**LAPORAN PENELITIAN**

**PREDIKSI DAN PERAMALAN SUKU BUNGA BANK INDONESIA (BI Rate) DENGAN JARINGAN SYARAF TIRUAN BACKPROPAGATION**



**DIAJUKAN OLEH :**

**RINI SOVIA 1005047601 (Ketua)**

**Musli Yanto 1007078901 (Anggota)**

**Rizki Fernando (Anggota)**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**UNIVERSITAS PUTRA INDONESIA YPTK**

**PADANG**

**2018**

**Ringkasan**

BI *Rate* merupakan suku bunga kebijakan yang memegang peranan dalam mengarahkan pergerakan ekonomi nasional. Prediksi tingkat suku bunga bank dapat dilakukan dengan berbagai teknik dan metode, salah satunya dengan menggunakan algoritma jaringan saraf tiruan *backpropagation*. Penelitian ini menggunakan metode Jaringan Syaraf Tiruan (JST) dengan faktor terkait yaitu *Kurs Dollar*, Jumlah Uang Beredar , Inflansi, dan *IHSG*. Data inilah yang akan dilatih dan diuji dengan menggunakan software *Matlab* dengan target atau keluaran berupa tingkat suku bunga naik, tetap atau turun. Penerapan metode JST untuk memprediksi dan meramalkan BI *Rate* menggunakan algoritma *backpropagation*. Arsitektur jaringan syaraf tiruan yang digunakan yaitu 4 input layer, 2 hidden layer , dan 1 output layer. Dengan pembuatan aplikasi Jaringan Syaraf Tiruan ini dapat bermanfaat bagi kita dalam menentukan BI *Rate* yang ditampilkan dalam bentuk website menggunakan pemrograman *PHP* dengan database *MySQL*.

**Kata Kunci : Jaringan Syaraf Tiruan, *Backpropagation,* Prediksi, *BI Rate*, *PHP, MySql***

**KATA PENGANTAR**

Segala puji dan syukur kepada sumber-sumber suara hati yang bersifat mulia. Sumber ilmu pengetahuan, Sumber segala kebenaran, Sang maha cahaya, Sang pembuka wawasan, Penabur cahaya ilham, pilar nalar kebenaran dan kebaikan yang terindah, Sang kekasih tercinta yang tak terbatas pencahayaan cinta-Nya bagi umat, Allah Subhanahu wa Ta’ ala, yang karna-Nya penulis dapat meyelesaikan tugas akhir ini dan dapat disajikan dalam bentuk tulisan. Penelitian ini adalah tindak lanjut dari ilmu yang didapatkan dari proses perkuliahan. Hasil dari penelitian ini diharapkan dapat membantu pihak-pihak yang membutuhkan terutama untuk dapat meningkatkan efisien dan efektifitas dalam menyajikan informasi.

Salawat serta salam teruntuk Nabi Muhammad SAW, yang telah memberikan serta menyampaikan kepada kita semua jalan menuju kebenaran dan kebaikan yang haqiqi.

Dalam Penulisan skripsi ini, tidak terlepas dari dukungan berbagai pihak dan pada kesempatan ini penulis ingin mengucapkan terima kasih kepada pihak yang telah membantu, terutama kepada:

1. Bapak H. Herman Nawas selaku Ketua Yayasan Perguruan Tinggi Komputer “YPTK” Padang.
2. Bapak Prof. Dr. Sarjon Defit S.Kom, M.Sc sebagai Rektor Universitas Putra Indonesia “YPTK” Padang.
3. Bapak Dr. Julius Santony S.Kom, M.Kom. sebagai Dekan Fakultas Ilmu Komputer Universitas Putra Indonesia “YPTK Padang.

Akhir kata, semoga penelitian ini dapat memberikan konstribusi terhadap masyarakat dan khususnya terhadap Universitas Putra Indonesia “YPTK” Padang. Penulis mengucapkan terima kasih dan mudah-mudahan penelitian ini berguna dan dapat bermanfaat bagi masyarakat banyak.

Padang, Januari 2018

**Penulis**

**DAFTAR ISI**

**RINGKASAN**

**KATA PENGANTAR**

**DAFTAR ISI**

**BAB I PENDAHULUAN** 1

* 1. Latar Belakang 1
  2. Rumusan Masalah 4
  3. Hipotesis 4
  4. Batasan Masalah 5
  5. Tujuan Penelitian 6
  6. Manfaat Penelitian 6

**BAB II LANDASAN TEORI**

1. Rekayasa Perangkat Lunak (RPL) 15
2. Tujuan Rekayasa Perangkat Lunak 17
3. Sejarah rekayasa Perangkat Lunak 17
4. Tahapan Umum Rekayasa Perangkat Lunak 20
5. Model Proses Rekayasa Perangkat Lunak 21
   * + 1. Model *Waterfall* 21
       2. Model *Prototipe* 23
       3. Model *Rapid Application Developmen* (RAP) 23
       4. Model *Iteratif* 24
       5. Model *Spiral* 25
6. UML (*Unified Modelling Language*) 26
   * + 1. *Use-Case Diagram* 27
       2. *Class Diagram* 30
       3. *Sequence Diagram* 33
       4. *Collaboration Diagram*  35
       5. *Statechart Diagram* 36
       6. *Activity Diagram...........................................................*  37
       7. *Deployment Diagram*  39
7. Kecerdasan Buatan (Artifical Intelligence) 40
   * 1. Defenisi Kecerdasan Buatan 40
     2. Sejarah Kecerdasan Buatan 42
     3. Kecerdasan Alami dan Kecerdasan Buatan 41
     4. Komputasi Kecerdasan Buatan dan Pemograman 48
8. Jaringan Syaraf Tiruan 48

2.4.1. Defenisi Jaringan Syaraf Tiruan 48

2.4.2. Sejarah Jaringan Syaraf Tiruan 50

2.4.2. Komponen Jaringan Syaraf Tiruan 51

2.4.2. Arsitektur Jaringan Syaraf Tiruan 53

1. Jaringan Syaraf Tiruan Algoritma *Backpropagation* 55

2.5.1. Konsep Dasar Metode *Backpropagation* 56

2.5.2. Jaringan Syaraf Tiruan Untuk Prediksi 61

* 1. Fungsi Aktivasi 61
     1. Pelatihan Standar *Backpropagation* 62
  2. Suku Bunga Bank Indonesia (*BI RATE*) 79
     1. Jadwal Penetapan dan Penentuan 80
     2. Besar Perubahan *BI Rate* 81

**BAB III METODOLOGI PENELITIAN**

1. Kerangka Penelitian 82
2. Tahapan Penelitian 82

3.2.1. Penelitian Pendahuluan 83

3.2.2. Pengumpulan Data 83

3.2.3 Metode Penelitian 84

3.2.4. Analisa 86

3.2.5. Perancangan 87

* + 1. Perancangan Model 87
    2. Perancangan Interface 89

3.2.6. Implementasi 89

3.2.7. Pengujian 90

**BAB IV HASIL DAN PEMBAHASAN**

4.1 Analisa 91

* 1. Proses Analisa 91
     1. Proses Analisa Data 91
     2. Proses Analisa Sistem 93
        1. Proses Pembagian Data 94
        2. Proses Transformasi 95
        3. Algoritma Dan Arsitektur Jaringan 102
        4. Arsitektur Jaringan Untuk Memprediksi *BI Rate* 104
        5. Perancangan Algoritma *Backpropagation* 106
  2. Perancangan Sistem 115
     1. Perancangan Sistem Dengan Menggunakan *UML* 115
        1. *Use Case Diagram* 115
        2. *Class Diagram* 121
        3. *Sequence Diagram*  125
        4. *Collaboratiom Diagram*  127
        5. *Statechar Diagram* 128
        6. *Activity Diagram* 129
        7. *Deployment Diagram* 130
     2. Desain *Interface* atau Antar Muka 132

**BAB V IMPLEMENTASI DAN PENGUJIAN**

5.1. Implementasi 136

* + 1. Spesifikasi Sistem 136
       1. Perangkat Keras (*Hardware)* 136
       2. Perangkat Lunak (*Software)* 137
    2. Instalasi Sofrware Pendukung 137
       1. Instalasi *Mowes* *Portable* II 137
  1. Pengujian 141
     1. Proses Pembuktian Dengan MATLAB 142
     2. Pengujian Manual 148
     3. Pengujian Aplikasi 149
     4. Pengujian Interface 151
        1. Tampilan Perancangan Database 151
        2. Tampilan Berbasis Website 152

**BAB VI PENUTUP**

1. Kesimpulan 157
2. Keterbatasan Sistem 158
3. Saran 158

**DAFTAR PUSTAKA** xxii

**LAMPIRAN** xxiii

**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang Masalah**

Perkembangan dunia teknologi informasi sudah sangat maju. Hal ini membuat banyak perubahan pada kehidupan manusia. Seiring dengan maju dan berkembangnya teknologi berbasis komputerisasi berbentuk *hardware* dan *software,* baik itu dimanfaatkan untuk sarana pendidikan, kesehatan, promosi, politik, bisnis, dan masih banyak realisasinya dimanfaatkan oleh manusia untuk mendapatkan kemudahan-kemudahan dalam menyelesaikan suatu permasalahan dalam pekerjaannya.

Dengan semakin pesatnya perkembangan teknologi komputer para ahli komputer mencoba menggantikan komputer menjadi suatu alat yang dapat meniru kepintaran manusia (otak manusia), yang diharapkan akan tercipta komputer yang dapat mengambil keputusan sendiri seperti layaknya manusia. Hal inilah yang mendorong lahirnya teknologi kecerdasan buatan (*Artificial Intelligence*). Salah satu teknik komputasi yang dikelompokkan dalam teknologi kecerdasan buatan (*Artificial Intelligence*) adalah jaringan syaraf tiruan (*Artificial Neural Network*).

Jaringan syaraf tiruan merupakan salah satu sistem pemrosesan yang dirancang dan dilatih untuk memiliki kemampuan seperti yang dimiliki oleh manusia dalam menyelesaikan persoalan yang rumit dengan melakukan proses belajar. Jaringan syaraf buatan mensimulasikan struktur proses pada otak manusia yaitu fungsi syaraf biologis dan kemudian membawanya kepada perangkat lunak kelas baru yang dapat mengenali pola yang kompleks serta belajar dari pengalaman-pengalaman masa lalu. Data masa lalu akan dipelajari oleh jaringan saraf tiruan sehingga mempunyai kemampuan untuk memberikan keputusan terhadap data yang belum pernah dipelajari (Kiki dan Kusumadewi,2004). Di dalam peramalan disebut peramalan kausal. Diantaranya regresi dan Jaringan Syaraf Tiruan (JST). Regresi merupakan peramalan yang bersifat parametrik sedangkan JST peramalan yang bersifat non parametrik.

Pada penelitian JST tentang peramalan dan prediksi biasanya menerapkan algoritma *backpropagation. Backpropagation* merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya(Mira Febrina, Faula Arina dan Ratna Ekawati, 2013).

Dari hasil penelitian Nazla Nurmila, Aris Sugiharto, dan Eko Adi Sarwoko (2007) Algoritma *Backpropagation Neural Network* Untuk Pengenalan Pola Karakter Huruf Jawa, dengan keakuratan algoritma *backpropagation* dalam mengenali pola karakter huruf Jawa adalah sebesar 99.563% untuk data sampel berupa data pelatihan, 61.359% untuk data sampel diluar data pelatihan, dan 75% untuk data sampel data pelatihan dan di luar data pelatihan. Dan juga tidak lupa penelitian dua tahun belakang oleh Dahriani Hakim Tanjung (2015) Jaringan Saraf Tiruan dengan *Backpropagation* untuk Memprediksi Penyakit Asma, dengan hasil metode backpropagation dapat digunakan untuk menyelesaikan masalah prediksi penyakit asma dengan menggunakan empat pola tampilan output yang terdiri dari Asma Akut Persisten Ringan, Asma Akut Persisten Sedang, Asma Akut Persisten Berat, Asma Akut Mengancam Jiwa.

Suku bunga Bank Indonesia (BI Rate) adalah suku bunga kebijakan yang mencerminkan sikap atau stance kebijakan moneter yang ditetapkan oleh Bank Indonesia dan diumumkan kepada publik (Bank Indonesia, 2012). BI Rate diumumkan oleh Dewan Gubernur Bank Indonesia setiap Rapat Dewan Gubernur (RDG) bulanan dan diimplementasikan pada operasi moneter yang dilakukan Bank Indonesia melalui pengelolaan likuiditas (liquidity management) di pasar uang untuk mencapai kebijakan moneter. Sasaran operasional kebijakan moneter dicerminkan pada perkembangan suku bunga Pasar Uang Antar Bank Overnight (PUAB O/N). Pergerakan di suku bunga PUAB ini diharapkan akan diikuti oleh perkembangan di suku bunga deposito, dan pada gilirannya suku bunga kredit perbankan. BI Rate sebagai respon (stance) kebijakan moneter ditetapkan untuk mengarahkan pergerakan inflasi dan ekonomi ke depan agar tetap berada pada jalur pencapaian sasaran inflasi yang telah ditetapkan (konsistensi). (Bank Indonesia, 2012). Perkiraan terhadap kenaikan atau penurunan BI Rate dilakukan berdasarkan pada tingkat inflasi. Inflasi memberi pengaruh positif terhadap return pasar. Saat inflasi naik, harga output produksi juga mengalami kenaikan, sehingga tingkat pinjaman di bank akan mengalami kenaikan. Hal ini menyebabkan jumlah uang yang beredar di Indonesia mengalami kenaikan sehingga nilai tukar Rupiah melemah. Selain itu, tingginya harga output produksi mengakibatkan investor lebih memilih menanamkan modalnya di pasar modal sehingga harga saham turun dan IHSG (Indeks Harga Saham Gabungan) secara umum akan mengalami penurunan. Dengan demikian, BI akan menaikkan BI Rate sebagai respon kebijakan moneter. Demikian pula sebaliknya, saat inflasi mengalami penurunan, maka nilai tukar Rupiah akan menguat, jumlah uang yang beredar mengalami penurunan, dan IHSG menguat, sehingga respon kebijakan moneter yang ditempuh oleh BI adalah dengan menaikkan BI Rate. Dengan kata lain, faktor-faktor yang sangat berpengaruh dalam penetapan BI Rate adalah tingkat inflas dan jumlah uang beredar di Indonesia, kurs Rupiah terhadap Dolar AS, serta IHSG.

Oleh karena itu melalui perancangan dan pembangunan aplikasi yang dapat meramalankan dan memprediksi BI Rate diharapkan dapat membantu baik pemerintah maupun instansi tertentu dalam membuat kebijakan yang berhubungan dengan BI Rate.

Dari latar belakang tersebut, maka saya sebagai penulis akan melakukan penelitian skripsi dengan judul “**PREDIKSI DAN PERAMALAN SUKU BUNGA BANK INDONESIA (BI RATE) DENGAN JARINGAN SYARAF TIRUAN BACKPROPAGATION**”.

**1.2 Perumusan Masalah**

Dari latar belakang masalah diatas, maka penulis merumuskan:

1. Bagaimana merancang dan membangun aplikasi Jaringan Syaraf Tiruan yang dapat melakukan prediksi dan peramalan BI Rate ?
2. Bagaimana menerapkan algoritma *backpropagation* pada aplikasi Jaringan Syaraf Tiruan yang dapat meramalkan dan memprediksi BI Rate dengan cepat dan akurat?
3. Bagaimana merancang aplikasi Jaringan Syaraf Tiruan yang dapat membantu instansi pemerintah, swasta maupun individu tertentu dalam membuat kebijakan yang berhubungan dengan BI Rate?

**1.3 Hipotesa**

Berdasarkan rumusan masalah yang telah diuraikan diatas, maka hipotesa dari penelitian ini adalah sebagai berikut :

1. Dengan merancang dan membangun aplikasi Jaringan Syaraf Tiruan diharapkan mampu melakukan prediksi dan peramalan BI Rate.
2. Dengan menerapkan algoritma *Backpropagation*  diharapkan mampu melakukan prediksi dan peramalan BI Rate dengan cepat dan akurat.
3. Dengan merancang Aplikasi Jaringan Syaraf Tiruan diharapkan dapat membantu instansi pemerintah, swasta maupun individu tertentu dalam membuat kebijakan yang berhubungan dengan BI Rate?

**1.4. Batasan Masalah**

Agar permasalahan yang dihadapi sesuai dan tidak menyimpang dari yang diharapkan, maka penulis memberikan batasan-batasan penilitian agar masalah tidak terlalu luas dan terarah dalam pembahasan dan penulisan skripsi ini. Batasan masalah akan difokuskan sebagai berikut :

1. Metode pengenalan pola yang digunakan adalah Jaringan Syaraf Tiruan menerapkan algoritma b*ackpropagation*.
2. Penelitian ini berfokus pada peramalan dan prediksi BI Rate satu bulan kedepan.
3. Data kurs dollar dengan rupiah (USD/IDR) yang digunakan adalah data bulanan.
4. Data jumlah uang beredar (miliaran) perbulan.
5. Data yang digunakan adalah data inflansi perbulan,
6. Data Indek Harga Saham Gabungan (IHSG) yang digunakan adalah data bulanan.

