**IMPLEMENTASI JARINGAN SARAF TIRUAN UNTUK PREDIKSI INFLASI DI KOTA YOGYAKARTA MENGGUNAKAN METODE BACKPROPAGATION**

**PROYEK TUGAS AKHIR**

Diajukan untuk memenuhi salah satu syarat

Mencapai derajat Sarjana S-1 Program Studi Teknik Informatika



Disusun oleh:

Herman Yogi Nasrin

5130411111

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS BISNIS DAN TEKNOLOGI INFORMASI**

**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**2017**

**IMPLEMENTASI JARINGAN SARAF TIRUAN UNTUK PREDIKSI INFLASI DI KOTA YOGYAKARTA MENGGUNAKAN METODE BACKPROPAGATION**

**PROYEK TUGAS AKHIR**

****  
Disusun oleh

Herman Yogi Nasrin

5130411111

Telah dipertanggungjawabkan di dalam Sidang Proyek Tugas Akhir  
pada tanggal, tgl-bln-thn (Pelaksanaan Sidang)

Tim Penguji:

Nama, Gelar (tanda tangan ketua)  
Ketua

Nama, Gelar (tanda tangan anggota) Anggota

Nama, Gelar (tanda tangan anggota) Anggota

Tugas akhir ini telah diterima sebagai salah satu syarat untuk mencapai derajat Sarjana S-1 Program Studi Teknik Informatika

Yogyakarta ,…………….

Ketua Program Studi Teknik Informatika

Fakultas Bisnis dan Teknologi Informasi, Universitas Teknologi Yogyakarta

Yuli Asriningtias, S.Kom., M.Kom.

# LEMBAR PERNYATAAN

Saya yang bertanda tangan dibawah ini:

**Nama : Herman Yogi Nasrin**

**NIM : 5130411111**

**Program Studi : Teknik Informatika**

Menyatakan bahwa Proyek Tugas Akhir yang berjudul :

Implementasi Jaringan Saraf Tiruan Untuk Prediksi Inflasi di Kota Yogyakarta Menggunakan Metode Backpropagation,

merupakan karya ilmiah asli saya dan belum pernah dipublikasikan oleh orang lain, kecuali yang tertulis sebagai acuan dalam naskah ini dan disebutkan dalam daftar pustaka. Apabila di kemudian hari, karya saya disinyalir bukan merupakan karya asli saya, maka saya bersedia menerima konsekuensi apa yang diberikan Program Studi Teknik Informatika Fakultas Bisnis dan Teknologi Informasi Universitas Teknologi Yogyakarta kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta

Pada tanggal : 31 Juli 2017  
Yang menyatakan

Herman Yogi Nasrin

# ABSTRAK

Inflasi merupakan suatu keadaan dimana harga barang secara umum mengalami kenaikan secara terus menerus atau terjadi penurunan nilai uang dalam negeri. Inflasi umumnya disebabkan oleh kenaikan permintaan dan kenaikan biaya produksi. Sehingga inflasi di setiap daerah atau khususnya di Kota Yogyakarta inflasi merupakan kasus yang harus bisa diprediksi agar memudahkan pemerintah dalam menanggulangi tingkat inflasi pada bulan selanjutnya. Inflasi bisa diprediksi dengan memanfaatkan data-data inflasi pada bulan-bulan sebelumnya yaitu dengan memanfaatkan data time series dalam melakukan prediksi tingkat inflasi. Adapun metode yang digunakan dalam memprediksi inflasi di Kota Yogyakarta menggunakan jaringan saraf tiruan dengan metode backpropagation. Tujuan dari penelitian ini merupakan untuk membuat sistem yang dapat meramalkan atau memprediksi tingkat inflasi yang terjadi di Kota Yogyakarta. Sehingga dapat membantu untuk memperkirakan tingkat inflasi di Kota Yogyakarta. Persen rata-rata kesalahan yang didapatkan menggunakan data yang sudah dilatih sebanyak 100 data dalam bentuk pola adalah 0.26% dengan tingkat akurasi 99.74% dan menggunakan data yang belum dilatih atau data uji memiliki persen rata-rata kesalahan adalah 2.93% dengan tingkat akurasi 97.07%.

Kata Kunci : Jaringan Saraf Tiruan, Prediksi, Inflasi, *Backpropagation.*.

# ABSTRACT

Inflation is a condition where the price of goods in general has increased continuously or there is a decline in the value of money in the country. Inflation is generally caused by increased demand and increased production costs. So inflation in every region or especially in Yogyakarta City inflation is a case that should be predictable in order to facilitate the government in tackling the inflation rate in the next month. Inflation can be predicted by utilizing inflation data in previous months by utilizing time series data in predicting inflation rate. The method used in predicting inflation in Yogyakarta City using artificial neural network with backpropagation method. The purpose of this study is to create a system that can predict or predict the inflation rate that occurred in the city of Yogyakarta. So it can help to estimate the inflation rate in Yogyakarta City. The mean percentage of errors obtained using the trained data of 100 data in the form of patterns is 0.26% with 99.74% accuracy and using untrained data or test data having an average percentage of error is 2.93% with an accuracy of 97.07%.

Keywords: Neural Network, Prediktion, Inflation, Backpropagation.

# KATA PENGANTAR

Puji syukur dipanjatkan atas kehadirat Allah SWT, karena dengan limpahan karunia-Nya penulis dapat menyelesaikan Proyek Tugas Akhir dengan judul Implementasi Jaringan Saraf Tiruan Untuk Prediksi Inflasi Di Kota Yogyakarta Menggunakan Metode *Backpropagation*.

Penyusunan Proyek Tugas Akhir ini diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana pada Program Studi Teknik Informatika Fakultas Bisnis dan Teknologi Informasi Universitas Teknologi Yogyakarta.

Proyek Tugas Akhir ini dapat diselesaikan tidak lepas dari segala bantuan, bimbingan, dorongan dan doa dari berbagai pihak, yang pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada:

1. Bapak Dr. Bambang Moertono, MM. Selaku Rektor Universitas Teknologi Yogyakarta.
2. Bapak Dr. Bambang Moertono, MM. selaku Dekan Fakultas Bisnis dan Teknologi Informasi Universitas Teknologi Yogyakarta.
3. Ibu Yuli Asriningtias, S.Kom., M.Kom. selaku Ketua Program Studi Teknik Informatika Fakultas Bisnis dan Teknologi Informasi Universitas Teknologi Yogyakarta.
4. Bapak Dr. Arief Hermawan, M.T. selaku pembimbing yang selalu berkesempatan dan berkenan memberikan pengarahan serta nasihat untuk kelancaran pengerjaan proyek kerja peraktik ini sampai terselesaikan.
5. Rekan-rekan Teknik Informatika Universitas Teknologi Yogyayakarta kelas TI.B angkatan 2013, yang telah memberi dukungan dan semangat juang dalam penyusunan laporan ini.
6. Keluarga yang selalu mendoakan dan memotivasi pengerjaan laporan ini.
7. Puji Nuniastuti yang selalu memberi dukungan, doa dan semangat dalam menyelesaikan laporan ini.
8. Semua pihak yang tidak dapat disebutkan satu persatu, baik langsung maupun tidak langsung membantu dalam penulisan Kerja Praktik ini.

Akhir kata, penulis menyadari bahwa sepenuhnya akan terbatasnya pengetahuan penyusun, sehingga tidak menutup kemungkinan jika ada kesalahan serta kekurangan dalam penyusunan Kerja Praktik, untuk itu saran dari pembaca sangat diharapkan sebagai bahan pelajaran berharga dimasa yang akan datang.

Yogyakarta, Agustus 2016

Herman Yogi Nasrin

# DAFTAR ISI

**HALAMAN JUDUL i**

**HALAMAN PENGESAHAN ii**

**LEMBAR PERNYATAAN iii**

**ABSTRAK iv**

**ABSTRACT v**

**KATA PENGANTAR vi**

**DAFTAR ISI viii**

**DAFTAR GAMBAR xi**

**DAFTAR TABEL xiii**

**BAB I PENDAHULUAN 1**

1.1 Latar Belakang Masalah 1

1.2 Rumusan Masalah 2

1.3 Batasan Masalah 2

1.4 Tujuan Peneliatan 3

1.5 Manfaat Penelitian 3

1.6 Sistematika penulisan 3

**BAB II KAJIAN PUSTAKA DAN LANDASAN TEORI 5**

2.1 Kajian Hasil Penelitian 5

2.2 Kajian Teori 6

2.2.1 Inflasi 6

2.2.2 *Data Time Series* 6

2.3 Jaringan Saraf Tiruan 7

2.4 Arsitektur Jaringan Saraf Tiruan 7

2.4.1 Jaringan Lapisan Tunggal (*Single Layer Network*) 7

2.4.2 Jaringan dengan Banyak Lapisan (*Multilayer Network*) 8

2.4.3 Jaringan dengan Lapisan Kompetitif (*Competitive Layer Network*) 8

2.5 *Backpropagation*  9

2.5.1 Perhitungan Maju 9

2.5.2 Perhitungan Mundur 11

2.6 (UML) 12

2.6.1 *Use Case Diagram*  13

2.6.2 *Class Diagram*  14

2.6.3 *Activity Diagram*  14

2.6.4 *Squence Diagram* 15

2.7 Diagram Alir (*Flowchar*) 16

2.9 Node Js 17

2.8 JSON 17

2.9 NPM 18

2.10 Angular Js 18

2.11 MongoDB 18

**BAB III METODE PENELITIAN 19**

3.1 Objek Penelitian 19

3.2 Metode Penelitian 19

3.2.1 Analisis 19

3.2.2 Pembuatan Sistem 19

3.3 Perangkat Pendukung Penelitian 22

3.3.1 Perangkat Keras (*Hardware*) 22

3.3.2 Perangkat Lunak (*Software*) 22

**BAB IV ANALISA DAN PERANCANGAN SISTEM 23**

4.1 Analisa Sistem 23

4.1.1 Data masukan 23

4.1.2 Proses Perhitungan 24

4.1.3 Hasil Prediksi 24

4.2 Perancangan Sistem 24

4.2.1 UML 25

4.2.2 Arsitektur Jaringan Saraf Tiruan 27

4.2.3 *Flowchart* Program 28

4.3 Perancangan Antar Muka Sistem 33

4.3.1 Perancangan Tampilan *Login* 33

4.3.2 Perancangan Tampilan Buat Akun Baru 33

4.3.3 Perancangan Tampilan Home Admin 34

4.3.4 Perancangan Tampilan Data Latih 35

4.3.5 Perancangan Tampilan Data Uji 35

4.3.6 Perancangan Tampilan Proses Pengujian 36

4.3.7 Perancangan Tampilan *User* 37

4.3.8 Perancangan Tampilan Prediksi 37

**BAB V IMPLEMTASI SISTEM 38**

5.1 Implementasi 38

5.2 Perangkat Keras (*Hardware*) 38

5.3 Perangkat Lunak (*Software*) 39

5.4 Implementasi Web 39

5.4.1 Halaman Awal 39

5.4.2 Halaman Login Admin 39

5.4.3 Halaman Data Latih dan Pelatihan 39

5.4.4 Halaman Data Uji 42

5.4.5 Halaman Pengujian JST 43

5.4.6 Halaman Prediksi 45

5.5 Hasil Pengujian *Black Box* 46

**BAB VI PENUTUP 48**

6.1 Kesimpulan 48

6.2 Saran 48

**DAFTAR PUSTAKA 61**

**LAMPIRAN 62**

# DAFTAR GAMBAR

Gambar 2.1. Jaringan Lapisan Tunggal 8

Gambar 2.2. *Multilayer Network* 8

Gambar 2.3. *Competitive Layer Network* 9

Gambar 4.1. Grafik Data Inflasi Januari 1979 Sampai Oktober 1982 24

Gambar 4.2. *Use Case Diagram* 25

Gambar 4.3. *Activity Diagram* 26

Gambar 4.4. Arsitektur Jaringan Saraf Tiruan 27

Gambar 4.5. *Flowchart Login* 28

Gambar 4.6. *Flowchart* Ganti *Password* 29

Gambar 4.7. *Flowchart* Pelatihan 30

Gambar 4.8. *Flowchart* Pengujian 31

Gambar 4.9. *Flowchart* Prediksi 32

Gambar 4.11. Tampilan Buat Akun Baru 34

Gambar 4.12. Tampilan *Home* Admin 34

Gambar 4.13. Tampilan Data Latih 35

Gambar 4.14. Tampilan Data Uji 36

Gambar 4.15. Tampilan Proses Pengujian 36

Gambar 4.16. Tampilan *User* 37

Gambar 4.17. Tampilan Prediksi 37

Gambar 5.1. Tampilan *Homepage* 39

Gambar 5.2. Tampilan Halam *Login* 40

Gambar 5.3. Tampilan Data Latih dan Pelatihan 41

Gambar 5.4. Halaman Data Uji 42

Gambar 5.5. Halaman Pengujian JST 44

Gambar 5.6. Halaman Prediksi Inflasi Kota Yogyakarta 46

# DAFTAR TABEL

Tabel 2.1. Komponen dan Simbol *Use Case* 13

Tabel 2.2. Komponen dan Simbo-Simbol *Class Diagram* 14

Tabel 2.3. Komponen dan Simbo-Simbol *Activity Diagram* 15

Tabel 2.4. Komponen dan Simbo-Simbol *Squence Diagaram* 16

Tabel 2.5. Komponen dan Simbo-Simbol *Flwochart* 16

Tabel 3.1. Perangkat Keras Pembuatan Sistem 22

Tabel 3.2. Perangkat Lunak Pembuatan Sistem 22

Tabel 5.1. Pengujian *Black Box* 46

# Pendahuluan

## Latar Belakang

Kota Yogyakarta merupakan kota tujuan wisata setelah Bali yang tidak bisa dipungkiri pula memiliki siklus inflasi yang naik turun setiap bulannya. Kota Yogyakarta atau kota yang dijuluki kota pelajar ini memiliki jumlah penduduk sebesar 388.627 jiwa juga memiliki pengaruh terhadap naik turunnya inflasi yang terjadi di Kota Yogyakarta selain dari komponen-komponen lainnya yang membuat inflasi di Kota Yogyakarta memiliki siklus inflasi yang naik turun setiap bulannya.

Komponen-komponen selain jumlah penduduk Kota Yogyakarta yang menjadi pengaruh inflasi juga memiliki komponen yang menyebabkan inflasi adalah kebutuhan masyarakat seperti bawang merah, telur ayam ras, cabai merah, cabai rawit, listrik dan lain-lain merupakan komponen-komponen yang berpengaruh terjadinya inflasi di Kota Yogyakarta. Kebutuhan masyarakat setiap harinya semakin meningkat seiring jumlah penduduk Kota Yogyakarta juga meningkat, dimana kebutuhan masyarakat atau keinginan masyarakat untuk mengkonsumsi barang dan jasa meningkat baik keinginan masyarakat yang disertai oleh kemampuan untuk membeli ataupun keinginan masyarakat yang tidak disertai oleh kemampuan untuk membeli. Kemampuan masyarakat untuk membeli kebutuhan sehari-hari tidak jarang kebutuhan masyarakat tersebut tidak terpenuhi akibat terjadinya inflasi di Kota Yogyakarta, hal ini disebabkan oleh ketidak pastian atau tidak bisa diprediksi inflasi yang akan terjadi di Kota Yogyakarta.

Inflasi yang tidak stabil dan sulit untuk diprediksi merupakan masalah yang berpengaruh besar terhadap ekonomi ataupun efisiensi ekonomi, inflasi juga dapat berdampak terhadap distribusi pendapatan dan kekayaan, selain itu inflasi yang akan terjadi pada periode selanjutnya tidak bisa diterka-terka dengan melakukan pemikiran sederhana karena laju inflasi tidak menentu dan terkadang sulit untuk dapat dikendalikan. Inflasi yang tidak menentu dan terkadang sulit untuk dikendalikan berdampak buruk terhadap perekonomian. Karena hal ini bisa menjadi

tahap awal terjadinyan level inflasi yang lebih atas yaitu hiperinflasi yang dimana inflasi terjadi perhari lebih dari 1% yang artinya perbulan bisa mencapai lebih dari 30%. Ketidak setabilan laju inflasi ini bisa diselesaikan dengan menerapkan teknologi yang tepat untuk melakukan prediksi atau memperkirakan laju inflasi yang terjadi.

Teknologi yang bisa digunakan untuk melakukan prediksi terhadap inflasi di Kota Yogyakarta adalah dengan memanfaatkan Jaringan Saraf Tiruan dan memanfaatkan metode *Backpropagation*. Dengan memanfaatkan teknologi Jaringan Saraf Tiruan ini diharapakan dapat memberikan alternatif lain dalam memprediski inflasi di Kota Yogyakarta setiap bulannya. Sehingga kebutuhan masyarakat bisa terpenuhi. Dari permasalahan tersebut maka penulis ingin mengangkat judul penelitian yaitu, “Implementasi Jaringan Saraf Tiruan untuk Prediksi Inflasi di Kota Yogyakarta Menggunakan Metode Bacpropagation”.

