untuk mencapai tujuan sebagai berikut.

1. Menerapkan algoritme *Fuzzy Time Series* (FTS) untuk peramalan dan menentukan parameter algoritme *Particle Swarm Optimization* (PSO) yang optimal untuk model *Fuzzy Time Series*.
2. Mendapatkan tingkat kesalahan dari hasil peramalan Produk Domestik Bruto (PDB) menurut lapangan usaha menggunakan algoritme *Fuzzy Time Series* dengan *Particle Swarm Optimization* (FTSPSO)?

1.4 Manfaat

Dilakukannya penelitian ini dengan maksud agar dapat memberikan manfaat yaitu berupa informasi mengenai peramalan Produk Domestik Bruto (PDB) di Indonesia pada tahun berikutnya.

1.5 Batasan Masalah

Penelitian ini memiliki Batasan masalah sebagai berikut:

1. Data Produk Domestik Bruto (PDB) yang digunakan adalah data Produk Domestic Bruto (PDB) menurut lapangan usaha di Indonesia atas harga berlaku yang didapat dari website Badan Pusat Statistika (BPS).

2. Data yang dianalisis adalah data Produk Domestik Bruto (PDB) triwulan dari tahun 2000 triwulan I sampai tahun 2014 triwulan IV sehingga menghasilkan total 60 data.

1.6 Sistematika Pembahasan

Sistematika pembahasan dalam skripsi ini sebagai berikut.

BAB I Pendahuluan

Berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan yang digunakan untuk membuat laporan.

BAB II Landasan Kepustakaan

Tinjauan pustaka berisi tentang penjelasan kajian pustaka dan dasar teori yang digunakan untuk mengaitkan referensi mengenai penelitian yang dilakukan.

BAB III Metodologi Penelitian

Metodologi penelitian adalah pembahasan terhadap metode yang digunakan pada penulisan implementasi *Fuzzy Time Series* dengan *Particle Swarm Optimization* (FTS-PSO) untuk peramalan pengeluaran Produk Bruto (PDB) di Indonesia.

BAB IV Perancangan dan Implementasi

Pembahasan tentang perancangan *Fuzzy Time Series* dengan *Particle Swarm Optimization* (FTS-PSO) untuk peramalan pengeluaran Produk Bruto (PDB) di Indonesia.

BAB V Pengujian dan Analisis

Berisi tentang proses dan hasil pengujian terhadap sistem yang telah direalisasikan.

BAB VI Penutup

Memuat kesimpulan dari keseluruhan uraian bab yang telah dikerjakan dan berisi saran untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Rujukan teori apa saja yang digunakan dalam penelitian ini sebagai alat yang digunakan untuk membedah sebuah permasalahan yang akan diangkat oleh peneliti akan dibahas pada bab ini. Adapun penelitian yang sudah dilakukan sebelumnya yang masih terkait dengan penelitian ini cukup banyak. Salah satunya yaitu penelitian yang dilakukan Khoslah, et. al, (2012) yang melakukan empat kali percobaan terhadap data Mackey-Glass Time Series dengan menggunakan data Times Series yang tidak berpola. Parameternya didesain agar dapat dimodifikasi dikaranakan terdapat fakta bahwa kinerja algoritme evolusioner tergantung pada pemilihan parameter. Percobaan tersebut menghasilkan bahwa algoritme evolusioner yang bisa mengoptimalkan nilai parameter agar dapat menghasilkan nilai akurasi dan konvergensi lebih baik adalah algoritme *Particle Swarm Optimization* (PSO).

Penelitian lainnya yang dilakukan Qiu, et al (2015) yang melakukan penelitian dengan menggunakan metode *Generalized Fuzzy Time Series Forcecasting Model* dengan *Particle Swarm Optimization* terhadap data Pendaftaran *University of Allabama* dari tahun 1971-1992 dan data harga index komposit bursa efek Shanghai (SSECI) tahun 1997 sampai 2006. Penelitian tersebut bertujuan untuk menoptimalkan panjang interval pada himpunan semesta menggunakan algoritme *Particle Swarm Optimization* agar dapat meningkatkan akurasi peramalan. Dalam percobaan didapatkan hasil menghasilkan *Root Mean Square Error* (RMSE) sebesar 2,59, *Mean Absolute Error* (MAE) sebesar 0,64 dan *Mean Absolute Percentage Error* (MAPE) sebesar 0.0004.

Pada penelitian lainnya yang dilakukan Prasojo, et al (2018) yang melakukan peramalan terhadap data penduduk di kabupaten Probolinggo menggunakan *Fuzzy Time Series* yang dioptimasi menggunakan *Particle Swarm Optimization*. Hasil percobaan dengan menggunakan metode yang diusulkan menghasilkan nilai error RMSE sebesar 14,503 lebih kecil dibandingkan menggunakan metode *Fuzzy Time Series* tanpa optimasi yang memiliki nilai RMSE sebesar 52,051.

Penelitian terkait lainnya yang sudah dilakukan Rifandi, et al (2018), dalam penelitian tersebut dilakukan peramalan terhadap data permintaan darah menggunakan *Fuzzy Time Series* dan *Modified Turbelence Particle Swarm Optimization* yang merupakan metode *Particle Swarm Optimization* yang telah dimodifikasi untuk optimasi interval *Fuzzy Time Series*. Dalam percobaan yang dilakukan metode yang diusulkan menghasilkan nilai *error MAPE* sebesar 7,5%.

Penelitian lainnya yang telah dilakukan Wanayasa, et al (2012). Dalam penelitian tersebut dilakukan perbandingan nilai *Average Forcasting Error Rate* (AFER) antara metode pemulusan *exponensial Holt-Winter Aditif* dengan *Fuzzy Time Series*, menghasilkan peramalan menggunakan metode *Fuzzy Time Series* memiliki nilai AFER yang lebih baik yaitu 0,64%.



Tabel 2.1 Perbandingan Penelitian Terdahulu dengan usulan

NO	JUDUL	OBJECT	METODE	HASIL
1	<i>Generalized Fuzzy Time Series Forecasting Model Enhanced with Particle Swarm Optimization</i> (Qiu, et al, 2015).	1. Data Pendaftaran University of Allabama dari tahun 1971-1992 2. data harga index komposit bursa efek Shanghai (SSECI) tahun 1997 sampai 2006.	<i>Fuzzy Time Series dengan Particle Swarm Optimization</i>	hasil menghasilkan <i>Root Mean Square Error</i> (RMSE) sebesar 2,59, <i>Mean Absolute Error</i> (MAE) sebesar 0,64 dan <i>Mean Absolute Percentage Error</i> (MAPE) sebesar 0.0004.
2	<i>Identification of type-2 Fuzzy Model for Time Series Forecasting using Particle Swarm Optimization</i> (Khoslah, et al, 2012).	<i>Mackey-Glass Time Series</i>	<i>Fuzzy Time Series dengan Particle Swarm Optimization.</i>	
3	Optimasi Fuzzy Time Series Menggunakan Algoritme Particle Swarm Optimization Untuk Peramalan Jumlah Penduduk Di Kabupaten Probolinggo (Prasojo, et al, 2018).	data jumlah penduduk perkecamatan setiap bulan dan setiap tahunnya dari 4 tahun terakhir mulai tahun 2013-2016	<i>Fuzzy Time Series dengan Particle Swarm Optimization (FTS-PSO).</i>	-Metode <i>Fuzzy Time Series dengan Particle Swarm Optimization (FTS-PSO)</i> menghasilkan RMSE 14,50370333. - Metode <i>Fuzzy Time Series dengan Particle Swarm Optimization (FTS-PSO)</i> menghasilkan



NO	JUDUL	OBJECT	METODE	HASIL
				error yang lebih kecil dari metode <i>Fuzzy Time Series</i> .
4	Optimasi Fuzzy Time Series Menggunakan Algoritme Particle Swarm Optimization pada Peramalan Permintaan Darah: Studi Kasus Unit Transfusi Darah Cabang- PMI Kota Malang (Rifandi, et al, 2018).	Data permintaan darah bulanan yang Kota Malang yang dihimpun oleh Unit Transfusi Darah Cabang – PMI Kota Malang dari tahun 2010 sampai 2015.	<i>Fuzzy Time Series dengan Modified Turbulent Particle Swarm Optimization.</i>	metode yang diusulkan menghasilkan nilai error MAPE sebesar 7,5%.
5	Peramalan Produk Domestik Regional Bruto (PDRB) Provinsi Bali Menggunakan Fuzzy Time Series	Data Produk Domestik Regional Bruto Provinsi Bali dari tahun 1991-2010	Fuzzy Time Series	- Metode Fuzzy Time Series menghasilkan nilai error AFFER sebesar 0,64% lebih dibandingkan menggunakan metode pemulusan <i>Exponensial Holt Winter Aditif</i> dengan AFFER 7,13%.
6	Optimasi Fuzzy Time Series Menggunakan Algoritme Particle Swarm Optimization Untuk Peramalan Pengeluaran	Data Produk Domestik Bruto Indonesia Tahun 2000 triwulan I - 2014 triwulan IV	<i>Fuzzy Time Series dengan Particle Swarm Optimization.</i>	- Partikel terbaik dari hasil optimasi menggunakan <i>Particle Swarm Optimization</i> . - Hasil peramalan Produk

NO	JUDUL	OBJECT	METODE	HASIL
	Produk Domestik Regional Bruto Di Kota Malang (usulan)			Domestik Bruto Indonesia. -Nilai Mean Absolute Percentage Error (MAPE)

2.2 Produk Domestik Bruto

Pertumbuhan ekonomi yang menunjukkan pertumbuhan produksi barang dan jasa dalam periode tertentu di suatu wilayah dapat dijadikan alat ukur untuk mengetahui perkembangan ekonomi suatu negara. Produksi tersebut dihitung menggunakan konsep nilai tambah (*value added*) yang diperoleh dari hasil seluruh unit atau sector-sektor ekonomi secara total dalam wilayah yang bersangkutan yang dikenal sebagai produk domesik bruto (PDB) (Badan Pusat Statistika, 2018). Sehingga dapat dikatakan bahwa PDB dijadikan sebagai salah satu indikator untuk mengukur kinerja perekonomian karena dapat menggambarkan pertumbuhan atau pergerakan sektor-sektor ekonomi yang telah dicapai oleh suatu pemerintahan. Ada 2 macam perhitungan PDB yaitu berdasarkan harga berlaku dimana perhitungannya menggunakan harga pada tahun yang bersangkutan dan harga konstan yang perhitungannya menggunakan harga pada tahun tertentu. Kegunaan data statistik PDB atas harga konstan digunakan untuk menggambarkan pertumbuhan ekonomi dari suatu tahun ke tahun atau dari periode ke periode, sedangkan PDB atas harga berlaku dapat menggambarkan pergeseran struktur ekonomi.

PDB juga dihitung menggunakan 3 pendekatan yaitu:

- Menurut Pendekatan Lapangan Usaha atau Produksi
Perhitungan PDB menggunakan pendekatan Lapangan usaha dalam penyajiannya dapat dikelompokkan menjadi 17 kategori lapangan, dimana kemudian diuraikan lagi menjadi sub-sub kategori lapangan usaha.
- Menurut Pendekatan Pengeluaran
Perhitungan PDB menggunakan pendekatan pengeluaran merupakan jumlah total dari semua komponen yang berhubungan dengan permintaan akhir berupa pengeluaran konsumsi akhir pemerintah, pengeluaran konsumsi akhir lembaga non-profit yang melayani rumah tangga, pengeluaran konsumsi akhir rumah tangga, perubahan inventori, dan ekspor neto (ekspor dikurangi impor) dan pembentukan modal tetap domestik bruto.
- Menurut Pendekatan Pendapatan
PDB menurut pendekatan ini berkaitan dengan proses produksi di suatu negara yang didalamnya terdapat jumlah balas jasa terhadap faktor-faktor produksi yang terlibat dapat berupa dalam upah dan gaji, bunga modal, sewa tanah dan sebagainya dalam jangka waktu tertentu (biasanya satu tahun).

2.3 Peramalan

Peramalan merupakan Teknik memperkirakan sesuatu pada masa mendatang dengan memperhatikan informasi pada masa lampau. Meramalkan nilai yang akan datang dalam melakukan analisis runtun waktu merupakan hal yang penting. Peramalan perlu dilakukan, untuk memperbesar peluang memperoleh keuntungan dan memperkecil terjadinya kerugian. Menurut Berutu (2013), menjelaskan bahwa peramalan dapat dibagi menjadi 2 macam berdasarkan sifatnya, pertama yaitu peramalan kumulatif yang datanya tidak jelas serta hanya berupa pendapat dari suatu pihak sehingga datanya biasanya tidak berwujud angka atau nilai dan yang kedua adalah peramalan kuantitatif yang datanya berupa angka dan nilai yang bersifat kuantitatif.

2.4 Fuzzy Time Series

Data times series adalah data yang diperoleh dari urutan waktu tertentu dan dalam rentan waktu tertentu atau dapat disebut data historis. *Fuzzy time series* adalah suatu metode memprediksi atau meramalkan probabilitas suatu keadaan masa mendatang untuk mengambil keputusan dengan menggunakan data-data historis dan fungsi-fungsi fuzzy. Pada peramalan menggunakan fuzzy time series, himpunan fuzzy yang berasal dari bilangan riil digunakan sebagai nilai dalam perhitungan.

Fuzzy time series dalam melakukan peramalan, terdapat dapat didefinisikan sebagai berikut.

1. Misalnya, jika terdapat $Y(t)$ untuk $t = (\dots, 0, 1, 2, \dots)$, yang adalah himpunan bagian dari bilangan riil, dimana menjadi bagian dari himpunan semesta dari *Fuzzy set* $f_i(t)$ untuk $i = (1, 2, 3, \dots)$, jika $f(t)$ adalah himpunan $f_1(t), f_2(t), \dots$. Maka $f(t)$ merupakan himpunan *Fuzzy time series* dari fungsi $Y(t)$.
2. Apabila jika $f(t)$ hanya dikarenakan oleh $f(t-1)$ dan dijelaskan dengan $f(t) \rightarrow f(t-1)$, maka dapat disimpulkan bahwa terdapat *Fuzzy Relation* antara $f(t)$ dan $f(t-1)$.

Tahapan-tahapan dalam proses peramalan menggunakan algoritme *fuzzy time series* dapat dilakukan sebagai berikut (Prasojo, 2018):

1. Menentukan himpunan semesta dengan cara menentukan nilai maksimum (D_{max}) dan nilai minimum (D_{min}) pada data historis, kemudian menentukan himpunan semesta (U) dengan menggunakan Persamaan 2.1.

$$U = [D_{min} - d_1, D_{max} + d_2] \quad (2.1)$$

Dimana untuk d_1 dan d_2 merupakan dua bilangan interger positif untuk menyesuaikan nilai batas bawah dan batas atas dari U .

2. Membagi himpunan semesta menjadi beberapa sub himpunan dengan ukuran ukuran rentang yang sama panjang kemudian membentuk

himpunan *fuzzy* sama dengan sub himpunan $u_1, u_2, u_3, \dots, u_m$. Misalnya $A_1, A_2, A_3, \dots, A_k$ adalah himpunan *fuzzy* dari himpunan semesta U , dimana nilai A berupa nilai linguistik, sehingga dapat didefinisikan sebagai persamaan 2.2 (Uslu, 2014).

$$A_i = \frac{a_{i1}}{u_1} + \frac{a_{i2}}{u_2} + \frac{a_{i3}}{u_3} + \dots + \frac{a_{im}}{u_m} \quad (2.2)$$

Dimana a_{im} merupakan nilai derajat keanggotaan berdasarkan persamaan 2.3

$$a_i = \begin{cases} 1 & , m = i \\ 0.5 & , m = i-1, i+1 \\ 0 & , i+1 < m < i-1 \end{cases}, i = 1, 2, \dots, m \quad (2.3)$$

3. Proses Fuzzifikasi

Selanjutnya mencari nilai keanggotaan berdasarkan himpunan *fuzzy* yang terbentuk sebelumnya. Selanjutnya melakukan proses fuzzyifikasi yaitu dengan cara memilih nilai yang memiliki keanggotaan tertinggi pada masing-masing variabel linguistik yang terbentuk.

4. Menentukan *Fuzzy Logical Relationship* (FLR), misal terdapat 2 himpunan *fuzzy* secara berurutan $A_i(t-1)$ dan $A_j(t)$ maka dapat dinyatakan sebagai $A_i \rightarrow A_j$.
5. Menentukan *Fuzzy Logical Relationship Group* (FLRG)
Misal: $A_i \rightarrow A_j, A_i \rightarrow A_k, A_i \rightarrow A_m, \dots$ Maka akan menjadi $A_i \rightarrow A_j, A_k, A_m, \dots$, semua grup yang sama bisa dijadikan sebagai matrik.
6. Memberikan nilai bobot pada grup *Fuzzy Logic Relation*. Contoh:
 $(t=1) A_1 \rightarrow A_1$, dengan bobot 1,
 $(t=2) A_2 \rightarrow A_1$, dengan bobot 1,
 $(t=3) A_2 \rightarrow A_3$, dengan bobot 2,
 $(t=4) A_3 \rightarrow A_1$, dengan bobot 3,
 $(t=5) A_3 \rightarrow A_2$, dengan bobot 4.
 Dimana t menyatakan waktu.
7. Melakukan perhitungan hasil peramalan, dengan cara menggunakan berdasarkan aturan yang telah didijabarkan oleh Xihao dan Yimin (2008) yaitu:
 - Jika FLR A_1 bernilai kosong ($A_i \rightarrow \#$), maka $F(t) = m_i$ yang merupakan nilai tengah dari sub himpunan u_i .
 - Jika FLR A_1 mempunyai relasi *one to one* ($A_i \rightarrow A_j$) atau himpunan *fuzzy* A_1 hanya memiliki satu himpunan FLR atau FLGR A_1 , maka $F(t) = m_i$ yang merupakan nilai tengah dari sub himpunan u_j .
 - Jika FLR A_1 mempunyai relasi *one to many* ($A_i \rightarrow A_{j1}, A_{j2}, A_{j3}, \dots, A_{jk}$) atau himpunan *fuzzy* A_1 hanya memiliki lebih dari satu himpunan FLR atau FLGR A_1 , maka mencari nilai $F(t)$ menggunakan persamaan 2.8.



$$F(t) = \frac{\sum_{i=1}^n m_{ij}}{n} \quad (2.4)$$

Dimana n adalah jumlah himpunan fuzzy yang berelasi dan m_{ij} merupakan nilai tengah dari sub himpunan u_{ij} .

2.5 Particle Swarm Optimization

Particle Swarm Optimization atau dapat disingkat menjadi PSO adalah sebuah Teknik optimasi yang tergolong dalam salah satu perkembangan teknik *heuristic*. PSO pertama kali dikemukakan oleh Kennedy dan Eberhart yang cara kerja dari PSO tersebut didasarkan pada perilaku individu (partikel atau gen) dari sekelompok burung, yang optimasi ruang pencarinya berdasarkan populasi. *Particle Swarm Optimization* (PSO) sendiri hampir menyerupai algoritme genetika hanya saja PSO tidak bergantung pada operasi mutasi, *crossover* dan seleksi. PSO lebih bergantung pada pertukaran informasi antar individu. Didalam PSO untuk mencapai solusi optimal pertama-tama dimulai dari kelompok solusi acak kemudian mencari berulang kali sampai kondisi berhenti terpenuhi. Fungsi *fitness* dalam PSO digunakan untuk menilai baik atau buruk sebuah individu.

Menurut Misra Hartati (2012), dalam mencari nilai optimum pada algoritme PSO prosesnya dapat dilakukan sebagai berikut.

1. Proses pertama yaitu menentuan nilai parameter, dimana nilai parameter yang dibutuhkan pada algoritme *Particle Swarm Optimization* ada 5 nilai yang harus ditentukan di awal antara lain inisialisasi kecepatan, posisi, nilai $pbest$, nilai $gbest$, nilai *fitness* yang akan dicari dan sebagai dari jawaban atas permasalahan yang ada.
2. Inisialisasi posisi dan kecepatan Partikel awal
Inisialisasi partikel awal dilakukan dengan cara mengambil nilai secara acak dari sekumpulan partikel, dapat dilihat pada persamaan 2.5 dan untuk insialisasi kecepatan partikel awal bernilai 0, dapat dilihat pada persamaan 2.6.

$$x_i^t = x_i(t) \quad (2.5)$$

$$v_i^t = \mathbf{0} \quad (2.6)$$

Keterangan:

v_i^t = kecepatan partikel ke i iterasi ke t

x_i^t = posisi partikel ke i pada iterasi ke t

3. Menghitung nilai *fitness*

Nilai fitness digunakan untuk menilai seberapa baik dan buruknya sebuah individu atau partikel. Dalam penelitian ini persamaan yang digunakan untuk meghitung nilai *fitness* dari setiap partikel dengan menggunakan persamaan 2.7.

$$\text{fitness} = \text{MAPE} = \frac{100\%}{n} \left(\sum_{i=1}^n \frac{|x_i - \hat{x}_i|}{x_i} \right) \quad (2.7)$$



4. *Update* Kecepatan dan Posisi
Untuk *update* kecepatan dan posisi dari masing-masing dimensi menggunakan rumus (Cholissodin & Riyandani, 2016). Untuk *update* kecepatan dapat dilihat pada persamaan 2.8 dan *update* posisi pada persamaan 2.9.

$$v_{ij}^{t+1} = w \cdot v_{ij}^t + c_1 \cdot r_1 (Pbest_{ij}^t - x_i^t) + c_2 \cdot r_2 (Gbest_g^t - x_i^t) \quad (2.8)$$

Keterangan:

$v_{i,j}^t$ = kecepatan partikel i dimensi j pada iterasi ke – t

$pbest_{i,j}^t$ = posisi terbaik dari partikel i dimensi j pada iterasi ke – t

$gbest_{g,j}^t$ = global optimal dari dimensi g pada iterasi ke-t

c= konstanta kecepatan

r= nilai acak $\in [0,1]$

w= bobot inersia

$$x_j^{k+1} = x_j^k + v_j^{k+1} \quad (2.9)$$

Keterangan:

$x_{i,j}^t$ = posisi partikel i dimensi j pada iterasi ke – t

5. Setelah kecepatan dan posisi ter-*update* maka selanjutnya melakukan perhitungan *fitness*.
6. Selanjutnya melakukan *update pbest*, dengan aturan jika partikel baru memiliki *fitness* yang lebih baik dari pada partikel yang lama, maka partikel yang baru dapat menggantikan partikel yang lama sebagai *pbest* yang baru.
7. Kemudian melakukan *update gbest* dengan cara menbandingkan nilai *fitness* antar masing masing *pbest*. *pbest* yang memiliki nilai *fitness* yang lebih baik akan menjadi *gbest*.
8. Proses akan terus mengulang dari tahap 4 sampai kondisi terpenuhi atau sampai iterasi maksimal.

2.6 Nilai Evaluasi

Nilai evaluasi adalah nilai yang digunakan untuk melihat akurasi dari hasil nilai peramalan yang didasarkan pada nilai aktual dan nilai ramalan. *Mean Absolute Percentage error* (MAPE) merupakan salah satu contoh dari beberapa metode yang bisa digunakan untuk menghitung nilai evaluasi. Sebagai contoh jika terdapat 2 model estimasi, maka model yang memiliki MAPE lebih kecil merupakan model yang lebih baik. Rumus MAPE yang digunakan dapat dilihat pada persamaan 2.10 (Vagarda, 2016).

$$MAPE = \frac{100\%}{n} \left(\sum_{t=1}^n \frac{|x_t - \hat{x}_t|}{x_t} \right) \quad (2.10)$$

Keterangan:

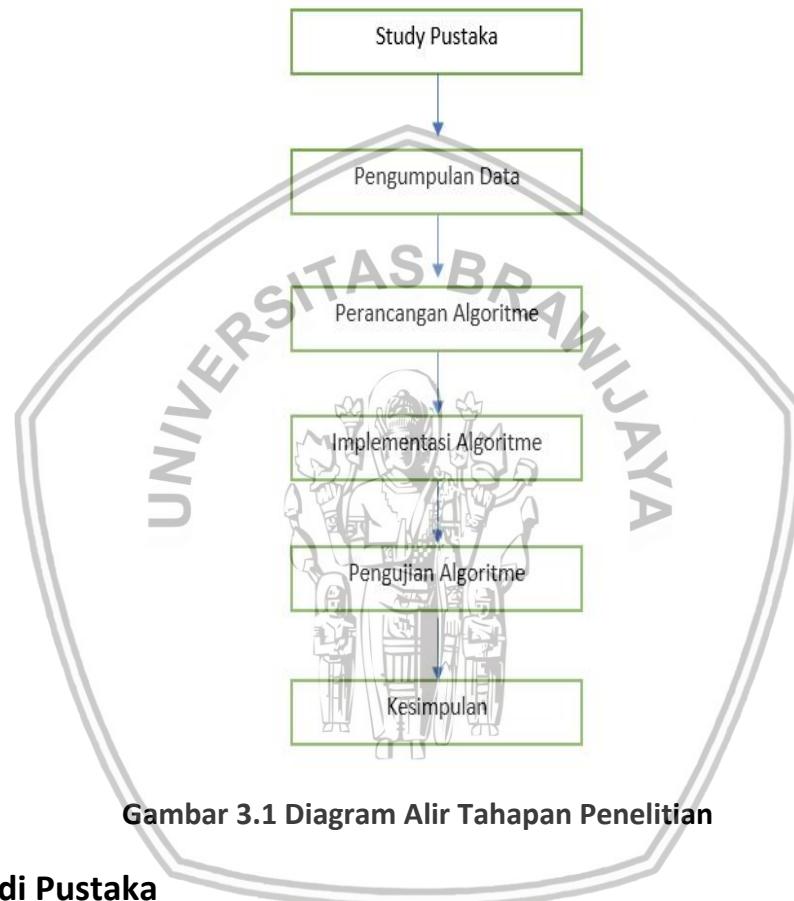
\hat{x}_t = data ramalan

x_t = data aktual



BAB 3 METODOLOGI

Penjelasan tentang metode apa saja yang akan dibutuhkan serta apa saja tahapan yang harus dijalankan dalam penelitian ini akan dibahas pada bab ini. Metode pelaksanaan yang digunakan meliputi kajian pustaka, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem dan kesimpulan. Gambar 3.1 ini menunjukkan diagram yang menggambarkan alur dari tahapan-tahapan yang terdapat pada metode pelaksanaan.