**1.5 Tujuan Penelitian**

Sesuai dengan pokok permasalahan yang telah dirumuskan diatas maka tujuan dari penelitian ini adalah :Untuk menghasilkan aplikasi yang dapat meramalkan dan memprediksi BI Rate dengan cepat dan akurat.

1. Untuk menghasilkan aplikasi yang dapat membantu masyarakat, pemerintah dan penulis sendiri dalam kehidupan sehari-hari.
2. Untuk mempermudahkan pihak tertentu dalam membuat kebijakan yang berhubungan dengan BI Rate. Seperti para investor asing dan perbankan.

**1.6 Manfaat Penelitian**

1. Masyarakat

Dapat membantu pada kehidupan kita yakni menjaga inflasi dan menjaga suku bunga bank seperti bunga deposito dan bungan tabungan.

1. Pemerintah

Dapat menjadi alternatif sebagai acuan pembuatan kebijakan yang berhubungan BI Rate .

1. Bagi Peneliti

Dapat membantu penulis dalam memahami lebih dalam mengenai *Artificial Intelligence* (AI) tentang JST dan algoritma *backpropagation* yang penulis terapkan pada aplikasi peramalan dan prediksi BI Rate yang penulis rancang

**BAB II**

**LANDASAN TEORI**

**2.1 Rekayasa Perangkat Lunak**

Menurut (Soetam Rizky, 2011), rekayasa perangkat lunak merupakan langkah awal untuk mempelajari materi-materi selanjutnya seperti OOP (*Object Oriented Programming),* OOD (*Object Oriented Design*), Analisa dan Desain Sistem Infomasi, Perancangan Sistem Informasi hingga materi-materi pemograman.

Karena menerapkan konsep RPL (baik utuh maupun hanya sebagian), maka perangkat lunak yang dibangun dapat lebih dan tertata sehingga proses pengembangan dapat menjadi lebih efektif dan efisien. Konsep dasara RPL layaknya fondasi sebuah gedung yang tidak terlihat tetapi harus sangat kuat agar gedung tersebut dapat berdiri dengan kokoh.

Dan sekali lagi , sangat penting diperhatikan bahwa apapun teori dan konsep yang ada dalam RPL bukanlah untuk dihapalkan, melainkan untuk dipahami dan diimplementasikan. Dengan gambaran dari Siklus Pembelajaran RPL menurut (Soetam Rizky, 2011) sebagai berikut :

*Sumber : (Soetam Rizky, 2011).*

**Gambar 2.1 Contoh Siklus Pembelajaran RPL**

Rekayasa perang lunak yang memenuhi kriteria sebagai berikut (Rosa A.S dan M.Shalahuddin, 2014) :

1. Dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi an lingkungan (*maintainabilty*).
2. Dapat diandalkan dengan proses bisnisyang dijalankan dan perubahan yang terjadi (*dependability* dan *robust*).
3. Efisien dari segi sumber daya dan penggunaan.
4. Kemampuan untuk dipakai sesuai dengan kebutuhan.

Dari kriteria diatas maka perangkat lunak yang baik adalah perangkat lunak yang dapat memenuhi kebutuhan pelanggan (*customer*) atau *user* (pemakai perangkat lunak) atau berorientasi kepada pelanggan atau pemakai perangkat lunak, bukan berorientasi pada pembuat atau pengembang perangkat lunak. Pekerjaan yang terkait dengan rekayasa perangkat dapat dikategorikan menjadi tiga buah kategori umum tanpa melihat area dari aplikasi, ukuran proyek perangkat lunak, atau kompleksitas perangkat lunak yang akan dibuat (Rosa A.S dan M.Shalahuddin, 2014).

Tantangan yang dihadapi dalam proses rekayasa perangkat lunak adalah sebagai berikut (Rosa A.S dan M.Shalahuddin, 2014) :

1. Tantangan warisan dimana perangkat lunak dikembangkan selama bertahun-tahun oleh orang-orang yang berbeda, hal ini dapat menyebabkan ketidakpahaman atau perubahan tujuan pembuatan perangkat lunak.
2. Tanangan heterogenitas dimana perangkat lunak harus dapat beradaptasi dengan teknologi yang terus berkembang dengan semangkin luasnya lingkungan distribusi perangkat lunak.
3. Tantangan pengiriman bahwa perangkat lunak dengan sekala besar dan kompleks sekalipun dapat sampai ke tangan pelanggan (*customer*) atau *user* dengan cepat dan kualitas tetap terjaga.

**2.1.1 Tujuan Rekayasa Perangkat Lunak**

Secara umum tujuan Rekayasa Perangkat Lunak tidak berbeda dengan bidang rekayasa yang lain. Bidang rekayasa akan selalu berusaha menghasilkan output yang kinerjanya tinggi waktu penyelesaian yang tepat dan Berbiaya rendah. Secara lebih khusus kita dapat menyatakan tujuan RPL adalah sebagai berikut (Cecep lupi, 2012) :

1. Menghasilkan perangkat lunak yang kinerjanya tinggi, andal serta tepat waktu.
2. Memperoleh biaya produksi perangkat lunak yang rendah.
3. Menghasilkan perangkat lunak yang biaya perawatan rendah.
4. Menghasilkan perangkat lunak yang bisa bekerja di berbagai jenis platform.

**2.1.2 Sejarah rekayasa Perangkat Lunak**

Industri perangkat lunak telah berkembang melalui empat era, yaitu pada tahun 1950-an sampai tahun 1960-an, tahun 1960-an sampai pertengahan tahun 1970-an, pertengahan tahun 1070-an sampai 1980-an, dan prtengahan tahun 19-80an sampai sekarang. Setiap era memiliki karakteristik khusus dan setia tahunnya perangkat lunak mengalami peningkatan. Pada awal tahun 1970-an, banyak program komputer yang mulai mengalami kegagalan dan bnayak yang kehilangan keyakinan sehingga krisis industri diumumkan. Berbagai alasan mengarah pada krisis meliputi hal-hal sebagai berikut (Janner Simarmata , 2010:) :

1. Perkembangan perangkat keras yang terlalu cepat.
2. Kemampuan untuk membangun yang dituntut untuk memenuhi kebutuhan secara cepat.
3. Penigkatan ketergantungan pada perangkat lunak.
4. Pembangunan perangkat lunak berkualitas tinggi dan dapat di percaya
5. Desain yang kurang dan minimnya sumber daya

Untuk menghindari kegagalan dan meningkatkan pencatatan, pemahaman yang lebih baik dari proses tersebut, teknik-teknik *estimasi* yang terbaik untuk mengukur waktu dan biaya maupun kualitas akan sangat dibutuhkan. Saat ini kebanyakan masalah terjadi karena adanya proses perangkat lunak yang kacau dan kadang-kadang keberhasilan perangkat lunak tergantung pada usaha perorangan. Selain itu alasan tersebut mungkin berkaitan dengan prinsip-prinsip rekayasa perangkat lunak apabila kebutuhan teridentifikasi dengan benar, maka akan memudahkan didalam mengidentifikasi praktik-praktik terbaik yang dapat diterapkan karena satu proses mungkin cocok untuk satu organisasi dan mungkin tidak cocok untuk sebagian besar lainnya.

Perkembangan perangkat lunak sampai sekarang dibagi kedalam beberapa era (Janner Simarmata, 2010) :

1. Era Pemula

Pada awalnya, bentuk perangkat lunak merupakan sambungan kabel ke dalam antarbagian dalam kabel komputer. Pengakses komputer dilakukan dengan *punched card* yaitu kartu yang dilubangi. Penggunaan komputer saat itu masih dilakukan secara langsung. Pada era ini, perangkat lunak merupakan satu kesatuan dengan perangkat kerasnya.

1. Era stabil

Pada era ini, penggunaan komputer sudah banyak digunakan, tidak hanya oleh kalangan peneliti atau akademis saja, tetapi juga oleh kalangan industri dan juga perusahaan. Pada era ini perangkat lunak mulai bergeser menjadi sebuah produk. Sebuah perangkat lunak mampu digunakan oleh banyak pengguna secara cepat atau langsung, pada era ini sistem basis data yang memisahkan program dengan data mulai dikenal

1. Era mikro

Sejalan dengan semangkin luasnya PC dan jaringan komputer di era ini, perangkat lunak juga berkembang untuk memenuhi kebutuhan perseorangan. Perangkat lunak dapat dibedakan menjadi perangkat lunak sistem yang bertugas menangani sisi internal dan perangkat lunak aplikasi yang digunakan secara langsung oleh penggunanya untuk keperluan tertentu.

1. Era modren

Saat ini perangkat lunak sudah terdapat dumana-mana, tidak hanya pada sebuah superkomputer dengan 25 prosesnya, sebuah komputer genggampun telah dilengkapi dengan perangkat lunak yang dapat disingkronkan dengan PC, mesin cuci, AC, dan oven. Telah ditanami perangkat lunak untuk mengatur operasi peralatan itu, lebih hebat lagi, setiap peralatan itu mungkin suatu saat kelak akan dapat saling terhubung.

* + 1. **Tahapan Umum Rekayasa Perangkat Lunak**

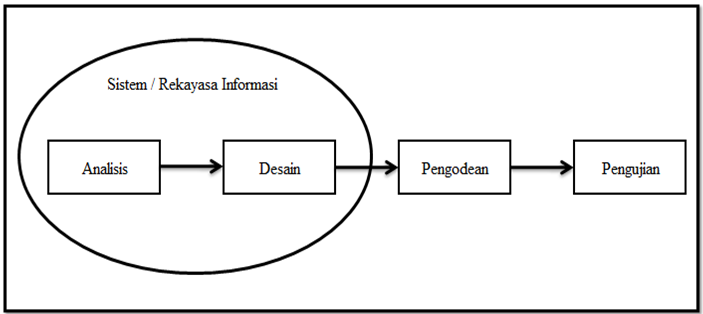
Menurut (Rosa A.S dan M. Salahuddin , 2013) rekayasa perangkat lunak dapat dibagi menjadi tiga bagian proses tanpa melihat area dari aplikasi, ukuran proyek-proyek perangkat lunak, dan kompleksitas perangkat lunak yang dibuat, yaitu:

1. Proses pendefenisian yang fokus pada “*what*”. Yaitu mengidentifikasi informasi apa yang harus diproses, seperti apa fungsi dan performansi yang diinginkan., seperti apa perilaku sistem yang diinginkan, apa kriteria validasi yang dibutuhkan atau mendefenisikan sistem.
2. Proses pengembangan yang fokus pada “*how*” , yaitu selama tahap pembangunan perangkat lunak berusaha mendefenisikan hal berikut:
3. Bagaimana data di strukturkan.
4. Bagaimana fungis-fungsi yang dibutuhkan diimplementasikan didalam arsitektur perangkat lunak.
5. Bagaimana detail prosedural diimplementasikan.
6. Bagaimana karakter antarmuka tampilan.
7. Bagaimana desain ditranslasikan ke bahasa pemograman.
8. Bagaimana pengujian akan dilakukan.
9. Proses pendukung. Artinya terasosiasi pada perbaikan kesalahan (*error*), adaptasi yang dibutuhkan pada lingkungan perangkat lunak yang terlibat dan perbaikan yang terjadi akibat perubahan kebutuhan *user.* Proses pendukung terdiri dari empat fase, yaitu:
10. Koreksi.
11. Adaptasi
12. Perbaikan
13. Pencegahan.

**2.1.4 Model Proses Rekayasa Perangkat Lunak**

Menururt (Rosa A.S dan M.Shalahuddin, 2014), SDLC ( *System Developmen Life Cycle*) adalah proses pengembangan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metedologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best prectice* atau cara-cara yang sudash teruji baik).

**2.1.4.1 Model *Waterfall***

Model SDLC air terjun (water fall) sering juga disebut model sekuensial linier (sequential linear) atau alur hidup klasik (classic life cycle)”. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (Rosa A.S dan M.Shalahuddin, 2014). Berikut ini adalah gambar model air terjun :

*Sumber : (Rosa A.S dan M.Shalahuddin, 2014).*

**Gambar 2.2 Ilustrasi Model *Waterfall***

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

1. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

1. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

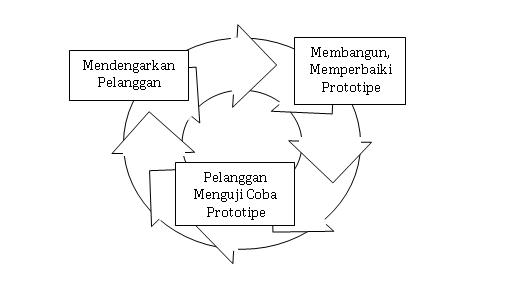
1. Pengujian

Pengujian fokus pada perangkat lunak secara segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

1. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari tahap analisis spesifikasi untuk perubahan perangkat lunak baru.

**2.1.3.2 Model Prototipe**

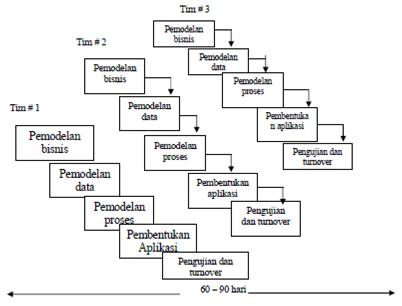
Model ini digunakan untuk manyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang di inginkan oleh pelanggan kepada pengembang perangkat lunak. Model ini dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat, lalu dibuatlah program prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program protipe biasanya program yang belum jadi. Program ini biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Model prototipe cocok digunakan untuk menjabarkan kebutuhan pelanggan secara lebih detail karena pelanggan sering kali kesulitan menyampaikan kebutuhannya secara detail tampa melihat gambaran yang jelas Berikut adalah gambar dari model prototip (Rosa A.S dan M.Shalahuddin, 2014) :

*Sumber : (Rosa A . S dan M. Shalahudin, Rekayasa Perangkat Lunak, 2014:32).*

**Gambar 2.3 Ilustrasi Model *Prototipe***

**2.1.3.3 Model *Rapid Application Developmen* (RAP)**

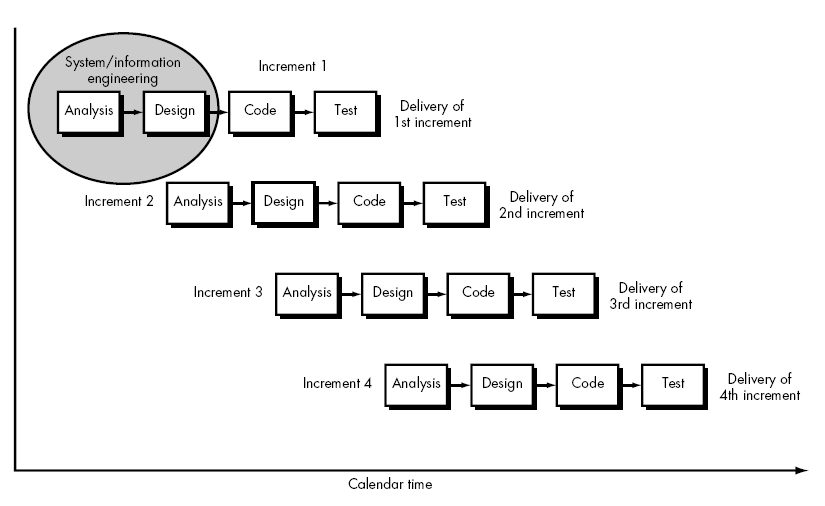
*Rapid application development*  (RAD) adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik incremental (bertingkat). RAD menekankan pada siklus pembangunan pendek, singkat, dan cepat. Model RAP adalah adaptasi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak (Rosa A.S-M.Shalahuddin, Rekayasa Perangkat Lunak,2014).