## Rumusan Masalah

Berdasarkan uraian latar belakang di atas maka dapat dirumuskan, yaitu bagaimana Mengimplementasikan Jaringan Saraf Tiruan untuk Prediksi Inflasi di Kota Yogyakarta Menggunakan Metode Backpropagation?

## Batasan Masalah

Permasalahan yang akan dibahas dan dikaji pada penelitian ini memiliki batasan-batasan yang mencakup:

### Sistem prediksi yang dibangun hanya mencakup data inflasi pada Kota Yogyakarta.

### Sistem yang dibangun memiliki data latih dan data uji dari tahun 1979 sampai dengan tahun 2016.

### Sistem yang dibangun menggunakan data perbulan, bukan pertahun, perminggu ataupun perhari.

### Sistem yang dibangun tidak mencakup laporan.

### Sistem yang dibangun hanya memprediksi inflasi di Kota Yogyakarta.

f. Data yang digunakan berbentuk *time series.*

## Tujuan penelitian

Tujuan penelitian ini adalah mengimplementasikan jaringan saraf tiruan untuk memprediksi inflasi yang akan terjadi di Kota Yogyakarta menggunakan metode *backpropagation.* Sehingga dapat membantu memperkirakan inflasi yang akan terjadi pada bulan selanjutnya khususnya di Kota Yogyakarta.

## Manfaat Penelitian

Dengan mengimplementasikan jaringan saraf tiruan untuk memprediksi inflasi yang ada di Kota Yogyakarta ini diharapkan:

### Dapat mengetahui jumlah inflasi yang akan terjadi di Kota Yogyakarta pada bulan berikutnya.

### Dapat membantu masyarakat untuk menyesuaikan target perekonomian dengan jumlah inflasi yang terjadi di Kota Yogyakarta.

### Dapat mendorong mengurangi inflasi di Kota Yogyakarta.

## Sistematika Penulisan

Penulisan laporan tugas akhir ini terdiri dari beberapa bab dengan maksud untuk mempermudah pembacaan yang lebih akurat. Adapun sistematika penulisan laporan tugas akhir ini adalah sebagai berikut:

**BAB I : PENDAHULUAN**

Bab ini menjelaskan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

**BAB II : KAJIAN PUSTAKA DAN KAJIAN TEORI**

Bab ini menjelaskan mengenai kajian hasil penelitian dan kajian teori yang digunakan sebagai pembanding dan dasar teori yang terkait dengan penelitian tugas akhir sebagai bahan untuk penyusunan laporan tugas akhir.

**BAB III : METODE PENELITAN**

Bab ini menjelaskan mengenai obyek penelitian yang akan dilakukan beserta metode penelitian yang akan digunakan.

**BAB IV : ANALISIS DAN PERANCANGAN**

Bab ini menjelaskan tentang analisis sistem yang sedang berjalan, analisis ini menentukan solusi untuk sistem yang akan dibangun, dan perancangan antarmuka sistem yang sesuai dengan analisis solusi sistem tersebut.

**BAB V : IMPLEMENTASI**

Bab ini menjelaskan mengenai tools yang digunakan dalam membangun sistem, cara instalasi sistem yang telah dibangun dan tutorial dari sistem yang telah dibangun.

**BAB VI : PENUTUP**

Bab ini berisi kesimpulan dari keseluruhan hasil penelitian dan saran yang berguna untuk peneliti berikutnya.

**DAFTAR PUSTAKA**

**LAMPIRAN**

# BAB II KAJIAN PUSTAKA DAN KAJIAN TEORI

## 2.1 Kajian Hasil Penelitian

Dalam penyusunan tugas akhir ini, kajian yang digunakan diambil dari berbagai referensi hasil skripsi, jurnal, dan naskah publikasi yang sudah pernah dilakukan dengan tema yang hampir sama.

Penelitian oleh M.F Andrijiasa dan Mistianingsih (2010), dengan judul Penerapan Jaringan Syaraf Tiruan Untuk Memprediksi Jumlah Pengangguran di Provinsi Kalimantan Timur dengan Menggunakan Algoritma Pembelajaran *Backpropagation*. Penelititan tersebut membahas bagaimana menerapkan JST untuk memprediksi jumlah pengangguran di Provinsi Kalimantan Timur dengan menggunakan algoritma pembelajaran *backpropagation* berdasarkan data survey lapangan sebagai tolok ukur perlu tidaknya ditinjau kembali peningkatan lapangan pekerjaan yang bersesuaian. Data yang digunakan adalah data dari tahun 2004 samapai dengan tahun 2008. Hasil pengujian dengan JST *backpropagation* dapat memprediksi jumlah penganguran dan target *output* yang diingikan lebih mendekati ketepatan dalam melakukan pengujian yaitu dengan dilakukannya pengujian jumlah pengangguran di tahun 2009 adalah 133.104, sedangkan hasil prediksi pengguran tahun 2008 yang dilakukan BPS Provinsi Kalimantan Timur adalah 139.830.

Penelitian oleh Nanik Susanti (2014), dengan judul Penerapan Model *Neural Network Backpropagation* untuk Prediksi Harga Ayam. Penelitan tersebut membahas tentang prediksi harga secara tepat di sektor unggas yang menyebabkan optimalisasi alokasi sumber daya, peningkatan efisiensi dan meningkatkan pendapatan industri unggas. Pada penelitian tersebut digunakan metode *neural network bacpropagation* untuk memprediksi harga ayam. Data yang digunakan adalah data rentet waktu dari bulan Agustus 2010 sampai dengan bulan Mei 2013, sebanyak 1015. Arsitektur yang digunakan pada penelitian tersebut adalah 1 lapisan input dengan 4 *neuron,* 1 lapisan *hidden* dengan 10  *neuran* dan 1 lapisan *output* dengan 1 *neuron*. Nilai MSE yang dihasilkan dalam penelitian tersebut adalah 0.0113 dan nilai koefisien korelasi untuk data pelatihan sebesar 0.961661 serta nilai koefisien korelasi untuk data pengujian sebesar 0.8696.

Penelitian oleh Amrin (2014), dengan judul Peramalan Tingkat Inflasi Indonesia Menggunakan *Neural Network Backpropagation* Berbasis Metode *Time Series*. Penelitian tersebut membahas tentang prediksi tingkat inflasi bulanan di Indonesia. Analisis data yang dilakukan disimpulkan bahwa performa model *neural network* dengan *backpropagation* yang dibentuk dari data *Training* dan divalidasi pada data *testing* memberikan tingkat akurasi prediksi yang cukup baik dengan nilai *Man Square Error* (MSE) 0.0171. Data yang digunakan adalah data sekunder, yaitu data inflasi bulanan Indonesia menurut kelompok pengeluaran tahun 2006-2014.

**2.2 Kajian Teori**

Dalam kajian teori ini berisi pembahasan mengenai konsep dasar teori yang digunakan sebagai landasan dalam mengerjakan tugas akhir ini. Pembahasan ini bermaksud untuk menguraikan teori dan algoritma yang digunakan dalam sistem.

**2.2.1 Inflasi**

Inflasi adalah meningkatnya harga-harga umum secara terus menerus. Kenaikan harga berlangsung dalam waktu lama yang terjadi hampir di seluruh barang dan jasa, hal ini disebut dengan inflasi. Apabila terjadi kenaikan harga satu atau dua hari pada barang atau jasa hal ini tidak dapat dikatakan inflasi.

Menurut Suwardjono (2008), Inflasi merupakan gejala kenaikan tingkat harga umum dari waktu ke waktu terus menaik. Inflasi ditunjukkan oleh indeks harga dari periode satu ke periode berikutnya disebut dengan laju inflasi (*rate of inflation*). Gejala ini ditunjukkan oleh statistik yang menujukkan bahwa daya beli uang menurun yang berarti penurunan indeks harga terefleksi pula perubahan daya beli atau nilai tukar uang, sedangkan kenaikan indeks harga berarti penurunan daya beli.

**2.2.2 Data *Time Series***

Data *Time Series* merupakan data yang berbentuk deret waktu baik dalam bentuk hari, Minggu, bulan ataupun tahun atau bisa juga *time series* dikatakan himpunan observasi data terurut dalam waktu. Data *time series* biasa digunakan untuk melakukan peramalan atau prediksi data selanjutnya yang dianalisis dari data-data sebelumnya.

**2.3 Jaringan Saraf Tiruan**

Jaringan saraf tiruan merupakan salah satu sistem yang dapat mengubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut atau bisa disebut pemodelan cara kerja atau fungsi sistem saraf manusia dalam melakukakan tugas-tugas tertentu.

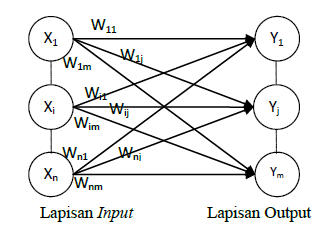
Menurut Hermawan, A. (2006), jaringan saraf tiruan (*artificial neural network*), atau disingkat JST, adalah sistem komputasi dimana arsitektur dan operasi diilhami dari pengetahuan tentang sel saraf biologi di dalam otak, yang merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba mentimulasi proses pembelajaran pada otak manusia tersebut.

**2.4 Arsitektur Jaringan Saraf Tiruan**

Arsitektur jaringan saraf tiruan merupakan seuah pengaturan neuron di dalam lapisan atau biasa disebut layer beserta dengan hubungan antar lapisan. Arsitektur inilah yang akan menentukan keberhasilan dari sebuah target yang dinginkan. Hal ini dikarenakan tidak semua permasalahan dapat diselesaikan dengan arsitektur yang sama. Adapun jenis-jeni arsitektur jaringan saraf tiruan adalah sebagi berikut:

**2.4.1 Jaringan Lapisan Tunggal (*Single Layer Network*)**

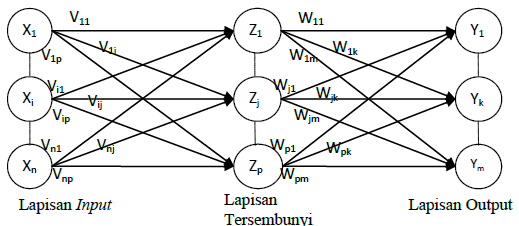
Jaringan lapisan tunggal merupakan arsitektur jaringan saraf tiruan yang dimana memiliki neuron-neuron diorganisasi dalam bentuk layer-layer yaitu layer *input* dan layer *output*. Adapun gambar dari lapisan tunggal pada jaringan saraf tiruan ini dapat dilihat pada Gambar 2.1 dibawah:



**Gambar 2.1** Jaringan Lapisan Tunggal

**2.4.2 Jaringan dengan Banyak Lapisan (*Multilayer Network*)**

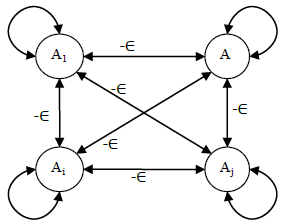
Terdapat perbedaan pada *Multilayer Network* dengan *Single Network* adalah pada banyak layer yang dimiliki yaitu, pada *Multilayer Network* memiliki *Hidden Layer* diantara layer *input* dan layer *output*. Adapun gambar dari *Multilayer Network* dapat dilihat pada Gambar 2.2 di bawah:



**Gambar 2.2** *Multilayer Network*

**2.4.3 Jaringan dengan Lapisan Kompetitif (*Competitive Layer Network*)**

Pada lapisan kompetitif ini hubungan antar *neuron* tidak diperlihatkan pada diagram arsitektur. Adapun gambar dari *Competitive Layer Netowork* dapat dilihat pada Gambar 2.3 di bawah:



**Gambar 2.3** *Competitive Layer Network*

**2.5 *Backpropagation***

Algoritma *Backpropagation* merupakan algoritma *learning* yang populer untuk memecahkan kasus-kasus yang rumit. Algoritma ini melakukan dua bentuk perhitunga, yaitu, perhitungan maju untuk menghitung *error* antara keluaran *actual* dan target dan perhitungan mundur yang mempropagasikan balik *error*  tersebut untuk memperbaiki bobot-bobot pada semua *neuron* yang ada.

Menurut Dillak, R. Y. dkk (2013), *Backpropagation* adalah bentuk jaringan saraf tiruan yang terdiri dari beberapa layer. JST *backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakanselama pelatihan serta kemampuan jaringan untuk memberikan respons yang besar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan.

Adapun alur perhitungan darialgoritma *backpropagation* baik proses hitung maju maupun proses hitung mundur sebagai berikut:

**2.5.1 Perhitungan Maju**

Perhitungan Maju atau dikenal dengan *feedforward* merupakan proses perhitungan pada algoritma *backpropagation* dilakukan setelah dilakukannya inisialisai bobot dengan cara mengacak angka kecil rentang 0 sampai 1 dan *load data* dari *database* diperlukan perhitungan maju yang melibatkan fungsi aktivasi yaitu yang digunakan di sini adalah fungsi aktivasi sigmoid, untuk mengaktifkan atau menonaktifkan *neuron* pada *layer* yang sedang dihitung pada langkah pertama hitung maju akan dilakukan yaitu dari setiap *neuron input layer* ke setiap *neuron hidden layer* dan dilanjutkan perhitungan dari *hidden layer* ke *output layer.* Adapun langkah-langkah dari perhitungan maju ini adalah:

Pertama dilakukan perhitungan fungsi aktivasi Sigmoid *neuron input layer* ke *neuron hidden layer*

Dimana:

Keterangan :

Z = *neuron hidden layer*

Wi1 = bobot ke i diantara *input layer* dan *hidden layer*

Xi = nilai *neuron input layer* ke i

b1 = nilai bias

Lalu dilanjutkan dengan proses perhitungan dari *neuron hidden layer* ke *neuron output layer*

Dimana :

w

**2.5.2 Perhitungan Mundur**

Perhitungan mundur merupakan proses yang dilakukan untuk mempogpragasikan balik *error* untuk memperbaiki bobot-bobot pada semua *neuron* arsitektur jaringan saraf tiruan. Dimana Hitung mundur dilakukan setelah proses penghitungan maju selesai yaitu, dimulai dari *neuron output layer* lalu ke *neuron hidden layer* dan dari *neuron hidden layer* ke *neuron input layer.*

Pertama dilakukan perhitungan *error* yaitu selisih dengan target pada data latih.

Keterangan:

E = *Error training*

y = *output neuron*

t = target data latih

Menghitung faktor koreksi pada *output layer* untuk setiap *neuron output layer*

Menghitung delta untuk bobot antara *output layer* dengan *hidden layer*

Keterangan:

α = *learning rate*

Menghitung faktor koreksi pada *hidden layer*

Dimana:

Menghitung delta untuk bobot antara *hidden layer* dengan *input layer*

Adapun perhitungan untuk perubahan bobot antara *input layer* dan *hidden layer* dihitung dengan rumus:

Perubahan bobot antara *hidden layer* dan *output layer* dihitung dengan rumus:

Hitung MSE (*Mean Squared Error*)

Keterangan:

= Selisih antara nilai target – dengan nilai prediksi (*Error Training*)

n = Jumlah data latih

**2.6 MAPE (*Mean Absolute Percentage Error*)**

Merupakan persentase rata-rata kesalahan antara data aktual atau asli dengan data hasil peramalan atau prediksi. Adapun rumus yang digunakan dalam MAPE adalah

Keterangan:

Aktual = Data asli

Ramalan = Hasil prediksi

n = Jumlah data

**2.7 Aproksimasi**

Aproksimasi merupakan cara yang digunakan untuk melakukan pembatasan nilai toleransi dari hasil prediksi yang dilakukan, yaitu, batas bawah maupun batas atas dari hasil prediksi. Adapun rumus yang digunakan untuk menentukan batas bawah dan batas atas adalah:

*Batas Atas Pengukuran = x + Δx*

*Batas Bawah Pengukuran = x – Δx*

Keterangan:

x = Hasil Prediksi

**2.8 UML**

UML adalah singkatan dari *Unified Modelling Languange* yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau dalam artian UML sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem *software*.

Komponen-komponen utama dari UML adalah *view(use case view, logical view, component view, concurrency view* dan *Devloytment view), model element*, *general mechanism* dan diagram. Adapun jenis-jenis diagram UML adalah sebagai berikut:

**2.8.1 *Use Case Diagram***

*Use Case Diagram* merupakan salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, *use case diagram* juga dapat mendeskripsikan tipe interaksi antara pemakai sistem dengan sistem itu sendiri. Berikut komponen dan simbol-simbol pembentuk diagram *use case* dapat dilihat pada Tabel 2.1.