3.1 Studi Pustaka

Study Literatur dibutuhkan untuk dapat memperoleh sebanyak-banyaknya informasi yang berfungsi sebagai acuan dalam penelitian. Literatur tersebut diperoleh dari buku, jurnal, artikel dan dokumentasi proyek. Literatur diperoleh dari beberapa bidang ilmu yang berkaitan dengan penelitian yang akan dilakukan, diantaranya:

- a) Produk Domestik Bruto
- b) Logika *Fuzzy*
- c) Algoritme *Fuzzy Time Series* (FTS)
- d) Algoritme *Particle Swarm Optimization* (PSO)



3.2 Pengumpulan Data

Dalam penelitian ini data yang digunakan berupa data time series karena data dikumpulkan terurut dalam rentan waktu selama beberapa tahun sebelum-sebelumnya sehingga nantinya akan digunakan untuk peramalan tahun selanjutnya. Sumber datanya diperoleh dari website Badan Pusat Statistik (BPS). Data yang digunakan data tahun 2000 triwulan I sampai tahun 2014 triwulan IV sehingga menghasilkan total data berjumlah 60 data. Data selengkapnya dapat dilihat pada Lampiran A.

3.3 Perancangan Algoritme

Pada Bab ini dilakukan perancangan implementasi metode bertujuan untuk mengidentifikasi komponen yang dibutuhkan dalam proses mengimplementasikan *fuzzy time series* dengan *particle swarm optimization* dalam peramalan pengeluaran Produk Domestik Bruto (PDB) di Indonesia.

3.4 Implementasi Algoritme

Implementasi metode *fuzzy time series* dengan *particle swarm optimization* dalam peramalan pengeluaran Produk Domestik Bruto (PDB) di Indonesia diterapkan sesuai perancangan yang didefinisikan sebelumnya.

3.5 Pengujian Algoritme

Pada tahap ini dilakukan pengujian nilai parameter dari algoritme PSO, sehingga diharapkan nantinya menghasilkan solusi yang terbaik dan akurat. Proses pengujian ini bertujuan untuk mendapatkan kombinasi parameter PSO yang tepat sehingga proses optimasi akan menghasilkan hasil yang optimal. Macam-macam parameter PSO yang akan diujikan antara lain banyak partikel, iterasi maksimum, koefisien partikel *cognitive* (c_1), koefisien partikel *social* (c_2), bobot *inertia* (w) dan Panjang dimensi partikel. Setelah pengujian dilakukan analisis dapat dilakukan untuk masing-masing pengujian.

Pada Penelitian ini rancangan pengujian terhadap parameter yang diujikan sebagai berikut:

1. Pengujian Jumlah Iterasi Maksimum

Jumlah iterasi yang diujikan adalah 50, 100, 150, 200, 250, 300, 350, 400, 450, 500 serta akan dilakukan 5 kali percobaan. Dilakukannya pengujian ini bertujuan agar dapat mengetahui berapa jumlah iterasi maksimum yang akan menghasilkan panjang interval yang paling optimum dalam fuzzy time series. Nilai parameter yang digunakan yaitu panjang dimensi sebanyak 15, jumlah partikel sebanyak 10, bobot inersia (w) sama dengan 0.3, nilai cognitive c_1 dan c_2 berturut-turut sama dengan 1.5 dan 0.5. Solusi terbaik diambil dari nilai *fitness* yang terendah. Pada Tabel 3.4 menunjukkan perancangan pengujian jumlah iterasi maksimum.

Tabel 3.1 Perancangan Pengujian Jumlah Iterasi Maksimum

Panjang dimensi partikel=15, Banyak partikel= 10, w= 0.3, c1= 1.5, c2= 0.5						
Banyak Iterasi	Nilai <i>fitness</i> percobaan ke -					Rerata <i>fitness</i>
	1	2	3	4	5	
50						
100						
150						
200						
250						
300						
350						
400						
450						
500						

2. Pengujian Panjang dimensi partikel

Jumlah partikel yang diujikan adalah 5, 10, 15, 20, 25, 30, 35, 40 serta akan dilakukan sebanyak 5 kali. Tujuan Pengujian ini dilakukan untuk mengetahui berapa jumlah panjang dimensi partikel yang akan menghasilkan interval yang paling optimum dalam *fuzzy time series*. Nilai parameter banyak partikel sebanyak 10, iterasi maksimum sebanyak hasil pengujian sebelumnya, bobot *inersia* (w) sebesar 0.3, nilai *cognitive* c1 dan c2 berturut-turut sama dengan 1.5 dan 0.5. Solusi terbaik diambil dari nilai *fitness* yang terendah. Perancangan pengujian Panjang dimensi dapat dilihat pada Tabel 3.2.

Tabel 3.2 Perancangan Pengujian Panjang Dimensi Partikel

itermaks=..., Banyak partikel= 10, w= 0.3, c1 = 1.5, c2 = 0.5						
Panjang dimensi partikel	Nilai <i>fitness</i> percobaan ke -					Rerata <i>fitness</i>
	1	2	3	4	5	
5						
10						
15						
20						

25						
30						
35						
40						

3. Pengujian Jumlah Partikel

Jumlah partikel yang diujikan adalah 5, 10, 15, 20, 25, 30, 35, 40 serta akan dilakukan 5 kali percobaan. Tujuan dari pengujian ini dilakukan agar mendapatkan jumlah partikel yang akan menghasilkan interval yang paling optimum dalam fuzzy time series. Sedangkan nilai parameternya yaitu Panjang dimensi sama dengan hasil dari pengujian sebelumnya, iterasi maksimum sebanyak hasil dari pengujian sebelumnya, bobot inersia (w) sebesar 0.3, nilai *cognitive* c_1 dan c_2 berturut-turut sama dengan 1.5 dan 0.5. Solusi terbaik diambil dari nilai *fitness* yang terendah. Perancangan pengujian banyak partikel dapat dilihat pada Tabel 3.3.

Tabel 3.3 Percangan pengujian banyak partikel

Banyak partikel	Nilai <i>fitness</i> percobaan ke -					Rerata <i>fitness</i>
	1	2	3	4	5	
5						
10						
15						
20						
25						
30						
35						
40						

4. Pengujian Kombinasi nilai c_1 dan c_2

Kombinasi c_1 dan c_2 yang diujikan adalah 0.5, 1, 1.5, dan 2 secara bergantian akan dikombinasikan serta akan dilakukan 5 kali percobaan. Tujuan dari dilakukannya pengujian ini agar mendapatkan kombinasi nilai c_1 dan c_2 yang akan menghasilkan interval yang paling optimum dalam fuzzy time series. Sedangkan untuk nilai parameternya adalah Panjang dimensi sama dengan hasil dari pengujian sebelumnya, banyak partikel sebanyak hasil dari pengujian sebelumnya, iterasi maksimum sama dengan

hasil dari pengujian sebelumnya, bobot inersia (w) bernilai 0.3. Solusi terbaik diambil dari nilai *fitness* yang terendah. Perancangan pengujian kombinasi c1 dan c2 dapat dilihat pada Tabel 3.5

Tabel 3.4 Perancangan Pengujian Kombinasi Nilai c1 dan c2

Panjang dimensi partikel =..., Banyak partikel=..., itermaks=...., w=0.3							
Nilai c1	Nilai c2	Nilai <i>fitness</i> percobaan ke-					Rerata <i>fitness</i>
		1	2	3	4	5	
0.5	0.5						
	1						
	1.5						
	2						
1	0.5						
	1						
	1.5						
	2						
1.5	0.5						
	1						
	1.5						
	2						
2	0.5						
	1						
	1.5						
	2						

5. Pengujian Besar Nilai Bobot *Inersia* (w)

Nilai bobot inersia yang diujikan adalah 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 serta akan dilakukan sebanyak 5 kali percobaan. Tujuan dari dilakukannya pengujian ini untuk mendapatkan nilai bobot *inersia* yang akan menghasilkan interval yang paling optimum dalam fuzzy time series, dimana untuk nilai parameternya yaitu Panjang dimensi, Banyak partikel, Iterasi maksimum, nilai *cognitive* c1 dan c2 bernilai sama dengan hasil dari pengujian sebelumnya. Solusi terbaik diambil dari nilai *fitness* yang terendah. Perancangan pengujian besar nilai bobot *inersia* dapat dilihat pada Tabel 3.5.

Tabel 3.5 Perancangan Pengujian Bobot Inersia

Panjang dimensi partikel=..., Banyak partikel=..., itermaks=...., c1 =, c2 =						
Nilai bobot (w)	Nilai <i>fitness</i> percobaan ke -					Rerata <i>fitness</i>
	1	2	3	4	5	
0.1						
0.2						
0.3						
0.4						
0.5						
0.6						
0.7						
0.8						
0.9						

6. Pengujian Tingkat Kesalahan Algoritme

Pada pengujian ini akan dilakukan untuk mengetahui tingkat kesalahan MAPE peramalan metode usulan dengan menggunakan parameter PSO yang optimal hasil pengujian sebelumnya terhadap 10 data uji yang sudah ditetapkan. Kemudian dilakukan perbandingan tingkat kesalahan dengan metode lainnya yaitu dengan metode *fuzzy time series* tanpa optimasi dan *regresi linier*. Perancangan pengujian Kesalahan algoritme dapat dilihat pada Tabel 3.6 dan perancangan perbandingan tingkat kesalahan dengan metode lainnya dapat dilihat pada Tabel 3.7.

Tabel 3.6 Perancangan Pengujian Tingkat Kesalahan Agoritme

Panjang dimensi partikel=..., Banyak partikel=..., itermaks=..., c1=..., c2=..., w=...					
No	Tahun	Triwulan	Data uji	Hasil Peramalan	Absolute Percentage Error
1	3.147023879	3	2116302.1		
2	1.052972902	4	2094027.1		
3	1.569550847	1	2143260.1		
4	2.913640186	2	2212148.1		

5	6.225332885	3	2359100.3		
6	0.157037171	4	2372768.0		
7	0.566459197	1	2387055.8		
8	3.336721533	2	2478081.7		
9	5.311487159	3	2622612.6		
10	0.487958835	4	2607178.8		
MAPE					

Tabel 3.7 Perancangan Pengujian Perbandingan MAPE

Metode	MAPE
Metode usulan	
Fuzzy Time Series interval berbasis rata2	
Regresi Linier	

3.6 Kesimpulan

Pengambilan kesimpulan dapat dilakukan apabila semua tahapan selesai dilakukan mulai dari Perancangan sampai pengujian metode yang terapkan. Kesimpulan didapat berdasarkan analisis dari hasil pengujian menggunakan metode yang telah diterapkan.



BAB 4 PERANCANGAN

4.1 Formulasi Permasalahan

Semua tentang yang berkaitan tentang perancangan *fuzzy time series* dengan *particle swarm optimization* akan dijelaskan pada bab ini, dimana perancangan akan dijelaskan dalam bentuk alur algoritme, manualisasi data aktual dan perancangan antarmuka. Fokus utama penelitian ini yaitu penentuan panjang interval pada sub himpunan *fuzzy time series* karena dapat mempengaruhi akurasi hasil prediksi. Pada sub himpunan tersebut nantinya akan dilakukan proses optimasi menggunakan algoritme *particle swarm optimization*. Pada Penelitian ini untuk datanya diambil dari website Badan Pusat Statistika (BPS) merupakan data total PDB di Indonesia menurut lapangan usaha dilihat pada Tabel 4.1 berdasarkan periode tahun 2000 sampai 2014.

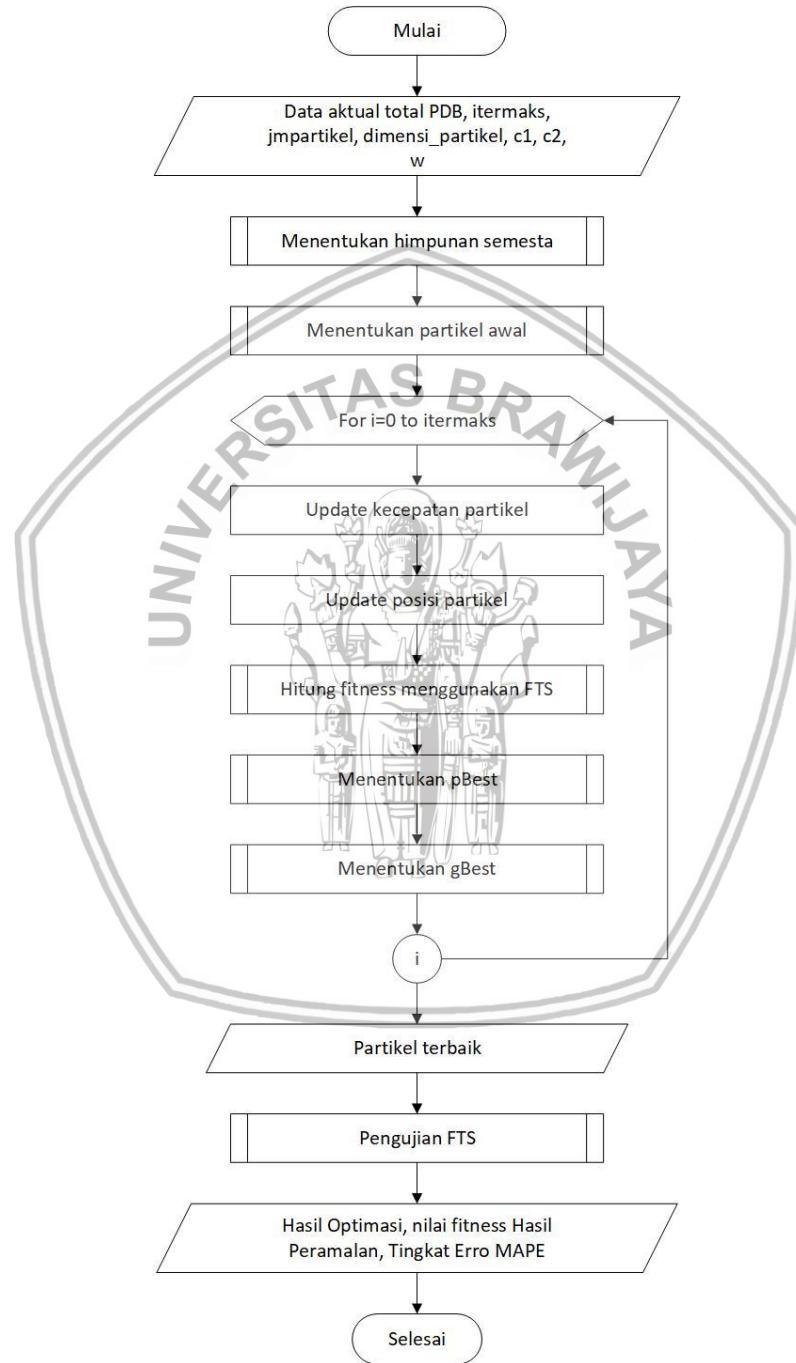
Tabel 4.1 Tabel data aktual total PDB Indonesia menurut lapangan usaha

NO	Tahun	Triwulan	PDB (Miliar Rupiah)
1	2000	Triwulan 1	325958.6
2	2000	Triwulan 2	336967.1
3	2000	Triwulan 3	360701.6
4	2000	Triwulan 4	366142.6
5	2001	Triwulan 1	386648.8
6	2001	Triwulan 2	416069.9
7	2001	Triwulan 3	426828.3
8	2001	Triwulan 4	416775.0
9	2002	Triwulan 1	436975.1
10	2002	Triwulan 2	450640.4
.....
54	2013	Triwulan 2	2212148.1
55	2013	Triwulan 3	2359100.3
56	2013	Triwulan 4	2372768.0
57	2014	Triwulan 1	2387055.8
58	2014	Triwulan 2	2478081.7
59	2014	Triwulan 3	2622612.6
60	2014	Triwulan 4	2607178.8

Sumber: Badan Pusat Statistika (2015)

4.2 Siklus Model Fuzzy Time Series dengan Particle Swarm Optimization

Dalam usaha untuk menyelesaikan masalah peramalan PDB Indonesia, maka dibuatlah digram alir yang dapat dilihat pada Gambar 4.1 yang merupakan siklus model dari Fuzzy Time Series dengan Particle Swarm Optimization.

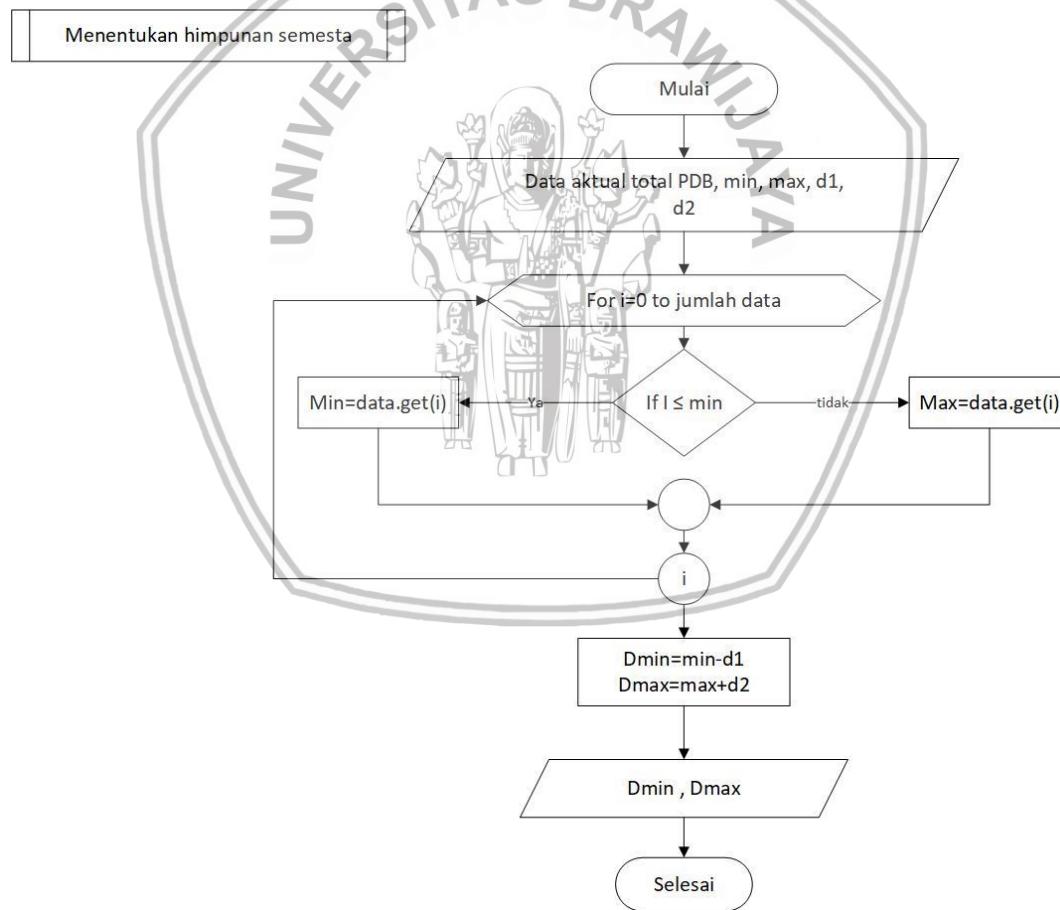


Gambar 4.1 Diagram Alir Optimasi Fuzzy Time Series menggunakan Particle Swarm Optimization

Sesuai apa yang ditunjukkan oleh Gambar 4.1 yang merupakan diagram alir yang menggambarkan tentang tahapan proses yang ada dalam algoritme optimasi *fuzzy time series* menggunakan *particle swarm optimization* untuk peramalan total PDB Indonesia, dimana masukan yang digunakan berupa data aktual total PDB Indonesia, banyaknya iterasi digunakan atau itermax, banyak jumlah partikel, jumlah dimensi partikel, nilai koefisien kognitif 1 atau c_1 dan nilai kognitif 2 atau c_2 , nilai bobot inersia atau nilai w . Dan juga terdapat 7 sub proses didalamnya yaitu menentukan himpunan semesta, menentukan sub himpunan *fuzzy*, hitung kecepatan partikel, hitung posisi partikel, hitung nilai *fitness*, menetukan *pbest* dan menetukan *gbest*. Sedangkan untuk keluaran yang dihasilkan berupa hasil peramalan.

4.2.1 Menentukan Himpunan Semesta

Tahapan proses yang ada dalam menentukan himpunan semesta dapat dilihat pada diagram alir pada Gambar 4.2. Himpunan semesta ditentukan dengan menggunakan persamaan 2.5.



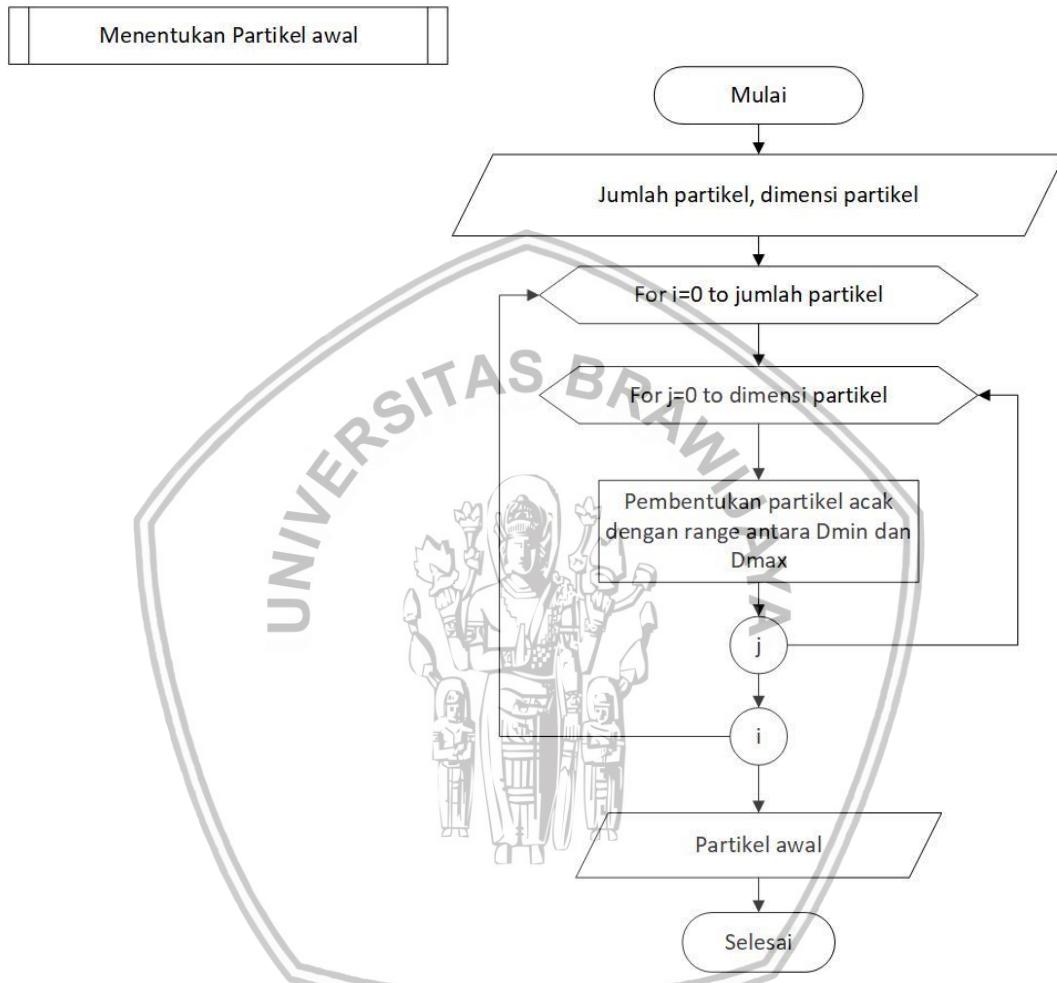
Gambar 4.2 Diagram Alir Menentukan Himpunan Semesta

Penjelasan untuk tahapan-tahapan yang ada dalam menentukan himpunan semesta sesuai Gambar 4.2 yaitu masukan berupa data aktual total PDB, nilai min, max dari data historis, serta nilai d_1 dan d_2 yang merupakan nilai random positif.

Keluaran berupa nilai Dmin dan Dmax pada himpunan semesta. Proses perhitungannya sesuai dengan persamaan 2.5.

4.2.2 Menentukan Partikel Awal

Tahap pertama dalam algoritme PSO adalah inisialisasi partikel awal, pada Gambar 4.3 menggambarkan proses yang ada dalam menentukan partikel awal.