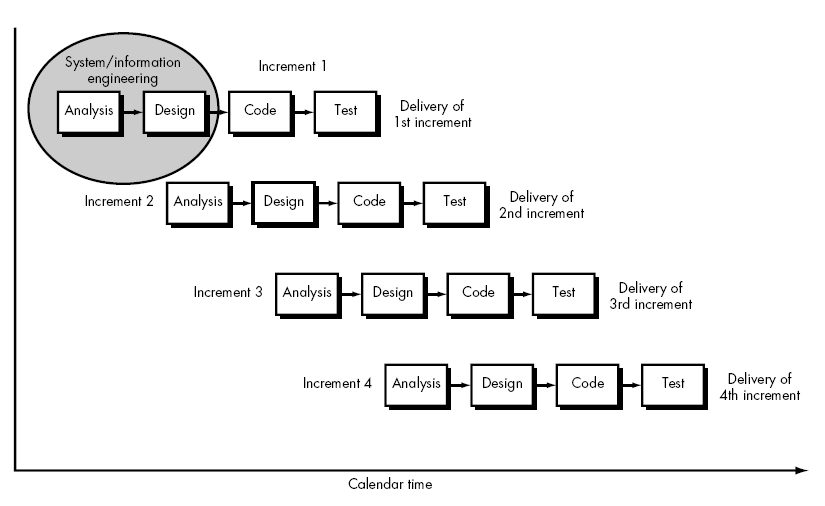
 Jika kebutuhan perangkat lunak dipahami dengan baik dan lingkup perangkat lunak dibatasi dengan baik sehingga tim dapat menyelesaikan pembuatan erangkat lunak dengan waktu yang pendek. Model RAD membagi tim pengembang menjadi beberapa tim untuk mengerjakan beberapa komponen masing-masinng tim pengerjaan dapat dilakukan secara paralel. Berikut adalah gambar dari model RAD (Rosa A.S-M.Shalahuddin, 2014) :

*Sumber : (Rosa A . S dan M. Shalahudin, 2014).*

**Gambar 2.4 Ilustrasi Model *Rapid Application Development***

**2.1.3.4 Model *Iteratif***

Model iteratif mengkombinasikan proses-proses pada model air terjun dan iteratif pada model prototipe. Model ini dibuat untuk mengtasi kelemahan dari model air terjun yang tidak mengakomodasi iterasi dan mengatasi kelemahan dari model prototipe yang memiliki proses terlalu pendek dan setiap iteratif prosesnya tidak selalu menghasilkan produk. Berikut adalah gambar dari model iteratif (Rosa A.S dan M.Shalahuddin, 2014:) :



*Sumber : (Rosa A . S dan M. Shalahudin, 2014).*

**Gambar 2.5 Ilustrasi Model *Iteratif***

**2.1.3.5 Model *Spiral***

Model spiral dibagi menjadi beberapa kerangka aktifitas atau disebut juga wilayah kerja (task region). Banyaknya wilayah kerja biasanya diantara tiga sampai enam wilayah sebagai berikut :

1. Komunikasi dengan pelanggan

Aktifitas ini diperlukan untuk membangun komunikasi yang efektif antara pengembang dengan pelanggan.

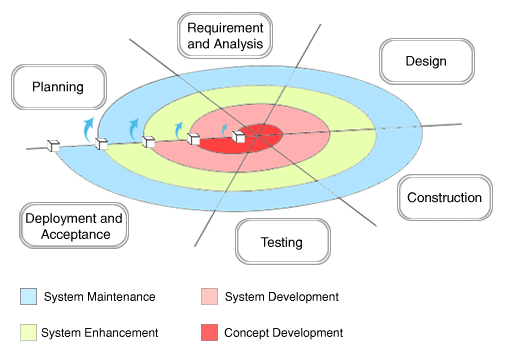
1. Perencanaan (*planning*)

Aktifitas ini diperlukan untuk mendefinisikan sumber daya, waktu, dan informasi yang terkait dengan proyek.

1. Analisis resiko (*risk analisys*)

Aktifitas ini diperlukan untuk memperkirakan resiko dari segi teknis maupun manajemen.

Berikut adalah gambar dari model spiral: (Rosa A . S dan M. Shalahudin, 2014)



*Sumber : (Rosa A . S dan M. Shalahudin, Rekayasa Perangkat Lunak, 2014).*

**Gambar 2.6 Ilustrasi Model *Spiral***

**2.2 UML (*Unified Modelling Language*)**

Menurut (Rosa A . S dan M. Shalahudin, 2014). UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metedologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metedologi berorientasi objek.

**2.2.1 *Use Case diagram***

Menurut (Rosa A . S dan M. Shalahudin, 2014). *Use case* atau *diagram use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi aPa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case yaitu pendefenisian* apa yang disebut aktor dan *use case.*

1. Aktor merupakan orang, proses, dan sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *use case* (Rosa A . S dan M. Shalahudin, 2014) :

**Tabel 2.1 Simbol – Simbol *Use Case Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Deskripsi** |
| 1 |  | *Use Case* | Fungsionalitas yang disediakan system sebagai unit-unit yang saling bertukar pesan antar unit atau actor, biasanya dinyatakan dengan menggunakan kata kerja di awal di frase nama *use case .* |
| 2 | ­  nama actor | Aktor atau *actor* | Orang, proses, atau system lain yang berinteraksi dengan system informasi yang akan dibuat diluar system informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang , tapi actor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor |
| 3 |  | Asosiasi atau *association* | Komunikasi antara actor dan use case yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan actor |
| 4 | <<extend>> | Ekstensi atau *extend* | Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu, mirip dengan prinsip inheritance pada pemograman berorientasi objek, biasanya usecase tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, missal    arah panah mengarah pada use case yang ditambahkan. |
| 5 |  | Generalisasi / *generalization* | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainya, missal:  Arah panah mengarah pada use case yang menjadi generalisasinya (umum) |
|  | <<include>>  <<uses>> | Menggunakan / include/ uses | Dijalankan use case ini ada dua sudut pandang yang cukup besar mengenai include di use case :   * Include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan, isalnya pada kasus berikut:   <<include>>   * Include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang ditambahkan telah dijalankan sebelum use case tambahan dijalankan   <<include>>  Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan. |

*Sumber : (Rosa A . S dan M. Shalahudin, 2014).*

**2.2.2 *Class Diagram***

*Classs Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi((Rosa A . S dan M. Shalahudin, 2014).

Berikut ini simbol-simbol yang sering digunakan pada saat pembuatan *class diagram*, dapat kita lihat pada tabel berikut ini :

**Tabel 2.2 Simbol – Simbol *Class Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Deskripsi** |
| 1 | |  | | --- | | Nama\_kelas | | +atribut | | +opration() | | Kelas | Kelas pada struktur sistem. |
| 2 | Nama\_interface | GFF Antarmunka / *interface* | Sama dengan konsep interface dalam pemograman berorientasi objek |
| 3 |  | Asosiasi / *association* | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity |
| 4 |  | Asosiasi berarah / directed association | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus) |
| 5 |  | Ketergantungan/ dependency | Relasi antar kelas dengan makna ketergantungan antar kelas |
| 6 |  | agregasi / aggregation | Relasi antar kelas dengan makna semua-bagian (*whole-part*) |

*Sumber : ((Rosa A . S dan M. Shalahudin, 2014).*

**2.2.3 *Sequence Diagram***

*Sequence diagram*menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message*  yang dikirim dan diterima antar objek.Simbol-simbol yang ada pada sequence diagram adalah sebagai berikut :

**Tabel 2.3 Simbol – Simbol *Sequence Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Deskripsi** |
| 1 | Nama aktor  nama aktor | Aktor Atau Tanpa waktu aktif | Orang, proses, atau system lain yang berinteraksi dengan system informasi yang akan dibuat itu sendiri, jadi walaupun symbol actor adalah gambar orang, tapi actor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal frase nama actor |
| 2 |  | Garis hidup atau *lifeline* | Menyatakan kehidupan suatu objek |
| 3 | Nama objek : nama kelas | Objek | Menyatakan objek yang berinteraksi pesan |
| 4 |  | Waktu aktif | Menyatakan objek dalam keadaan aktif dan berirentaksi pesan |
| 5 | <<create>> | Pesan tipe *create* | Menyatakan suatu objek membuat objek yang lain, arah pana mengarah pada objek yang dibuat |
| 6 | 1: nama\_metode() | Pesan tipe *call* | Menyatakan suatu objek memanggiloperasi /metode yang ada pada objek lain atau dirinya sendiri.  1 : nama\_metode  Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggiloperasi/metode maka oprasi /metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi. |
| 7 | 1: keluaran | Pesan tipe return | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu embalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian |
| 8 | <<destroy>>  ­ |  | Menyatakan suatu objek mengahiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy |

*Sumber : (Rosa A . S dan M. Shalahudin, 2014).*

**2.2.4 *Collaboration Diagram***

*Collabration diagram* menggambarkan interaksi antar objek/bagian dalam bentuk urutan pengiriman pesan. *Collaboration diagram* menunjukkan *message-message* obyek yang dikirimkan satu sama lain.*Collaboration diagram* adalah bentuk lain dari *squence diagram*. Bila *squence diagram* diorganisir menurut waktu maka *collabration* *diagram* diorganisir menurut ruang atau *space* ((Rosa A . S dan M. Shalahudin, 2014).

*Collaboration diagram* merupakan asosiasi diantara objek-objek. Panah didekat garis asosiasi menunjukkan *message,* sedangkan *content message* ditunjukkan dengan *label*. Angka pada *message* menunjukkan urutan *message.*

Berikut ini simbol-simbol yang sering digunakan pada saat pembuatan *collaboration diagram*, dapat kita lihat pada tabel berikut ini :

**Tabel 2.4 Simbol – Simbol *Collaboration Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Deskripsi |
| 1. | Nama objek : nama kelas | Objek | Objek yang melakukan interaksi pesan |
| 2. |  | Link | Relasi antar objek yang menghubungkan objek satu dengan ang lainya atau dengan dirinya sendiri  Nama\_objek : nama\_kelas |
| 3. |  | Arah pesan / stimulasi | Arah pesan yang terjadi,jika pada suatu link ada dua arah pesan yang berbeda maka arah juga digambarkan dua arah pada dua sisi *link* |

*Sumber : (Rosa A . S dan M. Shalahudin, 2014).*

**2.2.5 *Statechart Diagram***

*Statechart diagram* atau dalam bahasa indonesia disebut diagram mesin status digunakan untuk menggambarkan perubahan status atau transaksi status dari sebuah mesin atau sistem perubahan tersebuh digambarkan dalam bentuk *graf*  berarah. *Statechart* *diagram* cocok digunakan untuk menggambarkan alur interaksi pengguna denagn sistem (Rosa A . S dan M. Shalahudin, 2014).

Berikut ini simbol-simbol yang sering digunakan pada saat pembuatan *statechart diagram*, dapat kita lihat pada tabel berikut ini :

**Tabel 2.5 Simbol-Simbol *Statechart Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Deskripsi** |
| 1. |  | *Initial Pseudo State* | Bagaimana objek dibentuk atau diawali |
| 2. |  | *Final State* | Bagaimana objek dibentuk dan dihancurkan |
| 3. |  | *Event* | Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya |
| 4. | State | *State* | Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek. |

*Sumber : (Rosa A . S dan M. Shalahudin, 2014).*

**2.2.6 *Activity Diagram***

Diagram aktifitas atau *activity diagram* menggambarkan *workflow*(aliran kerja)atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa A . S dan M. Shalahudin, 2014).

Berikut ini simbol-simbol yang sering digunakan pada saat pembuatan *activity diagram*, dapat kita lihat pada tabel berikut ini :

**Tabel 2.6 Simbol – Simbol Activity Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Deskripsi** |
| 1. |  | Status awal | Status awal aktivitas system, sebuah diagram aktifitas memiliki sebuah status awal |
| 2. | aktivitas | Aktivitas | Aktivitas yang dilakukan system, aktivitas biasanya diawali dengan kata kerja |
| 3. |  | Percabangan atau *decision* | Asosiasi percabangan dimana jika ada pilihan ativitas lebih dari satu |
| 4. |  | Penggabungan | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu |
| 5 |  | Status akhir | Status akhir yang dilakukan system,sebuah diagram aktivitas memiliki sebuah status akhir |
| 6. | Nama swimlane | Swimlane | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi |

*Sumber : (Rosa A . S dan M. Shalahudin, 2014)*

**2.2.7 *Deployment Diagram***

*Deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan sistem tambahan (*embedded system)*, sistem *clien/server*, sistem terdistribusi murni dan rekayasa ulang aplikasi (Rosa A . S dan M. Shalahudin, 2014).

Berikut ini simbol-simbol yang sering digunakan pada saat pembuatan *Deployment diagram*, dapat kita lihat pada tabel berikut ini :

**Tabel 2.7 Simbol – Simbol Deployment Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Deskripsi** |
| 1. | package | Package | Package merupakan sebuah bungusan dari satu atau lebih node |
| 2. | Nama node | Node | Biasanya mengacu pada perangkat keras (hardware), perangkat unak yang tidak dibuat sendiri(software), jika da dalam node disertakan komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefenisikan sebelumnya pada diagram komponen |
| 3. |  | Ketergantungan / dependency | Kebergantungan antar node, arah panah mengarah pada node yang dipakai |
| 4. |  | Link | Relasi antar node |

*Sumber : (Rosa A . S dan M. Shalahudin, 2014).*

* 1. **Kecerdasan Buatan**
     1. **Defenisi Kecerdasan Buatan**

Menurut (Suyanto, 2014), Kecerdasan buatan berasal dari bahasa Inggris “*Artificil Intelligence”*  atau disingkat AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan *atificial* artinya buatan. Kecerdasan buatan yang dimaksud disini merujuk pada mesin yang mampu berfikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia.

1. *Thingking hummanly*: *the cognitive modeling approach*

Pendekatan ini dilakukan dengan dua cara sebagai berikut (Suyanto, 2014 ):

1. Melalui introspeksi, yaitu mencoba menangkap pemukiran-pemikiran kita sendiri pada saat kita berpikir. Tetapi, seorang psikolog mengaatakan “Bagaimana anda tau bahwa Anda mengerti? Karena pada saat Anda menyadari pemikiran Anda, ternyata pemikiran tersebut sudah lewat dan digantikan kesadaran Anda. Sehingga, defenisi ini terkesan mengada-ada dan tidak mungkin dilakukan.
2. Melalui eksperimen psikologi.
3. *Acting hummanly*: *the Turing test approach*

Pada tahun 1950, Alan turing merancang suatu ujian bagi komputer berintelejensia untuk menguji apakah komputer tersebut mampu mengelabuhi seorang manusia yang mengintograsinya melalui *teletype* (komunikasi berbasis teks jarak jauh). Jika *interrogator* tidak dapat membedakan yang diintrogasi adalah manusia atau komputer, maka komputer ber-*intelejensia* tersebut lolos dari *Turung test*. Komputer tersebut perlu memiliki kemampuan *Natural Language Processing, Knowledge Representation, Automated Reasonin*, *Machine Learning, Computer Vision, Robotics. Turing test* sengaja menghindari interaksi fisik antara *interogator* dan komputer karena simulasi fisik manusia tidak memerlukan *intelejensia* (Suyanto, 2014 ).

1. *Thinking rationally*: *the laws of thought approach*

Terdapat dua masalah dalam pendekatan ini yaitu(Suyanto, 2014 ):

1. Tidak mudah untuk membuat pengetahuan informal dan menyatakan pengetahuan tersebut ke dalam *formal term* yang diperlukan oleh notasi logika, khususnya ketika pengetahuan tersebut memiliki kepastian kurang dari 100%.
2. Terdapat perbedaan dasar antara dapat memecahkan masalah “dalam prinsip” dan memecahkannya “dalam dunia nyata”.
3. *Acting rationally*: *the rational agent approach*

Membuat interferensi yang logis merupakan bagian dari suatu *rational agent*. Hal ini disebabkan satu-satunya cara untuk melakukan aksi secara rasional adalah dengan menalar secara logis. Dengan menalar secara logis, maka bisa di dapatkan kesimpulan bahwa akasi yang diberikan akan mencapai tujuan, maka *agent* dapat melakukan aksi berdasarkan kesimpulan tersebut. (Suyanto, 2014 ).

Kecerdasan buatan (*Artificial Intelligence*) adalah bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia. Menurut John McCarthy, 1956, AI merupakan “Untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia.” (Sri Kusumadewi, 2004).

Defenisi AI untuk saat ini adalah *acting rationally* dengan pendekatan *rational agent.* Hal ini berdasarkan pemikiran bahwa komputer bisa melakukan penalaran secara logis dan juga bisa melakukan aksi secara rasional berdasarkan hasil penalaran tersebut.

* + 1. **Sejarah Kecerdasan Buatan**

Istilah AI pertama kali dikemukakan pada tahun 1956 di konferensi Darthmouth. Sejak saat itu, AI terus dikembangkan sebab berbagai penelitian mengenai teori-teori dan prinsip-prinsip juga terus berkembang. Meskipun istilah AI baru muncul pada tahun 1956, tetapi teori-teori yang mengarah ke AI sudah muncul sejak tahun 1941. Secara lengkap, berikut ini tahapan-tahapan sejarah perkembangan AI (Suyanto, 2014) :

1. Era komputer elektronik (1941)

Pada tahun 1941 telah ditemukan alat penyimpanan dan pemprosesan informasi. Penemuan tersebut dinamakan komputer elektronik yang dikembangkan di USA dan jerman. Komputer pertama ini memerlukan ruangan yang luas dan ruang AC yang terpisah. Saat itu komputer melibatkan konfigurasi ribuan kabel untuk menjalankan suatu program. Hal ini sangat merepotkan para *programmer*. Pada tahun 1949, berhasil dibuat komputer yang mampu menyimpan program sehingga membuat pekerjaan untuk memasukkan program menjadi lebih mudah. Penemuan ini menjadi dasar pengembangan program yang mengarah ke AI.

1. Era persiapan AI (1943-1959)

Pada tahun 1943, Warren McCulloch dan Walter Pitts menggunakan tiga hal: pengetahuan fisiologi dasar dan fungsi sel syaraf dalam otak, analisis formal tentang logika proposisi (*propositional logic),* dan teori komputasi Turing. Mereka berhasil membuat suatu meodel sel syaraf tiruan (*artificial neuron*) di mana setiap *neuron* digambarkan sebagai ‘*on’* dan ‘*off’.* Mereka menunjukkan bahwa setiap fungsi dapat dihitung dengan suatu jaringan sel syaraf tiruan dan bahwa semua hubungan logis dapat diimplementasikan dengan struktur jaringan yang sederhana.