**Tabel 2.1** Komponen dan Simbol *Use Case*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Notasi | Keterangan | Simbol |
| 1 | Aktor | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan sistem. |  |
| 2 | *Use Case* | Menggambarkan fungsi tertentu berupa aktivitas atau kejadian yang terdapat pada sistem. |  |
| 3 | Sistem | Menggambarkan paket yang menampilkan sistem secara terbatas |  |
| 4 | Asosisasi | Menghubungkan antara objek satu dengan objek yang lain |  |
| 5 | *Include* | Menspesifikasikan bahwa *use case*  sumber secara *eksplisit* |  |
| 6 | *Extend* | *Use case* yang terdiri dari langkah yang diekstraksi dari *use case* yang lebih *kompleks* |  |

**2.8.2 *Class Diagram***

*Class Diagram* merupakan salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan dalam artian diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi-relasi yang terdapat pada sistem tersebut. Berikut komponen dan simbol-simbol pembentuk diagram *Class Diagram* dapat dilihat pada Tabel 2.2.

**Tabel 2.2** Komponen dan Simbol *Class Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Notasi | Keterangan | Simbol |
| 1 | *Package* | Merupakan sebuah bungkusan dari satu atau lebih kelas | |  | | --- | |  | |  | | |
| 2 | Kelas | Merupakan blok-blok pembangun pada pemrograman berorientasi objek | |  | | --- | | Nama kelas | | atribut | | Method() | |
| 3 | Asosiasi | Merupakan sebuah *relationship* yang paling umum untuk menghubungkan dua kelas | 1..\* 1 |
| 4 | *Dependency* | Relasi antar kelas dengan maksud ketergantungan antar kelas |  |
| 5 | Generalisasi | Relasi antar kelas dengan maksud generalisasi-spesialisasi (khusus) |  |
| 6 | *Agregasi* | Relasi antar tabel dengan maksud semua bagian |  |

**2.8.3 *Activity Diagram***

*Activity* *Diagram* adalah salah satu jenis diagram pada UML yang dapat memodelkan proses apa saja yang terjadi pada sistem. Berikut komponen dan simbol-simbol pembentuk diagram *Activity Diagram* dapat dilihat pada Tabel 2.3.

**Tabel 2.3** Komponen dan Simbol *Activity Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Notasi | Keterangan | Simbol |
| 1 | *Activity* | Aktivitas yang dilakukan sistem yang menggunakan bagaimana masing-masing kelas antarmuka saling berintraksi. |  |
| 2 | *Initial Node* | Menggambarkan bagaimana objek dibentuk atau diawali |  |
| 3 | *Activity Final*  *Node* | Menggambarkan bagaimana objek dibentuk dan diawali |  |
| 4 | *Decision Node* | Digunakan untuk menggambarkan suatu keputusan yang harus diambil pada kondisi tertentu |  |
| 5 | *Swimlane* | Menunjukan siapa yang bertanggung jawab dalam melakukan aktivitas dalam suatu diagram |  |
| 6 | *Line Connetor* | Digunakan untuk menghubungkan satu simbul dengan simbol yang lain |  |
| 7 | *Fork* | Digunakan untuk menunjukkan yang dilakukan secara parallel |  |
| 8 | *Join* | Digunakan untuk menunjuk kegiatan yang digabungkan |  |

**2.8.4 *Squence Diagram***

*Squence Diagram* merupakan salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu, *squence diagram* juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan suatu seperti pada *use case diagram.* Berikut komponen dan simbol-simbol pembentuk diagram *Squence Diagram* dapat dilihat pada Tabel 2.4.

**Tabel 2.4** Komponen dan Simbol *Squence Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Notasi | Keterangan | Simbol |
| 1 | *Objeck* | Merupakan *instance* dari sebuah kelas dan dituliskan tersusun secara horizontal | Objek |
| 2 | Aktor | Aktor dapat berkomunikasi dengan objek, maka aktor juga dapat diurutkan sebagai kolom |  |
| 3 | Aktivitas | Aktivitas dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah *lifeline* |  |
| 4 | *Message* | Mengindikasikan komunikasi antar objek |  |
| 5 | *Lifeline* | *Lifeline* menginikasikan keberadaan sebuah objek dalam basis waktu |  |
| 6 | *Self-message* | Mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri |  |

**2.9 Diagram Alir (*Flowchart*)**

Diagram alir atau *flowchart* merupakan bagan yang memiliki simbol-simbol tertentu untuk menggambarkan urutan proses secara mendetail dan hubungan antar suatu proses dengan proses lainnya dalam suatu sistem atau program.

Menurut Kadir, A. (2012), Diagram alir atau *flowchart* adalah suatu standar untuk menggambarkan suatu proses. Setiap langkah dalam algoritma dinyatakan dengan sebuah simbol dan aliran setiap langkah yaitu dari suatu langkah ke langkah yang lain dinyatakan dengan gari yang dilengkapi panah.

Adapun simbol-simbol dari diagram alir atau *flowchart* ini dapat dilihat pada tabel 2.5.

**Tabel 2.5** Komponen dan Simbol *Flowchart*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Notasi | Keterangan | Simbol |
| 1 | Terminator | Menyatakan titik awal atau titik akhir diagram alir. |  |
| 2 | *Input/Output* | *Input/output* atau terkadang disebut data, digunakan untuk menyatakan operasi pemasukan data atau penampilan data. |  |
| 3 | Proses | Menyatakan sebarang proses, misalnya untuk menyatakan susatu operasi aritmetika. |  |
| 4 | Keputusan | Digunakan untuk melakukan pengambilan keputusan. Dalam hal ini, yang ada dalam simbol ini berupasuatu pertanyaan yang jawabannya berupa dua kemungkinan, yaitu YA atau TIDAK. |  |
| 5 | Proses Terdefinisi | Menyatakan prosedur lain yang telah didiagramalirkan. |  |
| 6 | Konektor | Digunakan untuk menghubungkan ke berbagai bagian dalam diagram alir. |  |

**2.10 Node Js**

Node Js adalah sebuah *platform* yang dibangun diatas Chrom’s JavaScript Runtime (v8). Node Js menggunakan *event-drivent, non-blocking* I/O yang membuat Node Js ringan dan efisien. Node Js merupakan Suu perangkat lunak yang didesain untuk membangun aplikasi berbasis Wet yang dieksekusi sebagai aplikasi server, yang menggunakan bahasa JavaScript dalam implementasinya.

**2.11 JSON**

JSON adalah suatu format pertukaran data yang ringan, mudah dibaca ditulis oleh manusia. Serta mudah diterjemahkan oleh komputer. Format JSON dibuat dari bahasa JavaScript, standar ECMA-262 Edisi ke 3 Desember 1999. Format JSON tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh Programme seperti C, C++, C#, Java, JavaScript All. Sehingga JSON baik digunakan untuk pertukaran data.

**2.12 NPM**

NPM adalah manajer peket punuk JavaScript dan registri *software* terbesar untuk keperluan penginstalan berbagai paket yang dibutuhkan dalam Node Js serta mengelola dependensi dalam proyek kerja Node Js dan mendistribusikan kode program.

**2.13 Angular Js**

Menurut Julisman, A. (2014), Angular Js adalah *Framework* Javascript yang dikembangkan oleh Google dan banyak digunakan pada produk-produk yang dibuat oleh Google. AngularJs telah menjadi standarisasi untuk keperluan pembuatan Aplikasi Web Dinamis dari sisi *Client,* karena kemudahan dan kecepatannya dalam melakukan komunikasi *Server* ke *Client*.

**2.14 MongoDB**

MongoDB adalah basis data NoSql (*not only sql*) yang merupakan *Documen-Oriented Database* dan merupakan *open source project* yang menggunakan struktur data JSON untuk menyimpan datanya. MongoDB sering digunakan untuk aplikasi berbasis Cloud, Grid Computing, atau Big Data.

**BAB III**

**METODE PENELITIAN**

1. **Objek Penelitian**

Objek dari penelitian ini merupakan inflasi, yaitu khusunya di Kota Yogyakarta. Inflasi atau kenaikan harga barang umum secara terus menerus pada setiap periode yang dapat mengukur baik atau buruknya prekonomian pada suatu daerah atau cakupan lebih luasnya adalah pada suatu negara, sangat perlu diketahui jumlah inflasi yang akan terjadi pada periode selanjutnya dengan cara memprediksi tingkat inflasi yang akan terjadi pada periode selanjutnya apakah akan terjadi kenaikan yang signifikan ataukah akan terjadi penurunan tingkat inflasi yaitu khususnya pada Kota Yogyakarta.

1. **Metode Penelitian**

Metode Penelitian yang digunakan dalam penelitian ini adalah metode *Waterfall* yang merupakan metodologi yang digunakan untuk membuat atau membangun sistem. Dalam metode *waterfall* memiliki tahapan-tahapan pengerjaan sampai sistem dikatakan selesai, yaitu, *requirements analysis and definition, system and software design, implementation and unit testing, integratif and system testing* dan *operation and maintenance.*

**3.2.1 Analisis**

Analisi merupakan tahapan yang dilakukan sebelum dilakukannya pembangunan sistem. Hal ini bertujuan untuk mempermudah dalam pengerjaan atau pembangunan suatu sistem. Adapun tahapan dalam analisis ini adalah :

a. Pengumpulan Data (*Data Gathering*)

Tahap pengumpulan data merupakan tahap yang dilakukan untuk memenuhi aspek-aspek yang dibutuhkan dalam sistem prediksi inflasi di Kota Yogyakarta, yaitu diantaranya melakukan pencarian data inflasi Kota Yogyakarta di Badan Pusat Statistik yang mencakup data dari tahun 1979 sampai dengan tahun 2016 yang berbentuk *time series*, yaitu data yang digunakan adalah data perbulan dari semua data yang sudah didapatkan.

b. Pengolahan Data (*Data Processing*)

Tahap Pengolahan data ini merupakan tahapan yang mencakup pengelolaan data Inflasi Kota Yogyakarta setiap bulannya yang dipisahkan menjadi data *input* yaitu data yang digunakan sebagai data yang diolah pada *neuron input*  pada jaringan saraf tiruan dan data *output* yang menjadi target. Data inflasi Kota Yogyakarta yang sudah ditetapkan menjadi data *input* maupun data *output* akan dilakukan proses pelatihan dan pengujian pada sistem prediksi inflasi di Kota Yogyakarta.

**3.2.2 Pembuatan Sistem**

Pembuatan sistem merupakan tahapan yang dilakukan untuk membangun sistem setelah dilakukannya penelitian sistem. Adapun tahapannya adalah:

a. Perancangan Arsitektur Jaringan Saraf Tiruan

Tahapan ini dilakukan perancangan arsitektur jaringan saraf tiruan yang menggunakan arsitektur jaringan saraf tiruan *Multy Layer Perceptron* (MLP) dan algoritma yang digunakan adalah algoritma *backpropagation* untuk melakukan prediksi inflasi di Kota Yogyakarta, pada arsitektur yang digunakan dalam sistem prediksi inflasi ini digunakan 5 *neuron input* pada *layer input,* *hidden layer* yang disesuaikan *nueron hiddennya* sampai mendapatkan hasil yang optimal, dan terdapat 1 *nuron output* pada *layer output*. Arsitektur ini yang akan menentukan perhitungan yang dilakukan oleh algoritma *backpropagation* pada kasus prediksi inflasi di Kota Yogyakarta.

b. Perancangan Sistem

1. Perancangan Prilaku Sistem

Perancangan perilaku sistem merupakan gambaran bagaimana sistem ini digunakan oleh setiap yang berintraksi dengan sistem prediksi inflasi di Kota Yogyakarta atau dalam artian perancangan prilaku sistem ini digunakan untuk menggambarkan sistem yang digunakan oleh pengguna sistem. Pada tahapan ini dilakukan perancangan bagaimana sistem tersebut berprilaku dengan menggunakan *Use Case Diagram*. Terdapat pula perancangan yang menggambarkan bagaimana sistem berfungsi secara keseluruhan baik dalam pelatihan data inflasi Kota Yogyakarta maupun pengujian data inflasi Kota Yogyakarta dan komponen-komponennya akan digambarkan prosesnya dengan menggunakan *Actifity Diagram*.

2. Perancangan Proses

Perancangan proses merupakan penjabaran bagaimana setiap proses berjalan pada sistem prediksi inflasi di Kota Yogyakarta. Proses-proses yang berjalan akan digambarkan atau diseketsakan alur kerja dari sistem prediksi inflasi di Kota Yogyakarta dengan menggunakan *Flowchart* yang akan menggambarkan urutan proses-proses pada sistem prediksi inflasi Kota Yogyakarta bagaimana hubungan proses satu dengan proses lainnya. Proses yang dimaksud adalah instruksi-intruksi pada program sistem prediksi inflasi di Kota Yogyakarta.

3. Desain *Interface*

Desain *interface* pada sistem prediksi inflasi di Kota Yogyakarta dilakukan bertujuan untuk menyesuaikan kebutuhan pengguna (*user*) yang akan menggunakan sistem prediksi inflasi di Kota Yogyakarta dibuat sesederhana mungkin tapi tidak menghilangkan unsur penting di dalam sistem prediksi inflasi yang dibangun. Desain *interface* sistem ini dibuat menggunakan *HTML* sebagai kerangka dan tata letak *website*nya dan *CSS* digunakan untuk mempercantik tampilan dari sistem prediksi inflasi di Kota Yogyakarta.

4. Implementasi Sistem

Tahap Implementasi sistem ini dilakukan setelah melakukan proses penataan tampilan atau desain *interface* yaitu untuk melakukan pengaktifan fungsi-fungsi yang dibutuhkan pada sistem prediksi inflasi di Kota Yogyakarta. Sistem ini akan diberikan fungsi-fungsi dengan cara penulisan *syntax* program menggunakan bahasa pemrograman *NodeJs* yang ditulis menggunakan *editor* atau *tool Sublime*, serta penggunaan tempat penyimpanan datanya digunakan basis data *NoSQL* yaitu basis data *MongoDB.*

**3.3 Perangkat Pendukung Penelitian**

Dibutuhkan perangkat pendukung yaitu untuk membantu proses pembuatan sistem prediksi inflasi di Kota Yogyakarta sehingga dapat membantu kinerja pembuatan sistem dengan baik, baik itu dalam bentuk perangkat keras maupun perangkat lunak.

**3.3.1 Perangkat Keras (*Hardware*)**

Perangkat keras yang mendukung dalam pembuatan sistem prediksi inflasi di Kota Yogyakarta memiliki spesifikasi yang bisa dilihat pada Tabel 3.1.

**Tabel 3.1** Perangkat Keras Pembuatan Sistem

|  |  |
| --- | --- |
| Processor | Intel(R) Core(TM) i3-2350M CPU @2.30GHz (4 CPUs), ~2.3GHz |
| Memory | 4GB DDR3 |
| VGA | Intel ® HD Graphics 3000 |
| Monitor | LCD 14” |
| Hard Disk | 500 GB |

**3.3.2 Perangkat Lunak (*Software*)**

Perangkat lunak merupakan perangkat yang dapat mendukung dalam hal membangun sistem prediksi inflasi di Kota Yogyakarta dalam sisi kebutuhan aplikasi pendukung untuk membangun sistem. Adapun aplikasi yang dilibatkan bisa dilihat pada Tabel 3.2.

**Tabel 3.2** Perangkat Lunak Pembuatan Sistem

|  |  |
| --- | --- |
| Sistem Operasi | Linux Ubuntu 16.04 |
| Aplikasi | SublimeText, Mongo, NPM, Edraw Max, Microsoft Word 2010. |
| Bahasa Pemrograman | NodeJs, HTML (Bahasa Markup), CSS, Angularjs. |

**BAB IV**

**ANALISIS DAN PERANCANGAN**

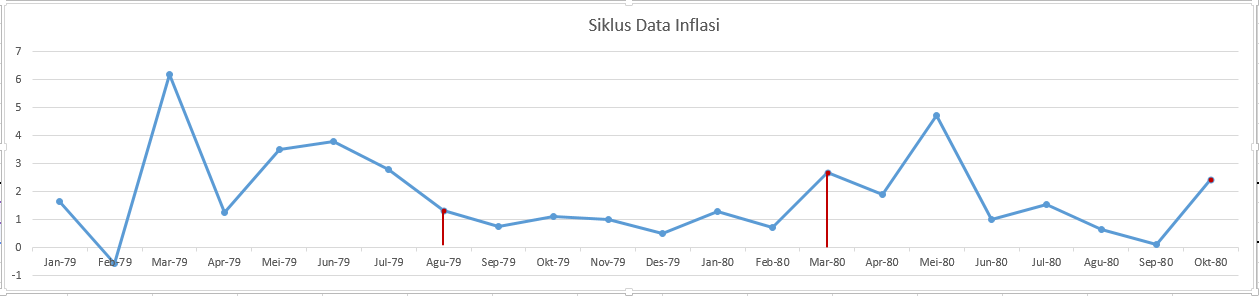
**4.1 Analisis Sistem**

Siklus inflasi sering digunakan untuk menentukan target inflasi yang akan terjadi pada periode selanjutnya dikarenakan dari siklus inflasi yang terjadi dapat menjadi patokan atau target inflasi yang harus ditempuh, atau siklus inflasi yang sudah terjadi bisa menjadi acuan untuk menghadapi laju inflasi pada periode selanjutnya atau sering disebut dengan paket kebijakan moneter.