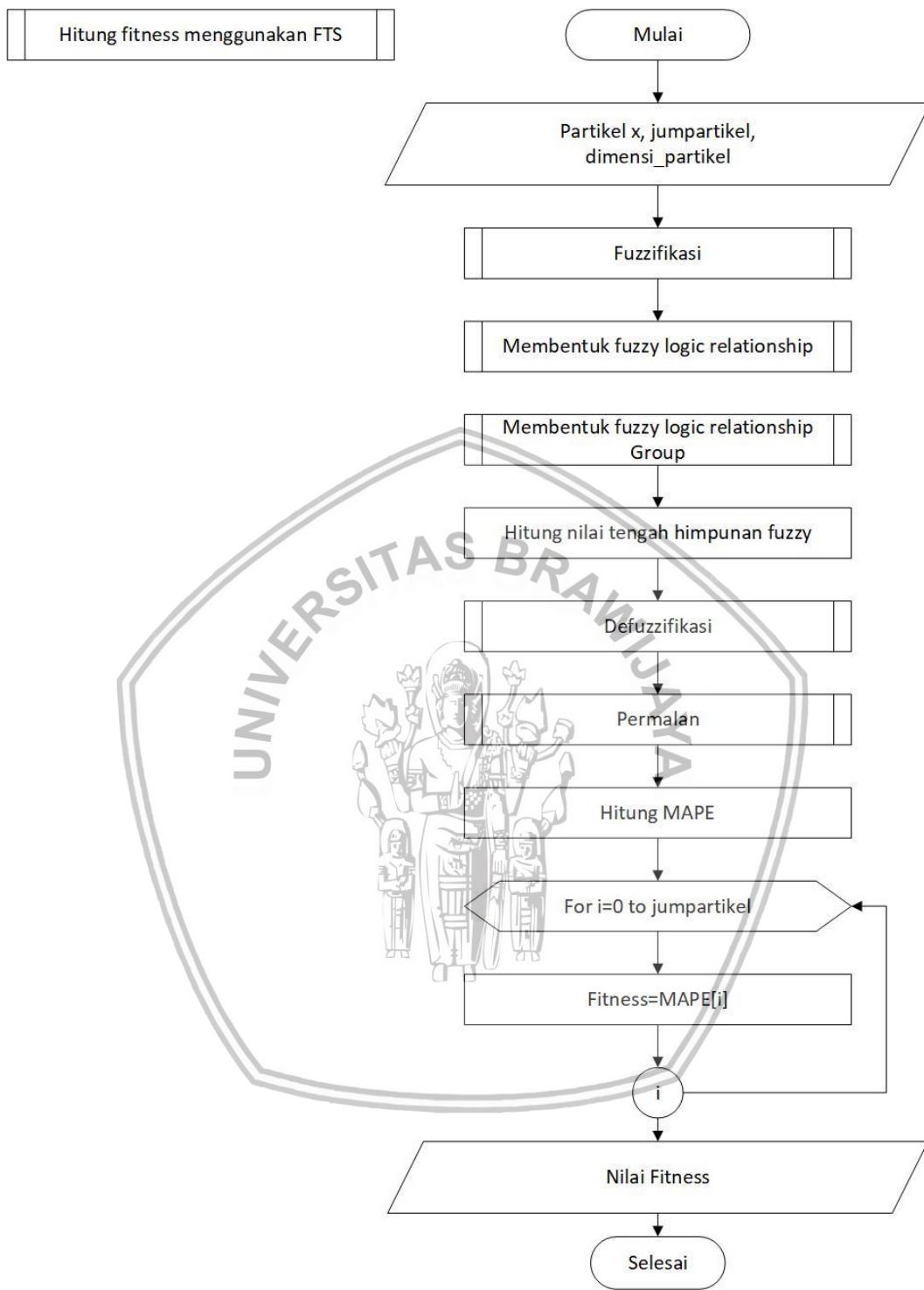


Gambar 4.3 Diagram Alir Menentukan Partikel awal

Pada tahap awal algoritme PSO yaitu inisialisasi partikel awal sesuai Gambar 4.3, masukan berupa nilai ukuran partikel atau jumlah partikel dan besar ukuran dimensi partikel. Kemudian dilakukan proses pembentukan partikel secara random atau acak dengan range antara nilai Dmin dan Dmax. Keluaran berupan nilai partikel awal.

4.2.3 Hitung *Fitness* Menggunakan FTS

Nilai *fitness* dihitung menggunakan metode *fuzzy time series* sampai pada tahap perhitungan *error* menggunakan *Mean Absolute Percentage Error* (MAPE). Pada Gambar 4.4 ditunjukkan proses perhitungan *fitness* secara lengkap.

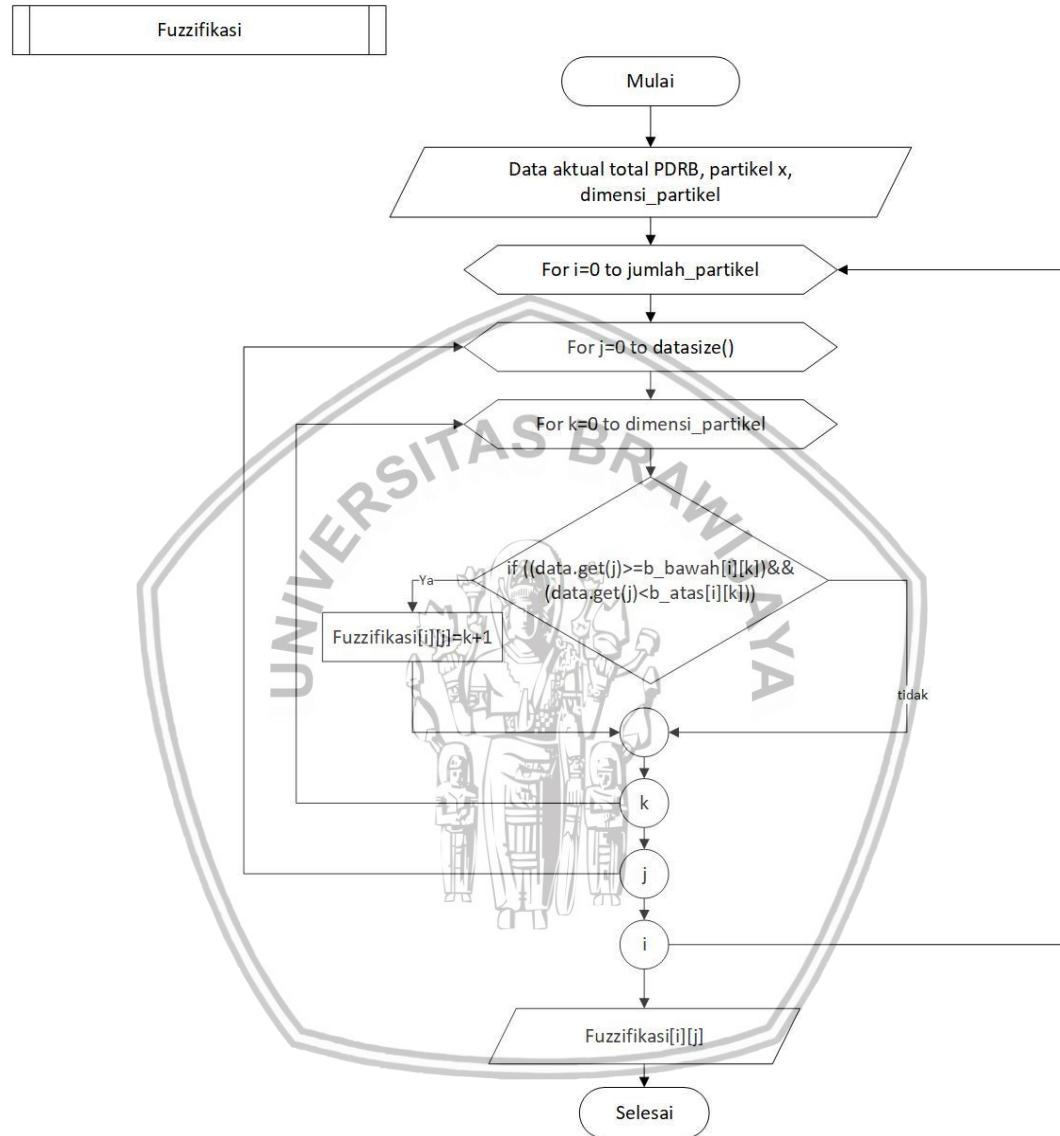


Gambar 4.4 Diagram Alir Hitung *Fitness* menggunakan FTS

Sesuai dengan apa yang ditunjukkan pada Gambar 4.4 proses perhitungan *fitness* diperoleh melalui proses FTS sampai proses perhitungan MAPE. Masukan dari proses tersebut berupa posisi partikel, jumlah partikel dan ukuran dimensi partikel, untuk keluarannya berupa nilai *fitness* pada masing masing posisi partikel. Dalam proses diatas didalamnya terdapat 5 sub proses yaitu fuzzifikasi membentuk FLR, membentuk FLGR, defuzzifikasi dan peramalan.

4.2.3.1 Fuzzifikasi

Fuzzifikasi merupakan proses yang berdasarkan pada tingkat derajat keanggotaan data aktual dengan sub himpunan *fuzzy* tertinggi. Untuk tahapan yang ada pada fuzzifikasi secara jelasnya ditunjukkan pada Gambar 4.5.



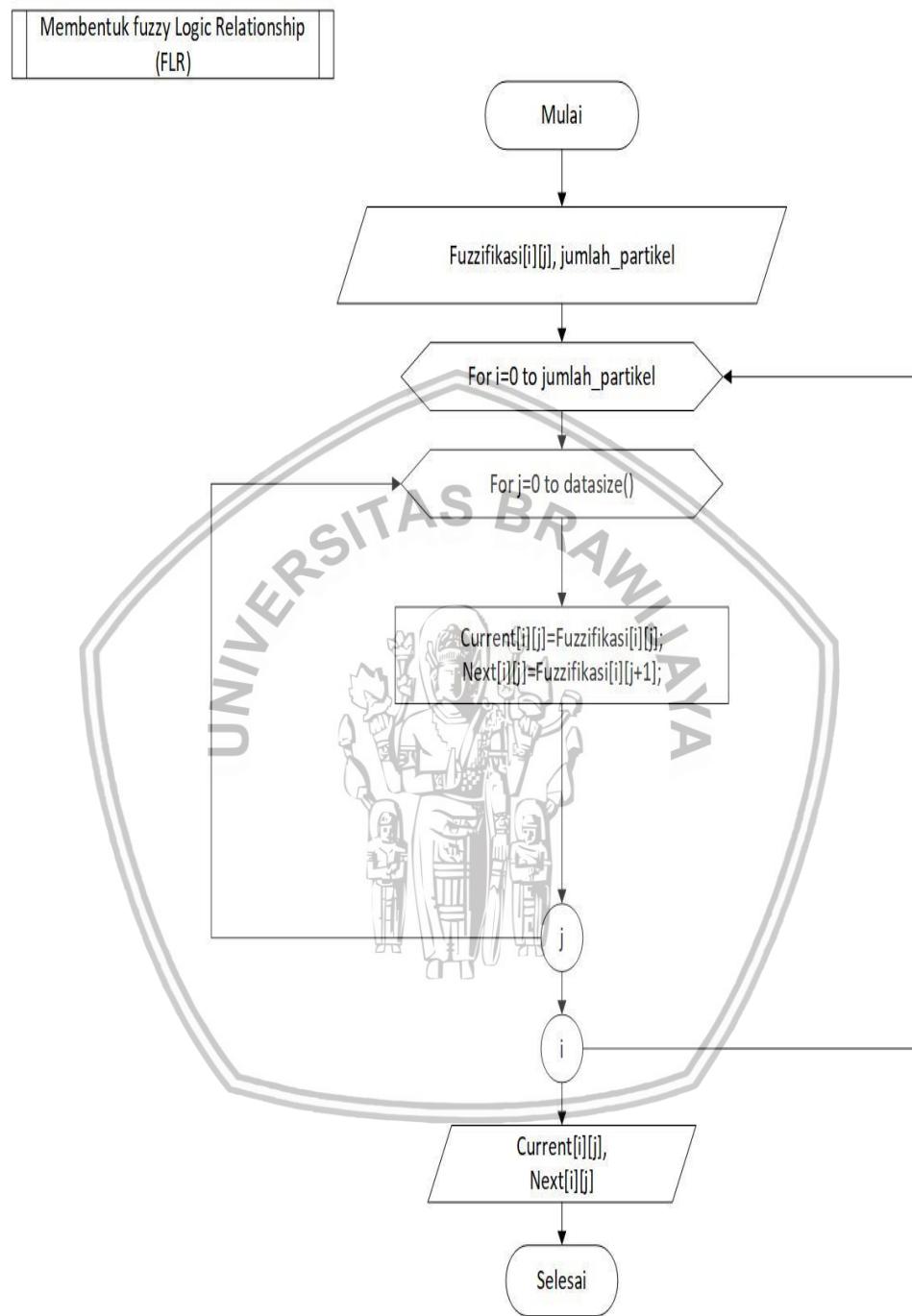
Gambar 4.5 Diagram Alir Fuzzifikasi

Proses fuzzifikasi pada FTS sesuai Gambar 4.5 menggunakan masukan berupa data aktual, posisi partikel, ukuran dimensi partikel, batas atas partikel, dan batas bawah partikel. Proses fuzzifikasi ini digunakan untuk menentukan keanggotaan data aktual dengan sub himpunan *fuzzy*. Sedangkan hasil Keluaran pada proses fuzzifikasi berupa hasil fuzzifikasi pada masing masing data.

4.2.3.2 Membentuk Fuzzy Logic Relationship (FLR)

Dalam menentukan nilai pada proses FLR ditentukan dengan cara membentuk sebuah relasi antara *current stage* $F(t)$ dengan *next stage* $F(t + 1)$ pada nilai

fuzzifikasi yang dihasilkan sebelumnya. Pada Gambar 4.6 telah digambarkan tahapan proses pembentukan FLR.

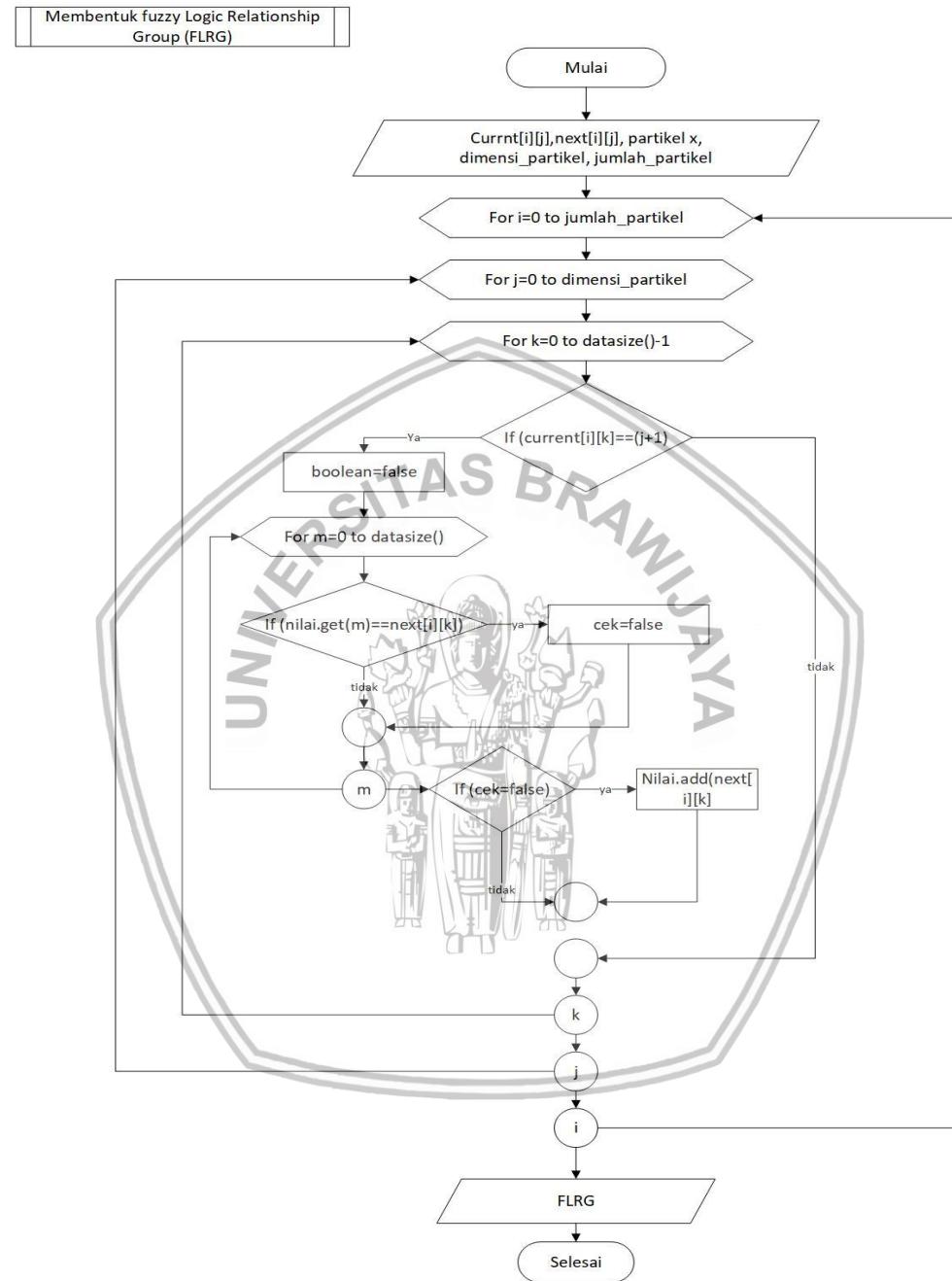


Gambar 4.6 Diagram Menentukan *Fuzzy Logic Relationship* (FLR)

4.2.3.3 Membentuk *Fuzzy Logic Relationship Group* (FLRG)

Dalam penelitian ini untuk membentuk *Fuzzy Logic Relationship Group* atau FLRG dilakukan dengan cara mengelompokan Hasil FLR pada proses sebelumnya. Pada proses pembentukan FLRG masukan yang digunakan adalah nilai current stage dan next stage hasil dari proses fuzzifikasi sebelumnya. Cara

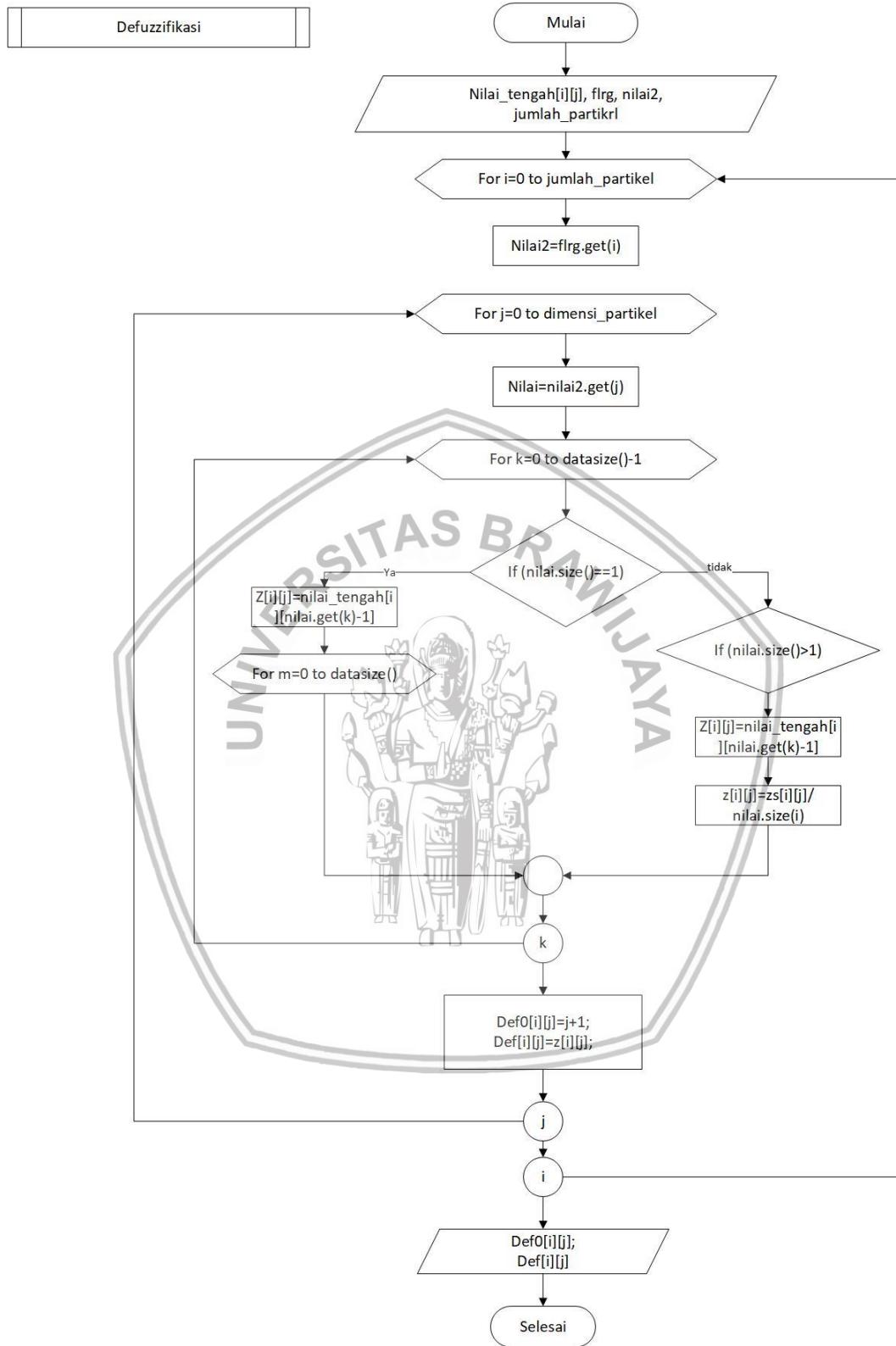
pembentukannya dengan mengelompokan relasi antara himpunan *fuzzy* yang sama pada current stage. Sedangkan keluaran berupa ralasi FLRG. Pada Gambar 4.7 telah digambaran proses pembentukan FLRG lebih jelasnya.



Gambar 4.7 Diagram Alir Menentukan *Fuzzy Logic Relationship Group* (FLRG)

4.2.3.4 Defuzzifikasi

Dalam penelitian ini pada proses defuzzifikasi digunakan untuk mengembalikan nilai *fuzzy* menjadi nilai *crips*, kemudian nantinya hasil dari proses defuzzifikasi dijadikan sebagai hasil peramalan. Pada Gambar 4.8 ditunjukkan proses yang ada dalam defuzzifikasi.

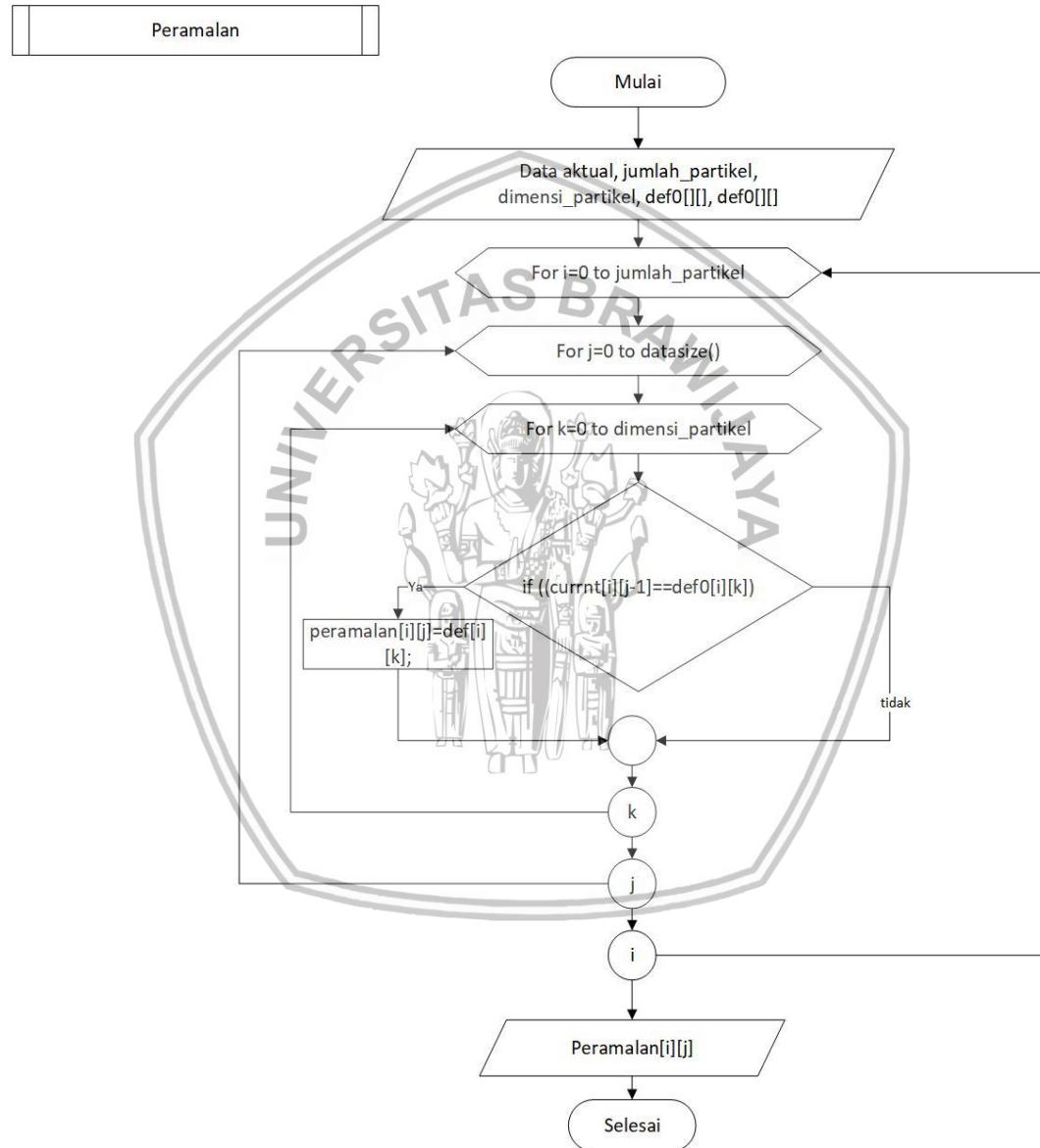
**Gambar 4.8 Diagram Alir Proses Defuzzifikasi**

Dalam melakukan perhitungan nilai pada proses defuzzifikasi dihitung berdasarkan pada nilai FLRG yang terbentuk dan nilai tengah pada sub himpunan fuzzy. Dengan beberapa ketentuan apabila hanya terdapat satu *next state* pada

FLR maka nilai defuzzifikasi sama dengan nilai tengah *next stage*. Jika terdapat lebih dari satu *next stage* pada FLR maka perhitungan defuzzifikasi menggunakan persamaan 2.8.