1. Awal perkembangan AI (1952-1969)

Pada tahun-tahun pertama perkembangannya, AI mengalami banyak kesuksesan. Diawali dengan kesuksesan Newell dan Simon dengan sebuah program yang disebut *General Problem Solver*. Program ini dirancang untuk memulai penyelesaian masalah secara manusiawi. Pada tahun 1958, McCarty di MIT AI Lab Memo No.1 mendefenisikan bahasa pemograman tingkat tinggi yaitu LISP, yang sekarang mendominasikan pembuatan program-program AI. Kemuadian, McCarty membuat program yang dinamakan *programs with Common Sense.* Di dalam program tersebut, dibuat rancangan untuk menggunakan pengetahuan dalam mencari solusi.

1. Perkembangan AI melambat (1966-1974)

Perdiksi Herbert Simon pada tahun 1957 yang menyatakan bahwa AI akan menjadi ilmu pengetahuan yang akan berkembang dengan pesat ternyata meleset. Pada 10 tahun kemudian, perkembangan AI melambat. Hal ini disebabkan adanya 3 kesulitan utama yang dihadapi AI, yaitu:

1. Program-program AI yang bermunculan hanya membangun sedikit atau bahkan tidak mengandung sama sekali pengetahuan (*knowledge*) pada asubjeknya. Program-program AI berhasil karena hanya menipulasi sintesis yang sederhana.
2. Banyak masalah yang harus diselesaikan oleh AI. Karena terlalu banyaknya masalah yang berkaitan, maka tidak jarang banyak terjadi kegagalan pada pembuatan program AI.
3. Beberapa batasanpada struktur dasar yang digunakan untuk menghasilkan perilaku intelejensia.
4. Sistem berbasis pengetahuan

Pengetahuan adalah kekuatan pendukung AI. Hal ini dibuktikan dengan program yang di buat oleh Ed Feigenbaum, Bruce Buchanan dan Joshua Lederberg yang membuat progrram untuk memecahkan masalah struktur molekul dan informasi yang di dapatkan dari *spectometer* massa. Program ini dinamakan *Dendral Programs* yang berfokus pada segi pengetahuan kimia.

1. AI menjadi sebuah industrti (1980-1988)

Industrilisasi AI diawali dengan ditemukannya *expert system* (sistem pakar) yang dinamakan R1 yang mampu mengkonfigurasi sistem-sistem komputer baru. Program tersebut mulai dioperasiakan di *Digital Equipment Corporation* (DEC), McDermott, pada tahun 1982. Pada tahun 1986, program ini telah berhasil menghemat US$ 40 juta per tahun. Pada tahun 1988, kelompok AI di DEC menjalankan 40 sitem pakar.

1. Kembalinya jaringan syaraf tiruan (1986-sekarang)

Meskipun bidang ilmu komputer menolak jaringan syaraf tiruan setelah diterbitkan buku “*Percepron”* karangan Minsky dan papert, tetapi para ilmuan masih mempelajari bidang ilmu tersebut dari sudut pandang lain yaitu fisika. Para ahli fisika seperti Hopfield (1982) menggunakan teknik-teknik mekanika statistika untuk menganalisa sifat-sifat penyimpangan dan optimasi pada jaringan syaraf. Para ahli psikologi, David Rumelhart dan Geoff Hinton, melanjukan penelitian mengenai model jaringan syaraf pada memori. Pada tahun 1985-an sedikitnya empat kelompok riset menemukan kembali algoritma belajar propogasi balik (*Back-Propogation learing*). Algoritma ini berhasil diimplementasikan ke dalam bidang ilmu komputer dan psikoloogi.

Para ilmuan memiliki dua cara pandang yang berbeda tentang AI. Yang pertama memandang AI sebagai bidang ilmu yang hanya fokus pada proses berpikir. Sedangkan yang kedua memandang AI sebagai bidang ilmu yang fokus pada tingkah laku. Cara pandang kedua meliha AI secara lebih luas karena suatu ringkah laku pastilah di dahului dengan proses berpikir.

* + 1. **Kecerdasan Alami dan Kecerdasan Buatan**

Jika dibandingkan dengan kecerdasan alami (kecerdasan yang dimiliki oleh manusia), kecerdasan buatan memiliki beberapa keuntungan secara komersial antara lain (Sutojo, 2011) :

1. Kecerdasan buatan lebih bersifat permanen. Kecerdasan alami akan cepat mengalami perubahan. Hal ini simungkinkan karena sifat manusia yang pelupa. Kecerdasan buatan tidak akan berubah sepanjang sistem komputer dan program tidak mengubahnya.
2. Kecerdasan buatan lebih mudah diduplikasi dan disebarkan. Mentransfer pengetahuan manusia dari satu orang ke orang lain membutuhkan proses yang sangat lama, dan juga suatu keahlian itu tidak akan pernah dapat diduplikasi dengan lengkap. Oleh karena itu, jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut dan dapat dipindahkan dengan mudah ke komuputer yang lain.
3. Kecerdasan buatan lebih murah dibandingkan dengan kecerdasan alami. Menyediakan layanan komputer akan lebih mudah dan lebih murah dibandingkan dengan harus mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.
4. Kecerdasan buatan bersifat konsisten. Hal ini disebabkan karena kecerdasan buatan adalah bagian dari teknolgi komputer. Sedangkan kecerdasan alami akan senantiasa berubah-ubah.
5. Kecerdasan buatan dapat di dokumentasi. Keputusan yang dibuat oleh komputer dapat didokumentasi dengan mudah dengan cara melacak setiap aktivasi dari sistem tersebut. Kecerdasan alami sangat sulit untuk diproduksi.
6. Kecerdasan buatan dapat mengerjakan pekerjaan lebih cepat dibandingkan dengan kecerdasan alami.
7. Kecerdasan buatan dapat mengerjakan pekerjaan lebih baik dibandingkan dengan kecerdasan alami.

Sedangkan keuntungan dari kecerdasan alami adalah:

1. Kreatif, kemampuan untuk menambah atau memenuhi pengetahuan itu sangat melekat pada jiwa manusia. Pada kecerdasan buatan, untuk menambah pengetahuan harus dilakukan melalui sistem yang dibangun.
2. Kecerdasan alami memungkinkan orang untuk menggunakan pengalaman secara langsung. Sedangkan pada kecerdasan buatan harus bekerja dengan input-input simbolik.
3. Pemikiran manusia dapat digunakan secara luas, sedangkan kecerdasan buatan sangat terbatas.
   * 1. **Komputasi Kecerdasan Buatan dan Pemograman**

Seperti telah diaktakan sebelumnya, bahwa pada awal diciptakannya, komputer hanya diperlukan sebagai alat hitung *(komputasi konvensional*). Untuk itu, ada beberapa peredaan antara komputasi yang dilakukan pada kecerdasan buatan dengan komputasi konvensional tersebut perbedaan Komputasi kecerdasan buatan dengan *komputasi konvensional* terihat pada Tabel berikur ini (Sutojo, 2011 ):

**Tabel 2.8 Perbadaan Kecerdasan Buatan dan Pemograman Konvensional**

|  |  |  |
| --- | --- | --- |
| **Dimensi** | **Kecerdasan Buatan** | **Pemograman Konvensional** |
| Pemrosesan | Mengandung konsep-konsep *simbolik* | *Algoritmatik* |
| Sifat *input* | Bisa tidak lengkap | Harus lengkap |
| Pencarian | Kebanyakan bersifat *heuristic* | Biasanya didasarkan pada algoritma |
| Keterangan | Disediakan | Biasanya tidak disediakan |
| Fokus | Pengetahuan | Data & infprmasi |
| Struktur | Kontrol dopisahkan dari pengetahuan | Kontrol terintegrasi dengan infofrmasi (Data |
| Sifat *output* | Kuantitatif | *Kualitatif* |
| Pemeliharaan&*update* | Relatif mudah | Sulit |
| Kemampuan menalar | Ya | Tidak |

*Sumber : (Sutojo, 2011)*

* 1. **Jaringan Syaraf Tiruan**
     1. **Defenisi Jaringan Syaraf Tiruan**

Pada [HAY94] dinyatakan bahwa kecepatan sebuah *neuron* manusia dalam memproses suatu kejadian jauh lebih lambat bila dibandingkan dengan kecepatan gerbang logika silikon. JST adalah *prosesor* tersebar paralel (*paralel distributed processor)* sangat besar yang memilliki kecendrungan untuk menyimpan pengetahuan yang bersifat pengalaman dan membuatnya siap untuk digunakan.

Jaringan syaraf tiruan mempunyai struktur tersebar *paralel* yang sangat besar dan mempunyai kemampuan belajar, sehingga bisa melakukan *generalization,* yaitu bisa menghasilkan *output* yang benar untuk *input* yang belum pernah dilatihkan. Dengan kedua kemampuan pemprosesan ini, JST mampu menyelesaikan masalah-masalah yang sangat kompleks (Suyanto, 2014 ).

Pengertian jaringan syaraf tiruan adalah (Wilis Kaswidjanti, Fani Widiastuti, Heru Cahya Rustamaji, 2013) sistem pemproses informasi yang meiliki karakteristik untuk kerja tertentu yang menyerupai jaringan saraf biologis. JST telah dikembangkan sebagai generalisasi model matematika dari aspek kognitif manusia atau saraf biologis, yaitu didasarkan pada asumsi-asumsi bahwa :

1. Pemprosesan informasi terjadi pada elemen-elemen yang disebut *neuron*.
2. Sinyal-sinyal merambat di antara neuron melalui interkoneksi.
3. Setiap interkoneksi memiliki bobot yang bersesuaian yang pada kebanyakan jaringan saraf berfungsi untuk mengalikan sinyal yang dikirim.
4. Setiap *neuron* menerapkaan fungsi aktivasi (biasanya tidak *linear*) pada masukan jaringan untuk menentukan sinyal keluarannya.

Jaringan syaraf tiruan disusun dengan asumsi yang sama seperti jaringan syaraf biologis (Asrul Huda, 2014) :

1. Pengolahan informasi terjadi pada elemen-elemen pemrosesan (neuron-neuron).
2. Sinyal antara dua buah neuron diteruskan melalui link-link koneksi
3. Setiap link koneksi memiliki bobot terasosiasi.
4. Setiap neuron menerapkan sebuah fungsi aktivasi terhadap input jaringan (jumlah sinyal input berbobot). Tujuannya adalah untuk menentukan sinyal output. Fungsi aktivasi yang digunakan biasanya fungsi yang nonlinier.

Jaringan syaraf tiruan merupakan salah satu representasi buatan dan otak manusia selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan syaraf ini diimplimentasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama pmbelajaran. Robert Hecht-Nielsend mendefinisikan jaringan syaraf tiruan (*Neural Network*) adalah struktur pemprosesan informasi yang terdistribusi dan bekerja secara *paralel*, yang terdiri dari elemen pemrosesan (yang memiliki memori lokal dan beroperasi dengan informasi lokal) yang diinterkoneksikan bersama laur sinyal searah yang bercabang (*fanout*) kesejumlah koneksi *collateral* yang diinginkan (setiap koneksi membawa sinyal yang sama dari keluaran elemen pemprosesan tersebut). Keluaran dari elemen pemrosesan tersebut dapat merupakan sembarang jenis persamaan *matematis* yang diinginkan. Seluruh proses yang berlansung pada setiap elemen pemprosesan harus benar-benar dilakukan secara lokal yaitu keluaran hanya bergantung pada nilai masukan pada saat itu yang diperoleh melalui koneksi dan nilai yang tersimpan dalam memori lokal (Asrul Huda, 2014).

Jaringan syaraf tiruan dibayangkan seperti otak buatan di dalam cerita-cerita fiksi ilmiah. Otak buatan ini dapat berpikir seperti manusia, dan juga sepandai manusia dalam menyimpulkan sesuatu potongan-potongan informasi yang diterima. Salah satu contoh pengambilan ide dari jaringan syaraf tiruan biologis adalah adanya elemen-elemen pemrosesan pada jaringan syaraf tiruan yang paling terhubung dan berperasi secara *paralel*. Ini meniru jaringan syaraf biologis yang tersusun dari sel-sel syaraf (*neuron*). Cara kerja dari elemen-elemen pemrosesan jaringan syaraf tiruan juga sama seperti cara *neuron* meng-*encode* informasi yang diterimanya (Asrul Huda, 2014 ).

* + 1. **Sejarah Jaringan Syaraf Tiruan**

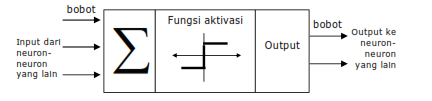
Menurut buku (Puspitaningrum, Diyah, 2006) Jaringan syaraf tiruan (JST) pertama kali diperkenalkan oleh McCulloch dan Pitts di tahun 1943. McCullotch dan Pitts menyimpulkan bahwa kombinasi beberapa*neuron* sederhana menjadi sebuah sistem neural yang akan meningkatkan kemampuan komputasinya. Bobot dalam jaringan yang akan diusulkan oleh McCulloch dan Pitts diatur untuk melakukan fungsi logika sederhana. Fungsi aktivasi yang diapakai adalah fungsu *threshold*.

Tahun 1958, Rosenblatt memperkenalkan dan memulai mengembangkan model jaringan yang disebut *Perceptron.* Metode pelatihan diperkenalkan untuk mengoptimalkan hasil iterasinya.

Widrow dan Hoff (1960) mengembangkan *perceptron* dengan memperkenalkan aturan peelatihan jaringan, yang dikenal sebagai aturan *delta* (kuadrat rata-rata terkecil). Aturan ini akan mengubah bobot *perceptron* apabila keluaran yang dihasilkan tidak sesuai dengan target yang diinginkan. Apa yang dilakukan peneliti terdahulu hanya menggunakan jaringan dengan layer tungal (*single layer*). Rumelhart (1986) mengembangkan *perceptron* menjadi *Backptopagatio,* yang memungkinkan jaringan diproses melalui beberapa layer.

Selain itu, beberapa model jaringan syaraf tiruan lain juga dikembangkan oleh Kohenen (1972), Hopfield (1982) dll. Pengembangan yang ramai dibicarakan sejak tahun 1990 an adalah aplikasi model-model jaringan syaraf tiruan untuk menyelesaikan masalah di dunia nyata.

* + 1. **Komponen Jaringan Syraf Tiruan**

 Ada beberapa tipe jaringan syaraf, namun demikian hampir semua memiliki komponen-komponen yang sama. Seperti halnya otak manusia, jaringan syaaraf terdiri dari beberapa *neuron,* dan ada hubungan antara *neuron-neuron* tersebut. *Neuron-neuron* tersebut akan mentranformasikan informasi yang diterima melalui sambungan menuju *neuron-nuron* yang lain. Pada jaringan syaraf, hubungan ini disebut degan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada *bobot* tersebut. Gambar 2.5 menunjukan struktur *neuron* pada jaringan syaraf.

*Sumber : (Maharani Dessy Wuryandari1, Irawan Afrianto, 2012)*

**Gambar 2.7 Struktur Jaringan Syaraf Tiruan**

Unsur-unsur dari suatu jaringan syaraf tiruan yaitu:

1. Unit/ Sel/ Node, tiga komponen utama: unit-unit *input*, unit-unit *hidden*, unit-unti *output*.
2. Bobot, informasi yang digunakan oleh jaringan untuk menyelesaikan persoalan. Setiap unit *input* dan unit *output* memiliki bobotnya masing-masing.
3. Bias, suatu bobot penghubung dari unit khusus dengan nilai aktivasi konstan tidak nol. Istilah ‘bias’ biasanya digunakan untuk merajuk pada ‘unit bias’ dengan nilai konstan 1. Sementara istilah ‘*thershold*’ biasanya digunakan untuk merujul pada ‘unit *threshold*’ dengan nilai konstan -1.

*Neuron* buatan ini sebenarnya mirip dengan *sel neuron biologis. Neuron-neuron*  buatan tersebut bekerja dengan cara yang sama pula dengan *neuron-neuron* biologis. Infofrmasi (disebut dengan : input) akan dikirim ke *neuron* dengan *bobot*  kedatangan tertentu. *Input* ini akan diproses oleh suatu proses perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap *neuron.* Apabila *input* tersebut akan mengitimkan *output* melalui *bobot-bobot output­*-nya ke semua *neuron* yang berhubungan dengannya.

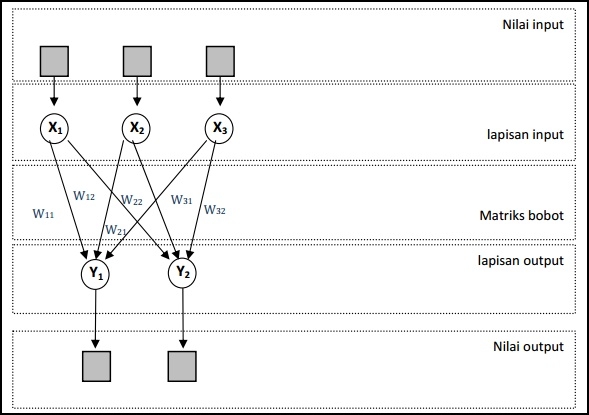
Pada jaringan syaraf, *neuron-neuron*  akan dikumpulkan dalam lapisan-lapisan (*layer)* yang disebut dengan lapisan *neuron* (*neuron layers).* Biasanya *neuron-neuron* pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahyna (kecuali lapisan *input* dan lapisan *output).* Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapaisan, mulai dari lapaisan *input* sampai ke lapiasan *output* melalui lapisan yang lailnnya, yang sering dikenal dengan nama lapisan tersembunyi (*hidden layer).* Tergantung pada algoritma pembelajarannya, bisa jadi informasi tersebut akan dirambatkan secara mundur pada jaringan.