Secara umum ada beberapa cara perhitungan yang dilakukan untuk menghitung laju inflasi sebelum menentukan kenaikan inflasi yang akan terjadi pada periode selanjutnya yaitu dengan cara menghitung Indeks Harga Konsumen (IHK) atau dalam kata lain yaitu memantau inflasi dari periode ke periode harus dilakukan secara cermat dengan melihat indikator-indikator perubahan harga pada komoditas tertentu hal ini memungkinkan perkiraan yang kurang tepat. Kemudian dengan cara melakukan perhitungan DGP *Deflator* (*Gross Domestic Product)* merupakan cara perhitungan dengan melihat indikator kenaikan harga barang tertentu dan nilai GDP *Deflator* diperoleh dari total jumlah barang dan jasa yang dihasilkan oleh unit-unit produksi di wilayah dalam negeri atau domestik yang dihitung dalam kurun waktu satu tahun dan memungkinkan akurasi yang diperoleh kurang baik karena masih menggunkan harga komoditi tertentu untuk melakukan perkiraan inflasi pada periode selanjutnya. Oleh karena itu diperlukan sistem prediksi inflasi yang menentukan akurasi yang tinggi dan cepat memperoleh hasil prediksi inflasi yang akan terjadi pada periode berikutnya, yaitu, dengan menggunakan pemilahan data yang bersifat terus menerus atau *Time series* atau dalam artian melakukan prediksi dengan menggunakan data-data inflasi sebelumnya.

**4.1.1 Data Masukan**

Data masukan pada sistem ini adalah data inflasi yang sudah terjadi pada bulan-bulan sebelumnya yaitu dari bulan Januari tahun 1979 sampai bulan Mei tahun 2016. Adapun dalam sistem ini harus melihat siklus data yang terjadi supaya proses perhitungan tergolong bagus yaitu dengan melakukan pemetaan pola data, adapun caranya sebagai berikut :



**Gambar 4.1** Grafik Data Inflasi Jan 1979 Sampai Des 1982

Pada gambar grafik di atas terdapat bentuk pola dari data bulan kebulan setiap tahunnya yang dimana memiliki pola data naik turunya yang tetap konsisten pada setiap 8 bulan. Adapun 8 bulan tersebut adalah data masukan yang digunakan dalam pembelajaran oleh sistem atau bisa dikatakan sistem lebih mudah mengenali pola data yaitu dari bulan ke n-7 sampai bulan ke n.

**4.1.2 Proses Perhitungan**

Proses perhitungan data yang dilakukan setelah menemukan pola data yang tepat akan dilakukan proses perhitungan yang dilakukan dengan jaringan saraf tiruan dengan metode *backpropagation*.

**4.1.3 Hasil Prediksi**

Hasil yang dikeluarkan olah sistem ini setelah melakukan perhitungan adalah prediksi inflasi di Kota Yogyakarta untuk periode selanjutnya.

**4.2 Perancangan Sistem**

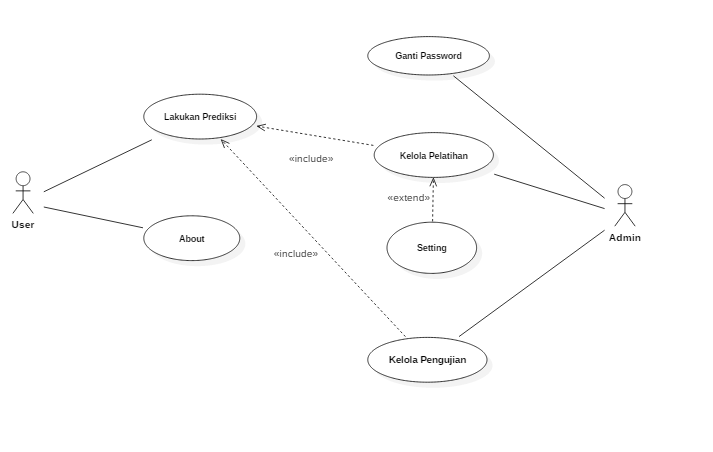
Perancangan sistem merupakan proses pengembangan sistem atau penyusunan sistem yang baru. Pada tahap perancangan sistem ini harus diperhatikan ketentuan dan struktur yang dapat membangun sebuah sistem. Perancangan sistem akan menghasilkan sebuah kesesuaian kebutuhan pemakai sistem, serta sistem yang dirancang dapat dikembangkan kembali.

**4.2.1 UML**

UML atau *Unified Modeling Language* merupakan bahasa pemodelan yang sering digunakan untuk standar industri dalam hal pemodelan perangkat lunak yang didalamnya terdapat notasi atau simbol-simbol dan konsep atau makna dari simbol-simbol yang dimiliki UML serta aturan yang sudah ditentukan oleh bahasa pemodelan ini untuk menyatukan simbol-simbol yang sudah ada pada UML. UML tidak mencakup metode pemodelan melainkan hanya sebuah bahasa pemodelan yang artinya UML tidak bergantung pada metode apapun atau bisa menggunakan metode apapun dalam bahasa pemodelan UML ini yang dimana UML memiliki diagram pemodelan perilaku dan diagram pemodelan struktur.

Perancangan sistem yaitu dalam hal ini adalah perancangan sistem prediksi untuk memprediksi jumlah inflasi di Kota Yogyakarta menggunakan UML hanya dalam segi pemodelan perilaku dari sistem prediksi ini.

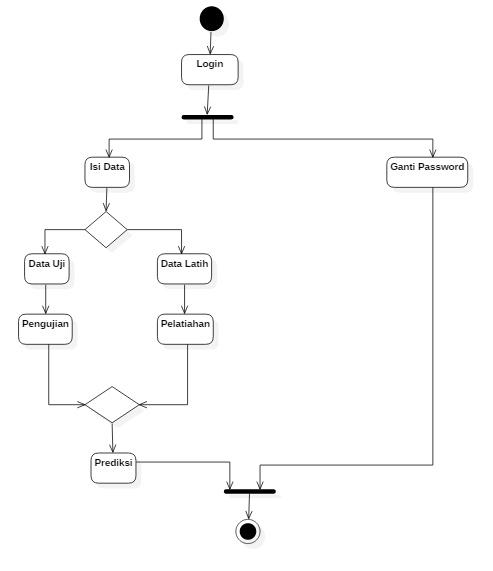
1. *Use Case Diagram*

*Use Case Diagram* menggambarkan sekelompok *use case* dan aktor yang disertai dengan hubungan diantara *use case* dan aktor yang ada. Dalam perancangan sistem prediksi jumlah inflasi di Kota Yogyakarta ini terdapat 2 aktor yaitu, amin dan *user*. Serta mememiliki beberapa *use case* yang menerangkan kegiatan-kegiatan yang dilakukan sistem prediksi ini. *Use case diagram* digambarkan pada Gambar 4.2.

**Gambar 4.2** *Use Case Diagram*

1. *Activity Diagram*

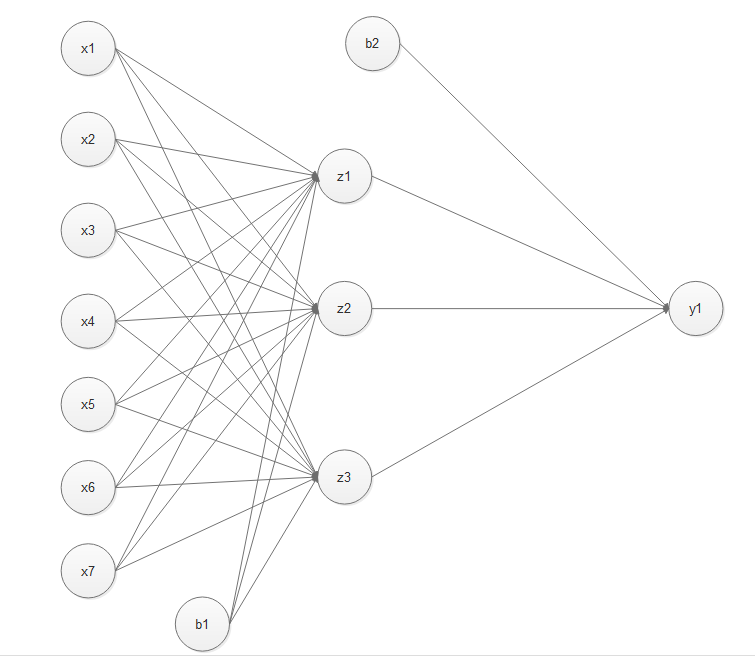
*Activity* *Diagram* merupakan diagram yang menggambarkan aktivitas yang terjadi pada sistem. *Activity diagram* menujukkan langkah dan proses kerja sistem yang dibuat dari peram sampai akhir. Adapun *activity diagram* dari sistem ini adalah pada Gambar 4.3.



**Gambar 4.3** *Activity Diagram*

**4.2.2 Arsitektur Jaringan Saraf Tiruan**

Arsitektur jaringan saraf tiruan yang digunakan pada sistem prediksi inflasi di Kota Yogyakarta ini adalah arsitektur *MultiLayer Perceptron* (MLP). Adapun gambar dari arsitektur jaringan saraf tiruan ini bisa dilihat pada Gambar 4.4.



**Gambar 4.4** Arsitektur Jaringan Saraf Tiruan.

x1 = Bulan periode 1 z1 .... zN = *Neuron Hidden Layer*

x2 = Bulan periode 2 y1 = *Neuron Output Layer*

x3 = Bulan periode 3 b1 = Bias *input layer*

x4 = Bulan periode 4 b2 = Bias *hidden layer*

x5 = Bulan periode 5

x6 = Bulan periode 6

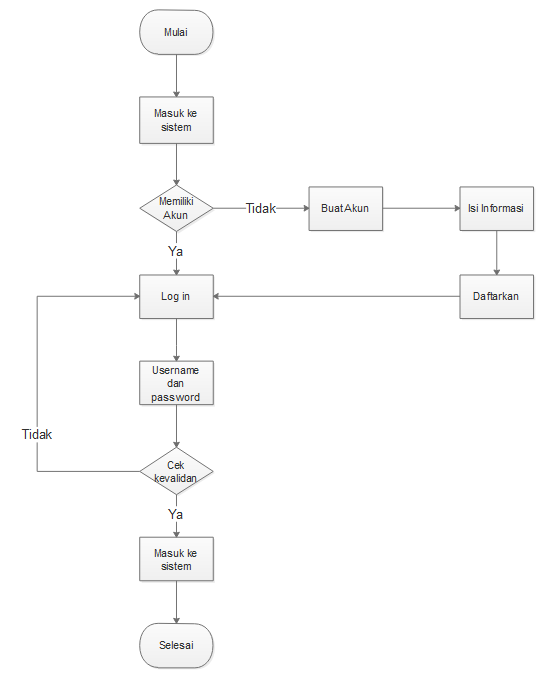
x7 = Bulan periode 7

**4.2.3 Flowchart Program**

*Flowchart* *Program* merupakan keterangan yang lebih rinci tentang bagaimana setiap langkah program atau prosedur dalam urutan yang tepat saat pemrosesan program.

1. *Flowchart Login*

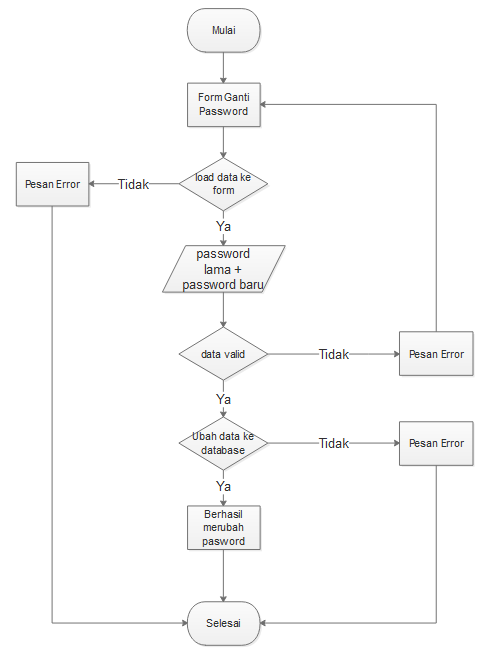
*Flowchart login* adalah diagram alir yang menggambarkan tentang proses *login* mulai dari awal sampai memasuki sistem. Adapun gambar dari *flowchart login* bisa dilihat pada Gambar 4.5.



**Gambar 4.5** *Flowchart Login*

1. *Flowchart* Ganti *Password*

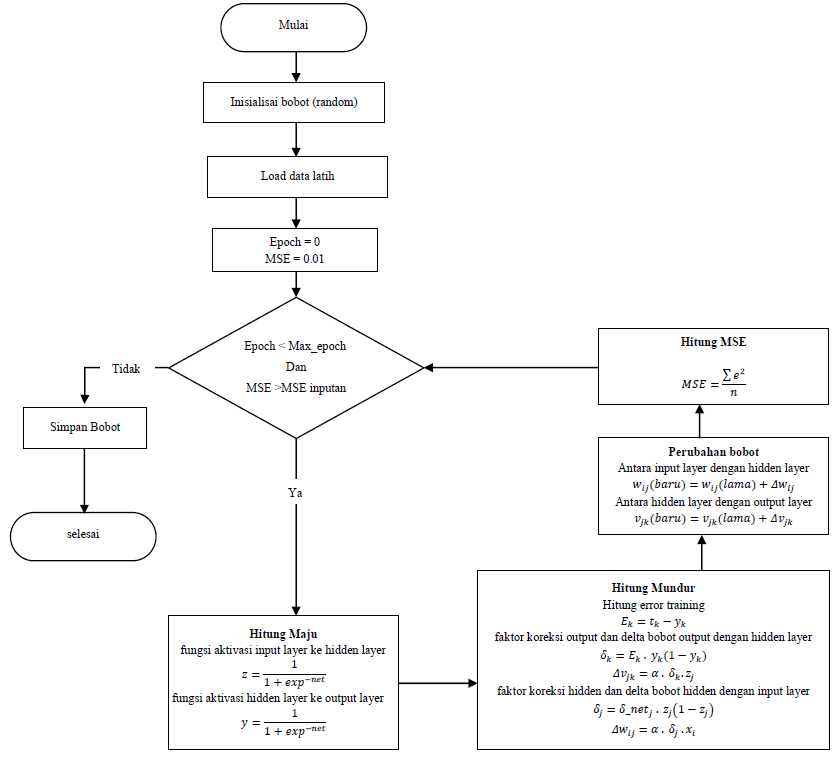
*Flowchart* ganti *password* merupakan diagram alir yang menggambarkan proses dari mengubah *password* *Login*. Adapun gambar dari *Flowchart* ganti *password* bisa dilihat pada Gambar 4.6.



**Gambar 4.6** *Flowchart* Ganti *Password*

1. *Flowchart* Pelatihan

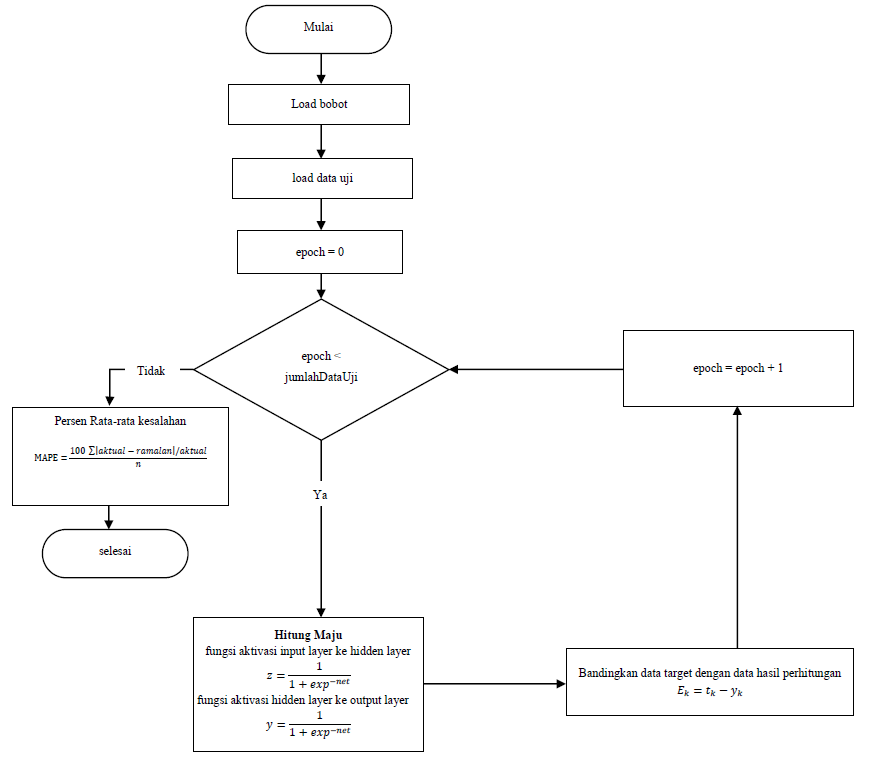
*Flowchart* pelatihan ini merupakan diagram alir yang menggambarkan proses pelatihan JST yaitu dengan metode *backpropagation*. Adapun gambar dari *Flowchart* pelatihan inibisa dilihat pada Gambar 4.7.