4.2.3.5 Peramalan

Pada proses peramalan ini digunakan untuk menghitung hasil peramalan pada masing-masing data. Diagram alir pada Gambar 4.9 menunjukkan proses peramalannya.



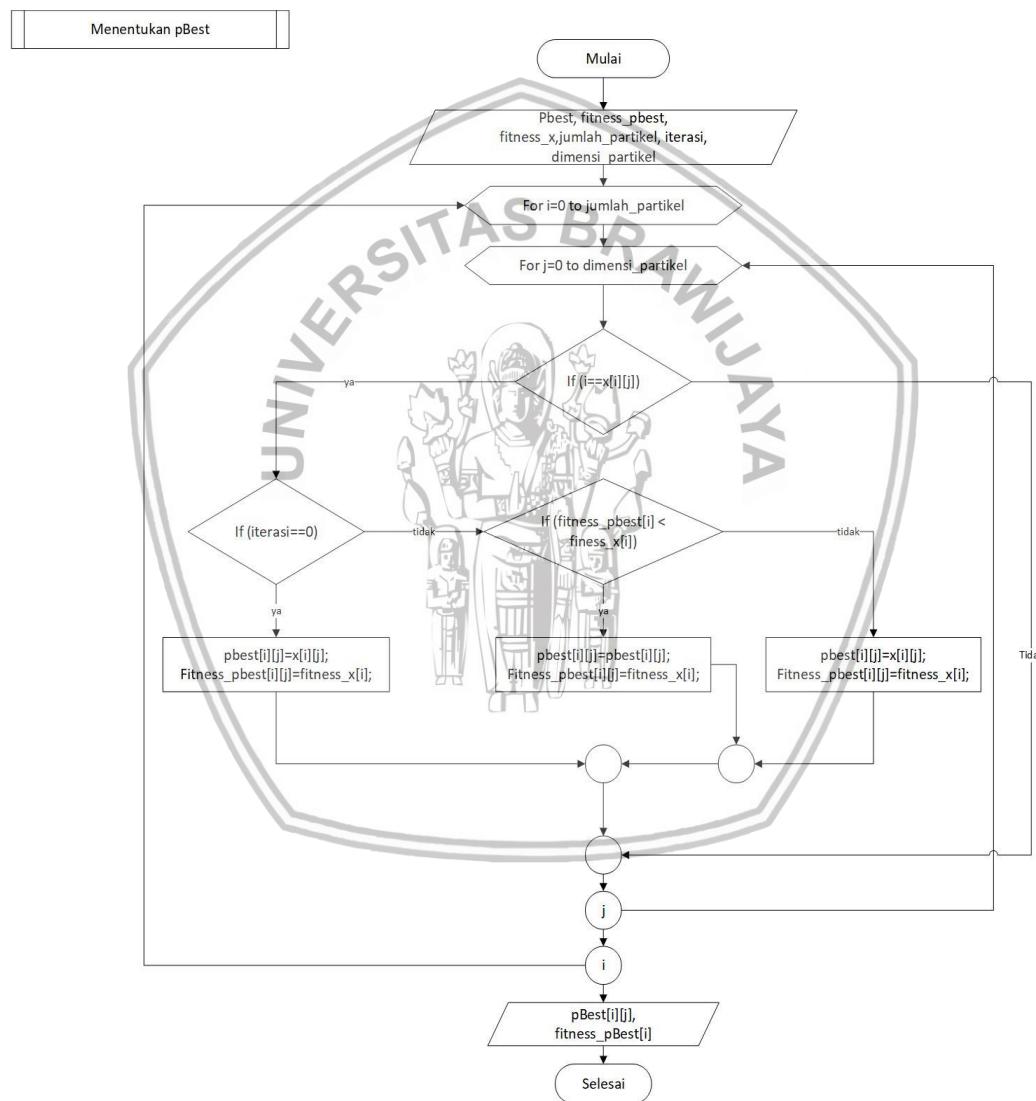
Gambar 4.9 Diagram Alir Proses Peramalan

Berdasarkan pada Gambar 4.9 menunjukkan masukan pada proses peramalan berupa data aktual, jumlah partikel, dimensi partikel dan nilai hasil defuzzifikasi. Proses peramalan dihitung dengan menggunakan nilai *current state* pada FLR dan nilai hasil defuzzifikasi. Sedangkan dalam menetukan jumlah nilai peramalan sama

dengan nilai $def[i][k]$, dimana keluaran pada proses diatas berupa hasil peramalan untuk masing-masing data.

4.2.4 Menentukan $pbest$

Penentuan $pbest$ digunakan untuk mencari posisi terbaik yang dapat dicapai oleh setiap partikel. Dalam menentukan partikel yang dapat ditetapkan sebagai $pbest$ dengan cara membandingkan $pbest$ pada iterasi sebelumnya dengan posisi partikel yang baru. Pada saat iterasi ke-0 (nol) $pbest$ akan bernilai sama dengan posisi partikel. Pada Gambar 4.10 menunjukkan gambaran proses menentukan $pbest$ lebih jelasnya.



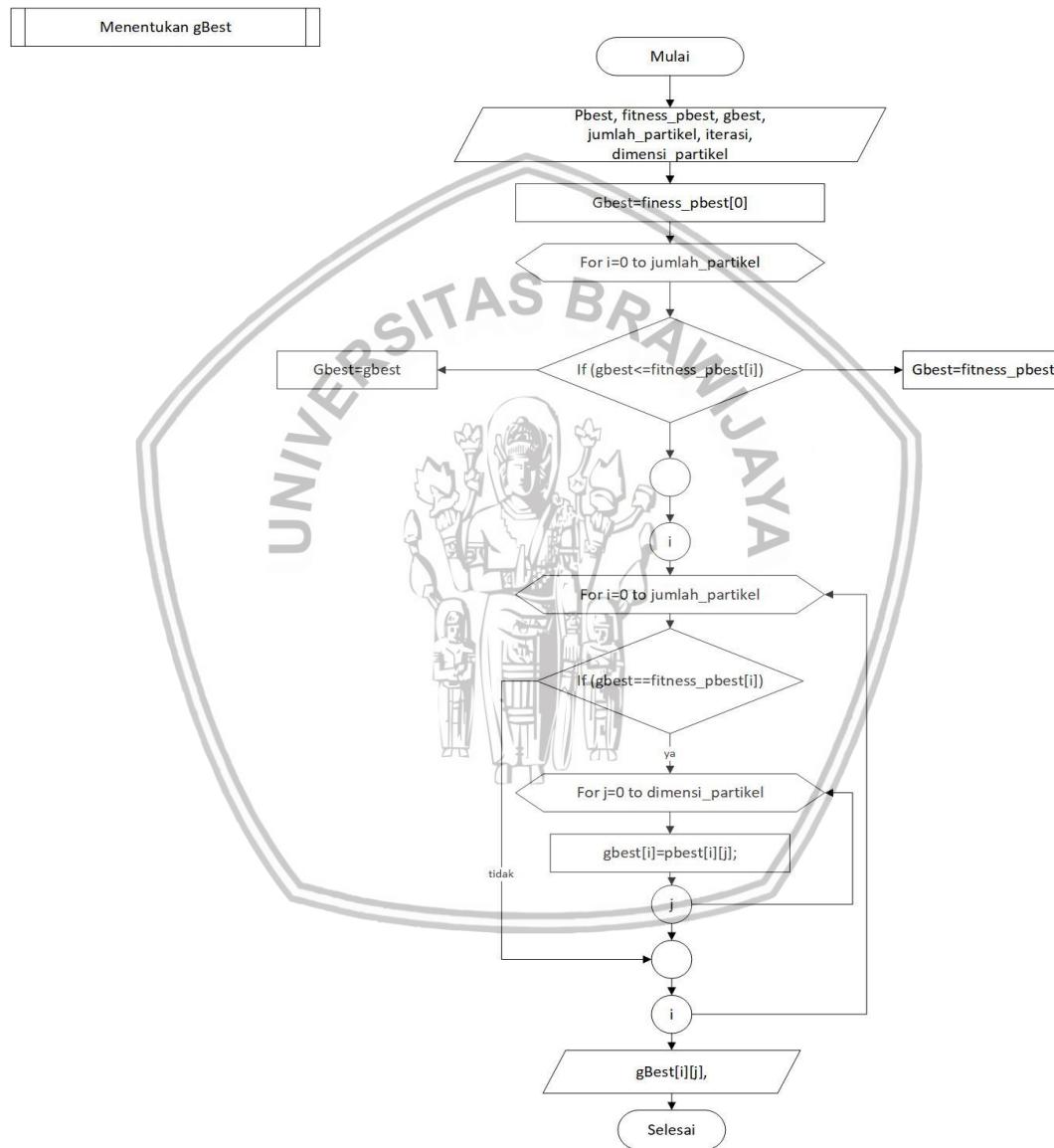
Gambar 4.10 Diagram Alir Menentukan $Pbest$

Berdasarkan apa yang ditunjukkan pada Gambar 4.10 masukan yang ada pada proses untuk menentukan $pbest$ berupa $pbest$ pada iterasi sebelumnya, nilai $fitness pbest$, nilai $fitness$ partikel, jumlah partikel, iterasi dan dimensi partikel. Jika iterasi 0 maka $pbest$ berniai sama dengan posisi partikel, sedangkan jika iterasinya selain 0 atau lebih dari 0 maka cara menentukan $pbest$ baru dengan cara

membandingkan nilai *fitness* antara *pbest* yang lama dengan posisi partikel yang baru. dimana yang mempunyai nilai *fitness* akan menjadi *pbest* yang baru.

4.2.5 Menentukan *gbest*

Menentukan *gbest* digunakan untuk mendapatkan posisi terbaik dari keseluruhan partikel. Penentuan *gbest* dilakukan dengan cara mencari *pbest* yang mempunyai nilai *fitness* terbaik atau tertinggi. Pada Gambar 4.11 menunjukkan gambaran proses secara rinci dalam menentukan *gbest*.



Gambar 4.11 Diagram Alir Menentukan *Gbest*

4.3 Perhitungan Manual

Perhitungan manual digunakan untuk menerapkan desain algoritme yang telah dilakukan terhadap objek yang digunakan. Pada perhitungan manual menggunakan *Microsoft Excel* dengan memakai 20 data latih dari tahun 2000

triwulan 1 sampai 2004 triwulan 4. Pada Tabel 4.2 menunjukkan contoh data latih yang digunakan pada perhitungan manual.

Tabel 4.2 Data Latih

NO	Tahun	Triwulan	PDB
1	2000	Triwulan 1	325958.6
2	2000	Triwulan 2	336967.1
3	2000	Triwulan 3	360701.6
4	2000	Triwulan 4	366142.6
5	2001	Triwulan 1	386648.8
6	2001	Triwulan 2	416069.9
7	2001	Triwulan 3	426828.3
8	2001	Triwulan 4	416775.0
9	2002	Triwulan 1	436975.1
10	2002	Triwulan 2	450640.4
11	2002	Triwulan 3	472136.1
12	2002	Triwulan 4	462081.8
13	2003	Triwulan 1	496247.8
14	2003	Triwulan 2	498023.8
15	2003	Triwulan 3	516103.7
16	2003	Triwulan 4	503299.3
17	2004	Triwulan 1	536605.3
18	2004	Triwulan 2	564422.1
19	2004	Triwulan 3	595320.6
20	2004	Triwulan 4	599478.2

4.3.1 Perhitungan Manual Optimasi *Fuzzy Time Series* Menggunakan *Particle Swarm Optimization*.

Berdasarkan apa yang dijelaskan pada bab sebelumnya untuk *Particle Swram Optimization* (PSO) digunakan untuk mengoptimasi jarak interval pada sub-himpunan *fuzzy time series*, serta diharapkan nantinya akan didapatkan hasil prediksi yang mendekati data aktual dengan partisi sub-himpunan yang optimal. Pada proses perhitungannya untuk langkah awal yang dilakukan yaitu inisialisasi parameter yang digunakan dalam algoritme PSO, pada perhitungan kali ini parameter PSO yang digunakan yaitu:



- Jumlah partikel : 4
- Dimensi partikel : 11
- Bobot *inersia* (w) : 0.6
- Koefisien *Cognitif* 1 (c_1) : 1.5
- Koefisien *Social* 2 (c_2) : 0.5

Kemudian langkah selanjutnya yaitu pembangkitan partikel secara acak yang dilakukan sebanyak jumlah dimensi partikel serta tidak lebih atau kurang dari batas minimum dan maksimum dan telah diurutkan dari yang terkecil ke nilai yang terbesar. Partikel yang terbentuk berupa nilai interval dari masing-masing sub-himpunan semesta pada *fuzzy time series*. Pada Gambar 4.12 menunjukkan pembangkitan partikel dalam penelitian ini.

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11
----	----	----	----	----	----	----	----	----	-----	-----

Gambar 4.12 Pembangkitan Partikel

Selanjutnya sesuai dengan algoritme yang telah dirancang langkah selanjutnya yaitu sebagai berikut:

1. Menentukan Himpunan Semesta (U)

Himpunan semesta dihitung menggunakan persamaan 2.5, dimana D_{\min} dan D_{\max} yang didapat secara berurutan adalah 691.58 dan 20747.51. sedangkan untuk nilai d_1 dan d_2 yaitu 25958.6 dan 30521.8. sehingga akan didiperoleh himpunan semesta (U) = [300000; 630000].

A. Iterasi Ke-0 (Nol)

2. Inisialisasi Kecepatan dan Partikel Awal

Pada iterasi ke-0 (nol) untuk inisialisasi kecepatan awal semua partikel sama yaitu bernilai nol (0). Sedangkan untuk inisialisasi partikel dilakukan secara acak sesuai batas bawah dan batas atas pada himpunan semesta dan telah diurutkan nilainya dari yang terkecil sampai terbesar. Secara berurutan pada Tabel 4.3 dan Tabel 4.4 menunjukkan hasil inisialisasi kecepatan awal dan inisialisasi partikel awal.

Tabel 4.3 Inisialisasi Kecepatan Awal

	X1	X2	X4	X3	X5	X6	X10	X11
V1(0)	0	0	0	0	0	0	0	0
V2(0)	0	0	0	0	0	0	0	0
V3(0)	0	0	0	0	0	0	0	0
V4(0)	0	0	0	0	0	0	0	0

Tabel 4.4 Inisialisasi Partikel Awal

Iterasi 0						
P1(0)	300716	380106	382631	400448	423803	541280
	563499	592771	607840	610371	620422	
P2(0)	387428	393208	403752	471562	490802	522222
	582860	586198	611680	624013	626146	
P3(0)	313933	343398	377639	398713	432622	454540
	458820	533205	547255	569795	601538	
P4(0)	303961	304024	310024	332756	386665	389785
	457057	497187	546926	556146	582085	

3. Hitung *Fitness*

Perhitungan *fitness* ditentukan berdasarkan nilai error MAPE menggunakan algoritme *fuzzy time series* dan lebih jelasnya akan dijelaskan pada sub bab 4.3.2. Pada Tabel 4.5 menunjukkan hasil dari perhitungan *fitness* masing-masing partikel pada iterasi ke- 0.

Tabel 4.5 Hasil Hitung *Fitness* Iterasi 0

Iterasi 0							<i>Fitness</i>
P1(0)	300716	380106	382631	400448	423803	541280	0.0575
	563499	592771	607840	610371	620422		
P2(0)	387428	393208	403752	471562	490802	522222	0.0651
	582860	586198	611680	624013	626146		
P3(0)	313933	343398	377639	398713	432622	454540	0.0288
	458820	533205	547255	569795	601538		
P4(0)	303961	304024	310024	332756	386665	389785	0.0403
	457057	497187	546926	556146	582085		

4. Menentukan Personal Best (*Pbest*)

Dalam menentukan *pbest* pada iterasi ke-0 (nol), nilai *Pbest* akan bernilai sama dengan partikel awal. Pada Tabel 4.6 menunjukkan hasil inisialisasi *Pbest* pada iterasi ke-0 yang telah dilakukan.

Tabel 4.6 *Pbest* pada Iterasi 0

Iterasi 0							<i>Fitness</i>
<i>Pbest</i> 1(0)	300716	380106	382631	400448	423803	541280	0.0575
	563499	592771	607840	610371	620422		
<i>Pbest</i> 2(0)	387428	393208	403752	471562	490802	522222	0.0572
	582860	586198	611680	624013	626146		
<i>Pbest</i> 3(0)	313933	343398	377639	398713	432622	454540	0.0288
	458820	533205	547255	569795	601538		
<i>Pbest</i> 4(0)	303961	304024	310024	332756	386665	389785	0.0431
	457057	497187	546926	556146	582085		

5. Menentukan Global Best (*gbest*)

Gbest ditentukan berdasarkan pada *pbest* yang memiliki nilai *fitness* terendah. Dalam kasus ini yang memiliki *fitness* terendah adalah *pbest* 2(0) dengan nilai *fitness* 0.27253. Pada Tabel 4.7 menunjukkan hasil penentuan *Gbest* lebih jelasnya.

Tabel 4.7 *Gbest* pada Iterasi 0

Iterasi 0							<i>Fitness</i>
<i>Gbest</i> (0)	313933	343398	377639	398713	432622	454540	0.0288
	458820	533205	547255	569795	601538		

B. Iterasi ke-1 (satu)

6. Update Kecepatan

Perhitungan *update* kecepatan akan dihitung menggunakan Persamaan 2.10. sebelum mencari kecepatan baru dilakukan perhitungan untuk mencari batas minimum dan maksimum kecepatan terlebih dahulu. Perhitungan batas minimum dan maksimum kecepatan sebagai berikut:

$$k = 0.6$$

$$v_{max} = k(x_{max_j} - x_{min_j})/2 = 0.6(630000 - 300000)/2 = 99000$$

$$v_{min} = -(v_{max}) = -99000$$

Dikarenakan dalam himpunan *fuzzy time series* terdapat rentang yang sama panjang sehingga nilai *vmax* dan *vmin* setiap dimensi partikel bernilai sama. Apabila nilai kecepatan baru kurang dari batas minimum maka nilai kecepatan baru akan diganti nilai batas minimum dan apabila kecepatan baru melebihi

batas maksimum maka nilai kecepatan baru akan diganti dengan nilai maksimum.

Selanjutnya *Update* kecepatan dilakukan pada masing-masing iterasi setiap partikel. Dalam menghitung kecepatan baru atau melakukan *update* kecepatan dengan menggunakan Persamaan 2.10. Sebagai contoh dalam salah satu proses perhitungan kecepatan baru pada partikel ke-1 untuk iterasi ke-1 sebagai berikut:

$$r1 = 0.3; r2 = 0.6$$

$$\begin{aligned} v_i^{t+1} &= w \cdot v_{ij}(t) + c_1 \cdot r_1 (Pbest_{ij}(t) - x_{ij}(t)) + c_2 \cdot r_2 (Gbest_{ij}(t) - x_{ij}(t)) \\ &= 0.6 \times 0 + 1.5 \times 0.3(300716 - 300716) + 0.5 \times 0.6(313933 - 300716) \\ &= -3965.1 \end{aligned}$$

Pada Tabel 4.8 menunjukkan hasil perhitungan *update* kecepatan untuk masing-masing partikel secara lengkap.

Tabel 4.8 Update Kecepatan Iterasi 1

Iterasi 1						
v1(1)	3965.1	-11012.4	-1497.6	-520.5	2645.7	-26022
	-31403.7	-17869.8	-18175.5	-12172.8	-5665.2	
v2(1)	-22048.5	-14943	-7833.9	-21854.7	-17454	-20304.6
	-37212	-15897.9	-19327.5	-16265.4	-7382.4	
v3(1)	0	0	0	0	0	0
	0	0	0	0	0	
v4(1)	2991.6	11812.2	20284.5	19787.1	13787.1	19426.5
	528.9	10805.4	98.7	4094.7	5835.9	

7. *Update* Posisi Partikel dan Hitung *Fitness*

Tujuan dilakukannya *update* posisi partikel yaitu agar dapat mengetahui posisi terbaru setiap partikel. Untuk perhitungan *update* posisi partikel menggunakan Persamaan 2.11 dan Perhitungan *fitness* ditentukan berdasarkan nilai *error MAPE* menggunakan algoritme *fuzzy time series* dan lebih jelasnya akan dijelaskan pada sub bab 4.3.2. contoh perhitungan *update* posisi partikel ke-1 pada iterasi ke-1 sebagai berikut:

$$\begin{aligned} x_j^{k+1} &= x_j^k + v_j^{k+1} \\ &= 300716 + (3965.1) = 304681.1 \end{aligned}$$

Pada Tabel 4.9 menampikan hasil lengkap *update* posisi partikel masing-masing pasrtikel.

Tabel 4.9 Update Posisi Partikel iterasi 1

Iterasi 1							Fitness
P1(1))	304681. 1	369093. 6	381133. 4	399927. 5	426448. 7	515258	0.0525
	532095. 3	574901. 2	589664. 5	598198. 2	614756. 8		
P2(1))	365379. 5	378265	395918. 1	449707. 3	473348	501917 .4	0.0261
	545648	570300. 1	592352. 5	607747. 6	618763. 6		
P3(1))	313933	343398	377639	398713	432622	454540	0.0288
	458820	533205	547255	569795	601538		
P4(1))	306952. 6	315836. 2	330308. 5	352543. 1	400452. 1	409211 .5	0.0454
	457585. 9	507992. 4	547024. 7	560240. 7	587920. 9		

8. Menentukan *Pbest* baru

Menentukan *Pbest* baru yaitu dengan cara membandingkan nilai *fitness* antara *Pbest* yang lama atau pada ietrasdi sebelumnya dengan posisi partikel yang baru hasil *update* posisi partikel, dimana *Pbest* yang baru dipilih dari partikel yang memiliki *fitness* terkecil. Hasil *update* *Pbest* ditunjukkan pada Tabel 4.10.

Tabel 4.10 *Pbest* Baru pada Iterasi 1

Iterasi 1							Fitness
Pbest 1(1)	304681. 1	369093. 6	381133. 4	399927. 5	426448. 7	515258	0.0525
	532095. 3	574901. 2	589664. 5	598198. 2	614756. 8		
Pbest 2(1)	365379. 5	378265	395918. 1	449707. 3	473348	501917. 4	0.0261
	545648	570300. 1	592352. 5	607747. 6	618763. 6		
Pbest 3(1)	313933	343398	377639	398713	432622	454540	0.0288
	458820	533205	547255	569795	601538		

<i>Pbest</i> 4(1)	303961	304024	310024	332756	386665	389785	0.0431
	457057	497187	546926	556146	582085		

9. Menentukan *Gbest* baru

Gbest baru ditentukan berdasarkan pada partikel *pbest* yang memiliki nilai *fitness* terendah. Dalam kasus ini yang memiki *fitness* terendah adalah *pbest2(1)* dengan nilai *fitness* 0.27253. dapat dilihat pada Tabel 4.11 menunjukkan hasil *Gbest* pada iterasi ke-1 secara jelasnya.

Tabel 4.11 *Gbest* pada Iterasi 1

Iterasi 1							<i>Fitnes</i> <i>s</i>
<i>Gbest(1)</i>	365379. 5	378265	395918. 1	449707. 3	473348	501917. 4	0.026 1
	545648	570300. 1	592352. 5	607747. 6	618763. 6		

10. Selanjutnya apabila telah masuk pada iterasi ke ($t = t + 1$), maka proses selanjutnya tinggal ulangi proses dari langkah 2 hingga seterusnya sampai tercapainya kondisi berhenti atau telah mencapai iterasi maksimum.