* + 1. **Arsitektur Jaringan**

Menurut (Maharani Dessy Wuryandari, Irawan Afrianto, 2012) Pada JST, neuron-neuron akan dikumpulkan dalam lapisan-lapisan (layer) yang disebut dengan lapisan neuron (neuron layers). Neuron-neuron pada satu lapisan akan dihubungkan dengan lapisanlapisan sebelum dan sesudahnya. Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapisan, mulai dari lapisan masukan sampai ke lapisan keluaran melalui lapisan tersembunyi (hidden layer). Gambar berikut ini jaringan syaraf dengan 3 lapisan dan bukanlah struktur umum jaringan syaraf karena beberapa jaringan syaraf ada yang tidak memiliki lapisan tersembunyi. Faktor terpenting dalam menentukan kelakuan suatu neuron adalah fungsi aktivasi dan pola bobotnya. Umumnya neuron-neuron yang terletak pada lapisan yang sama akan memiliki keadaan yang sama sehingga pada setiap lapisan yang sama neuron-neuron memiliki fungsi aktivasi yang sama. Bila neuron-neuron pada suatu lapisan (misal lapisan tersembunyi) akan dihubungkan dengan neuronneuron pada lapisan lain (misal lapisan keluaran) maka setiap neuron pada lapisan tersebut (lapisan tersembunyi) juga harus dihubungkan dengan setiap neuron pada lapisan lainnya (lapisan keluaran). Terdapat 3 macam arsitektur JST, yaitu::

1. *Single-Layer Feedforward Network*(Jaringan Layer Tunggal)

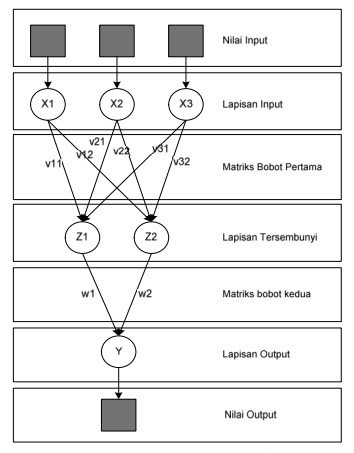
Suatu jaringan syaraf tiruan berlapis adalah jaringan *neuron* yang diorganisasikan dalam bentuk lapisan-lapisan. Pada bentuk jaringan berlapis yang sederhana, hanya terdapat *input layer* dengan *node* sumber yang terproyeksi ke dalam *output layer* dari *neuron* (*compulation nodes*), tetapi tidak sebaliknya. Dengan kata lain jaringan ini adalah jaringan jenis *Feedforward* yang tepat.Dalam jaringan ini sekumpulan input neuron dihubungkan langsung dengan sekumpulan *outputnya.* Dalam beberapa model (misal *perceptron)*, hanya ada satu buah unit *output.* Pada arsitektur ini semua unit *input* dihubungkan dengan semua unit *output* meskipun dengan bobot yang berbeda-beda. Tidak ada unit *input* yang dihubungkan dengan unit *input* lainnya. Seperti yang ada pada gambar 2.6 berikut :



*Sumber : (Maharani Dessy Wuryandari, Irawan Afrianto, 2012)*

**Gambar 2.8 *Feedforward Network* Dengan Satu lapisan *Neuron* Tunggal.**

1. ***Mulri-Layer Feedforward Network*** (Jaringan Layer Jamak)

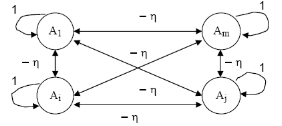
Jaringan layer jamak merupakan perluasan dari layer tunggal. Dalam jaringan ini, selain unit *input* dan *output* ada unit-unit lain yaitu *hidden layer* (layer tersembunyi). Dimungkinkan pula ada beberapa layer tersembunyi. Sama sepeti pada unit *input* dan *output*, unit-unit dalam satu layer tidak saling berhubungan. Seperti pada gambar 2.7 berikut:

*Sumber : Maharani Dessy Wuryandari, Irawan Afrianto, 2012*

**Gambar 2.9 Satu *hidden layer* dan satu *output layer.***

1. *Recurrent Network* (Jaringan Recurrent)

Recurrent Neural Network adalah jaringan yang mempunyai minimal satu *feedback loop.* Pada jaringan ini sekumpulan neuron bersainguntuk mendapatkan hak menjadi aktif. Umumnyahubungan antar neuron pada lapisan kompetitif initidak diperlihatkan pada diagram arsitektur. Gambar6 menunjukkan salah satu contoh arsitektur jaringan dengan lapisan kompetitif yang memiliki bobot -. Sebagai contoh, suatu *recurrent network* bisa terdiri dari satu lapaisan *neuron* tunggal dengan masing-masing *neuron* memberikan kembali *output* nya sebagai *input*  pada semua *neuron* yang lain, Gambar 2.8 menunjukkan salah satu contoh arsitektur jaringan dengan lapisan kompetitif yang memiliki bobot -.



*Sumber : (Maharani Dessy Wuryandari1, Irawan Afrianto, 2012)*

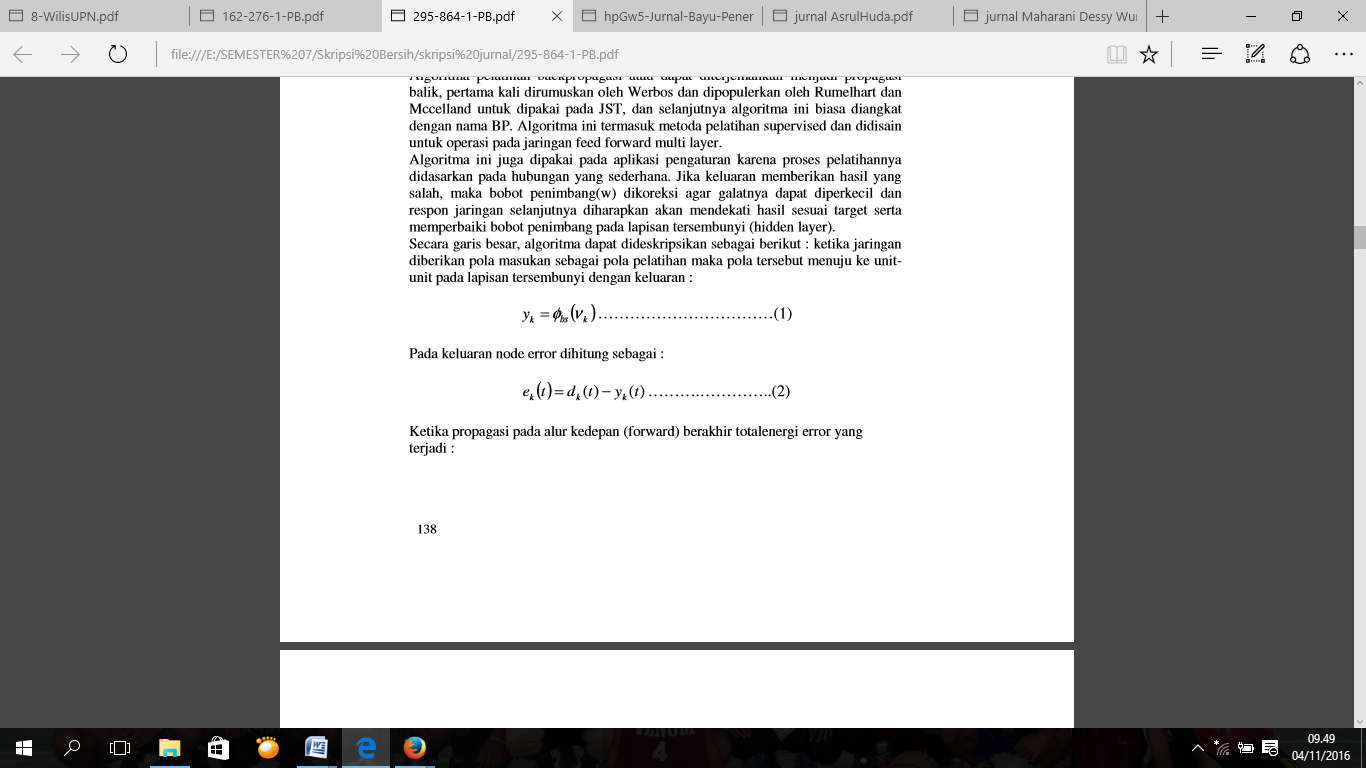
**Gambar 2.10 Recurrent Network Dengan *Neuron* Tunggal.**

* 1. **Jaringan Syaraf Tiruan Algoritma *Backpropagation***

Menurut (Maharani Dessy Wuryandari1, Irawan Afrianto, 2012) Salah satu arsitektur jaringan saraf tiruan yang sangat populer adalah multilayer feedforward networks. Secara umum, jaringan seperti ini terdiri dari sejumlah unit neuron sebagai lapisan masukan, satu atau lebih lapisan simpul-simpul neuron komputasi lapisan tersembunyi, dan sebuah lapisan simpul-simpul neuron komputasi keluaran. Sinyal masukan dipropagasikan ke arah depan (arah lapisan keluaran), lapisan demi lapisan. Jenis jaringan ini adalah hasil generalisasi dari arsitektur perceptron satu lapisan, jadi biasa disebut sebagai multilayer perceptron (MLPs). Error back propagation adalah algoritma MLPs yang menggunakan prinsip pembelajaran terawasi. Propagasi balik (ke arah lapisan masukan) terjadi setelah jaringan menghasilkan keluaran yang mengandung error. Pada fase ini seluruh bobot synaptic (yang tidak memiliki aktivasi nol) dalam jaringan akan disesuaikan untuk mengkoreksi/memperkecil error yang terjadi (error correction rule). Untuk pelatihan jaringan, pasangan fase propagasi ke depan dan balik dilakukan secara berulang untuk satu set data latihan, kemudian diulangi untuk sejumlah epoch (satu sesi lewatan untuk seluruh data

latihan dalam sebuah proses pelatihan jaringan) sampai error yang terjadi mencapai batas kecil toleransi tertentu atau nol.

* + 1. **Konsep Dasar Metode *Backpropagation***

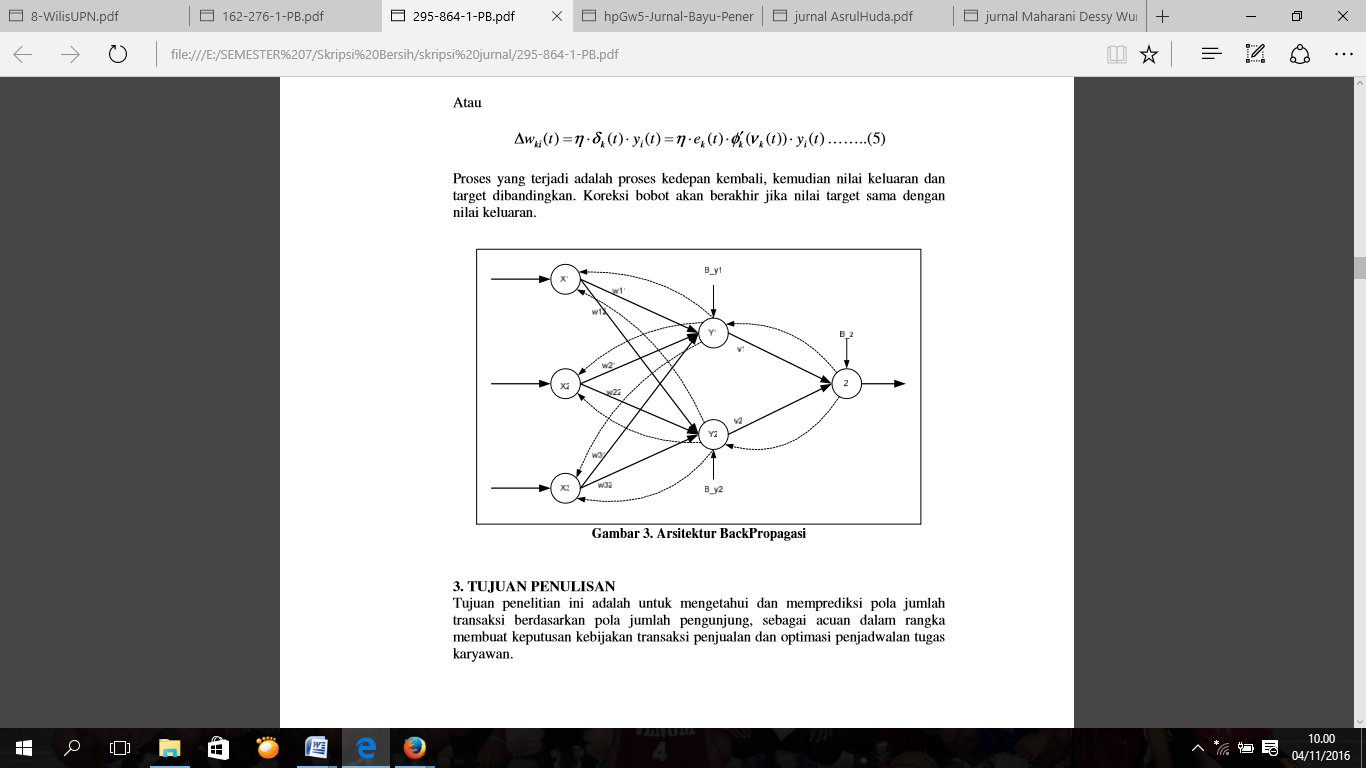
 Algoritma pelatihan backpropagasi atau dapat diterjemahkan menjadi propagasi balik, pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhart dan Mccelland untuk dipakai pada JST, dan selanjutnya algoritma ini biasa diangkat dengan nama BP. Algoritma ini termasuk metoda pelatihan supervised dan didisain untuk operasi pada jaringan feed forward multi layer. Algoritma ini juga dipakai pada aplikasi pengaturan karena proses pelatihannya didasarkan pada hubungan yang sederhana. Jika keluaran memberikan hasil yang salah, maka bobot penimbang(w) dikoreksi agar galatnya dapat diperkecil dan respon jaringan selanjutnya diharapkan akan mendekati hasil sesuai target serta memperbaiki bobot penimbang pada lapisan tersembunyi (hidden layer). Secara garis besar, algoritma dapat dideskripsikan sebagai berikut : ketika jaringan diberikan pola masukan sebagai pola pelatihan maka pola tersebut menuju ke unitunit pada lapisan tersembunyi dengan keluaran :

Pada keluaran mode error dihitung sebagai :

Ketika propagasi pada alur kedepan (*forward*) berakhir totalenergi error terjadi :

Koreksi bobot pada *backpropagation* :

Atau

Proses yang terjadi adalah proses kedepan kembali, kemudian nilai keluaran dan target dibandingkan. Koreksi bobot akan berakhir jika nilai target sama dengan nilai keluaran

*Sumber : (Iriansyah BM Sangadji, 2009)*

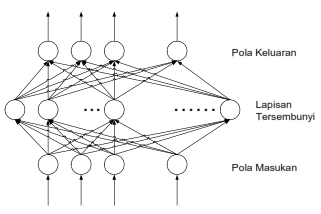
**Gambar 2.11 Metode *Backpropagation***

Kelemahan jaringan syaraf tiruan terdiri pada layer tungga membuat perkembangan jaringan syaraf tiruan menajdi terhenti pada sekitar tahun 1970 an. Penemuan backpropagation yang terdiri dari beberapa layar membuka kembali cakrawala. Terlebih setelah berhasil ditemukannya berbagai aplikasi yang dapat diselesaikan dengan *backpropagation,* membuat jaringan syaraf tiruan semakin diminati orang.

Jaringan syaraf tiruan tunggal memiliki keterbatasan dalam pengenalan pola. Kelemahan ini bisa ditanggulanggi dengan menambahkan satu atau beberapa layar tersembunyi diantara layar masukan dan keluaran. Meskipun pengguna lebih dari satu layar tersembunyi memiliki kelebihan manfaat untuk beberapa kasus, tapi pelatihannya memerlukan waktu yang lama. Maka umumnya orang mulai mencoba dengan sebuah layar tersembunyi lebih dahulu.