**Gambar 4.7** *Flowchart* Pelatihan

1. *Flowchart* Pengujian

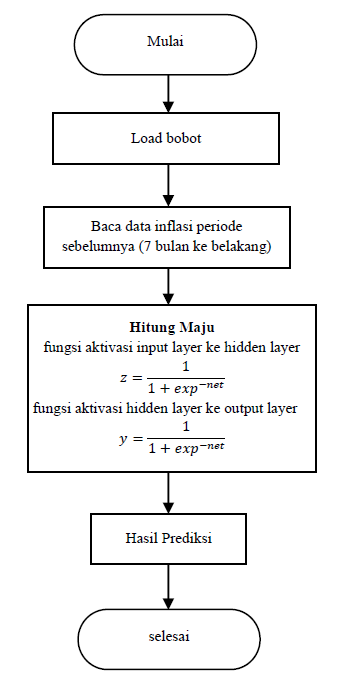
*Flowchart* Pengujian merupakan diagram alir yang menggambarkan proses pengujian JST yaitu dengan metode *backpropagation*. Adapun gambar dari *Flowchart* pelatihan inibisa dilihat pada Gambar 4.8.



**Gambar 4.8** *Flowchart* Pengujian

1. *Flowchart* Prediksi

*Flowchart* Prediksi merupakan diagram alir yang menggambarkan proses Prediksi Inflasi Kota Yogyakarta pada periode selanjutnya yaitu dengan metode *backpropagation*. Adapun gambar dari *Flowchart* pelatihan inibisa dilihat pada Gambar 4.9.



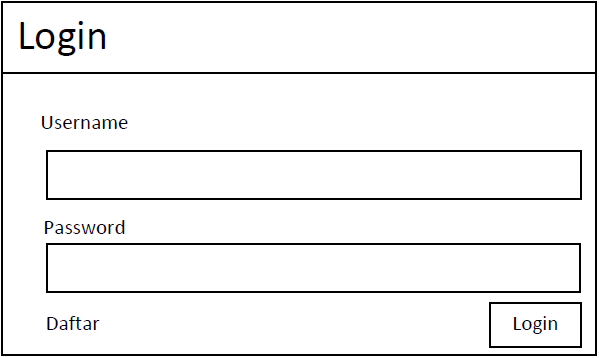
**Gambar 4.9** *Flowchart* Pengujian

**4.3 Perancangan Antar Muka Sistem**

Perancangan antar muka sistem atau *interface* digunakan untuk merancang sebuah tampilan dari sistem yang dibangun yaitu dalam hal ini adalah sistem prediksi inflasi di Kota Yogyakarta. Dimana terdapat tampilan-tampilan yang diperlukan seperti tampilan login, pengisian data latih ataupun data uji, tampilan prediksi dan lain-lain.

**4.3.1** **Perancangan Tampilan Login**

Tampilan *login* digunakan untuk melakukan *login* sebelum memasuki sistem yaitu di sini adalah untuk membatasi hak akses amin dengan pengguna lainnya. Adapun tampilan *login* bisa dilihat pada Gambar 4.10.

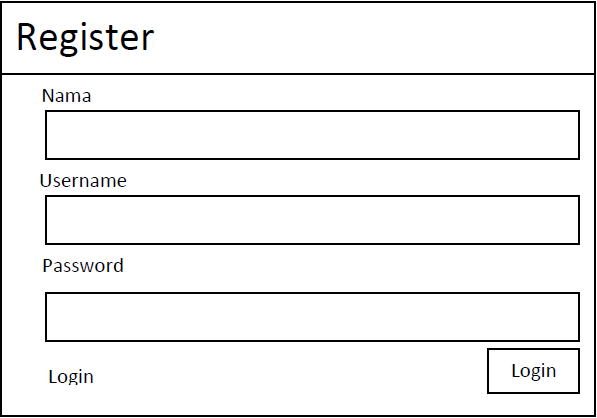


**Gambar 4.10** Tampilan *Login*

Pada tampilan *login* di atas terdapat buat aku baru, yaitu, untuk melakukan penambahan akun admin apabila ada user yang ingin menjadi seorang admin.

**4.3.2 Perancangan Tampilan Buat Akun Baru**

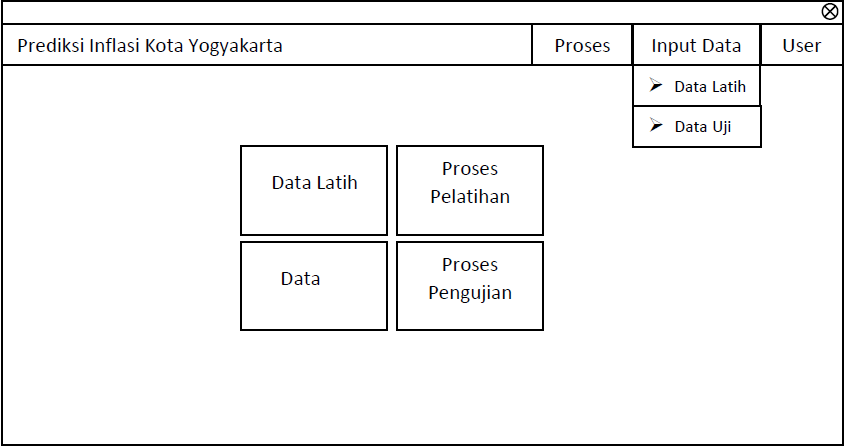
Tampilan buat aku baru merupakan tampilan yang digunakan untuk menambah pengguna dalam sisi admin. Adapun tampilannya pada Gambar 4.11.



**Gambar 4.11** Tampilan Buat Akun Baru

**4.3.3 Perancangan Tampilan Home Admin**

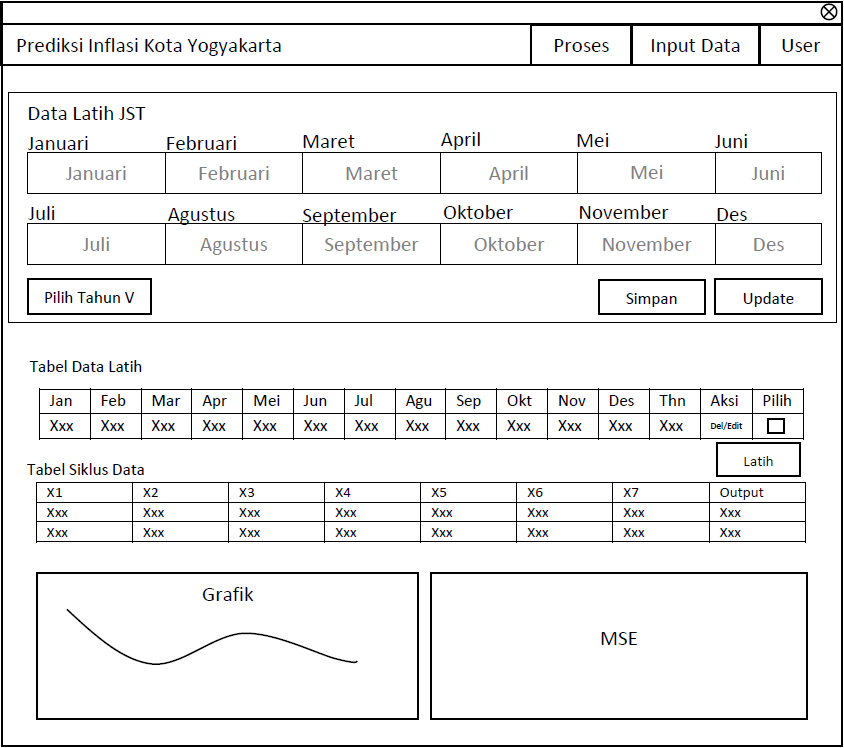
Tampilan *Home* Admin merupakan tampilan awal setelah dilakukannya login oleh seorang admin, di dalam tampilan ini terdapat beberapa fitur yaitu User yang berisi *logout*, ganti password, lalu terdapat Data yang berisi data latih dan data uji, lalu terdapat fitur proses yang dimana didalamnya terdapat proses pengujian JST. Adapun tampilan dari *home* admin ini bisa dilihat pada Gambar 4.12.



**Gambar 4.12** Tampilan *Home* Admin

**4.3.4 Perancangan Tampilan Data Latih**

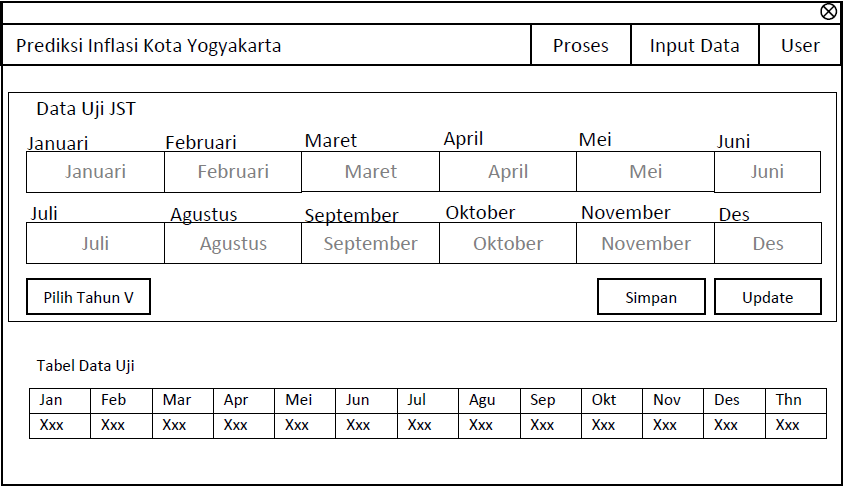
Tampilan Data Latih ini merupakan tempat melakukan pengisian data latih, perubahan data, ataupun penghapusan data bisa dilakukan pada halaman ini yang sudah terhubung ke *database* dan apabila ingin melakukan pelatihan pada tampilan ini sudah disediakan fungsi untuk melakukan pelatihan JST. Adapun gambar dari tampilan data latih ini bisa dilihat pada Gambar 4.13.



**Gambar 4.14** Tampilan Data Latih

**4.3.5** **Perancangan Tampilan Data Uji**

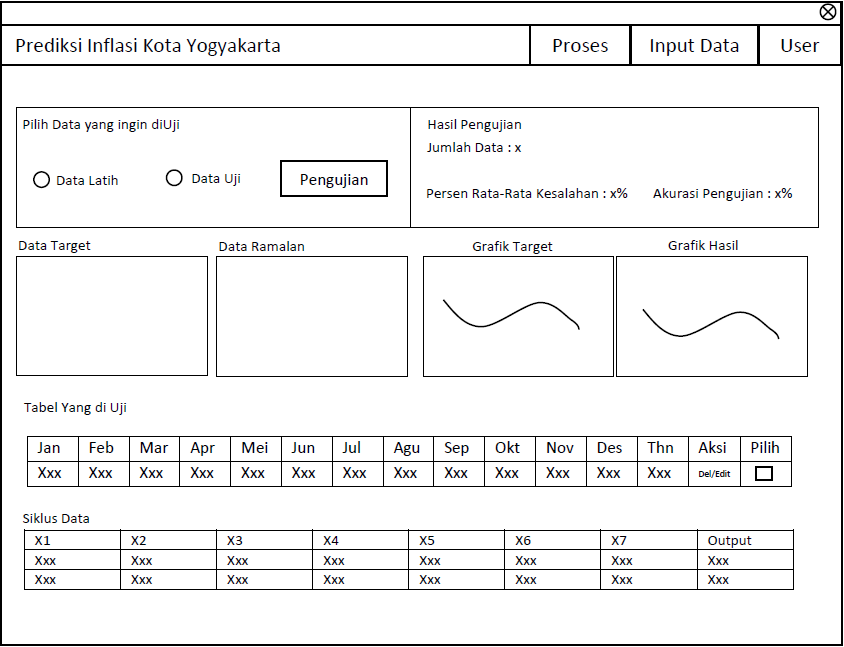
Perancangan Tampilan Data Uji ini merupakan tampilan tempat melakuakan pengisian data uji, pengubahan data uji dan penghapusan data uji. Pada tampilan ini bisa dikatakan hampir sama dengan tampilan data latih namun untuk tampilan data uji ini tidak terdapat fungsi untuk melakukan pelatihan data ataupun pengujian data. Adapun gambar dari perancangan tampilan data uji bisa dilihat pada Gambar 4.15.



**Gambar 4.15** Tampilan Data Uji

**4.3.6** **Tampilan Proses Pengujian**

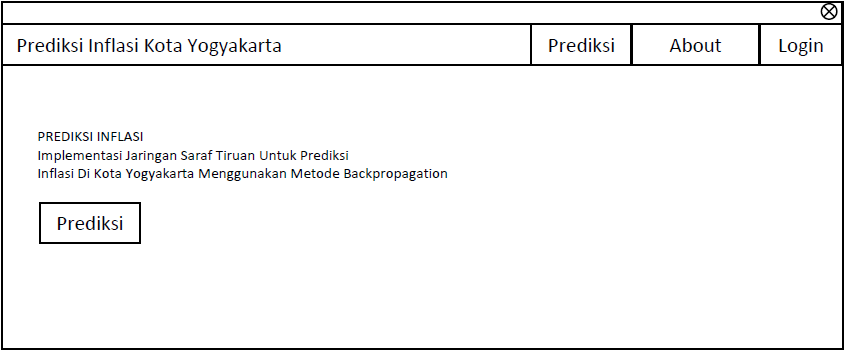
Tampilan Proses Pengujian ini merupakan tampilan yang digunakan untuk melakukan pengujian dari JST yaitu, bisa dilakukan pengujian data latih maupun data uji, di dalam tampilan ini hasil pengujian akan di tampilkan dalam bentuk grafik yaitu, perbandingan antara data asli dengan data hasil perhitungan dari JST. Adapun gambar untuk tampilan ini bisa dilihat pada Gambar 4.16.



**Gambar 4.16** Gambar Tampilan Proses Pengujian

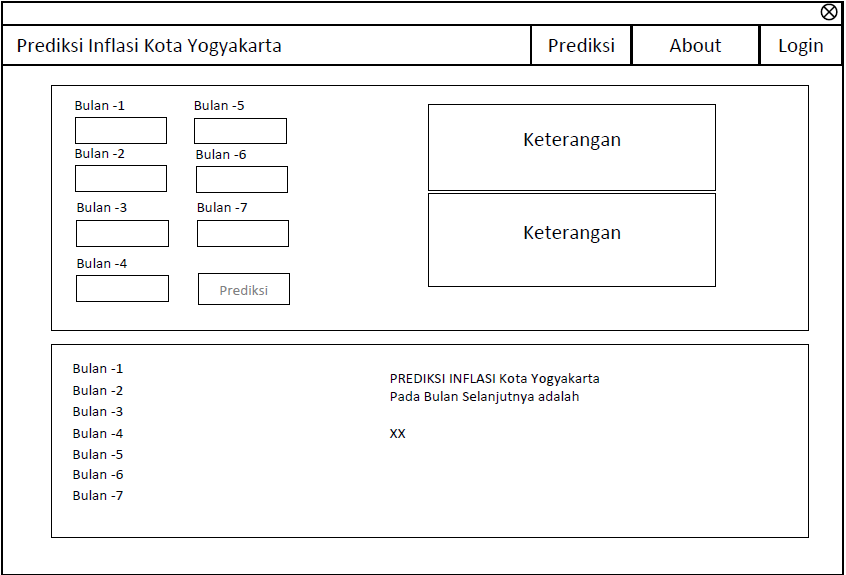
**4.3.7 Perancangan Tampilan User**

Tampilan *user* merupakan tampilan yang digunakan oleh pengguna yang bukan admin, dimana pada tampilan ini hanya memiliki sedikit fitur yaitu diantaranya adalah fitur untuk melihat tentang sistem dan fitur untuk melakukan prediksi inflasi. Adapun gambar dari tampilan *user* ini bisa dilihat pada Gambar 4.17.