4.3.2 Perhitungan Manual Nilai *Fitness* Menggunakan *Fuzzy Time Series*

Nilai *fitness* merupakan nilai yang digunakan untuk mengetahui partikel tersebut baik atau buruk. Pada penelitian ini nilai *fitness* dihitung menggunakan algoritme *fuzzy times series* (FTS) sampai perhitungan error MAPE sehingga dapat dikatakan bahwa nilai *fitness* sama dengan *Mean Absolute Percentage Error* (MAPE). Pada bab ini dilakukan contoh perhitungan *fitness* untuk partikel ke-1 pada iterasi ke-0. Berdasarkan desain algoritme yang telah dibuat perhitungan *fitness* dilakukan dengan tahapan sebagai berikut:

1. Menentukan Sub Himpunan

Sub himpunan dibentuk dengan cara mempartisi himpunan semesta berdasarkan jumlah interval yang sesuai dengan data inputan, dimana jumlah interval dapat dicari dengan cara jumlah dimensi partikel ditambah satu (1). Interval dinotasikan sebagai $u_1, u_2, u_3, \dots, u_m$. Batas interval tiap sub himpunan telah didapatkan pada saat optimasi interval menggunakan algoritme PSO. Sehingga sub-himpunan yang terbentuk pada partikel ke-1 untuk iterasi ke-0 ditunjukkan pada Tabel 4.12.

Tabel 4.12 Sub-Himpunan Partike ke-1 ilterasi ke-0

u	Batas bawah	Batas atas
1	300000	300716
2	300716	380106
3	380106	382631
4	382631	400448
5	400448	423803
6	423803	541280
7	541280	563499
8	563499	592771
9	592771	607840
10	607840	610371
11	610371	620422
12	620422	630000

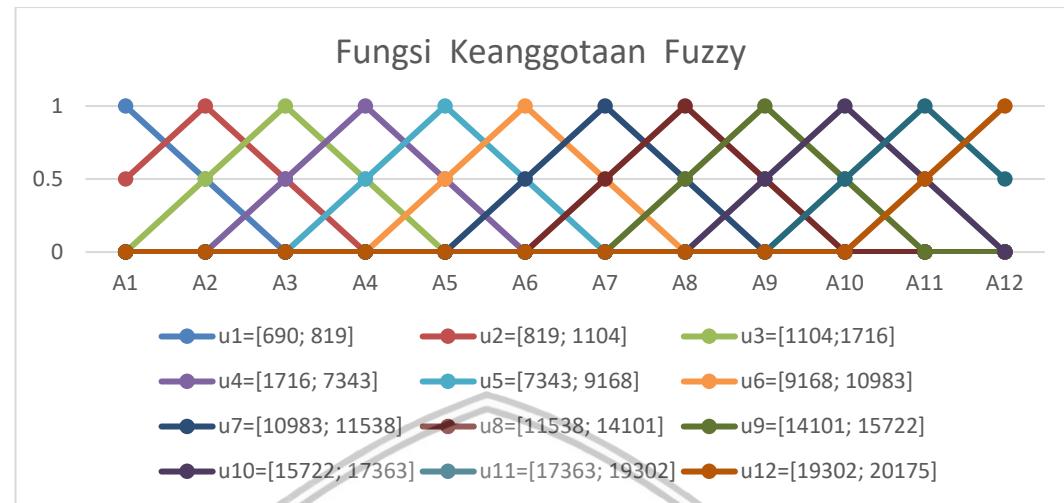
2. Membentuk Himpunan Fuzzy

Himpunan fuzzy dibentuk berdasarkan sub-himpunan semesta dan himpunan fuzzy dinotasikan sebagai A_1, A_2, \dots, A_m . pada penelitian ini himpunan fuzzy bernilai sama dengan nilai sub-himpunan. Pada Tabel 4.13 menunjukkan nilai linguistik yang digunakan pada penelitian ini dengan proses perhitungannya menggunakan rumus sesuai dengan Persamaan 2.6 dan Persamaan 2.7.

Tabel 4.13 Himpunan Fuzzy

	u1	u2	u3	u4	u5	u6	u7	u8	U9	u10	u11	u12
A1	1	0.5	0	0	0	0	0	0	0	0	0	0
A2	0.5	1	0.5	0	0	0	0	0	0	0	0	0
A3	0	0.5	1	0.5	0	0	0	0	0	0	0	0
A3	0	0	0.5	1	0.5	0	0	0	0	0	0	0
A5	0	0	0	0.5	1	0.5	0	0	0	0	0	0
A6	0	0	0	0	0.5	1	0.5	0	0	0	0	0
A7	0	0	0	0	0	0.5	1	0.5	0	0	0	0
A8	0	0	0	0	0	0	0.5	1	0.5	0	0	0
A9	0	0	0	0	0	0	0	0.5	1	0.5	0	0
A10	0	0	0	0	0	0	0	0	0.5	1	0.5	0
A11	0	0	0	0	0	0	0	0	0	0.5	1	0.5
A12	0	0	0	0	0	0	0	0	0	0	0.5	1

Pada Gambar 4.13 menunjukkan fungsi keanggotaan fuzzy yang telah terbentuk berdasarkan himpunan fuzzy.



Gambar 4.13 Fungsi Keanggoaan Fuzzy

3. Fuzzifikasi

Penentuan fuzzifikasi sesuai berdasarkan pada nilai keanggotaan data aktual terhadap himpunan fuzzy yang telah dibentuk. Dapat dilihat pada Tabel 4.14 yang menunjukkan secara jelas hasil fuzzifikasi yang didapat untuk kasus ini.

Tabel 4.14 Fuzzifikasi

No	Tahun	Triwulan	PDB	A
1	2000	Triwulan 1	325958.6	A2
2	2000	Triwulan 2	336967.1	A2
3	2000	Triwulan 3	360701.6	A2
4	2000	Triwulan 4	366142.6	A2
5	2001	Triwulan 1	386648.8	A4
6	2001	Triwulan 2	416069.9	A5
7	2001	Triwulan 3	426828.3	A6
8	2001	Triwulan 4	416775.0	A5
9	2002	Triwulan 1	436975.1	A6
10	2002	Triwulan 2	450640.4	A6
11	2002	Triwulan 3	472136.1	A6
12	2002	Triwulan 4	462081.8	A6
13	2003	Triwulan 1	496247.8	A6
14	2003	Triwulan 2	498023.8	A6



15	2003	Triwulan 3	516103.7	A6
16	2003	Triwulan 4	503299.3	A6
17	2004	Triwulan 1	536605.3	A6
18	2004	Triwulan 2	564422.1	A8
19	2004	Triwulan 3	595320.6	A9
20	2004	Triwulan 4	599478.2	A9

4. Membentuk *Fuzzy Logic Relationship* (FLR)

FLR dibentuk berdasarkan pada hasil pada proses fuzzifikasi, sehingga FLR dibentuk sesuai dengan nilai pada Tabel 4.14. misalkan untuk tahun 1988 ke tahun 1989 maka FLR nya sama dengan A1 ke A1. Hasil FLR lebih lengkapnya dapat dilihat pada Tabel 4.15.

Tabel 4.15 Hasil FLR

Tahun		FLR	
F(t) current	F(t+1) next stage	F(t) current	F(t+1) next stage
2000	Triwulan 1	2000	Triwulan 2
2000	Triwulan 2	2000	Triwulan 3
2000	Triwulan 3	2000	Triwulan 4
2000	Triwulan 4	2001	Triwulan 1
2001	Triwulan 1	2001	Triwulan 2
2001	Triwulan 2	2001	Triwulan 3
2001	Triwulan 3	2001	Triwulan 4
2001	Triwulan 4	2002	Triwulan 1
2002	Triwulan 1	2002	Triwulan 2
2002	Triwulan 2	2002	Triwulan 3
2002	Triwulan 3	2002	Triwulan 4
2002	Triwulan 4	2003	Triwulan 1
2003	Triwulan 1	2003	Triwulan 2
2003	Triwulan 2	2003	Triwulan 3
2003	Triwulan 3	2003	Triwulan 4
2003	Triwulan 4	2004	Triwulan 1
2004	Triwulan 1	2004	Triwulan 2

2004	Triwulan 2	2004	Triwulan 3	A8	A9
2004	Triwulan 3	2004	Triwulan 4	A9	A9

5. Membentuk *Fuzzy Logic Relationship Group* (FLRG)

Dari hasil proses FLR maka akan dapat dibentuk FLRG, dimana untuk proses pembentukannya dilakukan dengan cara mengelompokan relasi yang sama pada current stage menjadi satu kelompok relasi. Pada Tabel 4.16 memperlihatkan hasil FLGR untuk kasus ini.

Tabel 4.16 Hasil FLRG

F(t) current	F(t+1) next stage
A1	
A2	A2 , A4
A3	
A4	A5
A5	A6
A6	A5,A6,A8
A7	
A8	A9
A9	A9
A10	
A11	
A12	

6. Defuzzifikasi

Nilai tengah pada setiap himpunan fuzzy terlebih dahulu dicari sebelum masuk proses perhitungan defuzzifikasi. Sebagai contoh untuk mencari nilai tengah pada himpunan fuzzy A1.

$$\begin{aligned} \text{Nilai tengah } A1 &= \frac{\text{batas bawah } A1 + \text{batas atas } A1}{2} \\ &= \frac{300000 + 300716}{2} = 300358 \end{aligned}$$

Pada Tabel 4.17 menunjukkan hasil lengkap perhitungan nilai tengah untuk masing-masing himpunan fuzzy.

Tabel 4.17 Hasil Nilai Tengah

A	Batas bawah	Batas atas	Nilai Tengah
1	300000	300716	300358
2	300716	380106	340411
3	380106	382631	381368.5
4	382631	400448	391539.5
5	400448	423803	412125.5
6	423803	541280	482541.5
7	541280	563499	552389.5
8	563499	592771	578135
9	592771	607840	600305.5
10	607840	610371	609105.5
11	610371	620422	615396.5
12	620422	630000	625211

Selanjutnya membentuk defuzzifikasi berdasarkan prinsip-prinsip aturan defuzzifikasi yang dijelaskan oleh x_{hao} dan y_{min} sebagai berikut:

- Jika FLR A₁ bernilai kosong ($A_i \rightarrow \#$), maka $F(t) = m_i$ yang merupakan nilai tengah dari u_i.
- Jika FLR A₁ mempunyai ralasi one to one ($A_i \rightarrow A_j$) atau himpunan fuzzy A₁ hanya memiliki satu himpunan FLR atau FLGR A₁, maka $F(t) = m_j$ yang merupakan nilai tengah dari u_j.
- Jika FLR A₁ mempunyai ralasi one to many ($A_i \rightarrow A_{j1}, A_{j2}, A_{j3} \dots, A_{jk}$) atau himpunan fuzzy A₁ hanya memiliki lebih dari satu himpunan FLR atau FLGR A₁, maka mencari nilai F(t) menggunakan persamaan 2.4.

Dapat dilihat pada Tabel 4.18 yang menampilkan hasil defuzzifikasi yang telah dilakukan.

Tabel 4.18 Hasil Defuzzifikasi

F(t-1)	Defuzzifikasi	Hasil
A1	m1	300358
A2	(m2+m4)/2	365975.25
A3	m3	381368.5
A4	m5	412125.5
A5	m6	482541.5

A6	$(m_5+m_6+m_8)/3$	490934
A7	m_7	552389.5
A8	m_9	600305.5
A9	m_9	600305.5
A10	m_{10}	609105.5
A11	m_{11}	615396.5
A12	m_{12}	625211

7. Peramalan

Hasil peramalan diperoleh dengan cara memndapatkan nilai defuzzifikasi yang sesuai dengan FLR nya. Misalkan melakukan peramalan untuk tahun 2004 triwulan 4 adalah i, maka hasil peramalan didapat berdasarkan $(i-1)$ yaitu tahun 2004 triwulan 3, dimana tahun 2004 triwulan 3 termasuk himpunan fuzzy A9. Sehingga hasil peramalan tahun 2004 triwulan 4 sama dengan nilai defuzzifikasi A9 yaitu 600305.5. Pada Tabel 4.19 diperlihatkan hasil lengkap peramalan.

Tabel 4.19 Hasil Peramalan

Tahun	Triwulan	Data Aktual	Hasil Peramalan
2000	Triwulan 1	325958.6	
2000	Triwulan 2	336967.1	365975.25
2000	Triwulan 3	360701.6	365975.25
2000	Triwulan 4	366142.6	365975.25
2001	Triwulan 1	386648.8	365975.25
2001	Triwulan 2	416069.9	412125.5
2001	Triwulan 3	426828.3	482541.5
2001	Triwulan 4	416775.0	490934
2002	Triwulan 1	436975.1	482541.5
2002	Triwulan 2	450640.4	490934
2002	Triwulan 3	472136.1	490934
2002	Triwulan 4	462081.8	490934
2003	Triwulan 1	496247.8	490934
2003	Triwulan 2	498023.8	490934
2003	Triwulan 3	516103.7	490934

2003	Triwulan 4	503299.3	490934
2004	Triwulan 1	536605.3	490934
2004	Triwulan 2	564422.1	490934
2004	Triwulan 3	595320.6	600305.5
2004	Triwulan 4	599478.2	600305.5

8. Perhitungan Error

Mean Absolute Percentage Error (MAPE) pada penitian ini digunakan sebagai perhitungan tingkat kesalahan dari peramalan dengan perhitungan menggunakan persamaan 2.12. Pada Tabel 4.20 menunjukkan hasil perhitungan MAPE secara lengkapnya.

Tabel 4.20 Hasil Perhitungan Error

Tahun	Triwulan	Data Aktual	Hasil Peramalan	$(x-x')^2$	$ x-x' /x$
2000	Triwulan 1	325958.6			
2000	Triwulan 2	336967.1	365975.25	841472766.4	0.086086001
2000	Triwulan 3	360701.6	365975.25	27811384.32	0.014620534
2000	Triwulan 4	366142.6	365975.25	28006.0225	0.000457062
2001	Triwulan 1	386648.8	365975.25	427395669.6	0.053468548
2001	Triwulan 2	416069.9	412125.5	15558291.36	0.009480138
2001	Triwulan 3	426828.3	482541.5	3103960654	0.130528365
2001	Triwulan 4	416775.0	490934	5499557281	0.177935337
2002	Triwulan 1	436975.1	482541.5	2076296809	0.10427688
2002	Triwulan 2	450640.4	490934	1623574201	0.089414087
2002	Triwulan 3	472136.1	490934	353361044.4	0.039814579
2002	Triwulan 4	462081.8	490934	832449444.8	0.062439594
2003	Triwulan 1	496247.8	490934	28236470.44	0.010707957
2003	Triwulan 2	498023.8	490934	50265264.04	0.014235866
2003	Triwulan 3	516103.7	490934	633513798.1	0.048768687
2003	Triwulan 4	503299.3	490934	152900644.1	0.024568482
2004	Triwulan 1	536605.3	490934	2085867644	0.085111534
2004	Triwulan 2	564422.1	490934	5400500842	0.130200607
2004	Triwulan 3	595320.6	600305.5	24849228.01	0.008373471

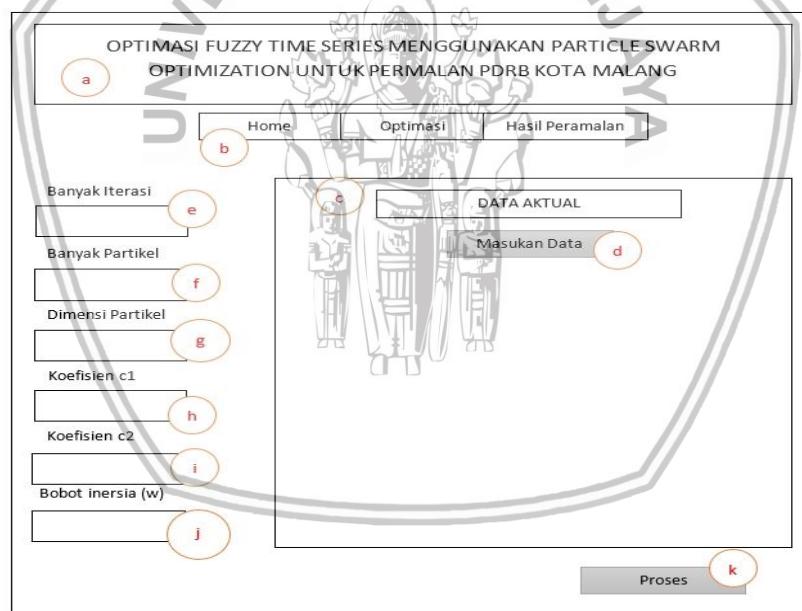
2004	Triwulan 4	599478.2	600305.5	684425.29	0.001380034
				Total	1.091867762
				HASIL MAPE	0.057466724

4.4 Rancangan Antarmuka

Dalam penelitian ini untuk program optimasi *Fuzzy Time Series* menggunakan *Particle Swarm Optimization* untuk peramalan Produk Domestik Bruto (PDB) Indonesia rancangan antarmukannya terdiri dari 3 halaman utama yaitu halaman home, optimasi, dan pengujian.

4.4.1 Rancangan Antarmukan Halaman Home

Rancangan antarmuka halaman home menampilkan data aktual yang digunakan dan juga digunakan untuk menginputkan parameter masukan untuk proses optimasi. Pada halaman ini juga terdapat tombol untuk memulai perhitungan peramalan yang terdapat pada tombol proses. Pada Gambar 4.12 menunjukkan rancangan antarmuka haaman home secara lengkapnya.



Gambar 4.14 Perancangan Antarmuka Halaman Home

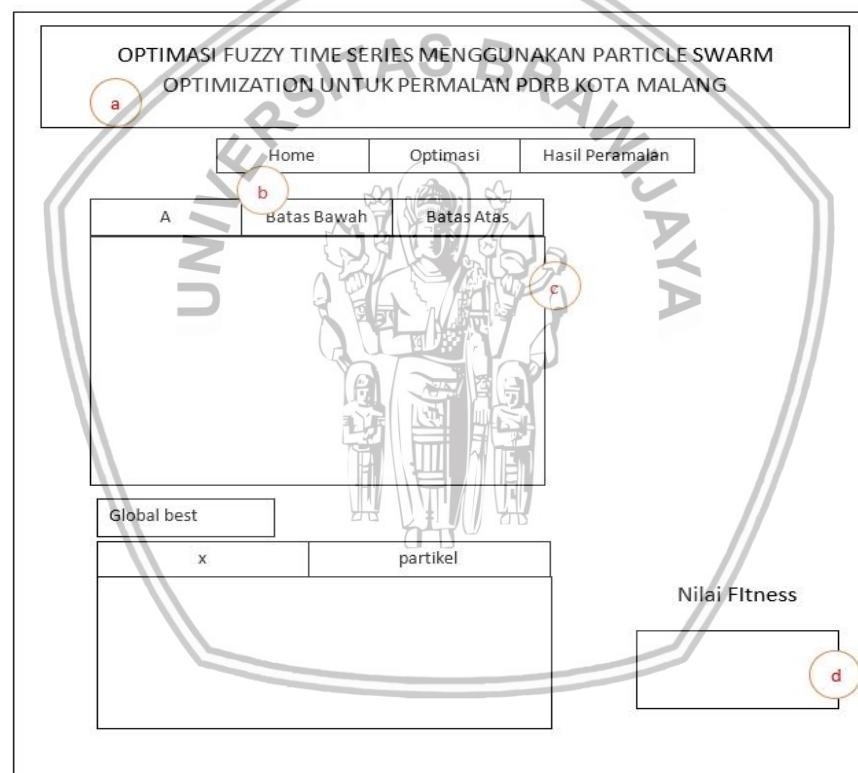
Keterangan:

- Judul atau titile dari program yang dibuat.
- Menu untuk berpindah halaman.
- Tabel yang yang digunakan untuk menampilkan data aktual
- Tombol untuk memasukan data.
- Textfield* untuk memasukan jumlah banyak iterasi.

- f. *Textfield* untuk memasukan jumlah banyak partikel.
- g. *Textfield* untuk memasukan jumlah banyak dimensi partikel.
- h. *Textfield* untuk memasukan niali koefisien c1.
- i. *Textfield* untuk memasukan nilai koefisien c2
- j. *Textfield* untuk memasukan nilai bobot inersia.
- k. Tombol untuk memulai peramalan .

4.4.2 Rancangan Antarmuka Halaman Optimasi

Pada rancangan antarmuka halaman optimasi ini digunakan agar dapat menampilkan hasil optimasi berupa menampilkan partikel yang merupakan global best dan nilai *fitnessnya* juga. Pada Gambar 4.13 menunjukkan secara jelas perancangan antarmuka halaman optimasi.



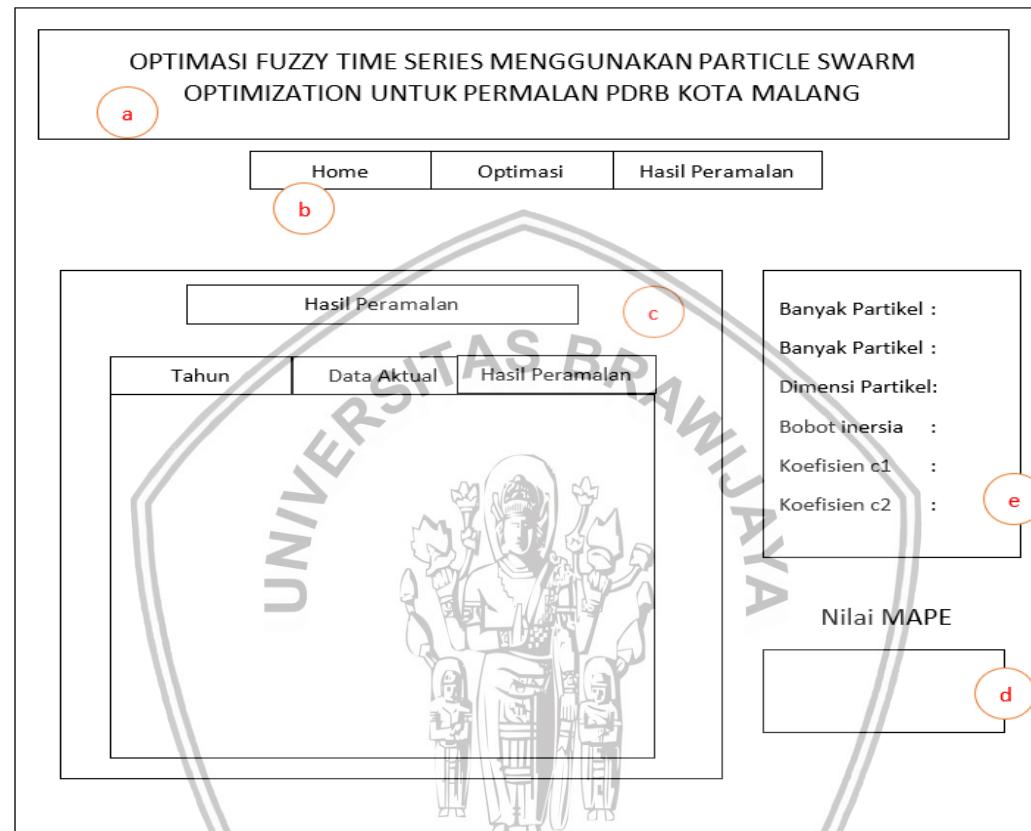
Gambar 4.15 Rancangan Antarmuka Halaman Optimasi

Keterangan:

- a. Judul atau titile dari program yang dibuat.
- b. Menu untuk berpindah halaman.
- c. Tabel untuk menampil partikel yang merupakan panjang interval yang opimal.
- d. *Textfield* untuk menampilkan nilai *fitness*.
- e. Tabel untuk menampilkan global best hasil optimasi.

4.4.3 Rancangan Antarmuka Halaman Hasil Peramalan

Pada sub bab ini akan dijelaskan mengenai rancangan antarmuka hasil peramalan digunakan untuk menampilkan hasil peramalan berdasarkan parameter optimasi yang telah masukan dan juga menampilkan hasil error peramalan berupa nilai MAPE. Pada Gambar 4.14 menunjukkan secara jelas perancangan antarmuka halaman hasil peramalan.



Gambar 4.16 Rancangan Antarmuka Halaman Hasil Peramalan

Keterangan:

- Judul atau title dari program yang dibuat .
- Menu untuk berpindah halaman .
- Tabel untuk menampilkan Hasil Peramalan.
- Textfield* untuk menampilkan nilai error MAPE.

BAB 5 IMPLEMENTASI

Pelaksanaan proses implementasi dari bab sebelumnya mengenai perancangan algoritme akan dibahas pada bab ini. Implementasi terdiri dari spesifikasi proses implementasi algoritme *Fuzzy Time Series* dengan *Particle Swarm Optimization* untuk peramalan PDB indonesia.

5.1 Implementasi Algoritme

Pemograman Java pada aplikasi *NetBean IDE* akan digunakan sebagai implementasi algoritme yang diusulkan. Pada penelitian ini akan ada dua implementasi algoritme yang akan dilakukan, yaitu implementasi algoritme *Particle Swarm Optimization* (PSO) dan implementasi algoritme *Fuzzy Time Series* (FTS).

5.1.1 Implemtasi Algoritme PSO

Implementasi algoritme PSO dilakukan untuk optimasi interval pada *Fuzzy Time Series*, Dimana prosesnya kan dilakukan berulang-ulang sampai kondisi bershenti terpenuhi atau sampai mencapai batas maksimum iterasi. Interval *Fuzzy Time Series* yang memiliki nilai *error* yang kecil merupakan tujuan dilakukannya impementasi algoritme PSO ini.