Seperti halnya model jaringan syaraf tiruan lain, *backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tetapi tidak sama) dengan pola yang dipakai saat pelatihan. Jaringan perambatan gelat mundur (*backpropagation)* merupakan salah satu algoritma jaringan syaraf tiruan yang sering digunakan dalam menyelesaikan masalah-masalah yang rumit berkaitan dengan identifikasi, prediksi pengenalan pola, dan sebagainya. Hal ini dimungkinkan karena jaringan dengan algoritma ini dilatih dengan menggunakan metode belajar terbimbing. Pelatihan dilakukan berulang-ulang sehingga dihasilkan jaringan yang memberikan tanggapan yang benar terhadap semua masukannya. Hal ini merupakan kelebihan dari jairnan syaraf tiruan *backpropagation* sehinga dapat mewujudkan sistem yang tahan akan kerusakan dan konsisten bekerja dengan baik (Badrul Anwar, 2011 : 112)

Jaringan *backpropagation* terdiri atas tiga lapisan atau lebih unit pengolah, yait lapisan masukan (*input*) terdiri atas variabel masukan unit sel saraf, lapisan tersembunyi dan keluaran (*output*). Seperti diperlihatkan pada gambar 2.9, ketiga lapisan terhubung secara penuh (Badrul Anwar, 2011 : 114).



*Sumber : (Badrul Anwar, 2011)*

**Gambar 2.12 Tiga Lapisan Jaringan *Backpropogation.***

Langkah yang dilakukan oleh Backpropogation adalah menghitung galat dan mengubah bobot pada semua interkoneksinya. Disini galat dihitung pada semua unit pengolah dan bobot pun diubah pada semua sambungan. Perhitungan dimulai dari lapisan masukan, hasil keluaran dari perambatan maju dibandingkan hasil keluaran yang diinginkan. Berdasarkan perbedaan ini kemudian dihitung galat untuk tiap-tiap lapisan pada jaringan (Badrul Anwar, 2011).

Algoritma pelatihan dengan metode *backpropagation* dimunculkan pada tahun 1969. *Neural Network* dengan metode *backpropogation* ini memiliki tahap jaringan multi *layer*, yaitu (Badrul Anwar, 2011) :

1. Nilai dikirim dari *input layer*  ke *hidden layer (forward)* sampai ke *output layer (actual output).*
2. *Actual output* dibandingkan dengan *output* yang diharapkan jika ada perbedaan maka dinyatakan sebagai *error.*
3. Selanjutnya *error* dikirim secara *backward* mulai dari *outpu*t *layer* ke *hidden layer* kemudian diteruskan ke *input layer.*

Untuk menentukan nilai pemberat dari *input (the net weighted input)* defenisikan terlebih dulu nilai *input*-nya. Jika nilai *input* telah ditentukan, maka selanjutnya tentukan fungsi aktvasi. *Neuron* pada *backpropagation* menggunakan fungsi aktivasi *sigmoid*. Algoritma pelatihan pada *backpropagation* sebagai berikut (Badrul Anwar, 2011) :

1. *Initiliazation*, memberikan nilai awal terhadap nilai-nilai yang diperlukan oleh *neuralnetwork* seperti *weight, threshold.*
2. *Activation*, nilai-nilai yang diberikan pada tahap *Initiliazation* akan digunakan pada tahap *Activation*. Dengan melakukan perhitungan:
3. Menentukan *actual output* pada *hidden layer*
4. Menghitung *actual output* pada *output layer.*
5. *Weight Training,* pada tahap *weight training* dilakukan dua kegiatan yaitu:
6. Menghitung *error gradient* pada *output layer.*
7. Menghitung *error gradient* pada *hidden layer.*
8. *Iteration,* pada tahap ini dilakukan proses pengulangan sampai mendapat *error* yang minimal.
   * 1. **Jaringan Syaraf Tiruan Untuk Prediksi**

Salah satu bidang dimana *backpropogation* dapat diaplikasikan dengan baik adalah bidang peramalan (*forecasting*). Peramalan yang sering kita dengar adalah peramalan besarnya penjualan, nilai tukar valuta asing, prediksi besarnya aliran air sungai, dll (Siang, 2005).

Secara umum, masalah peramalan dapat dinyatakan sebagai berikut: Diketahui sejumlah data runtun waktu (*time series*) *x1, x2,...., xn­.* Masalahnya adalah memperkirakan berapa harga *xn+1* berdasarkan *x1, x2,....,x­n.* Dengan *backpropagation*, *record* data dipakai sebagai data pelatihan untuk mencari bobot yang optimal. Untuk itu kita perlu menentukan besarnya periode dimana data berfluktuasi, periode ini kita tentukan secara *intuitif* (Siang, 2005).

* 1. **Fungsi Aktivasi**

Dalam *backpropogation*, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat:*kontinu,* ter-*diferensial,* dengan mudah dan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering dipakai adalah fungsi sigmoid biner yang memiliki *range* (0,1) (Siang, 2005).

*f(x)= dengan turunan f ‘(x) = f(x)(1-f(x))*

Fungsi lain yang sering dipakai adalah fungsi sigmoid bipolar yang bentuk fungsinya mirip dengan fungsi sigmoid biner, tapi dengan *range* (-1,1).

*f(x)=* dengan turunan *f’(x)* =

Fungsi sigmoid memiliki nilai maksimum =1. Maka untuk pola yang targetnya > 1, pola masukan dan keluaran harus terlebih dahulu ditransformasi sehingga semua pola nya memiliki *range* yang sama seperti fungsi sigmoid yang dipakai. Alternatif lain adalahh menggunakan fungsi aktivasi *sigmoid* hanya pada layar yang bukan layar keluaran. Pada layar keluaran, fungsi aktivasi yang dipakai adalah fungsi identitas : *f(x) = x*, (Siang,2005 ).

* + 1. **Pelatihan Standar *Backpropogation***

Pelatihan *backpropogation* meliputi 3 fase., fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yaang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi . kesalahan tersebut dipropogasikan mundur, mulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi (Siang, 2005 : 100).

1. Fase I : Propogasi maju

Selama propogasi maju, sinyal masukan *(= xi )* dipropogasikan ke layar tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit layar tersembunyi *(= z­i )* tersebut selanjutnya dipropogasikan maju lagi ke layar tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan *(= y­k )*. Berikutnya keluaran jaringan *(= yk )* dibandingkan dengan target yang harus dicapai *(= tk ).* Selisih *tk - yk* adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang diteentukan, maka iteraasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.

1. Fase II : Propogasi mundur

Berdasarkan kesalahan *tk - yk*, dihitung faktor *δk (k = 1,2,....m)* yang dipakai untuk mendistribusikan kesalahan di unit *yk* ke semua unit tersembunyi yang terhubung langsung dengan *yk. δk* juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran.

Dengan cara yang sama, dihitung faktor *δj* di setiap unit di layar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga semua faktor *δ* di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

1. Fase III : Perubahan bobot

Setelah semua faktor *δ* dihitung, bobot semua garis dimodifikasi secara bersmaan. Perubahan bobot suatu garis didasarkan atas faktor *δ neuron* di layar atasnya. Sebagai contoh perubahan bobot garis yang maju ke layar keluaran didasarkan atas *δk* yang ada di unit keluaran.

Ketiga fase tersebut dulang-ulang terus sehingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah *iterasi* atau kesalahan. *Iterasi* akan dihentikan jika jumlah iterasi yang akan dilakukan sudah melebihi jumlah maksimum *iterasi* yang ditetapkan, atau jika kesalahan yan terjadi sudah lebih kcil dari batas toleransi yang diizinkan.

Algoritma pelatihan untuk jaringan dengan satu layer tersembunyi (dengan fungsi aktivasi sigmoid biner) adalah sebagai berikut (Siang, 2005) :

*Langkah 0* : Inisialisasi semua bobot dengan bilangan acak kecil.

*Langkah 1* : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2-9.

*Langkah 2* : Untuk setiap pasang data pelatihan, lakukan langkah 3-8.

Fase I : Propogasi maju

*Langkah 3* : Tiap unit masukan menerima sinyal dan meneruskan ke unit tersembunyi diatasnya.

*Langkah 4* : Hitung semua keluaran di unit tersembunyi *zj­ (j = 1,2,....,p).*

*z\_netj = vjo + i vji*

*zj = f (z\_netj) =*

*Langkah 5* : Hitung semua keluaran jaringan di unit *yk  k(1,2,...,m).*

*y\_netk = wko + jwkj*

*yk = f (y\_netk) =*

Fase II : Propagasi mundur

*Langkah 6* : Hitung faktor *δ* unit keluaran berdasarkan kesalahan di setiap unit keluaran *yk (k = 1,2,...m).*

*δk (tk – yk ) f ’(y\_netk* ) = (tk­*– yk ) yk­ (1-yk )*

*δk* merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layar dibawahnya (langkah 7)

hitung suku perubahan bobot wkj (yang akan dipakai nanti untuk meubah bobot *wk j* ) dengan laju percepatan *α* .

*∆wk j = α δk zj ; k = 1,2,3,...,m ; j = 0,1,...,p*

*Langkah 7* : Hitung faktor *δ* unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi *zj (j=1,2,3...,p)*

*δ\_netj =kwk j*

faktor *δ* unit tersembunyi :

*δj = δ\_netj f’(z\_netj )= δ\_netj­ zj (1-zj )*

Hitung suku perubahan bobot vji (yang akakn dipakai nanti untuk merubah bobot vji).

*∆vji = α δj xi ; j = 1,2,...,p ; i = 0,1,...,n*

Fase III : Perubahan bobot

*Langkah 8* : Hitung semua perubahan bobot

Perubahan bobot garis yang menuju ke unit keluaran :

*Wkj (baru) = wkj (lama) + ∆wkj (k= 1,2,...,m ; j = 0,1,...,p)*

Perubahan bobot gaaris yang menuju ke unit tersembunyi :

*Vji (baru) = vji (lama) + ∆vji (j = 1,2,....,p ; i = 0,1,....,n)*

Seteleah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini hanya propogasi maju (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaaran jaringan (Siang,2005).

**2.7 Suku Bunga Bank Indonesia *(BI RATE)***

Suku bunga Bank Indonesia *(BI Rate)* adalah suku bunga kebijakan yang mencerminkan sikap atau stance kebijakan moneter yang ditetapkan oleh Bank Indonesia dan diumumkan kepada publik (Bank Indonesia, 2016). *BI Rate* diumumkan oleh Dewan Gubernur Bank Indonesia setiap Rapat Dewan Gubernur (RDG) bulanan dan diimplementasikan pada operasi moneter yang dilakukan Bank Indonesia melalui pengelolaan likuiditas (liquidity management) di pasar uang untuk mencapai kebijakan moneter. Sasaran operasional kebijakan moneter dicerminkan pada perkembangan suku bunga Pasar Uang Antar Bank Overnight (PUAB O/N). Pergerakan di suku bunga PUAB ini diharapkan akan diikuti oleh perkembangan di suku bunga deposito, dan pada gilirannya suku bunga kredit perbankan. Dengan mempertimbangkan pula faktor-faktor lain dalam perekonomian, Bank Indonesia pada umumnya akan menaikkan BI Rate apabila inflasi ke depan diperkirakan melampaui sasaran yang telah ditetapkan, sebaliknya Bank Indonesia akan menurunkan BI Rate apabila inflasi ke depan diperkirakan berada di bawah sasaran yang telah ditetapkan.

Bank Indonesia melakukan penguatan kerangka operasi moneter dengan memperkenalkan suku bunga acuan atau suku bunga kebijakan baru yaitu BI 7-Day Repo Rate, yang akan berlaku efektif sejak 19 Agustus 2016. Selain BI Rate yang digunakan saat ini, perkenalan suku bunga kebijakan yang baru ini tidak mengubah stance kebijakan moneter yang sedang diterapkan.

**BAB III**

**METODOLOGI PENELITIAN**

* 1. **Kerangka Penelitian**

Salah satu *metode* yang dapat memudahkan seorang peneliti dalam menyelesaikan laporan dan penelitian adalah dengan menyusun sebuah kerangka penelitian. Adapun kerangka penelitian yang penulis lakukan dapat tergambar seperti bagan dibawah ini.

**Pendahuluan Penelitian**

**Pengumpulan Data**

**Analisa**

**Perancangan**

**Pengujian**

**Implementasi**

Gambar 3.1 Kerangka Penelitian

* 1. **Tahapan penelitian**

Tahapan penelitian adalah tahapan yang akan dilakukan untuk mempermudah dalam melakukan penelitian. Adapun tahapan-tahapan dalam penelitian ini adalah sebagai berikut:

* + 1. **Pendahuluan Penelitian**

Pendahuluan penelitian ini yang harus dilakukan adalah identifikasi masalah dilakukan dengan mencari tahu di jurnal dan buku yang berhubungan dengan Peramalan dan Prediksi BI Rate yang nantinya dapat dijadikan acuan dalam merancang aplikasi JST*.* Tujuan dari tahap ini adalah untuk mengetahui permasalahan yang terjadi secara cepat, sehingga diharapkan penelitian dapat memberikan solusi yang paling optimal terhadap pemecahan permasalahan tersebut.

* + 1. **Pengumpulan Data**

Dalam melakukan pengumpulan data penulis mendapatkan data dari berbagai sumber. Data dalam penelitian ini di peroleh dari *jurnal*, buku, website resmi Bank Indonesia dan data yang di peroleh dari *referensi* lainnya.

**Tabel 3.1 Waktu Penelitian**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Kegiatan** | **Bulan Ke** | | | | | | | | | | | | | | | | | | | |
| **1** | | | | **2** | | | | **3** | | | | **4** | | | | **5** | | | |
| 1 | Pendahuluan  Penelitian |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Pengumpulan Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Analisa Data dan Sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Perancangan aplikasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Pengujian |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

* + - 1. **Metode Penilitian**

Dalam hasil ini metode yang digunakan adalah kualitatif, yaitu dengan cara membandingkan teori-teori yang ada dengan permasalahan yang terjadi beberapa bulan terdahulu atau menggunakan data masa lampau dan kemudian mengadakan pembahasan terhadap masalah yang ada berdasarkan teori-teori tersebut. Dalam pengumpulan data dan informasi untuk penulisan penilitian ini dilakukan dengan beberapa cara yaitu :

1. Penelitian perpustakaan (*Library Research*)

Metode ini, penulis lakukan dengan mengumpulkan data dari buku-buku pemograman, jurnal-jurnal bahasa pemograman, *PHP, MySQL,* rekayasa perangkat lunak dan kepustakaan lainnya yang berkaitan dengan analisa dan perancangan aplikasi JST untuk mendapatkan bahan-bahan yang secara ilmiah dapat dijadikan landasan dalam menyusun penelitian ini

1. Penelitian laboratorium(*Laboratory Reasearc*)

Suatu metode penilitian yang dilakukan dengan menggunakan alat bantu *personal komputer* (PC). Dalam hal ini penelitian dilakukan dengan merancang program atau peranghat lunak yang sesuai dengan topik dan permasalahan yang dihadapi dan juga dalam hal penyusunan laporan secara keseluruhan. Dengan adanya teknik-teknik pengumpulan data, maka penulis mempraktekannya dan berusahan mengumpulkan data-data yang dipelukan sebanyak mungkin.

Penelitan yang dilakukan dalam aplikasi ini menggunakan komputer dengan spesifikasi sebagai berikut :

1. Kebutuhan *Hardware*
2. *HP Pavilion G4*
3. *Processor Intel®* C*ore(TM)* i5-3210M CPU 2.5 *Ghz (4CPU, -) 2.25Ghz*
4. *Memory RAM 4 GB*
5. *Harddisk 500 GB*
6. *Keyboard internal* dan *mouse*
7. *Printer Canon IP 1270*
8. Media penyimpanan *flashdisk* 16 GB
9. Kebutuhan *Software*
10. *Sistem Operasi Windows 7 beserta aplikasinya*
11. *Microsoft Office 2007*
12. *Mowes Portable*
13. *PHP My SQL*
14. *Notepad ++*
15. *Adobe Dreamweaver CS6*
16. *Rational Rose*
17. *Mozila Firefox*
    * 1. **Analisa**

Pada tahap *analisa,* semua *aspek* harus dipertimbangkan dalam pembuatan sebuah *aplikasi.* Tahap analisa nya adalah sebagai berikut :

1. Analisa Data

Analisa ini dilakukan untuk membatasi objek yang akan diteliti agar menjadi sebuah informasi yang lebih sistematis dan mudah dimengerti. Tahap analisa data merupakan tahap yang paling penting dalam pengembangan sebuah system aplikasi. Pada tahap inilah akan dilakukan evaluasi dan identifikasi terhadap masalah yang ada.

1. Analisa proses

Analisa ini dilakukan untuk mengetahui bagaimana pemecahan masalah sehingga dapat mengahasilkan solusi dengan menggunakan metode yang tepat. Metode yang digunakan dalam penelitian ini adalah metode *Algoritma Backpropagation,* Algoritma ini akan membentuk sebuah prediksi dengan cara menggunakan fungsi interval [0.1, 0.9]. karena pembuatan aplikasi iniditujukan untuk prediksi dan permalan suku bunga Bank Indonesia (BI Rate) dengan jaringan syaraf tiruan backpropagation.

1. Analisa Sistem

Analisa ini dilakukan untuk mengetahui apa saja yang dibutuhkan dalam perancangan sistem. Sehingga menghasilkan sebuah sistem yang efektif dan efesian dalam pengimplementasiannya nanti.

* + 1. **Perancangan**

Pada tahap ini akan membuat sebuah perancangan sistem yang akan dijalankan, mulai dari menganalisa program yang sedang berjalan, dan merancang program yang akan kita jalankan tersebut. Pada tahap ini juga dilakukan pengumpulan data-data yang mendukung perancangan sistem.