**Gambar 4.17** Tampilan User

**4.3.8 Perancangan Tampilan Prediksi**

Tampilan Prediksi ini merupakan tampilan yang menyediakan proses prediksi inflasi kepada *user* yang ingin melakukan prediksi inflasi di Kota Yogyakarta. Bulan yang diinputkan mulai dari bulan -1 yang merupakan bulan terakhir sebelum bulan yang diprediksi, dan diikuti bulan-bulan sebelumnya.Adapun gambar untuk tampilan prediksi ini bisa dilihat pada gambar 4.18.

**Gambar 4.18** Tampilan Prediksi

# BAB V

# IMPLEMENTASI SISTEM

**5.1 Implementasi**

Tahap implementasi ini adalah tahap yang dilakukan setelah semua perancangan sistem dilakukan dan sudah memenuhi kriteria dan logika yang dapat dipahami dan siap diimplementasikan atau diterapkan. Penerapan atau implementasi sistem yang dilakukan pada kasus inflasi di Kota Yogyakarta dengan menggunakan metode *backpropagation* ini digunakan beberapa teknologi yang mendukung dalam pembangunan sistem prediksi inflasi di Kota Yogyakarta menggunakan metode *backpropagation* yaitu, diantaranya adalah penggunaan bahasa pemrograman Node Js sebagai *backend* dan Angular Js sebagai *front And* serta *database* yang digunakan adalah MongoDB. Adapun penggunaan perangkat keras maupun perangkat lunak dan tampilan-tampilan dari aplikasi selain bahasa pemrograman yang digunakan dalam tahap implementasi ini juga akan dijelaskan dengan singkat.

**5.2 Perangkat Keras (*Hardware*)**

Penggunaan perangkat keras untuk mendukung pengimplementasian dalam membangun sistem jaringan saraf tiruan untuk prediksi inflasi di Kota Yogyakarta menggunakan metode *backpropagation* ini digunakan spesifikasi prangkat keras sebagai berikug :

1. Laptop ASUS
2. Monitor 14” (inch)
3. Processor Intel Core I3
4. RAM 4 GB
5. Hardisk 500 GB

**5.3 Perangkat Lunak (*Software*)**

Penggunaan Perangkat lunak untuk mendukung pengimplementasian dalam membangun sistem jaringan saraf tiruan untuk prediksi inflasi di Kota Yogyakarta menggunakan metode *backpropagation* ini adalah :

1. Sistem Operasi Linux Ubuntu 16.04
2. Node Js
3. MongoDB
4. AngularJs
5. Sublime Text
6. Web Browser Firefox atau Google Chrome

**5.4 Implementasi Web**

**5.4.1 Halaman Awal**

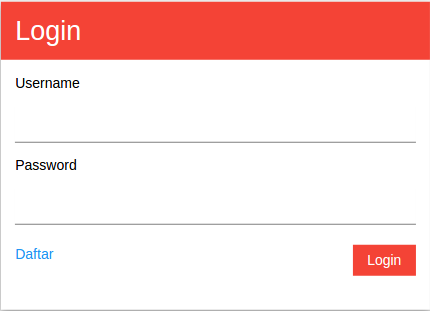
Halaman Awal akan menampilkan fitur-fitur yang bisa diakses oleh *public* *user* yaitu fitur prediksi, *about*, dan *Logiin Admin*, adapun tampilan dari halaman awal ini bisa dilihat pada Gambar



**Gambar 5.1** Tampilan *Homepage*

**5.4.2 Halaman Login Admin**

Halaman Login Admin ini merupakan halaman yang digunakan untuk membatasi hak akses pengguna, dan pengguna yang boleh memasuki halaman admin melalui login admin ini adalah pengguna yang memiliki hak akses sebagai admin, adapun fitur yang tersedia adalah tombol login, Daftar untuk mendaftar akun baru serta *username* dan *password*. Adapun tampilan dari halaman *login* ini bisa dilihat pada Gambar 5.2.



**Gambar 5.2** Tampilan Halaman Login

passport.serializeUser(function(user, done){

done(null, user);

});

passport.deserializeUser(function(user, done){

User.findById(user.\_id, function(err, user){

done(err, user);

});

});

passport.use(new LocalStrategy(

function(username, password, done){

User.findOne({username: username, password: password}, function(err, user){

if(err){

return done(err);

}

if(!user){

return done(null, false);

}

return done(null, user);

console.log(user);

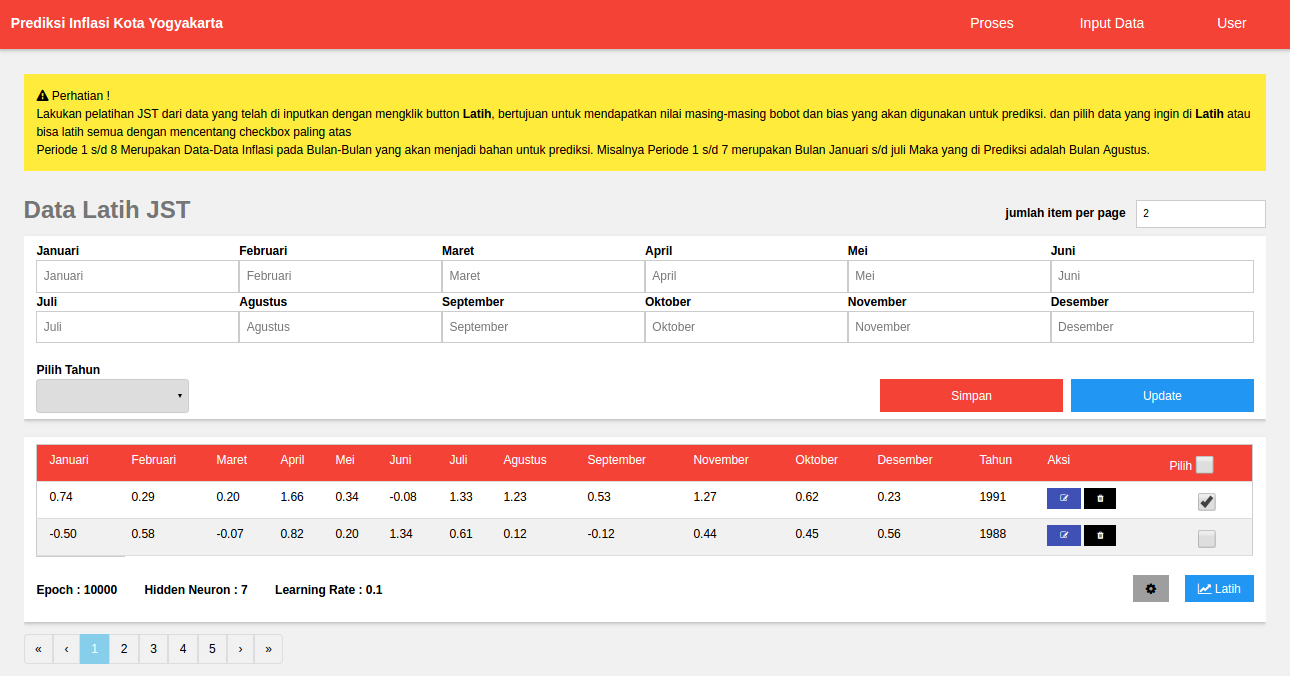
});

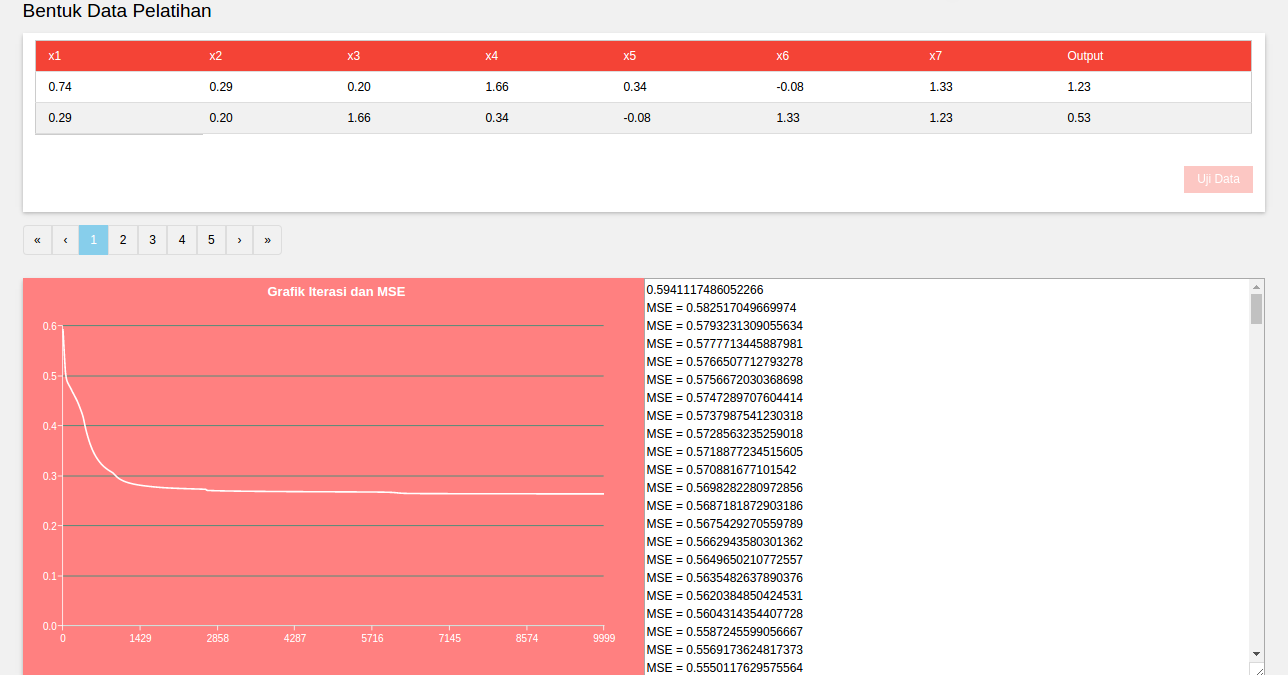
}

));

**Gambar 5.3** *Syntax* Halaman *Login*

**5.4.3 Halaman Data Latih dan Pelatihan**

Pada halaman Data Latih dan Pelatihan terdapat fitur untuk melakukan simpan, *update* dan penghapusan data serta terdapat fitur untuk melakukan Pengaturan *Epoch, Hidden* dan *Learning rate* dan data yang dilatih bisa dipilih dengan memanfaatkan fitur pilih pada tabel yang sudah ada dan pada gambar sudah diberikan contoh pelatihan data. Adapun tampilan dari halaman ini bisa dilihat pada Gambar 5.4.



**Gambar 5.4** Halaman Data Latih dan Pelatihan

app.get('/latih/train', function(req, res){

//Setting Epoch, Hidden, Learning rate

setting.find(function(err, data){

hneuron = data[0].neuron;

epoch = data[0].epoch;

lrate = data[0].lrate;

console.log(hneuron);

net = new ch\_ann.NeuralNetwork({

hiddenLayers: [hneuron],

learningRate: lrate

});

})

//Algoritma Pembuatan Pola data

Latih.find(function(err, data){

for (var i = 0; i < data.length; i++) {

dtSleksi(data[i].input);

for (var j = 1; j < 6; j++) {

datatrain.push({input:ArrInput[j], output:ArrOtput[j]});

}

}

var latih = net.train(datatrain, {

errorThresh: 0.00000000000000000001,

iterasi: epoch,

log: true,

logPeriod: 1

});

res.json(latih);

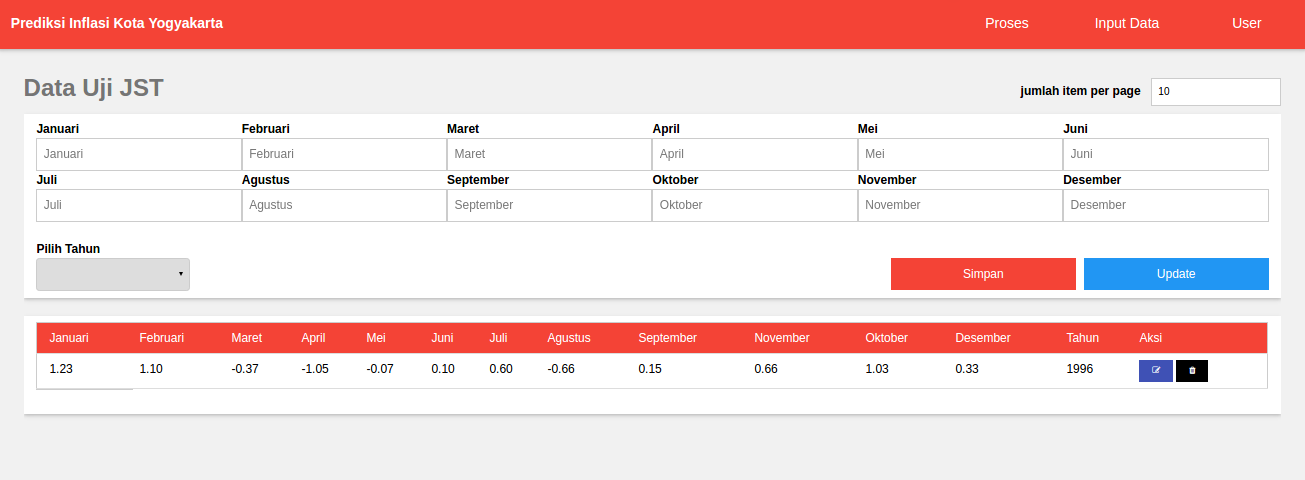
})

});

**Gambar 5.5** *Syntax* Program Pelatihan

**5.4.4 Halaman Data Uji**

Pada Halaman ini hanya tempat menyimpan data, mengubah ataupun menghapus data baru yang akan digunakan untuk melakukan pengujian pada sistem prediksi inflasi di Kota Yogyakarta. Adapun tampilannya bisa dilihat pada Gambar 5.6.



**Gambar 5.6** Halaman Data Uji

require('../data/datapengujian');

var Uji = mongoose.model('Uji');

module.exports.simpanUji = function(req, res){

var uji = new Uji(req.body);

uji.save();

res.json(req.body);

}

module.exports.tampilUji = function(req, res){

Uji.find({}).sort({tgl:-1}).exec(function(err,data){

res.json(data);

});

}

module.exports.UjiFilter = function(req, res){

Uji.find({tahun: req.params.id}).sort({tgl:-1}).exec(function(err,data){

res.json(data);

});

}

module.exports.hapusUji = function(req, res){

Uji.findById(req.params.id, function(err, data){

data.remove(function(er, doc){

res.json(doc);

})

})

}

module.exports.ambilUji = function(req, res){

Uji.findById(req.params.id, function(err, data){

res.json(data);

})

}

module.exports.ubahUji = function(req, res){

Uji.findById(req.params.id, function(err, data){

data.input = req.body.input;

data.output = req.body.output;

data.save();

res.json(data)

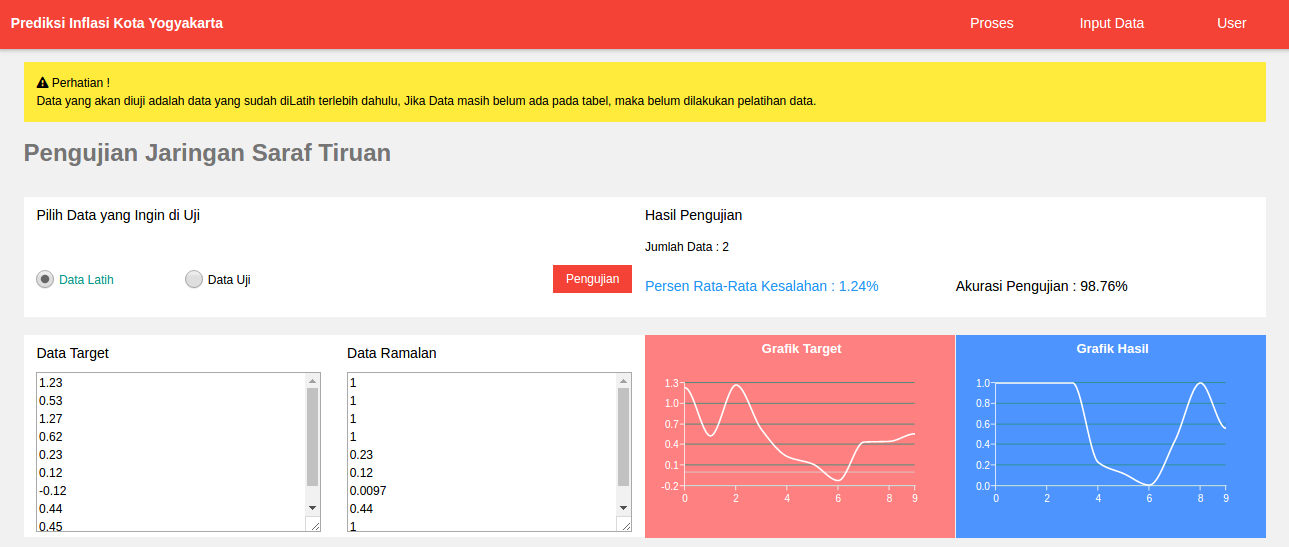
})