5.1.1.1 Implementasi Proses Update Kecepatan Partikel

Pada proses implementasi *update* kecepatan terdapat proses inisialisasi kecepatan awal yang di set dengan nilai 0 dan proses *update* kecepatan dengan menggunakan persamaan 2.8. Pada proses ini juga terdapat cek kecepatan agar kecepatan baru yang dihasilkan tidak lebih dari Vmax. Implementasi Proses *update* kecepatan ditunjukkan oleh Kode Program 5.1.

```

1  public double[][][] kecepatan(int iter) {
2      Random rand =new Random();
3      if (iter==0) {
4          Kecepatan=new
5          double[MAX_ITERATION+1][SWARM_SIZE][PROBLEM_DIMENSION];
6          for (int i=0 ; i<SWARM_SIZE; i++) {
7              for ( int j=0 ;j <PROBLEM_DIMENSION;j++) {
8                  Kecepatan[iter] [i] [j]= 0;
9              }
10         }
11     }else{
12         double pos;
13         double r1= rand.nextDouble();
14         double r2= rand.nextDouble();
15         //update kecepatan
16         double k =0.6;
17         double vmax =k * (Dmax-Dmin);
18         double vmin =-(vmax) ;
19         for ( int i=0; i<SWARM_SIZE; i++) {
20             for ( int j=0;j<PROBLEM_DIMENSION;j++) {

```

```

21         pos=partikel[i][j];
22         Kecepatan[iter][i][j] = (w*Kecepatan[iter -
23             1][i][j]) + ((C1*r1)*(partikelpbest[iter-1][i][j]-
24             pos)) + ((C2*r2)*(partikelgbest[iter-1][0][j]-pos));
25     }
26 }
27 //cek Kecepatan
28 for (int i=0; i<partikel.length; i++){
29     for (int j=0;j<partikel[i].length;j++){
30         if (Kecepatan[iter][i][j]>vmax ){
31             Kecepatan[iter][i][j]=vmax ;
32         }else if(Kecepatan[iter][i][j]<vmin ){
33             Kecepatan[iter][i][j]=vmin ;
34         }else{
35             Kecepatan[iter][i][j] = Kecepatan[iter][i][j] ;
36         }
37     }
38 }
39 }
40 }
41 return Kecepatan;
42 }
```

Kode Program 5.1 Implementasi Proses *update* Kecepatan Partikel

Penjelasan dari Kode Program 5.1 dapat dilihat dibawah ini.

1. Baris 6 sampai 10 merupakan perulangan untuk melakukan inisialisasi kecepatan awal.
2. Baris 19 sampai 26 merupakan perulangan untuk melakukan *update* kecepatan.
3. Baris 28 sampai 40 adalah untuk melakukan cek kecepatan agar tidak melebihi kecepatan maksimum yang telah ditentukan.

5.1.1.2 Implementasi Proses *Update Posisi* Partikel

Pada proses implementasi yang ada pada *update* posisi partikel terdapat proses inisialisasi posisi awal yang nilainya didapat dari nilai random dan proses *update* posisi partikel dengan menggunakan persamaan 2.9. Pada Kode Program 5.2 menunjukkan implementasi dari proses *update* posisi partikel.

```

1 public double[][] posisi(int iter) {
2     Random rand =new Random();
3     if (iter==0){
4         partikel=new
5         double[SWARM_SIZE] [PROBLEM_DIMENSION];
6         int cov=(int)Math.round(Dmax);
7         for ( int i=0 ; i<partikel.length ; i++){
8             for ( int j=0 ;j<partikel[i].length ; j++){
9                 partikel[i][j]=rand.nextInt(cov)+Dmin;
10            }
11        }
12        for ( int i=0; i<partikel.length; i++) {
```



```

13         Arrays.sort( partikel[i]);
14     }
15 }else{
16     //update posisi
17     double pos;
18     for (int i=0; i<partikel.length; i++){
19         for (int j=0; j<partikel[i].length; j++) {
20             partikel[i][j]=partikel[i][j]+Kecepatan
21 [iter][i][j] ;
22         }
23     }
24 }
25     return partikel;
26 }
```

Kode Program 5.2 Implementasi Proses *Update Posisi Partikel*

Penjelasan dari Kode Program 5.2 dapat dilihat dibawah ini.

1. Baris 7 sampai 11 merupakan perulangan untuk melakukan inisialisasi posisi partikel awal.
2. Baris 12 sampai 14 merupakan perulangan untuk melakukan sorting atau pengurutan terhadap partikel awal.
3. Baris 18 sampai 23 merupakan perulangan untuk melakukan *update* posisi partikel.

5.1.1.3 Implementasi Proses Hitung *fitness*

Nilai *fitness* digunakan untuk mengetahui seberapa bagus atau buruk partikel yang terbentuk, dimana dihitung dengan menggunakan metode *fuzzy time series* sampai pada tahap perhitungan error menggunakan *Mean Absolute Percentage Error* (MAPE). Pada Kode Program 5.3 menunjukkan implementasi dari Proses hitung *fitness*.

```

1 public double[] Fitness(int iterasi) {
2     fuzzifikasi();
3     FLR();
4     flgr();
5     nilaitengah();
6     defuzzifikasi();
7     peramalan();
8     Mape();
9     fitness=new double[SWARM_SIZE];
10    for(int k=0;k<SWARM_SIZE;k++) {
11        boolean ceks=false;
12        for(int l=0;l<PROBLEM_DIMENSION+1;l++) {
13            if(batas_bawah[k][l]>Dmax ||
14                batas_bawah[k][l]<Dmin) {
15                ceks=true;
16            }else if(batas_atas[k][l]>Dmax ||
```



```

17             batas_atas[k][l]<Dmin) {
18                 ceks=true;
19             }
20         }
21         if(ceks==true) {
22             fitness[k]=100;
23         }else{
24             fitness[k]=mape[k];
25         }
26     }
27     return fitness;
28 }
```

Kode Program 5.3 Implementasi Proses Hitung *fitness*

Penjelasan dari Kode Program 5.3 dapat dilihat dibawah ini.

1. Baris 2 sampai 8 perintah untuk memanggil method fuzzifikasi, FLR, FLGR, nilaitengah, defuzzifikasi, peramalan dan Mape.
2. Deklarasi variabel ceks yang benilai false
3. Baris 10 sampai 20 merupakan perulangan untuk melakukan cek kondisi jika ada parikel yang melebihi dmax dan kurang dari dmin, dimana jika kondisi benar maka variabel ceks akan bernilai true.
4. Baris 21 sampai 25 jdigunakan untuk cek kondisi jika varaibel ceks bernilai true maka variabel *fitness* sama dengan 100, jika tidak akan sama dengan hasil mape.

5.1.1.4 Implementasi Proses Update *Pbest*

Update Pbest dilakukan dengan cara membandingankan *pbest* pada iterasi sebelumnya dengan *fitness* partikel yang baru, dengan ketentuan jika *Pbest* sebelumnya memilki *fitness* yang lebih kecil dari *fitness* partikel baru maka *Pbest* baru bernilai sama dengan *Pbest* sebelumnya dan berlaku sebaliknya. Pada iterasi ke- 0 *pbest* bernilai sama dengan *fitness* partikel. Pada Kode Program 5.4 menunjukkan implementasi dari proses hitung *fitness* .

```

1 public double[][] pbest(int iter) {
2     p_pbest(iter);
3     if(iter==0) {
4         pbest=new double[MAX_ITERATION+1][SWARM_SIZE];
5         for (int i=0; i<SWARM_SIZE; i++) {
6             for (int j=0; j<PROBLEM_DIMENSION; j++) {
7                 pbest[iter][i]=fitness[i];
8             }
9         }
10    }else{
11        for (int i=0; i<SWARM_SIZE; i++) {
12            for (int j=0; j<PROBLEM_DIMENSION; j++) {
13                if (pbest[iter-1][i]<=fitness[i]) {
14                    pbest[iter][i]=pbest[iter-1][i];
15                }
16            }
17        }
18    }
19 }
```



```

15         }else{
16             pbest[iter][i]=fitness[i] ;
17         }
18     }
19 }
20 return pbest;
21 }
22 }
```

Kode Program 5.4 Implementasi Proses *Update Pbest*

Penjelasan dari Kode Program 5.4 dapat dilihat dibawah ini.

1. Baris 5 sampai 9 merupakan perulangan untuk melakukan *update Pbest* pada iterasi 0.
2. Baris 11 sampai 19 merupakan perulangan untuk melakukan *update Pbest* pada iterasi lebih dari 0.

5.1.1.5 Implementasi Proses *Update Gbest*

Update Gbest ditentukan dengan cara mencari *Pbest* yang mempunyai nilai *fitness* terbaik atau terendah. Pada Kode Program 5.5 menunjukkan implementasi dari proses *update Gbest*.

```

1 public void gbest(int iter){
2     partikelgbest=new
3     double[MAX_ITERATION+1][SWARM_SIZE][PROBLEM_DIMENSION];
4     gbest=new double[MAX_ITERATION+1];
5     double nilaiM=Double.MAX_VALUE;
6     double[][][] partikeltemp=new
7     double[MAX_ITERATION+1][SWARM_SIZE][PROBLEM_DIMENSION];
8     int k=0;
9     for (int i=0; i<partikel.length; i++){
10        for (int j=0; j<partikel[i].length; j++){
11            partikeltemp[iter][i][j]=partikelpbest[iter]
12 [i][j];
13            if(pbest[iter][i]<=nilaiM){
14                nilaiM=pbest[iter][i] ;
15                partikelgbest[iter][k][j]=partikeltemp[iter]
16 [i][j];
17            }
18        }
19        gbest[iter]=nilaiM;
20    }
21 }
```

Kode Program 5.5 Implementasi Proses *Update Gbest*

Penjelasan dari Kode Program 5.5 dapat dilihat dibawah ini.

1. Baris 2 deklarasi variabel partikel *gbest* untuk menempung nilai partikel *gbest*.
2. Baris 4 deklarasi variabel *gbest* untuk menampung nilai *fitness gbest*.

3. Baris 8 sampai 17 merupakan perulangan untuk melakukan proses *Update Gbest*.

5.1.2 Implementasi Algoritme Fuzzy Time Series (FTS)

Dalam penelitian ini algoritme *fuzzy time series* akan digunakan sebagai implementasi proses peramalan, dimana akan terjadi berbagai proses didalamnya diantaranya yaitu mencari himpunan semesta, membentuk range interval, menentukan FLR dan FLGR, menghitung defuzzifikasi, proses peramalan dan hitung error MAPE.

5.1.2.1 Implementasi Proses Menentukan Himpunan Semesta

Dalam sub bab ini akan dilakukan penentuan nilai maksimum (Dmax) dan nilai minimum (Dmin) pada data historis, kemudian menentukan himpunan semesta (U) dengan menggunakan Persamaan 2.1. Pada Kode Program 5.6 menunjukkan implementasi dari proses menentukan himpunan semesta.

```

1  public void Semesta(){
2      double min=9999999;
3      double max=-9999999;
4      double d1=1.58;
5      double d2=2.49;
6      for (int i=0; i<datapdrb.size(); i++) {
7          if (datapdrb.get(i).getNilai()<= min) {
8              min=datapdrb.get(i).getNilai();
9          } else{
10              min=min;
11          }
12          if (datapdrb.get(i).getNilai ()>= max) {
13              max=datapdrb.get(i).getNilai();
14          } else{
15              max=max;
16          }
17      }
18      Dmin=min-d1;
19      Dmax=max+d2;
20  }

```

Kode Program 5.6 Implementasi Proses Menentukan Himpunan Semesta

Penjelasan dari Kode Program 5.6 dapat dilihat dibawah ini.

1. Baris 6 sampai 17 merupakan perulangan untuk mencari nilai maximum dan minimum pada data yang digunakan.
2. Baris 18 sampai 19 merupakan perintah untuk menentukan Dmin dan Dmax.

5.1.2.2 Implementasi Proses Membentuk Range Interval

Range interval dibentuk berdasarkan nilai posisi partikel serta nilai Dmin dan Dmax, sehingga jumlah interval didapat dengan cara jumlah dimensi partikel ditambah satu (1). Pada Kode Program 5.7 menunjukkan implementasi dari proses menentukan panjang interval.

```
1 public void interval() {
2     batas_bawah=new
3     double[SWARM_SIZE] [PROBLEM_DIMENSION+1];
4     batas_atas=new
5     double[SWARM_SIZE] [PROBLEM_DIMENSION+1];
6     double[][] temp_in=new double[SWARM_SIZE]
7     [PROBLEM_DIMENSION+2];
8     for (int i=0; i<temp_in.length; i++) {
9         for (int j=0;j<temp_in[i].length; j++) {
10            if (j==0){
11                temp_in[i][j]=Dmin;
12            }else if (j>0 && j<temp_in[i].length-1) {
13                temp_in[i][j]=partikel[i][j-1];
14            }else{
15                temp_in[i][j]=Dmax;
16            }
17        }
18    }
19    for(int i=0; i<batas_bawah.length;i++){
20        Arrays.sort(temp_in[i]);
21    }
22    for (int i=0;i<batas_bawah.length;i++){
23        for(int j=0;j<batas_bawah[i].length;j++) {
24            batas_bawah[i][j]=temp_in[i][j];
25            batas_atas[i][j]=temp_in[i][j+1];
26        }
27    }
28 }
```

Kode Program 5.7 Implementasi Proses Membentuk Range Interval

Penjelasan dari Kode Program 5.7 dapat dilihat dibawah ini.

1. Baris 2 deklarasi variabel batas_bawah untuk menampung nilai batas bawah
2. Baris 4 deklarasi variabel batas_atas untuk menampung nilai batas atas
3. Baris 6 sampai 7 deklarasi variabel temp_in untuk menampung nilai interval.
4. Baris 8 sampai 18 perulangan yang digunakan untuk mendapatkan nilai interval
5. Baris 19 sampai 21 perulangan untuk mengurutkan nilai yang ada pada varaibel temp_in
6. Baris 6 sampai 17 merupakan perulangan untuk membentuk range interval.

5.1.2.3 Implementasi Proses Fuzzifikasi

Fuzzifikasi ditentukan berdasarkan pada nilai keanggotaan data aktual dengan himpunan fuzzy yang telah dibentuk, dimana himpunan fuzzy dibentuk berdasarkan persamaan 2.2. Pada Kode Program 5.8 menunjukkan implementasi dari proses fuzzifikasi.

```

1 public void fuzzifikasi() {
2     fuzzifikasi=new int[SWARM_SIZE][datapdrb.size()];
3     for(int i=0;i<fuzzifikasi.length;i++) {
4         for (int j=0; j<fuzzifikasi[i].length; j++) {
5             for (int k=0; k<PROBLEM_DIMENSION+1; k++) {
6                 if (batas_bawah[i][k]<=
7                     datapdrb.get(j).getNilai()
8                     &&datapdrb.get(j).getNilai()
9                     <batas_atas[i][k]){
10                     fuzzifikasi[i][j]=k+1;
11                 }
12             }
13         }
14     }
}

```

Kode Program 5.8 Implementasi Proses Fuzzifikasi

Penjelasan dari Kode Program 5.8 dapat dilihat dibawah ini.

1. Baris 1 deklarasi variabel fuzzifikasi untuk menampung nilai fuzzifikasi
2. Baris 4 sampai 17 merupakan perulangan untuk menentukan fuzzifikasi.

5.1.2.4 Implementasi Proses Menentukan FLR

Fuzzy Logical Relationship (FLR) dibentuk berdasarkan hasil fuzzifikasi, misal terdapat 2 himpunan fuzzy secara berurutan $A_i(t - 1)$ dan $A_j(t)$ maka dapat dinyatakan sebagai $A_i \rightarrow A_j$. Pada Kode Program 5.9 menunjukkan implementasi dari proses menentukan FLR .

```

1 public void FLR(){
2     current =new int[SWARM_SIZE][datapdrb.size()];
3     next=new int[SWARM_SIZE][datapdrb.size()];
4     for (int i=0; i<current.length; i++) {
5         for (int j=0; j<current[i].length-1; j++) {
6             current[i][j]=fuzzifikasi[i][j];
7             next[i][j]=fuzzifikasi[i][j+1];
8         }
9     }
10 }

```

Kode Program 5.9 Implementasi Proses Menentukan FLR

Penjelasan dari Kode Program 5.9 dapat dilihat dibawah ini.

1. Deklarasi variabel current dan next untuk menampung nilai current dan next.
2. Baris 4 sampai 17 merupakan perulangan untuk Menentukan nilai variabel current dan Next yang digunakan untuk menentukan FLR.

5.1.2.5 Implementasi Proses Menentukan FLGR

FLRG dibentuk berdasarkan hasil dari proses FLR. FLGR dibentuk dengan cara mengelompokan relasi yang sama pada current stage menjadi satu kelompok relasi. Pada Kode Program 5.10 menunjukkan implementasi dari proses menentukan FLGR.



```

1  public void flgr(){
2      flgr = new ArrayList<>() ;
3      for (int i=0; i<SWARM_SIZE; i++) {
4          nilai2= new ArrayList<>() ;
5          for (int j=0; j<PROBLEM_DIMENSION+1; j++) {
6              nilai = new ArrayList();
7              for (int k=0; k<datapdrb.size()-1; k++) {
8                  if (current[i][k]==j+1){
9                      boolean cek=false ;
10                     for (int l=0; l<nilai.size(); l++){
11                         if ((int)nilai.get(l)== next[i][k]){
12                             cek = true;
13                         }
14                     }
15                     if (cek==false){
16                         nilai.add(next[i][k]) ;
17                     }
18                 }
19                 nilai2.add(nilai);
20             }
21             flgr.add(nilai2) ;
22         }
23     }
24 }
```

Kode Program 5.10 Implementasi Proses Menentukan FLGR

Penjelasan dari Kode Program 5.10 dapat dilihat dibawah ini.

1. Deklarasi variabel flgr untuk menampung nilai flgr.
2. Baris 4 sampai 17 merupakan perulangan untuk menentukan FLGR.

5.1.2.6 Implementasi Proses Defuzzifikasi

Defuzzifikasi dibentuk berdasarkan prinsip-prinsip aturan defuzzifikasi yang dijelaskan oleh Xihao dan Ymin (2008) sebagai berikut:

- Jika FLR A1 bernilai kosong ($A_i \rightarrow \#$), maka $F(t) = m_i$ yang merupakan nilai tengah dari u_i .
- Jika FLR A1 mempunyai ralasi one to one ($A_i \rightarrow A_j$) atau himpunan fuzzy A1 hanya memiliki satu himpunan FLR atau FLGR A1, maka $F(t) = m_j$ yang merupakan nilai tengah dari u_j .
- Jika FLR A1 mempunyai ralasi one to many ($A_i \rightarrow A_{j1}, A_{j2}, A_{j3}, \dots, A_{jk}$) atau himpunan fuzzy A1 hanya memiliki lebih dari satu himpunan FLR atau FLGR A1, maka mencari nilai $F(t)$ menggunakan persamaan 2.4.

Pada Kode Program 5.11 menunjukkan implementasi dari proses Defuzzifikasi.



```
1  public void nilaitengah(){
2      midpoint=new
3      double[SWARM_SIZE] [PROBLEM_DIMENSION+1];
4      for (int i=0; i<SWARM_SIZE; i++){
5          for (int j=0;j<PROBLEM_DIMENSION+1; j++){
6              midpoint[i][j]= (batas_atas[i][j] +
7              batas_bawah[i][j])/2 ;
8          }
9      }
10 }
11 public void defuzzifikasi() {
12     def0 =new double[SWARM_SIZE] [nilai2.size()] ;
13     def =new double[SWARM_SIZE] [nilai2.size()];
14     double nTengah[][]= new double [SWARM_SIZE]
15     [nilai2.size()];
16     double plus[][]= new double[SWARM_SIZE]
17     [nilai2.size()] ;
18     for (int i=0; i<SWARM_SIZE; i++){
19         nilai2 =(ArrayList) flgr.get(i) ;
20         for (int j=0; j<nilai2.size(); j++){
21             nilai=(ArrayList) nilai2.get(j);
22             if(nilai.isEmpty()){
23                 nTengah[i][j]=midpoint[i][j];
24             }
25             for (int k=0; k<nilai.size(); k++){
26                 if (nilai.size()== 1){
27                     nTengah[i][j]=midpoint[i]
28                     [(int)nilai.get(k)-1 ];
29                 } else if (nilai.size()>1){
30                     plus[i][j]+= midpoint[i]
31                     [(int)nilai.get(k)-1 ];
32                     nTengah[i][j]=plus[i][j] /
33                     nilai.size();
34                 }
35             }
36             def0[i][j] = j+1 ;
37             def[i][j] = nTengah[i][j] ;
38         }
39     }
40 }
```

Kode Program 5.11 Implementasi Proses Defuzzifikasi

Penjelasan dari Kode Program 5.11 dapat dilihat dibawah ini.

1. Baris 2 deklarasi variabel midpoint untuk menampung nilai midpoint atau nilai tengah.
2. Baris 3 sampai 8 merupakan perulangan untuk melakukan perhitungan nilai tengah.
3. Baris 15 sampai 37 merupakan perulangan untuk membentuk defuzzifikasi.

5.1.2.7 Implementasi Proses Peramalan

Hasil peramalan diperoleh dengan cara memndapatkan nilai defuzzifikasi yang sesuai dengan FLR nya. Misalkan melakukan peramalan untuk tahun 2007 adalah i, maka hasil peramalan didapat berdasarkan (i-1) yaitu tahun 2006, dimana tahun 2006 termasuk himpunan fuzzy A11. Sehingga hasil peramalan tahun 2007 sama dengan niali defuzzifikasi A11 yaitu 20175. Pada Kode Program 5.12 menunjukkan implementasi dari proses peramalan.

```

1  public double[][] peramalan(){
2      prediksi=new
3      double[SWARM_SIZE][datapdrb.size()+1];
4      for (int i=0; i<SWARM_SIZE; i++){
5          System.out.println ("partike ke "+i );
6          System.out.println ("Bulan\t\tAktual\t\tHasil
7          Ramalan");
8          for(int j=0; j<=datapdrb.size(); j++){
9              if (j==0){
10                  System.out.println
11                  (datapdrb.get(j).getTahun ()+"\t\t"
12                  +datapdrb.get(j).getNilai ());
13              }
14              else if(j==datapdrb.size()){
15                  int temp=fuzzifikasi[i][datapdrb.size()-
16                  1];
17                  prediksi[i][j]=def[i][temp-1];
18                  int tahunberikutnya=datapdrb.get(j-
19                  1).getTahun ()+1;
20                  System.out.println
21
22                  (tahunberikutnya+"\t\t\t"+prediksi[i][j]);
23              }
24              else{
25                  for(int k=0;k<PROBLEM_DIMENSION+1;k++){
26                      if (current[i][j-1] == def0[i][k]){
27                          prediksi[i][j]=def[i][k];
28                          System.out.println(datapdrb.get(j).
29                          getTahun ()
30                          +"\t\t"+datapdrb.get(j).getNilai ()+
31                          "\t\t"+prediksi[i][j]);
32                      }
33                  }
34              }
35          }
36      }
37      return prediksi;
38  }

```

Kode Program 5.12 Implementasi Proses Peramalan

Penjelasan dari Kode Program 5.4 dapat dilihat dibawah ini.

1. Baris 2 inisialisasi variabel prediksi.
2. Baris 3 sampai 27 merupakan perulangan untuk melakukan peramalan.

3. Baris 18 sampai 24 merupakan perulangan untuk melakukan peramalan pada tahun depan.

5.1.2.8 Implementasi Proses Hitung MAPE

Perhitungan Tingkat error dihitung menggunakan *Mean Abosolute Percentage Error* (MAPE) menggunakan persamaan 2.10. Pada Kode Program 5.13 menunjukan implementasi dari proses hitung mape.

```

1  public void Mape() {
2      mape=new double[SWARM_SIZE];
3      for(int i=0;i<SWARM_SIZE;i++) {
4          double sum=0;
5          double selisih=0;
6          for(int j=1;j<datapdrb.size();j++) {
7              selisih=Math.abs((datapdrb.get(j).getNilai()-
8                  prediksi[i][j])/datapdrb.get(j).getNilai());
9              sum=selisih+sum;
10         }
11         int jum=datapdrb.size()-1;
12         mape[i]=sum/jum;
13     }
14 }
```

Kode Program 5.13 Implementasi Proses Hitung MAPE

Penjelasan dari Kode Program 5.4 dapat dilihat dibawah ini.