* + - 1. **Perancangan Model**

Pada tahap ini juga dilakukan pengumpulan fakta-fakta yang mendukung perancangan sistem. Dengan menggunakan UML *(­Unified Modelling language)* sebagai *tools* dalam menjelaskan alur analisa program.

Adapun UML *(­Unified Modelling language)* yang akan digunakan adalah sebagai berikut:

1. *Use Case Diagram*

Dalam aplikasi Jaringan Syaraf Tiruan akan di bangun use case diagramnya yang terdiri dari satu aktor yaitu admin. Dimana user ini akan menjadi pengguna aplikasi JST yang dapat melakukan proses pengoperasian aplikasi tersebut,

1. *Class Diagram*

Pada diagram *class* menggambarkan struktur sistem dari segi pendefenisisan kelas-kelas yang akan dibuat. Diagram ini akan memperlihatkan dengan jelas struktur dari setiap tabel yang ada serta menunjukkan hubungan antar tabel.

1. *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan langkah-langkah pada use case diagram yang akan dilakukan sebagai respon dari sebuah aplikasi yang menghasilkan outputnya. Pada aplikasi Jaringan Syaraf Tiruan Backpropagation ini yang akan dibangun sequence diagram untuk menggambarkan langkah-langkah yang terjadi serta respon dari sistem pada setiap prosesnya.

1. *Collaboration Diagram*

*Collaboration Diagram* lebih menekankan kepada peran setiap objek dan pada waktu penyampain pesan. Jadi pada aplikasi Jaringan Syaraf Tiruan Backpropagation ini akan di bangun collaboration diagram yang akan menggambarkan interaksi user dengan sistem sesuai dengan urutan proses yang terjadi pada saat user menggunakan aplikasi ini

1. *Statechart Diagram*

*Statechart diagram* akan menggambarkan perubahan status yang akan terjadi di aplikasi Jaringan Syaraf Tiruan Backpropagation ini saat aplikasi ini dijalankan. Perubahan yang terajadi pada objek akan digambarkan oleh diagram ini dalam bentuk graf berarah.

1. *Activity Diagram*

*Activity diagram* untuk menggambarkan jalur kerja aliran aktifitas dalam sistem yang dirancang, mulai dari aliran berawal, *deccision* yang mungkin terjadi , dan bagaimana aliran berakhir. Pada aplikasi Jaringan Syaraf Tiruan Backpropagation ini, *Activity diagram* akan menunjukan proses dari awal (*start*) pada saat aplikasi dijalakan kemudian proses yang terjadi selama aplikasi digunakan, hingga stop atau aplikasi selesai digunakan.

1. *Deployment Diagram*

*Deployment diagram* menggambarkan detail bagaimana komponen komponen di-*deploy* dalam infrastruktur system, dimana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaiman kemampuan jaringan pada lokasi tersebut, spesifikasi *server,* dan hal-hal lain yang bersifat fisikal. Deployment diagram dari aplikasi Jaringan Syaraf Tiruan Backpropagation ini akan menggambarkan perempatan komponen-komponen, mulai dair proses serta hubungan antara peralatan yang satu dengan yang lainnya.

* + - 1. **Perancangan *Interface***

Perancangan *interface* dibuat untuk memberikan penjelasan tentang tampilan yang dihadapkan pada pengguna saat menggunakan sistem. Pada tahap ini, peneliti melakukan perancangan terhadap tampilan atau desain antarmuka (interface) dari Jaringan Syaraf Tiruan Backpropagation ini. Perancangan yang akan dilakukan meliputi perancangan *layout* halaman-halaman yang ada pada aplikasi ini.

* + 1. **Pengujian**

Pengujian sistem akan meggambarkan bagaimana sebuah sistem berjalan. Tahapan pengujian ini adalah sebagai berikut :

1. Pungujian Aplikasi

Pengujian aplikasi dilakukan agar aplikasi web yang dibuat sesuai dengan harapan, dan bagaimana proses dari aplikasi ini memprediksi BI Rate dengan akurat.

1. Pengujian interface

Membuat tampilan sebuah aplikasi agar dapat semenarik mungkin dan mudah digunakan oleh user nantinya.

* + 1. **Implementasi**

Untuk melakukan kegiatan implementasi dan tindak lanjut implementasi, untuk mengimplementasikan prograam aplikasi yang telah dirancang, maka diperlukan sebuah alat bantu komputer, yang mana untuk mengoperasikan komputer itu sendiri memerlukan tuga buah komponen pendukung sebagai berikut :

1. Perangkat Keras (*hardware*)

*Hardware* yang digunakan untuk merangcang atau menjalankan program aplikasi yang telah dibuat dalam satu unit komputer yang lengkap dengan CPU, hardisk sebagai media penyimpanan data yang akan di instalkan pada komputer, memory standar dengan ukuran 1GB.

1. Perangkat Lunak (*Software*)

Untuk menjalankan program aplikasi yang dirancang harus menggunakan beberapa software pendukung. Software-software pendukung yang harus di instal adalah Mowes Portable dan *Adobe Dreamweaver CS6*  dan Notepad ++ yang berfungsi untuk merancang desain program aplikasi.

1. Manusia (*Brainware*)

*Brainware* merupakan operator yang berfungsi untuk mengoperasikan atau menjalankan program aplikasi

**BAB IV**

**HASIL DAN PEMBAHASAN**

* 1. **Pendahuluan**

Untuk mendapatkan suatu metode yang baik dalam peramalan digunakan tingkat ketelitian sebagai ukuran.Semakin tinggi tingkat ketelitian yang didapat maka semakin baik metode tersebut.Setelah dilakukan penelitian dan pengujian didapat hasil-hasil penelitian dengan *klasifikasi* terhadap *error* maksimum yang dapat tercapai. Besarnya nilai *error* yang dicapai tersebut menentukan seberapa besar nilai ketepatan yang dapat dilakukan dalam peramalan. Untuk serangkaian data yang akan ditentukan nilainya, berbeda tingkat penyimpangannya yang secara terakumulasi membentuk hasil peramalan tersebut. Untuk mendapatkan analisa, maka hasil-hasilnya ditampilkan dalam bentuk grafik. Pembentkan pola peramalan yang terbentuk secara implisit oleh Jaringan Syaraf Tiruan terlihat pada grafik.

* 1. **Proses Analisa**
     1. **Proses Analisa Data**

Analisa sistem merupakan suatu uraian dari sistem informasi yang utuh menjadi suatu bagian yang terbentuk dari kompenan yang berguna untuk mengidentifikasi dan mengevaluasi permasalahan, hambatan yang terjadi sehingga dapat diusulkan suatu perbaikan dalam mengembangkan sistem dengan menggunakan perangkat lunak (software) Jaringan Syaraf Tiruan dengan metode *backpropagation* dalam memprediksi dan meramalkan *BI rate* kedepanya.

Data-data atribut yang digunakan adalah sebagai berikut :

**Tabel 4.1 Data Atribut**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Variable** | **Simbol** | **Sumber** |
| Kurs IDR/USD | Input | X1 | [www.bi.go.id](http://www.bi.go.id) |
| Jumlah Uang Beredar (M2) | Input | X2 | [www.bi.go.id](http://www.bi.go.id) |
| Inflansi | Input | X3 | [www.bi.go.id](http://www.bi.go.id) |
| IHSG | Input | X4 | www.fbs.id |
| BI Rate | Output | T | [www.bi.go.id](http://www.bi.go.id) |

**Tabel 4.2 Data Makro Ekonomi**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tahun** | **Bulan** | **Kurs** | **JUB (M2)** | **Inflansi** | **IHSG** | **BI Rate** |
| **(IDR/USD)** | **(Miliaran)** | **(%)** | **(Juta Rp)** | **(%)** |
| 2016 | Mar | 13.276k | 4.562k | 3.61 | 4.845k | 5.5 |
| 2016 | Apr | 13.204k | 4.582k | 3.83 | 4.839k | 5.5 |
| 2016 | May | 13.615k | 4.614k | 3.49 | 4.797k | 5.5 |
| 2016 | Jun | 13.180k | 4.738k | 3.02 | 5.017k | 5.25 |
| 2016 | Jul | 13.094k | 4.730k | 3.58 | 5.216k | 5.25 |
| 2016 | Aug | 13.615k | 4.746k | 3.31 | 5.386k | 5.25 |
| 2016 | Sep | 12.998k | 4.738k | 3.07 | 5.365k | 4.75 |
| 2016 | Oct | 13.051k | 4.779k | 2.79 | 5.423k | 4.75 |
| 2016 | Nov | 13.563k | 4.869k | 3.21 | 5.149k | 4.75 |
| 2016 | Dec | 13.436k | 5.005k | 3.45 | 5.103k | 4.75 |
| 2017 | Jan | 13.343k | 4.937k | 3.33 | 5.303k | 4.75 |
| 2017 | Feb | 13.343k | 4.938k | 3.6 | 5.383k | 4.75 |

Data-data yang diperlukan dalam perancangan aplikasi Jaringan Syaraf Tiruan adalah hasil penelitian murni dengan mengambil variabel dari nilai-nilai yang bisa dijadikan prediktor untuk memprediksi *BI Rate* antara lain meliputi pergerakan kurs dollar, jumlah uang beredar, tingkat inflansi perbulan, ihsg perbulan. Selanjutnya data dibagi dua yaitu, data yang digunakan sebagai pelatihan dan data yang digunakan sebagai pengujian.

* + 1. **Proses Analisa Sistem**

Proses analisa sistem dilakukan berdarkan data yang telah diperoleh dari hasil penelitian yang telah diperoleh. Pada tahap ini data yang telah ada akan dibagi menjadi data pelatihan dan data pengujian.

**4.3 Data Jaringan Syaraf Tiruan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO | X1 | X2 | X3 | X4 | T |
| 1 | 13.276 | 4.582 |  | 4.839 | 5.5 |
| 3.61 |
| 2 | 13.204 | 4.614 |  | 4.797 | 5.5 |
| 3.83 |
| 3 | 13.615 | 4.738 |  | 5.017 | 5.5 |
| 3.49 |
| 4 | 13.18 | 4.73 |  | 5.216 | 5.25 |
| 3.02 |
| 5 | 13.094 | 4.746 |  | 5.386 | 5.25 |
| 3.58 |
| 6 | 13.615 | 4.738 |  | 5.365 | 5.25 |
| 3.31 |

* + - 1. **Proses Pembagian Data**

1. **Data Pelatihan**

Berikut merupakan data data yang akan dijadikan sebagai pelatihan dalam mengiplementasikan *Jaringan Syaraf Tiruan* untuk memprediksi Pergerakan Suku Bunga acuan BI. Data yang digunakan adalah data dari bulan maret sampai agustus 2016.

**Tabel 4.4 Data Input Pelatihan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO | X1 | X2 | X3 | X4 | T |
| 1 | 13.276 | 4.582 |  | 4.839 | 5.5 |
| 3.61 |
| 2 | 13.204 | 4.614 |  | 4.797 | 5.5 |
| 3.83 |
| 3 | 13.615 | 4.738 |  | 5.017 | 5.5 |
| 3.49 |
| 4 | 13.18 | 4.73 |  | 5.216 | 5.25 |
| 3.02 |
| 5 | 13.094 | 4.746 |  | 5.386 | 5.25 |
| 3.58 |
| 6 | 13.615 | 4.738 |  | 5.365 | 5.25 |
| 3.31 |

1. **Data Pengujian**

Berikut merupakan data data yang akan dijadikan sebagai pelatihan dalam mengplementasikan Jaringan Syaraf Tiruan untuk memprediksi BI rate .Data yang digunakan adalah data dari bulan September 2016 sampai Februari 2017:

**Tabel 4.5 Data Input Pengujian**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO** | **X1** | **X2** | **X3** | **X4** | **T** |
| 1 | 12.998 | 4.738 |  | 5.365 | 4.75 |
| 3.07 |
| 2 | 13.051 | 4.779 |  | 5.423 | 4.75 |
| 2.79 |
| 3 | 13.563 | 4.869 |  | 5.149 | 4.75 |
| 3.21 |
| 4 | 13.436 | 5.005 |  | 5.103 | 4.75 |
| 3.45 |
| 5 | 13.343 | 4.937 |  | 5.303 | 4.75 |
| 3.33 |
| 6 | 13.343 | 4.938 |  | 5.383 | 4.75 |
| 4.45 |

* + - 1. **Proses Transformasi**

Fungsi *aktivasi* yang digunakan untuk mengolah data diatas adalah fungsi *sigmoid* (*biner*). Maka data harus *ditransformasikan* dulu karena *range*keluaran fungsi *aktivasi sigmoid* adalah [0,1]. Data ditransformasikan ke *interval* yang lebih kecil, misalkan pada interval [(0.1), (0.9)] adalah :

X = + 0.1

Dimana:

x : nilai

a : data minimun

b : data maksimum

Dengan *transformasi* ini maka data terkecil akan menjadi 0.1 dan terbesar akan menjadi 0.9 .pencarian dibawah ini menunjukan hasil transformasi untuk data pelatihan yang nantinya dipakai sebagai data pelatihan backpropagation.

**Tabel 4.6 Proses Transformasi Data Input Pelatihan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO** | **X1** | **X2** | **X3** | **X4** | **T** |
| 1 | 13.276 | 4.582 |  | 4.839 | 5.5 |
| 3.61 |
| 2 | 13.204 | 4.614 |  | 4.797 | 5.5 |
| 3.83 |
| 3 | 13.615 | 4.738 |  | 5.017 | 5.5 |
| 3.49 |
| 4 | 13.18 | 4.73 |  | 5.216 | 5.25 |
| 3.02 |
| 5 | 13.094 | 4.746 |  | 5.386 | 5.25 |
| 3.58 |
| 6 | 13.615 | 4.738 |  | 5.365 | 5.25 |
| 3.07 |

a : data minimum = 3.02

b : data maksimum = 13.615

X1 = 0.8\*(13.276–3.02)/(13.615–3.02)+0.1= 0.8744

0.8\*(13.204–3.02)/(13.615 –3.02) + 0.1 =0.8689

0.8\*(13.615–3.02)/(13.615 –3.02)+0.1 = 0.9000

0.8 \* ( 13.18–3.02) / (13.615 –3.02) + 0.1 = 0.8671

0.8 \* (13.094–3.02) / (13.615 –3.02) + 0.1 = 0.8606

0.8 \* (13.615–3.02) / (13.615 –3.02) + 0.1 = 0.9000

X2 = 0.8\*(4.582–3.02)/(13.615 –3.02)+0.1= 0.2179

0.8\*(4.614–3.02)/(13.615 –3.02)+0.1=0.2203

0.8\*(4.738–3.02)/(13.615 –3.02)+0.1 =0.2297

0.8\*(4.73–3.02)/(13.615 –3.02) + 0.1 =0.2291

0.8 \* (4.746–3.02) / (13.615 –3.02) + 0.1 = 0.2303

0.8 \* (4.738–3.02) / (13.615 –3.02) + 0.1 = 0.2297

X3 = 0,8\*(3.61–3.02)/(13.615 –3.02)+0.1 = 0.1441

0.8\*(3.83–3.02)/(13.615 –3.02) + 0.1 =0.1661

0.8\*(3.49–3.02)/(13.615 –3.02)+ 0.1 = 0.1354

0.8\*(3.02–3.02)/(13.615 –3.02)+ 0.1 =0.1000

0.8 \* (3.58–3.02) / (13.615 –3.02) + 0.1 = 0.1422

0.8 \* (3.31–3.02) / (13.615 –3.02) + 0.1 = 0.1218

X4 = 0.8\*(4.839–3.02)/(13.615 –3.02)+0.1 = 0.2373

0.8\*(4.797–3.02)/(13.615 –3.02) + 0.1 =0.2341

0.8\*(5.017–3.02)/(13.615 –3.02)+ 0.1 = 0.2507

0.8\*(5.216–3.02)/(13.615 –3.02)+ 0.1 =0.2658

0.8 \* (5.386–3.02) / (13.615 –3.02) + 0.1 = 0.2786

0.8 \* (5.365–3.02) / (13.615 –3.02) + 0.1 = 0.2770

Selanjutnya data target pelatihan yang digunakan sebagai *output layer* dan dilakukan proses transformasi sebagai berikut :

T = 0.8\*(5.5 –3.02)/(13.615 –3.02)+0.1 = 0.2872

0.8\*(5.5 –3.02)/(13.615 –3.02) + 0.1 =0.2872

0.8\*(5.5 –3.02)/(13.615 –3.02)+ 0.1 = 0.2872

0.8\*(5.25 –3.02)/(13.615 –3.02)+ 0.1 =0.2683

0.8 \* (5.25 –3.02) / (13.615 –3.02) + 0.1 = 0.2683

0.8 \* (5.25 –3.02) / (13.615 –3.02) + 0.1 = 0.2683

Setelah mendapatkan nilai dari hasil proses tansformasi data pelatihan di atas maka di dapatkan nilai paling rendah 0.1 dan nilai tertinggi 0.9 .bisa di lihat dari tabel di bawah ini :

**Tabel 4.7 Hasil Transformasi Data Input Pelatihan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **X1** | **X2** | **X3** | **X4** | **T** |
| 0.8744 | 0.2179 | 0.1445 | 0.2373 | 0.2872 |
| 0.8689 | 0.2203 | 0.1661 | 0.2341 | 0.2872 |
| 0.9 | 0.2297 | 0.1354 | 0.2507 | 0.2872 |
| 0.8671 | 0.2291 | 0.1 | 0.2658 | 0.2683 |
| 0.8606 | 0.2303 | 0.1422 | 0.2786 | 0.2683 |
| 0.9 | 0.2297 | 0.1218 | 0.277 | 0.2683 |

Selanjutnya pencarian data input pengujian, dimana dibawah ini menunjukan hasil transformasi untuk data pengujian yang nantinya dipakai sebagai data pengujian *Backpropagation*.