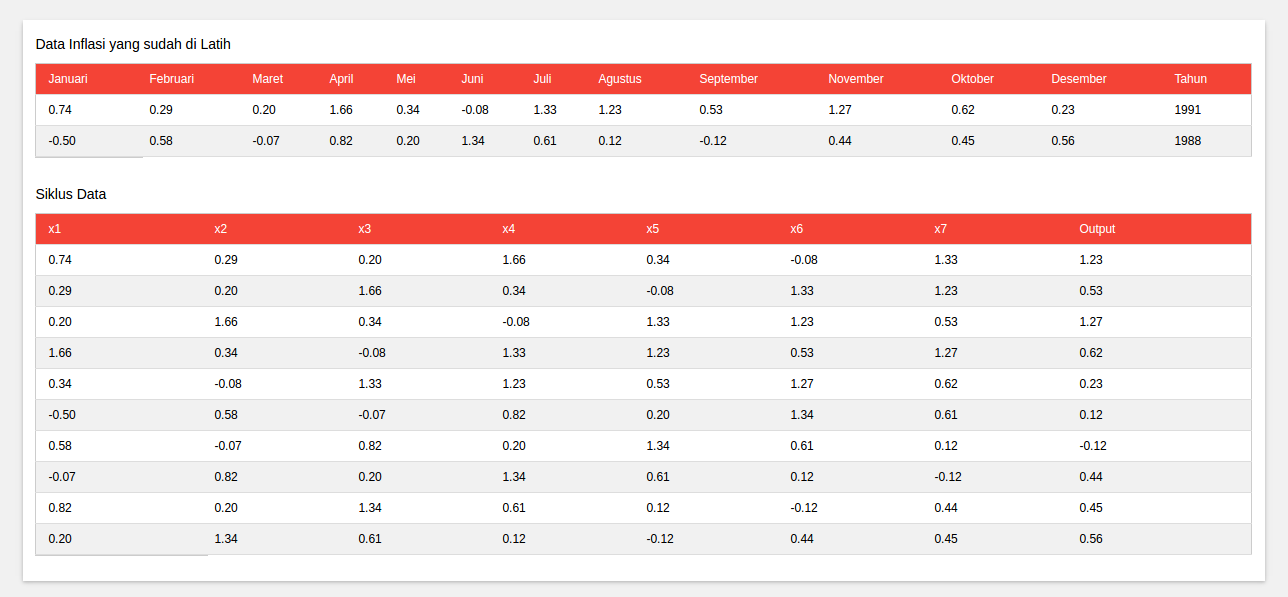
}

**Gambar 5.7** *Syntax* CRUD Data Uji

**5.4.5 Halaman Pengujian JST**

Halaman pengujian JST ini memiliki beberapa fitur yaitu tombol pengujian dan pemilihan data yang mau diujikan, ada dua jenis data yang bisa diuji, yaitu, data yang sudah dilatih dan data baru dari *database* uji. Pada halaman ini apabila dilakukan pengujian akan terlihat data target dan data hasil ramalan serta akurasi dan persen rata-rata kesalahan yang didapatkan oleh JST. Adapun tampilan dari halaman pengujian ini bisa dilihat pada Gambar 5.8.



****

**Gambar 5.8** Halaman Pengujian JST

var dtuji = [];

var akurasiUji = 0;

app.get('/latih/uji/:id', function(req, res){

dtuji = [];

var korError = 0;

var aktual = 0;

var ramalan = 0;

Latih.find({kondisi: req.params.id}, function(err, data){

for (var i = 0; i < data.length; i++) {

dtSleksi(data[i].input);

for (var j = 1; j < 6; j++) {

dtuji.push({input:ArrInput[j], output:ArrOtput[j]});

}

}

sumOutput = 0;

sumTarget = 0;

var isikan = [];

for (var i=0; i<dtuji.length; i++){

sumTarget += dtuji[i].output[0];

var oput = net.run(dtuji[i].input);

sumOutput += oput[0];

isikan.push((oput[0]).toPrecision(2));

aktual = dtuji[i].output[0];

ramalan = (oput[0]).toPrecision(2);

korError = korError + ((100 \* (aktual - ramalan) / aktual)/dtuji.length);

}

korError = korError/dtuji.length;

akurasiUji = korError;

console.log("Kor Error : "+korError);

//Kembalikan Aturan Array

var start = isikan.length - 5;

var stop = isikan.length - 5;

var ok = 5;

var isikan1 = [];

for (var i = 0; i < isikan.length; i++) {

if (i == ok) {

stop = stop-5;

start = 0;

start = stop;

ok = ok + 5;

}

isikan1.push(Number(isikan[start])+0);

start++;

}

var latihUji = {

korerror : korError,

isikan : isikan1

}

res.json(latihUji);

isikan = [];

isikan1 = [];

});

});

sumOutput += oput[0];

isikan.push((oput[0]).toPrecision(2));

aktual = dtuji[i].output[0];

ramalan = (oput[0]).toPrecision(2);

korError = korError + ((100 \* (aktual - ramalan) / aktual)/dtuji.length);

}

korError = korError/dtuji.length;

akurasiUji = korError;

console.log("Kor Error : "+korError);

//Kembalikan Aturan Array

var start = isikan.length - 5;

var stop = isikan.length - 5;

var ok = 5;

var isikan1 = [];

for (var i = 0; i < isikan.length; i++) {

if (i == ok) {

stop = stop-5;

start = 0;

start = stop;

ok = ok + 5;

}

isikan1.push(Number(isikan[start])+0);

start++;

}

var latihUji = {

korerror : korError,

isikan : isikan1

}

res.json(latihUji);

isikan = [];

isikan1 = [];

});

});

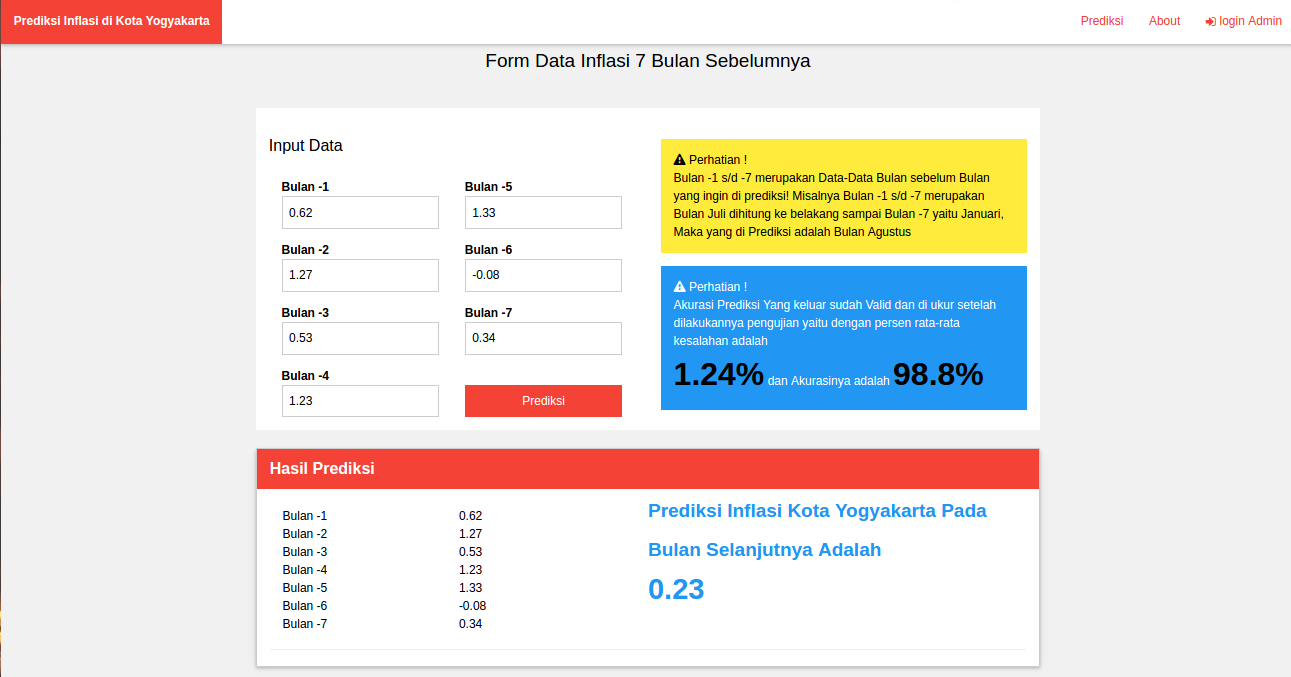
**5.4.8 Halaman Lis Admin**

Halaman ini hanya dapat diakses oleh super admin. Halaman ini menampilkan admin yang terdaftar baik admin biasa ataupun super admin. Pada halaman ini juga super admin dapat menambah dan menghapus admin, Sedangkan untuk mengedit dapat dilakukan pada akun super admin itu sendiri. Lihat pada gambar 5.10 dibawah ini

**Gambar 5.9** *Syntax* Program Pengujian JST

**5.4.6 Halaman Prediksi**

Halaman ini menampilkan keterangan-keterangan yang akan membantu pengguna dalam menggunakan sistem, karena halaman ini ditujukan untuk *public user* serta terdapat tombol prediksi yang akan melakukan proses prediksi apabila data-data sudah dimasukkan. Adapun tampilannya bisa dilihat pada Gambar 5.10.



**Gambar 5.10** Halaman Prediksi Inflasi Kota Yogyakarta

app.get('/prediksi', function(req, res){

var output = net.run(predData);

res.json(output[0]);

});

app.post('/prediksi', function(req, res){

var output = net.run(req.body);

res.json(output);

});

**Gambar 5.11** *Syntax* Prediksi Inflasi Kota Yogyakarta

**5.5 Hasil Pengujian Jaringan Saraf Tiruan**

Hasil Pengujian Jaringan Saraf Tiruan bertujuan untuk mengetahui seberapa baik sistem bisa mengenali data-data yang akan diujikan, yaitu, dalam hal ini terdapat dua tahap pengujian, diantaranya pengujian menggunakan data yang sudah dilatih dan data yang belum dilatih sama sekali atau menggunakan data baru.

**5.5.1 Pengujian Data Latih**

Adapun hasil pengujian dengan data yang sudah dilatih sebanyak 240 data yang belum dibentuk menjadi pola data bisa dilihat pada Tabel 5.1.

**5.6 Hasil Pengujian *Black Box***

Pengujian *black box* adalah pengujian yang dilakukan untuk memeriksa fungsional dari perangkat lunak. Adapun tabel pengujian *black box* pada tabel 5.6 di bawah ini.

**Tabel 5.6** Pengujaian *Black Box*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Pengujian** | **Keterangan** | **kesimpulan** |
| 1 | *homepage* | Menampilkan gambar home, menampilkan pilihan menu prediksi, login admin, dan tentang *website* | Berhasil |
| 2 | Prediksi | Menampilkan halaman prediksi | Berhasil |
| 3 | Login Admin | Menampilkan halaman Login admin | Berhasil |
| 4 | *About* | Menampilkan *About* *website* | Berhasil |
| 5 | Tombol *Login* | Masuk ke halaman Admin apabila *username* dan *password* sudah benar | Berhasil |
| 6 | TombolPengaturan | Menampilkan halaman pengaturan | Berhasil |
| 7 | Tombol Latih Data | Akan menampilkan bentuk pola Data yang dilatih dan grafik serta MSE dari pelatihan | Berhasil |
| 8 | *Checkbox* | Mencentang data | Berhasil |
| 9 | Tombol Simpan Data Latih | Data Akan tersimpan apabila data sudah terisi | Berhasil |
| 10 | Tombol Edit Pada Kolom Aksi | Mengambil data dari tabel lalu ditampilkan ke *textbox* | Berhasil |
| 11 | Tombol Hapus Pada Kolom Aksi | Menghapus data pada tabel dan *database* | Berhasil |
| 12 | Tombol *Update* | Mengubah data yang sudah dipilih setelah melakukan aksi pada tombol edit | Berhasil |
| 13 | Menu *Input Data* | Menampilkan halaman data latih dan data uji | Berhasil |
| 14 | Menu Pengujian | Menampilkan halaman pengujian | Berhasil |
| 15 | *Radio Button* Data Latih | Menampilkan data yang telah dilatih | Berhasil |
| 16 | *Radio Button* Data Uji | Menampilkan data uji | Berhasil |
| 17 | Tombol Pengujian | Menampilkan Hasil uji dan persentasi rata-rata kesalahan dan akurasi pengujian | Berhasil |
| 18 | Menu *User Logout* | Keluar dari admin dan menampilan halaman awal | Berhasil |
| 19 | Tombol Prediksi pada halaman prediksi | Menampilkan hasil prediksi inflasi kota Yogyakarta untuk bulan selanjutnya | Berhasil |

**5.7 Perbandingan Hasil Prediksi**

**Tabel 5.7** Perbandingan Hasil Prediksi

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Jumlah Tahun | Tahun  Latih | Tahun  Prediksi | Bulan  Prediksi | Data  Target | Hasil Prediksi | Selisih |
| 1 Tahun | 2009 | 2010 | Agustus | 0,43 | 0,41 | 0,02 |
| 2 Tahun | 2008-2009 | 2010 | Agustus | 0,43 | 0,096 | 0,334 |
| 3 Tahun | 2007-2009 | 2010 | Agustus | 0,43 | 0,54 | -0,11 |
| 4 Tahun | 2006-2009 | 2010 | Agustus | 0,43 | 0,13 | 0,3 |
| 5 Tahun | 2005-2009 | 2010 | Agustus | 0,43 | 0,43 | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Jumlah Tahun | Tahun Latih | Tahun Prediksi | Bulan Prediksi | Data Target | Hasil Prediksi | selisih |
| 1 Tahun | 2001 | 2002 | Agustus | 0,82 | 0,92 | -0,1 |
| 2 Tahun | 2000-2001 | 2002 | Agustus | 0,82 | 0,81 | 0,01 |
| 3 Tahun | 1999-2001 | 2002 | Agustus | 0,82 | 0,82 | 0 |
| 4 Tahun | 1998-2001 | 2002 | Agustus | 0,82 | 0,92 | -0,1 |
| 5 Tahun | 1997-2001 | 2002 | Agustus | 0,82 | 0,91 | -0,09 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Jumlah Tahun | Tahun Latih | Tahun Prediksi | Bulan Prediksi | Data Target | Hasil Prediksi | Selisih |
| 1 Tahun | 2006 | 2008 | Oktober | 0,62 | 0,56 | 0,06 |
| 2 Tahun | 2005-2006 | 2004 | Oktober | 0,54 | 0,4 | 0,14 |
| 3 Tahun | 2004-2006 | 2008 | April | 0,21 | 0,15 | 0,06 |
| 4 Tahun | 2003-2006 | 2008 | November | 0,07 | 0,002 | 0,068 |
| 5 Tahun | 2002-2006 | 2001 | Oktober | 0,67 | 0,98 | -0,31 |

Prediksi yang dilakukan dapat dilihat pada tabel-tabel di atas, dan sudah memiliki selisih. Pelatihan yang terlihat bagus adalah data yang dilatih yaitu 1 tahun dan 5 tahun.

# Bab VI

# PENUTUP

## 6.1 Kesimpulan

Kesimpulan yang didapatkan dalam sistem prediksi inflasi di Kota Yogyakarta ini adalah Sistem Jaringan Saraf Tiruan Untuk Prediksi Inflasi di Kota Yogyakarta Menggunakan Metode *Backpropagation* dapat berjalan dan Sistem dapat memprediksi tingkat inflasi pada bulan selanjutnya setelah di*input*kan data pada bulan-bulan sebelumnya. Persen kesalahan rata-rata pada pengujian dengan data latih yang berbentuk pola sebanyak 100 data sudah dilatih mencapai 0.26% dan data pengujian yang berbentuk pola sebanyak 35 data mencapai 2.93% dan data dengan pelatihan 1 tahun dan 5 tahun memiliki akurasi yang teliti pada hasil prediksi Inflasi di Kota Yogyakarta, yaitu, memiliki selisih 0.02 hasil pada bulan Agustus tahun 2010 dengan dilakukan pelatihan data setahun sebelumnya yaitu, Tahun 2009, oleh sebab itu dapat dijadikan patokan target tingkat Inflasi di Kota Yogyakarta, serta Sistem prediksi inflasi ini dapat mendorong mengurangi jumlah Inflasi di Kota Yogyakarta.