1. Baris 2 inialisasi variabel mape.
2. Baris 3 sampai 13 merupakan perulangan untuk melakukan perhitungan error MAPE.

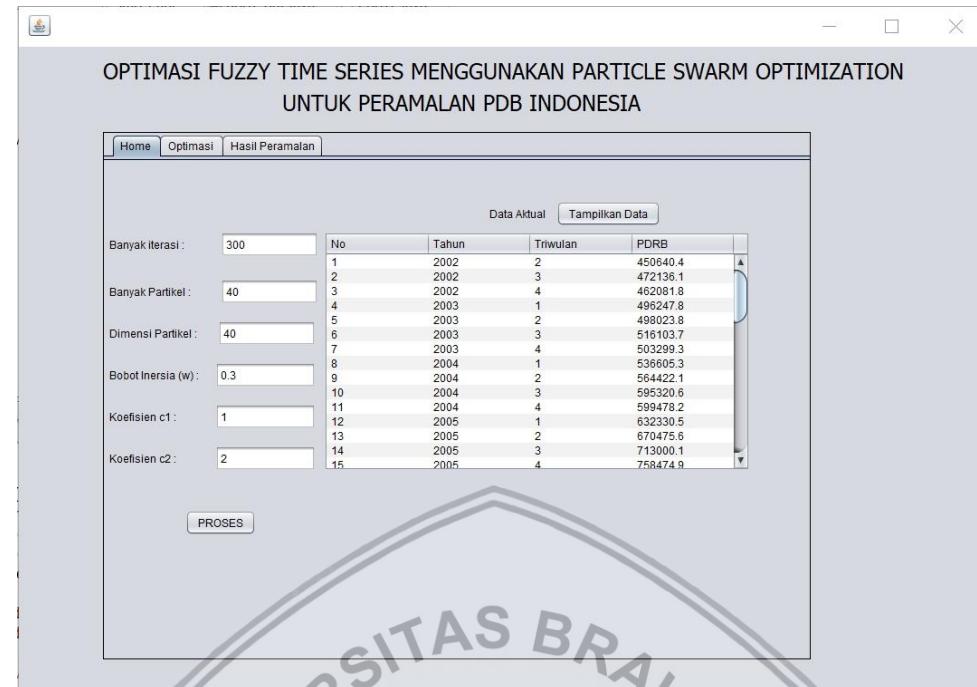
5.2 Implementasi Antarmuka

Sesuai dengan hasil rancangan antarmuka yang telah dilakukan sebelumnya, kemudian dilakukan pengimplementasian dalam bentuk nyata. Ada 3 halaman yang ditampilkan pada antarmuka program optimasi *Fuzzy Time Series* menggunakan *Particle Swarm Optimization* untuk peramalan Produk Domestik Bruto Indonesia, yaitu halaman Home, halaman Optimasi dan halaman Hasil Peramalan.

5.2.1 Implementasi Antarmuka Halaman Home

Pada awal saat menjalankan program, pertama muncul adalah halaman Home ini. Pada halaman home ini pengguna harus memasukan nilai parameter PSO yang dibutuhkan terlebih dahulu yaitu banyak Iterasi, banyak partikel, dimensi partikel, nilai bobot inersia, nilai koefisien c1 dan nilai koefisien c2 agar dapat menjalankan proses peramalan serta pengguna juga dapat menampilkan data aktual yang digunakan pada sistem dengan menekan tombol tampilkan data. Pada halaman ini juga terdapat tombol proses untuk menjalankan proses peramalan. Pada Gambar 5.1 menunjukkan implementasi dari rancangan antarmuka halaman home.

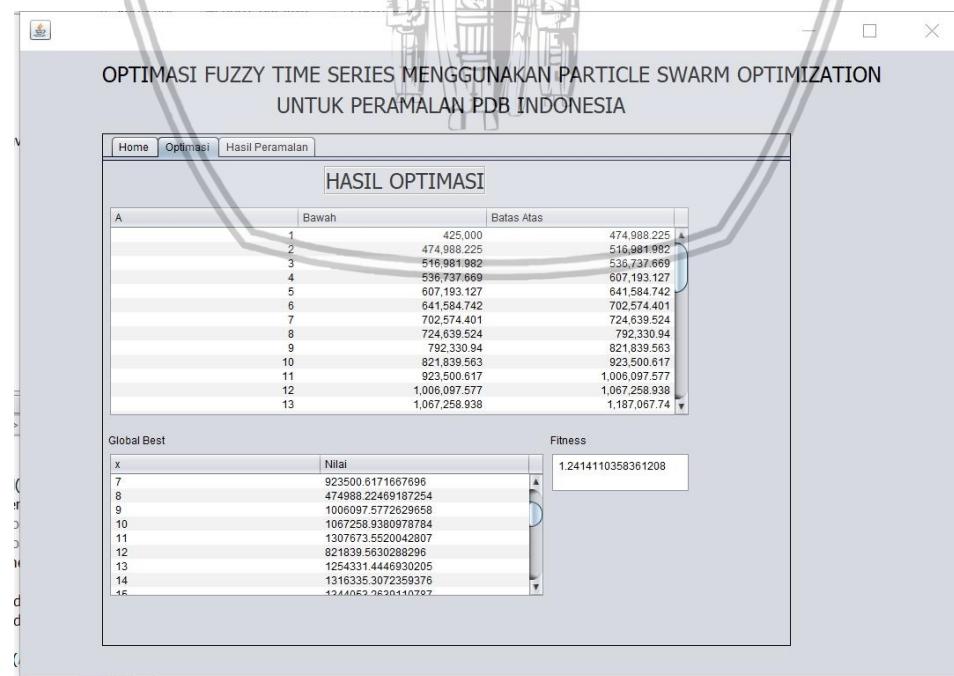




Gambar 5. 1 Implementasi Antarmuka Halaman Home

5.2.2 Implementasi Antarmuka Halaman Optimasi

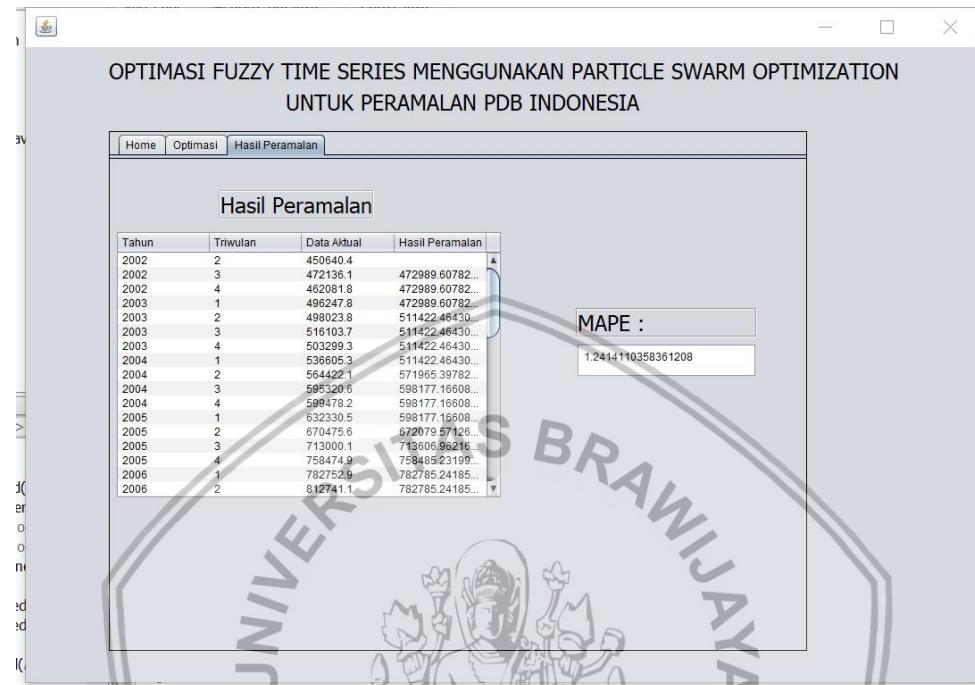
Pada antarmuka halaman ini akan menampilkan hasil dari panjang interval yang telah dioptimasi, serta menampilkan partikel global best (*gbest*) dan nilai *fitness* yang optimal. Pada Gambar 5.2 menunjukkan implementasi dari rancangan antarmuka halaman optimasi.



Gambar 5.2 Implementasi Antarmuka Halaman Optimasi

5.2.3 Implementasi Antarmuka Halaman Hasil Peramalan

Pada halaman ini akan menampilkan hasil akhir berupa hasil nilai peramalan dan nilai *error MAPE* yang dihasilkan oleh program. Pada Gambar 5.3 menunjukkan implementasi dari antarmuka halaman hasil peramalan.



Gambar 5.3 Implementasi Antarmuka Halaman Hasil Peramalan

BAB 6 PENGUJIAN DAN ANALISIS

Dari hasil perancangan pengujian pada bab sebelumnya, kemudian akan dilakukan pengujian yang nantinya hasil pengujian akan di analisis sehingga dapat menghasilkan suatu kesimpulan mengenai penelitian ini. Pengujian ini dilakukan untuk mendapatkan nilai rata-rata *fitness*, sehingga akan dilakukan 5 kali percobaan setiap pengujiannya.

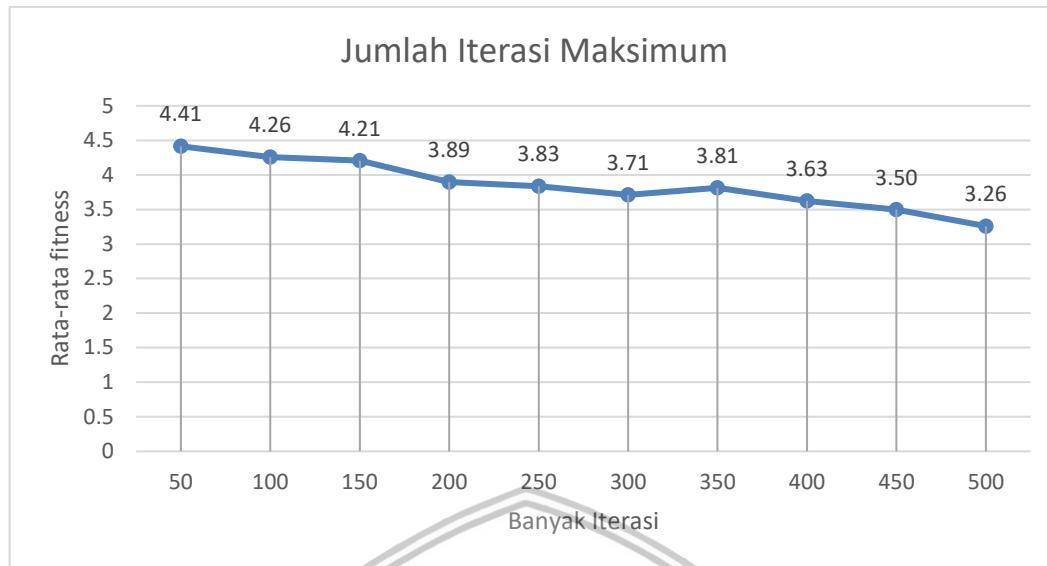
6.1 Hasil dan Pengujian Jumlah Iterasi Maksimum

Berdasarkan perancangan pengujian yang dilakukan pada bab sebelumnya, kemudian dilakukan pengujian jumlah iterasi maksimum ini dengan tujuan agar dapat mengetahui Jumlah Iterasi yang akan menghasilkan solusi terbaik. Jumlah iterasi yang diujikan adalah 50, 100, 150, 200, 250, 300, 350, 400, 450, 500 dan pengujian dilakukan sebanyak 5 kali percobaan. Solusi terbaik diambil dari nilai rata-rata *fitness* yang terendah. Tabel 6.1 menunjukkan hasil pengujian jumlah iterasi maksimum yang telah dilakukan.

Tabel 6.1 Hasil Pengujian Banyak Iterasi

Panjang dimensi partikel =15, Banyak Partikel =10, w =0.3, c1 = 1.5, c2 = 0.5						
Jumlah Iterasi	Nilai <i>fitness</i> percobaan ke-					Rerata <i>fitness</i>
	1	2	3	4	5	
50	4.38238	4.86697	4.07837	4.24272	4.50029	4.41415
100	4.57137	5.52120	3.46943	4.27420	3.44690	4.25662
150	5.04258	3.73664	4.02958	3.28722	4.93069	4.20534
200	3.46223	3.27440	4.65170	4.45866	3.62637	3.89467
250	3.90848	3.76276	3.33906	3.88778	4.27396	3.83441
300	4.00485	3.96664	3.54208	3.77878	3.26826	3.71212
350	4.61261	2.91223	3.68348	4.25262	3.59388	3.81096
400	3.08594	4.36458	2.77425	4.42159	3.48051	3.62537
450	3.18186	3.24739	3.59083	3.77078	3.68882	3.49594
500	3.30794	3.31466	3.08761	3.20372	3.38225	3.25924

Berdasarkan Pada Tabel 6.1, jumlah iterasi maksimum yang menghasilkan solusi terbaik adalah iterasi sebanyak 500 dengan nilai rata-rata *fitness* yang dihasilkan yaitu 3.25924. Sedangkan pengujian banyak iterasi terhadap nilai rata-rata *fitness* direpresentasikan dalam grafik yang ditunjukkan pada Gambar 6.1.



Gambar 6.1 Grafik Hasil Pengujian Jumlah Iterasi Terhadap Nilai Rata-rata fitness

Sesuai dengan Gambar 6.1 dapat dilihat nilai rata-rata *fitness* yang dihasilkan cenderung semakin turun seiring bertambahnya jumlah iterasi, namun terjadi kenaikan di tengah-tengah pada iterasi 350 kemudian mengalami penurunan kembali untuk iterasi selanjutnya dan menghasilkan nilai rata-rata *fitness* yang lebih baik dengan menggunakan 500 iterasi.

6.2 Hasil dan Analisis Pengujian Panjang Dimensi Partikel

Dilakukannya pengujian panjang dimensi partikel ini dengan tujuan agar mendapatkan panjang dimensi partikel yang dapat menghasilkan solusi terbaik. Jumlah panjang dimensi partikel yang diujikan berturut-turut sebanyak 5, 10, 15, 20, 25, 30, 35, 40 serta pengujian dilakukan sebanyak 5 kali percobaan. Solusi terbaik diambil dari nilai rata-rata *fitness* yang terendah. Pada Tabel 6.2 menunjukkan hasil dari pengujian panjang dimensi yang telah dilakukan.

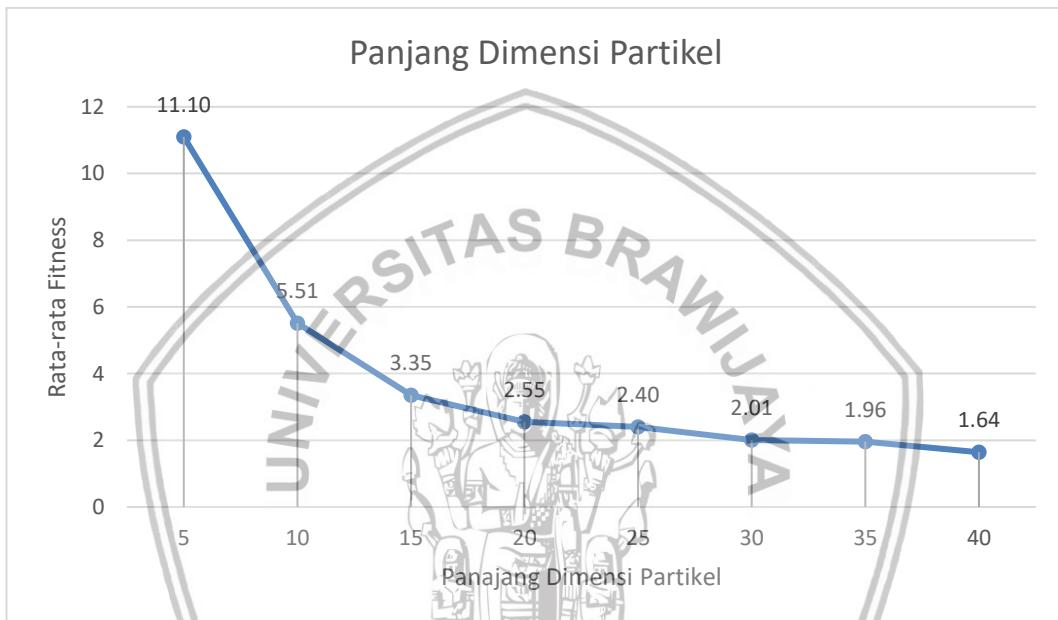
Tabel 6.2 Hasil Pengujian Panjang Dimensi Partikel

Banyak partikel =10, itermaks=500, w=0.3, c1=1.5, c2=0.5						
Panjang dimensi partikel	Nilai <i>fitness</i> percobaan ke-					Rerata <i>fitness</i>
	1	2	3	4	5	
5	10.88660	11.37532	11.24088	10.86715	11.12839	11.09967
10	5.69149	5.23086	5.25954	5.50042	5.85521	5.50750
15	3.35777	3.37940	3.44784	3.21726	3.34699	3.34985
20	2.13372	2.66119	2.75201	2.69388	2.52909	2.55398
25	2.48619	2.63250	2.35196	2.13518	2.39407	2.39998



30	1.82005	2.07135	2.20006	1.91527	2.04229	2.00980
35	1.72060	2.33582	1.94511	1.80317	1.99044	1.95903
40	1.52552	2.17874	1.60344	1.48172	1.42029	1.64194

Berdasarkan Tabel 6.2, panjang dimensi partikel yang menghasilkan solusi terbaik adalah panjang dimensi partikel sebanyak 40 dengan nilai rata-rata *fitness* yang dihasilkan yaitu 2.006582. Pada Gambar 6.2 menunjukkan grafik hasil dari pengujian panjang dimensi partikel terhadap nilai rata-rata *fitness*.



Gambar 6.2 Grafik Hasil Pengujian Panjang Dimensi Partikel Terhadap Nilai Rata-rata *fitness*

Berdasarkan grafik pada Gambar 6.2 dapat ditarik kesimpulan bahwa semakin besar nilai panjang dimensi partikel yang digunakan maka semakin baik pula rata-rata *fitness* yang dihasilkan dan ditunjukkan juga bahwa pengujian panjang dimensi partikel dengan nilai lebih besar dari 30 menghasilkan nilai rata-rata *fitness* yang cukup berdekatan dengan hasil rata-rata *fitness* tidak lebih dari 2.

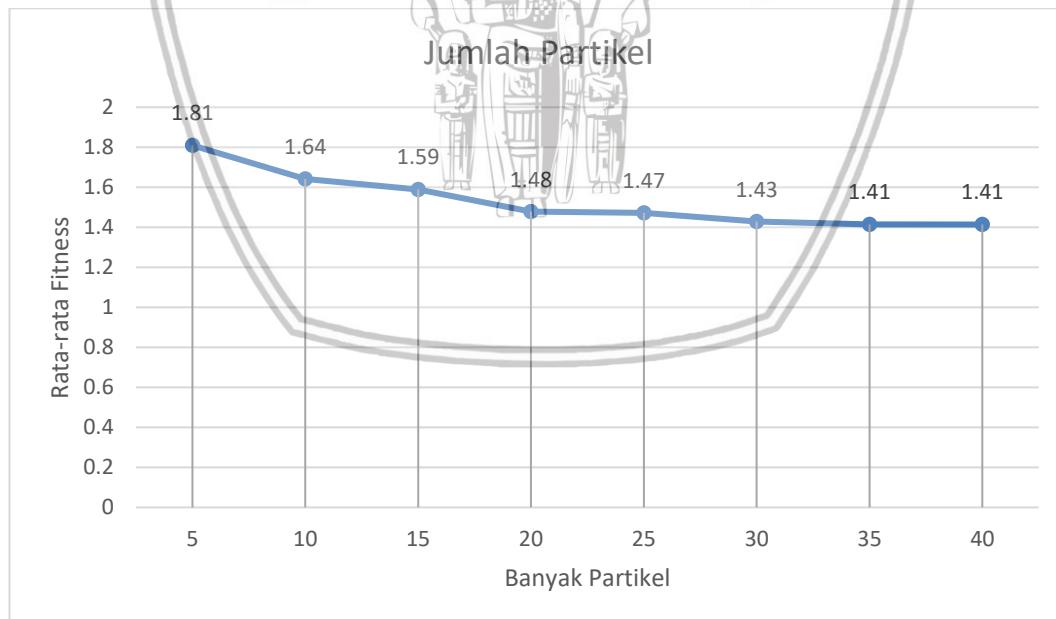
6.3 Hasil dan Analisis Pengujian Jumlah Partikel

Berdasarkan perancangan pengujian yang dilakukan pada bab sebelumnya, kemudian dilakukan pengujian jumlah partikel yang akan dilakukan sebanyak 5 kali percobaan dengan jumlah partikel yang diujikan berturut turut sejumlah 5, 10, 15, 20, 25, 30, 35, 40. Tujuan pengujian ini dilakukan untuk mengetahui jumlah partikel yang dapat menghasilkan solusi terbaik, dimana solusi terbaik diambil dari nilai rata-rata *fitness* yang terendah. Pada Tabel 6.3 menunjukkan hasil dari pengujian banyak partikel yang telah dilakukan.

Tabel 6.3 Hasil Pengujian Jumlah Partikel

Panjang dimensi partikel =40, itermaks=500, w=0.3, c1=1.5, c2=0.5						
Jumlah Partikel	Nilai <i>fitness</i> percobaan ke-					Rerata <i>fitness</i>
	1	2	3	4	5	
5	1.84521	1.78725	2.04641	1.64950	1.71825	1.80932
10	1.52552	2.17874	1.60344	1.48172	1.42029	1.64194
15	1.81865	1.74474	1.42841	1.41474	1.53696	1.58870
20	1.44803	1.50543	1.38962	1.42066	1.62509	1.47776
25	1.45261	1.57641	1.37136	1.41031	1.55028	1.47219
30	1.46268	1.49610	1.44198	1.30541	1.43559	1.42835
35	1.39323	1.48091	1.50037	1.46436	1.23590	1.41495
40	1.41672	1.41191	1.49394	1.23699	1.51040	1.41399

Berdasarkan Pada Tabel 6.3, jumlah partikel yang menghasilkan solusi terbaik adalah partikel sebanyak 40 dengan nilai rata-rata *fitness* yang dihasilkan yaitu 1.41399. Sedangkan pengujian jumlah partikel terhadap nilai rata-rata *fitness* direpresentasikan dalam grafik yang ditunjukkan pada Gambar 6.3.

**Gambar 6.3 Grafik Hasil Pengujian Jumlah Partikel Terhadap Nilai Rata-rata *fitness***

Sesuai dengan apa yang ditunjukkan grafik pada Gambar 6.3 dapat disimpulkan jumlah partikel berbanding terbalik dengan nilai rata-rata *fitness* yang dihasilkan, yaitu semakin banyak partikel yang digunakan maka semakin kecil atau baik pula rata-rata *fitness* yang dihasilkan. Pada Gambar 6.3 ditunjukkan juga

bahwa pengujian jumlah partikel dengan nilai lebih besar dari 25 menghasilkan rata-rata *fitness* dengan perubahan yang tidak terlalu signifikan bahkan dapat dikatakan konstan, dimana nilai rata-rata *fitness* yang dihasilkan memiliki nilai yang berdekatan.

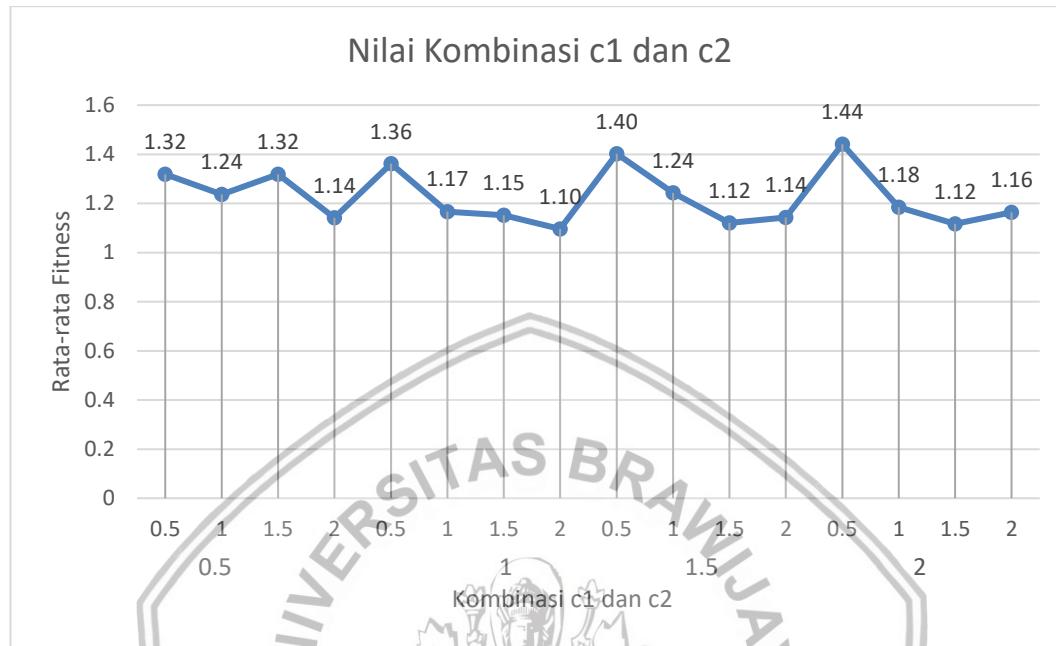
6.4 Hasil dan Analisis Pengujian Kombinasi Nilai variabel c1 dan c2

Pada penelitian ini akan dilakukan pengujian kombinasi nilai variabel c1 dan c2 sebanyak 5 kali percobaan dengan nilai kombinasi adalah 0.5, 1, 1.5 sesuai dengan perancangan pengujian pada bab sebelumnya. Tujuan dilakukannya pengujian agar mendapatkan nilai kombinasi nilai koefisien *kognitif* (c1) dan koefisien *social* (c2) yang akan menghasilkan solusi terbaik, dimana solusi terbaik akan diambil berdasarkan nilai rata-rata *fitness* yang terendah. Pada Tabel 6.4 menunjukkan hasil dari pengujian kombinasi nilai variabel c1 dan c2 yang telah dilakukan.