## 6.2 Saran

Sistem jaringan saraf tiruan untuk prediksi inflasi di Kota Yogyakarta dengan metode *backpropagation* ini tidak lepas dari berbagai kekurangan, untuk itu ada beberapa saran untuk pengembangan selanjutnya yaitu:

1. Diharapkan penentuan pola siklus data *time series* yang lebih baik, agar pelatihan maupun pengujian serta hasil prediksi lebih baik.
2. Diharapkan penentuan arsitektur jaringan dan normalisasi data bisa ditentukan dengan yang lebih baik, supaya hasil prediksi tidak jauh berbeda dengan data asli Inflasi.
3. Diharapkan penggunaan fungsi aktivasi kedepannya bisa memilih yang lebih baik.

# DAFTAR PUSTAKA

Amrin., (2014), *Peramalan Tingkat Inflasi Indonesia Menggunakan Neural Network Backpropagation Berbasis Metode Time Series,* Jurnal Techno Nusa Mandiri, Vol. XI No. 2, September 2014.

AndriJasa, M.F dan Mistianingsih., (2010), *Penerapan Jaringan Syaraf Tiruan Untuk Jumlah Penggangguran di Provinsi Kalimantan Timur dengan Menggunakan Algoritma Pembelajaran Backpropagation,* Jurnal Informatika Mulawarman, Vol 5 No. 1 Februari 2010 50.

Dillak, R.Y, Bintiri, M.G dan Harjoko A., (2013), *Sistem Deteksi Penyakit Menggunakan Jaringan Saraf Tiruan*, Yogyakarta: Graha Ilmu.

Heizer, J dan Render, B., (2005), *Opration Management,* Jakarta: Salemba Empat.

Hermawan, Arief., (2006), *Jaringan Saraf Tiruan, Teori dan Aplikasi*, Yogyakarta: Andi Offset.

Julisman, Agung., (2014), *Sistem Aplikasi Travel Dengan Angularjs dan Codeigniter*, Yogyakarta: Lokomedia.

Kadir, Abdul., (2012), *Algoritma dan Pemrograman Menggunakan C dan C++,* Yogyakarta: Andi Offset.

Susanti, Nanik., (2014), *Penerapan Model Neural NEtwork Backpropagation Untuk Prediksi Harga Ayam,* Prosiding SNATIF Ke-1, ISBN: 978-602-1180-04-4.

Suwardjono., (2008), *Teori Akuntansi Perekayasaan Pelaporan Keurangan,* Edisi Ketiga, BPFE, Yogyakarta.

# LAMPIRAN

**Data Mentah Inflasi Kota Yogyakarta Tahun 1979 – 2016**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Thn/Bln** | **Jan** | **Feb** | **Mar** | **Apr** | **Mei** | **Jun** | **Jul** | **Agu** | **Sep** | **Okt** | **Nov** | **Des** |
| **1979** | 1,66 | -0,56 | 6,19 | 1,25 | 3,53 | 3,8 | 2,8 | 1,32 | 0,77 | 1,11 | 1,02 | 0,52 |
| **1980** | 1,3 | 0,72 | 2,7 | 1,89 | 4,73 | 1,02 | 1,54 | 0,65 | 0,12 | 2,44 | 2,3 | 0,52 |
| **1981** | 1,75 | 1,2 | 0,15 | 0,3 | 0,82 | 0,69 | 1,88 | 0,89 | -0,66 | 0,73 | 0,08 | 1,07 |
| **1982** | 3,53 | 1,03 | 0,36 | -0,3 | -0,53 | -0,08 | 2 | 0,02 | 1,83 | 1,05 | 0,67 | 0,39 |
| **1983** | 3,02 | 0,72 | -0,48 | 2,76 | 1,47 | 2,43 | -0,01 | -0,42 | 1,18 | 0,18 | 0,76 | 1,06 |
| **1984** | 3,27 | 1,39 | 0,66 | 0,07 | 0,97 | -0,41 | 0,68 | -0,29 | 0,02 | -0,02 | -0,26 | 1,84 |
| **1985** | 0,9 | -0,66 | 0,42 | 3,28 | -0,26 | 0,25 | 0,55 | -0,22 | -0,21 | 0,19 | 1,08 | 0,44 |
| **1986** | 1,14 | 1,1 | -0,39 | -0,19 | 1,19 | 0,51 | -0,27 | -0,02 | 1,38 | 2,74 | 0,63 | 1,41 |
| **1987** | 0,08 | 0,78 | 0,04 | 1,06 | 1,37 | 0,41 | 0,22 | 0,25 | 1,05 | 2,35 | 1,92 | 0,84 |
| **1988** | -0,5 | 0,58 | -0,07 | 0,82 | 0,2 | 1,34 | 0,61 | 0,12 | -0,12 | 0,44 | 0,45 | 0,56 |
| **1989** | -0,62 | 1,52 | 0,85 | 0,34 | 1,13 | 0,18 | 0,44 | -1,05 | 1,04 | 0,67 | 0,96 | -0,25 |
| **1990** | 0,44 | 0,96 | 0,13 | 1,56 | 0,31 | 1,2 | 1,01 | 1,32 | 0,83 | 2,29 | 0,44 | 0,24 |
| **1991** | 0,74 | 0,29 | 0,2 | 1,66 | 0,34 | -0,08 | 1,33 | 1,23 | 0,53 | 1,27 | 0,62 | 0,23 |
| **1992** | 1,19 | 0,61 | 0,67 | 0,17 | -0,34 | 0,62 | -0,02 | -0,53 | 0,35 | 0,02 | 0,7 | 1,34 |
| **1993** | 2,94 | 2,05 | 1,1 | -0,17 | 0,32 | 0,41 | 0,62 | -0,06 | 0,17 | 0,48 | 1,21 | 0,94 |
| **1994** | 0,47 | 1,56 | 1,29 | -0,72 | 0,53 | -0,31 | 1,46 | 1,29 | 0,49 | 0,87 | 1,38 | 0,24 |
| **1995** | 1,32 | 1,36 | 1,59 | 1,64 | 0,08 | -0,41 | 1,21 | 0,73 | 0,64 | 0,34 | 0,3 | 0,84 |
| **1996** | 1,23 | 1,1 | -0,37 | -1,05 | -0,07 | 0,1 | 0,6 | -0,66 | 0,15 | 0,66 | 1,03 | 0,33 |
| **1997** | 0,17 | 0,88 | 0,53 | 0,01 | -0,4 | -0,11 | 0,95 | 1,24 | 1,76 | 1,67 | 2,81 | 3,21 |
| **1998** | 6,23 | 14,58 | 5,38 | 4,11 | 3,57 | 4,75 | 8,6 | 7,53 | 4,43 | -0,14 | -0,24 | 0,83 |
| **1999** | 2,46 | 0,31 | 0,28 | -0,51 | -0,14 | -0,46 | -0,61 | -0,1 | -0,39 | -0,05 | 0,47 | 2,51 |
| **2000** | 0,78 | -0,34 | 0,09 | 0,3 | 0,37 | 0,65 | 1,3 | 0,36 | 0,3 | 0,72 | 1,2 | 1,37 |
| **2001** | -0,08 | 1,31 | 1,26 | 0,48 | 0,9 | 1,16 | 1,75 | 0,32 | 1,08 | 0,67 | 1,49 | 1,57 |
| **2002** | 1,44 | 0,75 | 0,33 | -0,25 | 1,53 | 0,4 | 1,38 | 0,82 | 1,56 | 0,51 | 1,68 | 1,27 |
| **2003** | 0,88 | 0,1 | -0,02 | 0,22 | 0,11 | 0,67 | 1,06 | 0,06 | 0,53 | 0,75 | 0,67 | 0,57 |
| **2004** | 0,6 | -0,2 | 0,44 | 0,75 | 0,86 | 0,31 | 0,55 | 0,54 | 0,26 | 0,5 | 1,08 | 1,05 |
| **2005** | 1,2 | 0,14 | 0,95 | 0,3 | 0,47 | 0,66 | 1,09 | 0,87 | 1,06 | 6,53 | 1,4 | -0,45 |
| **2006** | 2,5 | 0,21 | -0,17 | 0,64 | 1,05 | 0,83 | 0,6 | 0,84 | 1,07 | 0,79 | 0,43 | 1,17 |
| **2007** | 0,89 | 0,54 | 0,42 | 0,02 | 0,07 | 0,08 | 0,77 | 1,4 | 0,96 | 1,09 | 1,01 | 0,47 |
| **2008** | 1,25 | 1,01 | 0,56 | 0,21 | 1,08 | 2,51 | 1,31 | 0,67 | 1,15 | 0,62 | 0,07 | -0,11 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **2009** | 0,09 | 0,32 | 0,18 | -0,34 | 0,27 | 0,18 | 0,32 | 0,77 | 0,8 | -0,03 | 0,09 | 0,24 |
| **2010** | 0,57 | 0,31 | 0,13 | 0,25 | 0,14 | 1,26 | 1,4 | 0,43 | 1,06 | 0,28 | 0,62 | 0,72 |
| **2011** | 0,84 | 0,1 | 0,21 | -0,28 | 0,13 | 0,26 | 0,9 | 0,63 | 0,19 | 0,04 | 0,33 | 0,48 |
| **2012** | 0,25 | 0,1 | 0,36 | 0,11 | 0,05 | 0,75 | 0,76 | 0,42 | 0,19 | 0,38 | 0,2 | 0,66 |
| **2013** | 0,96 | 0,93 | 0,79 | -0,3 | -0,29 | 0,84 | 2,58 | 0,87 | -0,24 | 0,61 | 0,2 | 0,17 |
| **2014** | 1,05 | 0,07 | 0,14 | 0,07 | 0,05 | 0,43 | 0,85 | 0,09 | 0,49 | 0,28 | 1,13 | 1,76 |
| **2015** | 0,13 | -0,4 | 0,15 | 0,38 | 0,36 | 0,35 | 0,63 | 0,33 | 0,04 | 0,01 | 0,13 | 0,96 |
| **2016** | 0,53 | -0,09 | 0,02 | -0,16 | 0,8 |  |  |  |  |  |  |  |

***Source Code* Program**

***Backpropagation***

var \_ = require("underscore"),

lookup = require("./lookup");

var NeuralNetwork = function(options) {

options = options || {};

this.learningRate = options.learningRate || 0.3;

this.momentum = options.momentum || 0.1;

this.hiddenSizes = options.hiddenLayers;

}

NeuralNetwork.prototype = {

initialize: function(sizes) {

this.sizes = sizes;

this.outputLayer = this.sizes.length - 1;

this.bias = [];

this.bobot = [];

this.output = [];

this.delta = [];

this.ganti = [];

this.errors = [];

for (var layer = 0; layer <= this.outputLayer; layer++) {

var size = this.sizes[layer];

this.delta[layer] = setnol(size);

this.errors[layer] = setnol(size);

this.output[layer] = setnol(size);

if (layer > 0) {

this.bias[layer] = setrandom(size);

this.bobot[layer] = new Array(size);

this.ganti[layer] = new Array(size);

for (var node = 0; node < size; node++) {

var prevSize = this.sizes[layer - 1];

this.bobot[layer][node] = setrandom(prevSize);

this.ganti[layer][node] = setnol(prevSize);

}

}

}

},

dtBobot: function(dtbobot, dtbias, dtoutput, dtsize){

this.sizes = dtsize;

this.outputlayer = this.sizes.length - 1;

this.bias = dtbias;

this.bobot = dtbobot;

this.output = dtoutput;

},

run: function(input) { //untuk testing

//console.log(this.bias);

//console.log("bobot : " + this.bobot + " bias : " + this.bias + " output : " + this.output + " size : " + this.size + " outputlayer : " + this.outputLayer);

if (this.inputLookup) {

input = lookup.toArray(this.inputLookup, input);

}

var output = this.runInput(input);

if (this.outputLookup) {

output = lookup.toHash(this.outputLookup, output);

}

return output;

},

runInput: function(input) {

this.output[0] = input; // set output state of input layer

for (var layer = 1; layer <= this.outputLayer; layer++) {

for (var node = 0; node < this.sizes[layer]; node++) {

var bobot = this.bobot[layer][node];

//menghitung input di hiden layer dan output layer

var sum = this.bias[layer][node];

for (var k = 0; k < bobot.length; k++) {

sum += bobot[k] \* input[k];

}

this.output[layer][node] = 1 / (1 + Math.exp(-sum));

}

var output = input = this.output[layer];

}

return output;

},

train: function(data, options) {

data = this.formatData(data);

options = options || {};

var iterasi = options.iterasi || 20000;

var errorThresh = options.errorThresh || 0.005;

var log = options.log || false;

var logPeriod = options.logPeriod || 10;

var callback = options.callback;

var callbackPeriod = options.callbackPeriod || 10;

var inputSize = data[0].input.length;

var outputSize = data[0].output.length;

var hiddenSizes = this.hiddenSizes;

if (!hiddenSizes) {

hiddenSizes = [Math.max(3, Math.floor(inputSize / 2))]; //floor untuk membulatkan bilangan ke integer terbawah

}

var sizes = \_([inputSize, hiddenSizes, outputSize]).flatten();

this.initialize(sizes);

var error = 1;

var isiChart=[];

for (var i = 0; i < iterasi && error > errorThresh; i++) {

var sum = 0;

for (var j = 0; j < data.length; j++) {

var err = this.trainPattern(data[j].input, data[j].output);

sum += err;

}

error = sum / data.length;

if (log && (i % logPeriod == 0)) {

// console.log(i);

isiChart.push(error);

}

if (callback && (i % callbackPeriod == 0)) {

callback({ error: error, iterasi: i });

}

}

console.log(this.sizes);

return {

bobot:this.bobot,

bias:this.bias,

output:this.output,

sizes:this.sizes,

outputlayer:this.outputlayer,

isiChart:isiChart,

error: error,

iterasi: i

};

},

trainPattern : function(input, target) {

// feed forwardnya yaitu dari input ke hiden layer kemudian layer output

this.runInput(input);

// back propagation .. kembali ke belakang,dan juga update bobot

this.hitungDelta(target);

this.updateBobot();

var error = mse(this.errors[this.outputLayer]);

return error; //mengembalikan nilai error yang bisa di tampilkan

},

hitungDelta: function(target) {

for (var layer = this.outputLayer; layer >= 0; layer--) {

for (var node = 0; node < this.sizes[layer]; node++) {

var output = this.output[layer][node];

var error = 0;

if (layer == this.outputLayer) {

error = target[node] - output;

}

else {

var delta = this.delta[layer + 1];

for (var k = 0; k < delta.length; k++) {

error += delta[k] \* this.bobot[layer + 1][k][node];

}

}

this.errors[layer][node] = error;

this.delta[layer][node] = error \* output \* (1 - output);

}

}

},

updateBobot: function() {

for (var layer = 1; layer <= this.outputLayer; layer++) {

var incoming = this.output[layer - 1];

for (var node = 0; node < this.sizes[layer]; node++) {

var delta = this.delta[layer][node];

for (var k = 0; k < incoming.length; k++) {

var change = this.ganti[layer][node][k];

change = (this.learningRate \* delta \* incoming[k])

+ (this.momentum \* change);

this.ganti[layer][node][k] = change;

this.bobot[layer][node][k] += change;

}

this.bias[layer][node] += this.learningRate \* delta;

}

}

},

formatData: function(data) {

// turn sparse hash input into arrays with 0s as filler

if (!\_(data[0].input).isArray()) {

if (!this.inputLookup) {

this.inputLookup = lookup.buildLookup(\_(data).pluck("input"));

}

data = data.map(function(datum) {

var array = lookup.toArray(this.inputLookup, datum.input)

return \_(\_(datum).clone()).extend({ input: array });

}, this);

}

if (!\_(data[0].output).isArray()) {

if (!this.outputLookup) {

this.outputLookup = lookup.buildLookup(\_(data).pluck("output"));

}

data = data.map(function(datum) {

var array = lookup.toArray(this.outputLookup, datum.output);

return \_(\_(datum).clone()).extend({ output: array });

}, this);

}

return data;

}

}

//fungsi untuk random bilangan

function randomWeight() {

return Math.random() \* 0.4 - 0.2;

}

//fungsi untuk set bilangan jadi 0 bila kurang dari size

function setnol(size) {

var array = new Array(size);

for (var i = 0; i < size; i++) {

array[i] = 0;

}

return array;

}

//funsi untuk set size jadi random weight

function setrandom(size) {

var array = new Array(size);

for (var i = 0; i < size; i++) {

array[i] = randomWeight();

}

return array;

}

//Jika MSE sekarang lebih kecil dari MSE sebelumnya, maka laju pembelajaran dinaikkan.

function mse(errors) {

// mean squared error

var sum = 0;

for (var i = 0; i < errors.length; i++) {

sum += Math.pow(errors[i], 2);

}

return sum / errors.length;

}

exports.NeuralNetwork = NeuralNetwork;