Tabel 6.4 Hasil Pengujian Kombinasi Nilai Varibel c1 dan c2

Panjang dimensi partikel=40, Banyak Partikel=40, itermaks=500, w=0.3							
Nilai c1	Nilai c2	Nilai <i>fitness</i> percobaan ke-					Rerata <i>fitness</i>
		1	2	3	4	5	
0.5	0.5	1.30958	1.35304	1.20172	1.13109	1.59568	1.31822
	1	1.19071	1.31503	1.23798	1.10258	1.33818	1.23690
	1.5	1.20700	1.31888	1.78609	1.37340	0.90707	1.31849
	2	1.04288	1.09900	1.26927	1.19855	1.10003	1.14195
1	0.5	1.09673	1.63851	1.50390	1.26352	1.30576	1.36168
	1	0.93760	1.27872	1.03889	1.22604	1.35151	1.16655
	1.5	0.97207	1.26175	1.15390	1.37392	0.99693	1.15171
	2	0.98255	1.27046	1.00894	1.11481	1.10408	1.09617
1.5	0.5	1.52820	1.57195	1.38341	1.34348	1.18471	1.40235
	1	1.29320	1.06575	1.88333	0.99684	0.97538	1.24290
	1.5	1.25176	1.06785	0.90707	1.17931	1.19484	1.12016
	2	0.99671	1.03979	1.13675	1.15450	1.38343	1.14224
2	0.5	1.22804	1.52366	1.33838	1.55546	1.55968	1.44104
	1	1.30023	1.10332	1.12310	1.03246	1.36497	1.18481
	1.5	1.33614	1.02649	1.07872	1.11897	1.02619	1.11730
	2	1.27481	1.17521	0.93813	1.24224	1.19076	1.16423

Pada Tabel 6.4 memperlihatkan bahwa kombinasi nilai c1 sama dengan 1 dan nilai c2 sama dengan 2 menghasilkan nilai rata-rata *fitness* terbaik dengan nilai rata-rata *fitness* yang dihasilkan sebesar 1.09617. Pada Gambar 6.4 menunjukkan grafik hasil pengujian kombinasi nilai variabel c1 dan c2 terhadap rata-rata *fitness*.



Gambar 6.4 Grafik Hasil Pengujian Kombinasi Nilai Variabel c1 dan c2 Terhadap Rata-rata *fitness*

Berdasarkan grafik pada Gambar 6.4 dapat dilihat bahwa nilai rata-rata *fitness* yang dihasilkan dari pengujian kombinasi nilai variabel c1 dan c2 cenderung tidak merata. Dimana terjadi peningkatan dan penurunan nilai rata-rata *fitness* secara tidak menentu. Berdasarkan grafik tersebut apabila nilai kombinasi c1 lebih besar dari c2 menghasilkan rata-rata *fitness* yang lebih tinggi dari pada kombinasi jika nilai c1 lebih kecil dari c2.

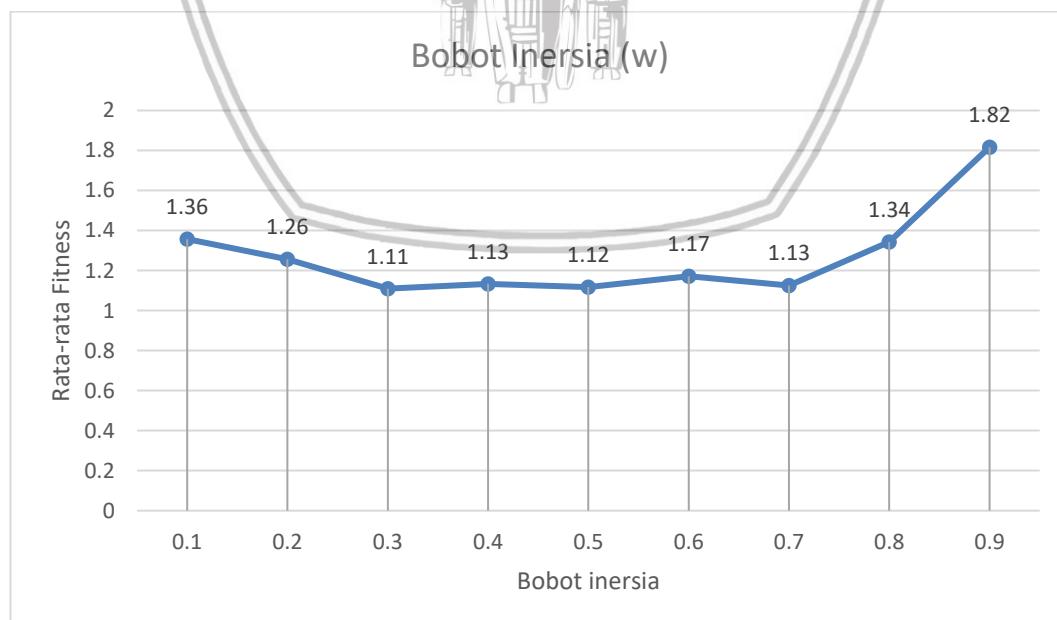
6.5 Hasil dan Analisis Pengujian Nilai Variabel Bobot Inersia

Pada penelitian ini percobaan pada pengujian nilai variabel bobot inersia (*w*) akan dilakukan sebanyak 5 kali percobaan sesuai dengan perancangan pengujian pada bab sebelumnya nilai bobot inersia yang diujikan adalah 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 dan 0.9. Tujuan dilakukannya pengujian ini adalah untuk mengetahui nilai variabel bobot *inersia* yang dapat menghasilkan solusi terbaik, dimana solusi terbaik diambil dari nilai rata-rata *fitness* yang terendah. Pada Tabel 6.5 menunjukkan hasil pengujian nilai variabel bobot inersia yang telah dilakukan.

Tabel 6.5 Hasil Pengujian Nilai Variabel Bobot Inersia

Panjang dimensi partikel=40, Banyak partikel=40, itermaks=500, c1=1, c2=2.						
Nilai bobot (w)	Nilai <i>fitness</i> percobaan ke-					Rerata <i>fitness</i>
	1	2	3	4	5	
0.1	1.47554	1.20562	1.33928	1.37511	1.38696	1.35650
0.2	1.12678	1.08494	1.14380	1.12207	1.80172	1.25586
0.3	1.02523	1.09010	1.19089	1.13402	1.10823	1.10969
0.4	1.08233	1.07453	1.12249	1.22072	1.16274	1.13256
0.5	1.10457	1.00650	1.09232	1.09197	1.28895	1.11686
0.6	1.20547	1.22704	1.03528	1.24414	1.14245	1.17088
0.7	0.97597	1.20636	1.00928	1.27933	1.15805	1.12580
0.8	1.46473	1.31657	1.26412	1.25464	1.41668	1.34335
0.9	1.55189	1.85847	2.01839	1.91571	1.73961	1.81681

Berdasarkan Tabel 6.5 nilai bobot inersia (w) sama dengan 0,3 menghasilkan nilai rata-rata *fitness* terbaik dengan nilai rata-rata *fitness* yang dihasilkan sebesar 1.071799. Pada Gambar 6.6 diperlihatkan hasil pengujian nilai bobot inersia (w) terhadap rata-rata *fitness*.

**Gambar 6.5 Grafik Hasil Pengujian Nilai Bobot inersia (w)**

Berdasarkan Gambar 6.5 dapat disimpulkan bahwa nilai rata-rata *fitness* yang dihasilkan dari pengujian nilai bobot inersia pada 0.1 sampai 0.3 cenderung

mengalami penurunan, kemudian mengalami kenaikan dan penurunan bergantian pada pengujian selanjutnya. Pada pengujian nilai bobot inersia lebih dari 0.7 terlihat menghasilkan kenaikan yang signifikan.

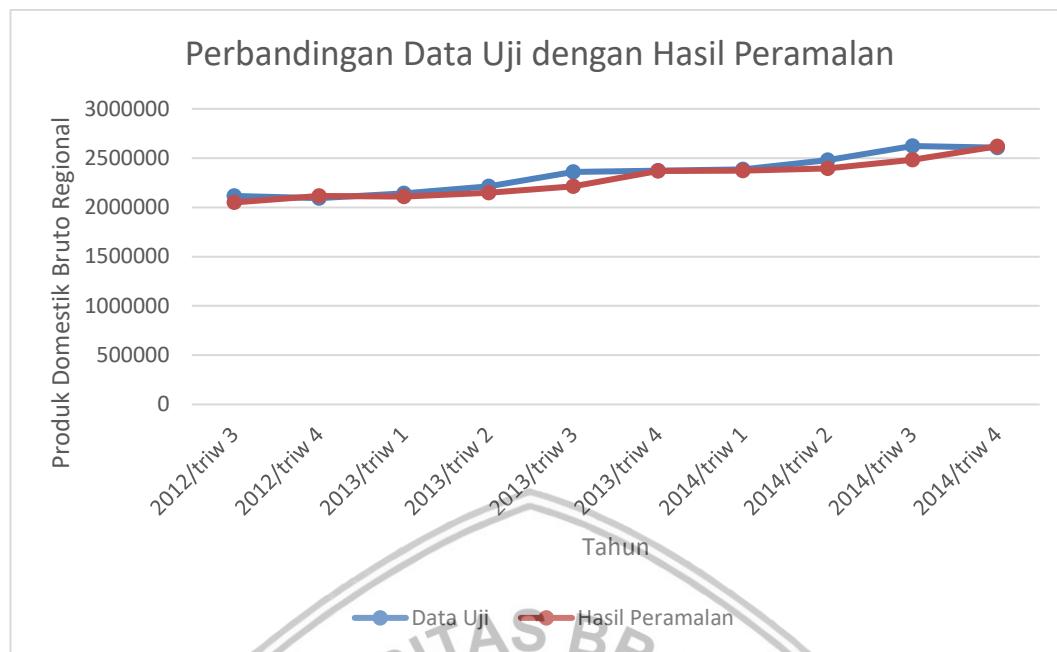
6.6 Pengujian Tingkat Kesalahan Algoritme

Pada pengujian ini akan menggunakan banyak jumlah partikel, dimensi partikel, nilai konstanta variabel c_1 dan variabel c_2 , nilai bobot inersia dan iterasi maksimal diperoleh dari hasil pengujian yang dilakukan sebelumnya. Data latih yang digunakan ada 50 data dan data uji yang digunakan adalah data PDB indonesia tahun 2012 triwulan 3 sampai 2014 triwulan 4, total ada 10 data uji. Tujuan dilakukannya pengujian ini untuk mengetahui tingkat kesalahan paramalan dengan metode yang telah diusulkan.

Tingkat kesalahan yang dihasilkan menggunakan *Mean Absolute Percentage Error* (MAPE) adalah 2.47% berdasarkan pengujian yang telah dilakukan terhadap 10 data uji. Pada Hasil pengujian tingkat kesalahan metode ditunjukkan pada Tabel 6.6, sedangkan pada Gambar 6.6 diperlihatkan grafik perbandingan hasil data uji dengan hasil peramalan.

Tabel 6.6 Hasil Pengujian Tingkat Kesalahan Sistem

Panjang dimensi partikel=40, Banyak partikel=40, itermaks=500, $c_1, c_2=2$, $w=0.3$					
No	Tahun	Triwulan	Data uji	Hasil Peramalan	Absolute Percentage Error
1	3.147023879	3	2116302.1	2049701.568	3.25724009
2	1.052972902	4	2094027.1	2116076.638	1.052972902
3	1.569550847	1	2143260.1	2109620.543	2.126811944
4	2.913640186	2	2212148.1	2147694.064	4.273751382
5	6.225332885	3	2359100.3	2212238.453	7.161423199
6	0.157037171	4	2372768.0	2369041.872	0.578485876
7	0.566459197	1	2387055.8	2373534.103	0.985385307
8	3.336721533	2	2478081.7	2395395.014	3.593045058
9	5.311487159	3	2622612.6	2483312.869	5.512626224
10	0.487958835	4	2607178.8	2619900.759	0.593529126
MAPE					2.47681846



Gambar 6.6 Grafik Perbandingan Data uji dengan Hasil Peramalan

Pada Gambar 6.6 dapat disimpulkan bahwa peramalan menggunakan fuzzy time series dengan pengoptimasian interval menggunakan *Particle Swarm Optimization* berhasil dilakukan terhadap data PDB, dimana dapat dibuktikan dengan hasil peramalan tidak terlalu berbeda dari data aktualnya atau saling berdekatan.

Hasil peramalan menggunakan metode *Fuzzy Time Series* dengan penentuan interval berbasis rata-rata dan metode *Regresi Linier* dapat dilihat pada Tabel 6.7. Pada Tabel 6.8 menunjukkan perbandingan tingkat kesaaman menggunakan MAPE antara metode yang diusulkan dengan metode *Fuzzy Time Series* dengan penentuan interval berbasis rata-rata dan metode *Regresi Linier*.

Berdasarkan Tabel 6.8 hasil tingkat kesalahan yang didapat menggunakan metode *Fuzzy Time Series* yang dioptimasi menggunakan *Particle Swarm Optimization* sedikit lebih baik, apabila dibandingkan dengan metode *Fuzzy Time Series* dengan penentuan interval berbasis rata-rata dengan tingkat kesalahan MAPE yang dihasilkan sebesar 2,66%. Namun apabila dibandingkan dengan metode *Regresi Linier* menghasilkan tingkat kesalahan yang tidak lebih baik, dimana tingkat kesalahan MAPE 1,52% dengan menggunakan metode *Regresi Linier*. Hal tersebut dikarenakan data PDB yang digunakan cenderung naik setiap tahunnya atau data bersifat linier dan pada metode yang diusulkan apabila terjadi penurunan atau peningkatan data aktual yang besar, maka hasil prediksi untuk periode berikutnya akan menghasilkan *error* yang besar karena nilai prediksi ditentukan berdasarkan himpunan *fuzzy* pada periode sebelumnya (A_{i-1}). Dari penyataan di atas dapat disimpulkan bahwa optimasi *fuzzy time series* menggunakan PSO dapat menghasilkan tingkat kesaaman lebih baik dari *fuzzy time series* tanpa optimasi tetapi tidak lebih baik dari *Regresi Linier* karena data yang digunakan bersifat linier, dimana lebih cocok menggunakan metode *Regresi Linier*.

Tabel 6.7 Hasil Peramalan *Fuzzy Time Series* dengan penentuan interval berbasis rata-rata dan metode *Regresi Linier*

No	Tahun	Triwulan	Data uji	Hasil Peramalan (FTS interval berbasis rata2)	Hasil Peramalan (regresi linier)
1	3.147023879	3	2116302.1	2039000	2057512.889
2	1.052972902	4	2094027.1	2122000	2120121.905
3	1.569550847	1	2143260.1	2092000	2182730.921
4	2.913640186	2	2212148.1	2152000	2245339.936
5	6.225332885	3	2359100.3	2214500	2307948.952
6	0.157037171	4	2372768.0	2361000	2370557.968
7	0.566459197	1	2387055.8	2381000	2433166.984
8	3.336721533	2	2478081.7	2381000	2495775.999
9	5.311487159	3	2622612.6	2478500	2558385.015
10	0.487958835	4	2607178.8	2612000	2620994.031

Tabel 6.8 Perbandingan Nilai MAPE dengan 10 data uji

Metode	MAPE
Metode usulan	2.476%
Fuzzy Time Series interval berbasis rata2	2.657%
Regresi Linier	1.525%



BAB 7 KESIMPULAN

7.1 Kesimpulan

Berdasarkan dari analisis, perancangan dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Nilai parameter pada algoritme particle swarm optimization (PSO) pada setiap pengujian sangat berpengaruh pada hasil *fitness* yang dihasilkan. Berdasarkan hasil pengujian parameter yang telah dilakukan pada penelitian ini, parameter *particle swarm optimization* (PSO) yang menghasilkan interval *fuzzy time series* yang optimal untuk peramalan *fuzzy time series* yaitu panjang dimensi sebesar 40, banyak partikel sebesar 40, 500 untuk iterasi maksimum, nilai variabel c_1 sebesar 1 dan variabel c_2 bernilai 2 dan untuk bobot *inertia* sebesar 0,3.
2. Tingkat kesalahan algoritme yang dihasilkan oleh program dalam penelitian optimasi *fuzzy time series* menggunakan particle swarm optimization untuk permasalahan peramalan Produk Domestik Bruto Indonesia, menghasilkan tingkat keasalahan menggunakan MAPE sebesar 2,9135%. Tingkat kesalahan tersebut didapat dari kesalahan hasil peramalan dengan data aktual pada 10 data uji.

7.2 Saran

Berdasarkan penelitian optimasi *fuzzy time series* menggunakan *particle swarm optimization* (PSO) untuk kasus peramalan Produk Domestik Bruto (PDB) di Indonesia yang telah dilakukan, saran untuk penelitian selanjutnya dapat menggunakan data nonlinier dan menerapkan metode penentuan nilai prediksi lainnya selain berdasarkan pada himpunan FLR pada periode sebelumnya agar dapat menghasilkan tingkat kesalahan yang lebih baik.

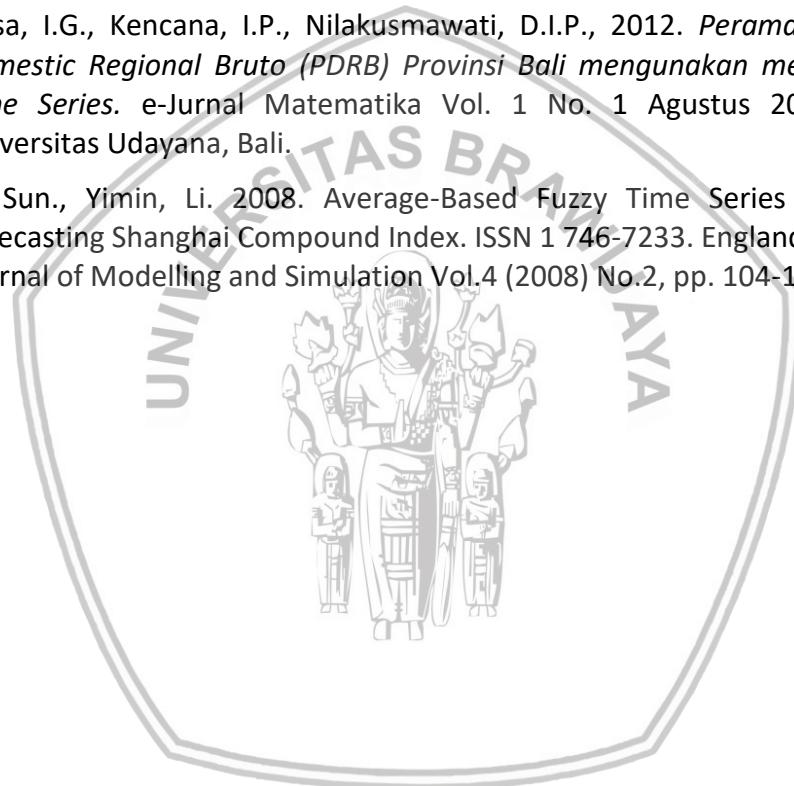


DAFTAR PUSTAKA

- Berutu, S.S., 2013. Peramalan *Penjualan dengan Metode Fuzzy Time Series Ruey Chyn Tsaur.* S2. Universitas Diponegoro. Tersedia di <http://eprints.undip.ac.id/41216/1/Sunn_eng_Sandino_B.pdf> [Diakses 11 Februari 2018].
- Badan Pusat Statistika, 2015. PDB Triwulan Atas Dasar Harga Berlaku Menurut Lapangan Usaha (Milliar Rupiah) 2000-2014. [pdf] Badan Pusat Statistika Indonesia. Tersedia di: <<https://bps.go.id/statictable/2009/08/24/-seri-2000-pdb-triwulanan-atas-dasar-harga-berlaku-menurut-lapangan-usaha-milliar-rupiah-2000-2014.html>> [Diakses 10 Februari 2018].
- Badan Pusat Statistika, 2018. PDB Indonesia Tahun 2014-2018. [pdf] Badan Pusat Statistika Indonesia. Tersedia di: <<https://bps.go.id/publication/2018/10/05/02d80724b61eb684620a2e88/pdb-indonesia-triwulanan-2014-2018.html>> [Diakses 12 Februari 2018].
- Cholissodin, I., Riyandani, E., 2016. Swarm Intelligence. Fakultas Ilmu Komputer. Universitas Brawijaya.
- Cai. Q., Zhang. D., Wu. B., dan Leung. A., 2013. *A Novel Stock Forecasting Model Based on Fuzzy Time Series and Genetic Algorithm. International Conference on Computational Science* doi: 10.1016/j.procs.2013.05.281.
- Dalimunthe, Y. D., 2017. *Analisis Peramalan Data Produk Domestik Regional Bruto Sebagai Tolak Ukur Kinerja Perekonomian Provinsi Bangka Belitung. Journal of Business and Economics*, Vol 1 Nomor 1, 2017. Universitas Bangka Belitung, Indonesia.
- Efendi, R., Ismail, Z. & Deris, M., 2015. *A New Linguistic Approach of Fuzzy Time Series for Daily Forecasting of Malaysian Electricity Load Demand. Applied Soft Computing*, pp.422-430.
- Elfajar, A.B., Setiawan, B.D., Candra, D., 2017. *Peramalan Jumlah Kunjungan Wisatawan Kota Batu Menggunakan Metode Invariant Fuzzy Time Series*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer. Vol. 1, No. 2, Februari 2017, hlm. 85-94.
- Jatipaningrum, M.T., 2016. *Peramalan Data Produk Domestik Bruto dengan Fuzzy Time Series Markov Chain*. Jurnal Teknologi, Volume 9 Nomor 1, Juni 2016, 31 – 38. Institut Sains & Teknologi AKPRIND, Yogyakarta.
- Khosla M., Kumar R.S., Uddin M. 2012. *Identification of Type-2 Fuzzy Models Time-Series Forecasting Using Particle Swarm Optimization*. IEEE. 2012 Internasional Coference on Communication Systems and Network Technologies.
- Misra Hartati, I. V. B. S., 2012. *Pengembangan Algoritme Particle Swarm Optimization untuk Optimalisasi Dispersi Batch Pada Proses Produksi*. Simposium Nasional RAPI XI FT UMS, pp. 116-123.



- Prasojo, C.A., Setiawan, B.D., Marji, 2018. *Optimasi Fuzzy Time Series Menggunakan Algoritme Particle Swarm Optimization Untuk Peramalan Jumlah Penduduk Di Kabupaten Probolinggo*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer. Vol. 2, No. 8, Agustus 2018, hlm. 2791-2799.
- Qiu W., Zhang C., Zang P. 2015. Generalized Fuzzy Time Series Forecasting Model Enhanced with Particle Swarm Optimization. International Journal of u- and e- Service, Science and Technology. Vol.8, No.5.
- Vagarda, V & Mahmudy, WF., 2016. *Pemodelan Fuzzy Time Series Dengan Algoritme Particle Swarm Optimization Untuk Peramalan Pemakaian Air PDAM Kota Malang*. S1. Universitas Brawijaya.
- Wanayasa, I.G., Kencana, I.P., Nilakusmawati, D.I.P., 2012. *Peramalan Produk Domestic Regional Bruto (PDRB) Provinsi Bali menggunakan metode Fuzzy Time Series*. e-Jurnal Matematika Vol. 1 No. 1 Agustus 2012, 12-19. Universitas Udayana, Bali.
- Xihaou. Sun., Yimin, Li. 2008. Average-Based Fuzzy Time Series Model for Forecasting Shanghai Compound Index. ISSN 1 746-7233. England, UK. Word Journal of Modelling and Simulation Vol.4 (2008) No.2, pp. 104-111.



LAMPIRAN A DATA PDB

NO	Tahun	Triwulan	Nilai
1	2000	Triwulan 1	325958.6
2	2000	Triwulan 2	336967.1
3	2000	Triwulan 3	360701.6
4	2000	Triwulan 4	366142.6
5	2001	Triwulan 1	386648.8
6	2001	Triwulan 2	416069.9
7	2001	Triwulan 3	426828.3
8	2001	Triwulan 4	416775.0
9	2002	Triwulan 1	436975.1
10	2002	Triwulan 2	450640.4
11	2002	Triwulan 3	472136.1
12	2002	Triwulan 4	462081.8
13	2003	Triwulan 1	496247.8
14	2003	Triwulan 2	498023.8
15	2003	Triwulan 3	516103.7
16	2003	Triwulan 4	503299.3
17	2004	Triwulan 1	536605.3
18	2004	Triwulan 2	564422.1
19	2004	Triwulan 3	595320.6
20	2004	Triwulan 4	599478.2
21	2005	Triwulan 1	632330.5
22	2005	Triwulan 2	670475.6
23	2005	Triwulan 3	713000.1
24	2005	Triwulan 4	758474.9
25	2006	Triwulan 1	782752.9
26	2006	Triwulan 2	812741.1
27	2006	Triwulan 3	870319.8
28	2006	Triwulan 4	873403.0
29	2007	Triwulan 1	920203.1
30	2007	Triwulan 2	963862.5
31	2007	Triwulan 3	1031408.7
32	2007	Triwulan 4	1035418.9
33	2008	Triwulan 1	1110032.3
34	2008	Triwulan 2	1220605.9
35	2008	Triwulan 3	1327509.6
36	2008	Triwulan 4	1290540.6
37	2009	Triwulan 1	1315272.0
38	2009	Triwulan 2	1381407.4
39	2009	Triwulan 3	1458209.4
40	2009	Triwulan 4	1451314.6
41	2010	Triwulan 1	1505857.0