**Tabel 4.8 Proses Transformasi Data Pengujian**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO** | **X1** | **X2** | **X3** | **X4** | **T** |
| 1 | 12.998 | 4.738 |  | 5.365 | 4.75 |
| 3.07 |
| 2 | 13.051 | 4.779 |  | 5.423 | 4.75 |
| 2.79 |
| 3 | 13.563 | 4.869 |  | 5.149 | 4.75 |
| 3.21 |
| 4 | 13.436 | 5.005 |  | 5.103 | 4.75 |
| 3.45 |
| 5 | 13.343 | 4.937 |  | 5.303 | 4.75 |
| 3.33 |
| 6 | 13.343 | 4.938 |  | 5.383 | 4.75 |
| 4.45 |

a : data minimum = 2.79

b : data maksimum = 13.563

X1 = 0.8 \* (12.998–2.79) / (13.563–2.79) + 0.1 = 0.8580

0.8 \* (13.051–2.79) / (13.563–2.79) + 0.1 = 0.8619

0.8 \* (13.563–2.79) / (13.563–2.79) + 0.1 = 0.9000

0.8 \* (13.436–2.79) / (13.563–2.79) + 0.1 = 0.9805

0.8 \* (13.343–2.79) / (13.563–2.79) + 0.1 = 0.8836

0.8 \* (13.343–2.79) / (13.563–2.79) + 0.1 = 0.8836

X2 = 0.8 \* (4.778–2.79) / (13.563–2.79) + 0.1 = 0.2446

0.8 \* (4.868–2.79) / (13.563–2.79) + 0.1 = 0.2477

0.8 \* (5.005–2.79) / (13.563–2.79) + 0.1 = 0.2543

0.8 \* (4.936–2.79) / (13.563–2.79) + 0.1 = 0.2644

0.8 \* (4.938–2.79) / (13.563–2.79) + 0.1 = 0.2594

0.8 \* (5.017–2.79) / (5017–2.79) + 0.1 = 0.2595

X3 = 0.8 \* (2.79–2.79) / (13.563–2.79) + 0.1 = 0.1207

0.8 \* (3.21–2.79) / (13.563–2.79) + 0.1 = 0.1000

0.8 \* (3.45–2.79) / (13.563–2.79) + 0.1 = 0.1311

0.8 \* (3.33–2.79) / (13.563–2.79) + 0.1 = 0.1490

0.8 \* (3.6–2.79) / (13.563–2.79) + 0.1 = 0.1401

0.8 \* (4.45–2.79) / (13.563–2.79) + 0.1 = 0.1605

X4 = 0.8 \* (5.422–2.79) / (13.563–2.79) + 0.1 = 0.2912

0.8 \* (5.148–2.79) / (13.563–2.79) + 0.1 = 0.2955

0.8 \* (5.102–2.79) / (13.563–2.79) + 0.1 = 0.2751

0.8 \* (5.302–2.79) / (13.563–2.79) + 0.1 = 0.2717

0.8 \* (5.382–2.79) / (13.563–2.79)+ 0.1 = 0.2866

0.8 \* (5.592–2.79) / (13.563–2.79) + 0.1 = 0.2925

Selanjutnya data target pengujian yang digunakan sebagai *output layer* dan dilakukan proses transformasi sebagai berikut :

T = 0.8\*(4.75–2.79)/(13.563–2.79) + 0.1 = 0.2455

0.8 \* (4.75–2.79) / (13.563–2.79) + 0.1 = 0.2455

0.8 \* (4.75–2.79) / (13.563–2.79) + 0.1 = 0.2455

0.8 \* (4.75–2.79) / (13.563–2.79) + 0.1 = 0.2455

0.8 \* (4.75–2.79) / (13.563–2.79) + 0.1 = 0.2455

0.8 \* (4.75–2.79) / (13.563–2.79) + 0.1 = 0.2425

Setelah mendapatkan nilai dari hasil proses tansformasi data pengujian di atas maka di dapatkan nilai paling rendah 0.1 dan nilai tertinggi 0.9 .bisa di lihat dari tabel di bawah ini :

**Tabel 4.9 Hasil Transformasi Data Input Pengujian**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO | X1 | X2 | X3 | X4 | T |
| 1 | 0.8580 | 0.2446 |  | 0.2912 | 0.2455 |
| 0.1207 |
| 2 | 0.8619 | 0.2477 |  | 0.2955 | 0.2455 |
| 0.1000 |
| 3 | 0.9000 | 0.2543 |  | 0.2751 | 0.2455 |
| 0.1311 |
| 4 | 0.8905 | 0.2644 |  | 0.2717 | 0.2455 |
| 0.1490 |
| 5 | 0.8836 | 0.2594 |  | 0.2866 | 0.2455 |
| 0.1401 |
| 6 | 0.8836 | 0.2595 |  | 0.2925 | 0.2455 |
| 0.1601 |

* + - 1. **Algoritma dan Arsitektur Jaringan**

Pada penelitian ini arsitektu Jaringan Syaraf Tiruan yang digunakan untuk prediksi *BI Rate* adalah Jaringan Syaraf Tiruan *Backpropagation* yang terdiri dari :

1. Lapisan *Input*

Dengan 4 simpul masing-masing untuk Kurs Dollar (*x1*), Jumlah Uang Beredar (*x2*), Inflansi(*x3*), IHSG(*x4*).

1. Lapisan Output

Dengan 1 simpul, yaitu hasil akhirnya berupa *BI Rate* (pola yang didapatkan) (*t*) sebagai nilai yang menjadi prediksi *BI Rate* .

1. Lapisan Tersembunyi

Denga jumlah simpul yang ditentukan oleh pengguna bentuk arsitektur   
jaringan syaraf tiruannya dapat dilihat pada gambar berikut :

v11

v12  **B**   
**X1 Z1** v21  **X2** v22 w1 v31 **Z2** Y v22 v32  **X3** v41 w2 *BI Rate*v42 **X4**    ***Input Layer Hidden Layer Output Layer***

**Gambar 4.1 Arsitektur Jaringan Syaraf Tiruan untuk permintaan Barang**

Keterangan :

*X* = Masukan (Input)

*J* = *1* s/d n( *n=4*)

*V* = Bobot pada lampisan tersembunyi

*W* = Bobot pada lapisan pengeluaran

*N* = Jumlah untuk pengelola pada lapisan tersembunyi

*B* = Bias pada lapisan tersembunyi dan lapisan keluaran

*K* = Jumlah unit pengelola pada lapisan keluaran

*Y* = Keluaran hasil

Algoritma pemrosesan yang digunakan adalah algoritma *Backpropagation* dengan fungsi aktivasi *sigmoid* yang memiliki rentang interval *0,1* dan *-1*

* + - 1. **Arsitektur Jaringan Untuk Memprediksi BI Rate**

Pola proses pelatihan ini, penulis akan mencoba melakuka pelatihan dnegan menggunakan beberapa pola yang mana sesuai dan akurat yan dapat menentukan kualifikasi factor yang tepat dalam memprediksi Bi Rate. pola-pola tersebut yaitu sebagai berikut :

Pola 1

1. Jumlah *neuron* pada input layer sebanyak 4 *neuron*
2. Jumlah *neuron* pada *hidden* layer sebanyak 1*neuron*
3. Toleransi *error* sebesar 0.01
4. *Learning Rate* sebesar 0.95
5. *Momentum* sebesar 0.5
6. *Max epoch* sebanyak 100,000

Pola 2

1. Jumlah *neuron* pada input layer sebanyak 4 *neuron*
2. Jumlah *neuron* pada *hidden* layer sebanyak 2 *neuron*
3. Toleransi *error* sebesar 0.01
4. *Learning Rate* sebesar 0.95
5. *Momentum* sebesar 0.5
6. *Max epoch* sebanyak 100,000

Pola 3

1. Jumlah *neuron* pada input layer sebanyak 4 *neuron*
2. Jumlah *neuron* pada *hidden* layer sebanyak 3*neuron*
3. Toleransi *error* sebesar 0.01
4. *Learning Rate* sebesar 0.95
5. *Momentum* sebesar 0.5
6. *Max epoch* sebanyak 100,000

Pola 4

1. Jumlah *neuron* pada input layer sebanyak 4 *neuron*
2. Jumlah *neuron* pada *hidden* layer sebanyak 4*neuron*
3. Toleransi *error* sebesar 0.01
4. *Learning Rate* sebesar 0.75
5. *Momentum* sebesar 0.5
6. *Max epoch* sebanyak 100,000

Pola 5

1. Jumlah *neuron* pada input layer sebanyak 4 *neuron*
2. Jumlah *neuron* pada *hidden* layer sebanyak 5 *neuron*
3. Toleransi *error* sebesar 0.01
4. *Learning Rate* sebesar 0.85
5. *Momentum* sebesar 0.5
6. *Max epoch* sebanyak 100,000
   * + 1. **Perancangan Algoritma *Backpropagation***

Pada tahap perancangan ini hasil analisi yang telah didapatkan akan diterapkan pada langkah langkah algoritma backpropagation menggunakan fungsi aktivasi *sigmoid*. Adapun langkah-langkah penggunaan algoritma backpropagation adalah :

1. Tahap *Initialization*

Merupakaan tahapan untuk mendefenisikan atau menset awal nilai untuk variabel-variabel yang diperlukan, seperti : nilai input, weigth, output yang diharapkan, *learning rate* dan sebagainya.

1. Tahap *Activation*

Pada tahap ini yang dilakukan adalah dua kegiatan yaitu menghitung *actualoutput* pada *hidden layer* dan menghitung *actual output* pada *output layer.*

1. Tahap *Weight Training*

Pada tahap ini yang dilkakukan ada dua kegiatan yaitu menghitung *error gradient* pada *output layer* dan menghitung *error gradient* pada *hidden layer*

1. Tahap *Iteration*

Tahapan terakhir ini adalah tahapan untuk pengujian dimana jika error yang dharapkan belum ditemukan maka akan kembali lagi ke tahapan ke dua yaitu tahap *activation*

Berikut ini diberikan contoh perhitungan pelatihan atau peramalan dengan jaringan syaraf tiruan untuk menentuknan prediksi *BI Rate*, sebagai contoh pelatihan menggunakan 4 buah variabel input *vector x,*yaitu :

X1 = Kurs Dollar

X2 = Jumlah Uang Beredar

X3 = Inflansi

X4 = IHSG

Data yang digunakan dalam proses Algoritma *Backpropagation* adalah data pelatihan yang sudah di Transformasi kedalam bentuk variabel data Jaringan Syaraf Tiruan Algoritma *Backpropagation.* Berikut data yang ada dibawah ini:

**Tabel 4.10 Data Input**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **X1** | **X2** | **X3** | **X4** | **T** |
| 0.8744 | 0.2179 | 0.1445 | 0.2373 | 0.2872 |
| 0.8689 | 0.2203 | 0.1661 | 0.2341 | 0.2872 |
| 0.9 | 0.2297 | 0.1354 | 0.2507 | 0.2872 |
| 0.8671 | 0.2291 | 0.1 | 0.2658 | 0.2683 |
| 0.8606 | 0.2303 | 0.1422 | 0.2786 | 0.2683 |
| 0.9 | 0.2297 | 0.1218 | 0.277 | 0.2683 |

Arsitektur jairngan yang akan dipilih dimisalkan terdiri dari jumlah input layer terdiri dari 4*neuron* yang variabelnya adalah inputan dari data data yang akan di gunakan untuk memprediksi dan nilai *BI Rate* sebagai target / output (T).

Untuk membentuk Jaringan Syaraf Tiruan, terlebih dahulu dilakukan inisialisasi bobot awal, bobot awal yang menghubungkan simpul-simpul pada lapisan input x= (x1,x2,x3,x4) dan lapisan tersembunyi untuk arsitektur di atas adalah v = (v11, v12, v21, v22, v31, v32, v41, v42) dan bobot bias di pilih secara acak, demikian pula bobot awal yang menghubungkan simpul-simpul pada lapisan tersembunyi dan lapisan output (w1, w2) juga dipilih secara acak.

**Tabel 4.11 Bobot awal dan bias *input* ke *hidden* dari matlab**

|  |  |  |
| --- | --- | --- |
|  | **Z1** | **Z2** |
| **X1** | 96.3588 | 97.0264 |
| **X2** | -362.88 | 314.055 |
| **X3** | 24.1552 | -57.359 |
| **X4** | -60.045 | 9.9249 |
| **Bias** | 5.3452 | -147.38 |

**Tabel 4.12 Bobot awal dan *hidden* ke *output* dari matlab**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Z1** | **Z2** | **BIAS** |
| **Y** | 0.7061 | 0.2441 | 0.0265 |

Diketahui

X1 = 0.8744 X2 = 0.2179 X3 = 0.1445 X4 = 2373

V11 = 96.3588 V12 = 97.0264

V21 = -362.878 V22 = 314.0554

V31 = 24.1552 V32 = -57.3593

V41 = -60.0448 V42 = 9.9249

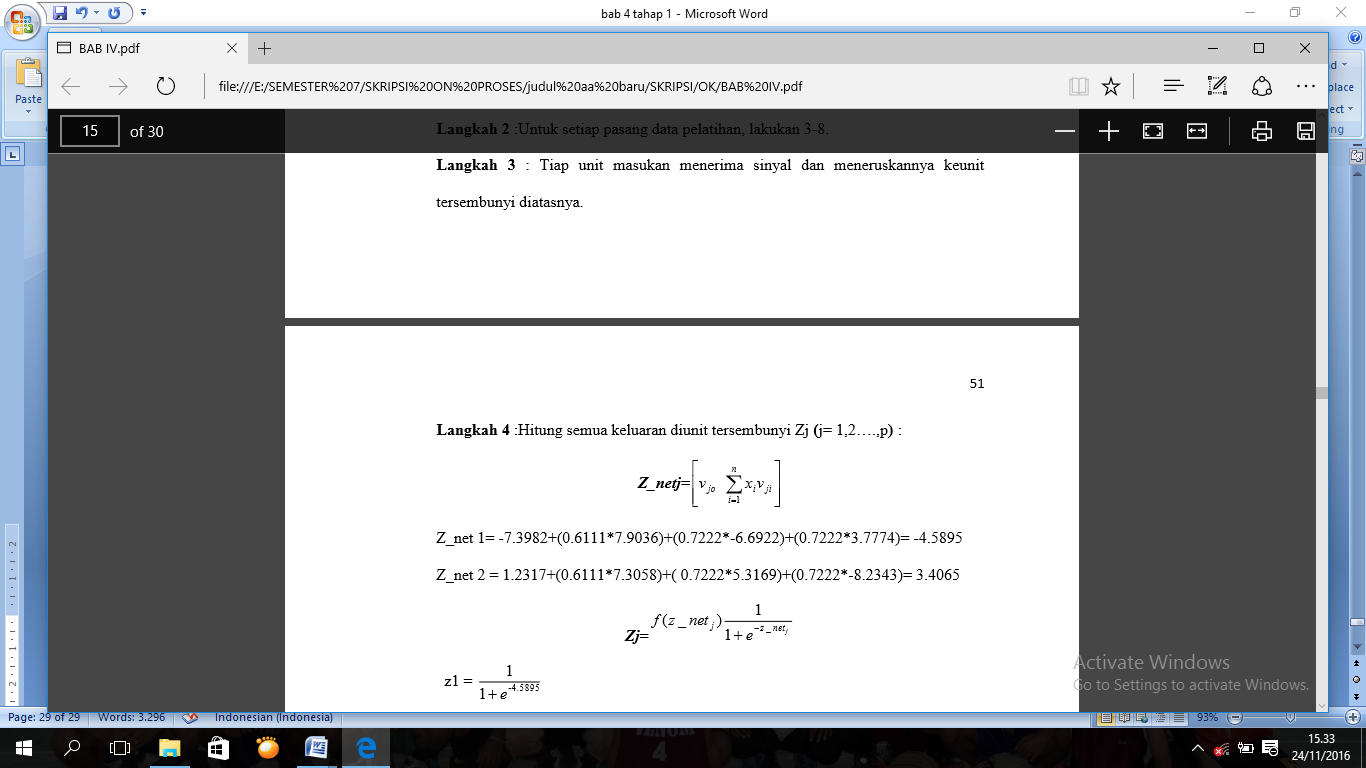
W11 = 5.3452 W12 = -147.375

Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil.

Langkah 1 : Jika kondisi perhitungan belum terpenuhi, lakukan langkah   
 2-9.

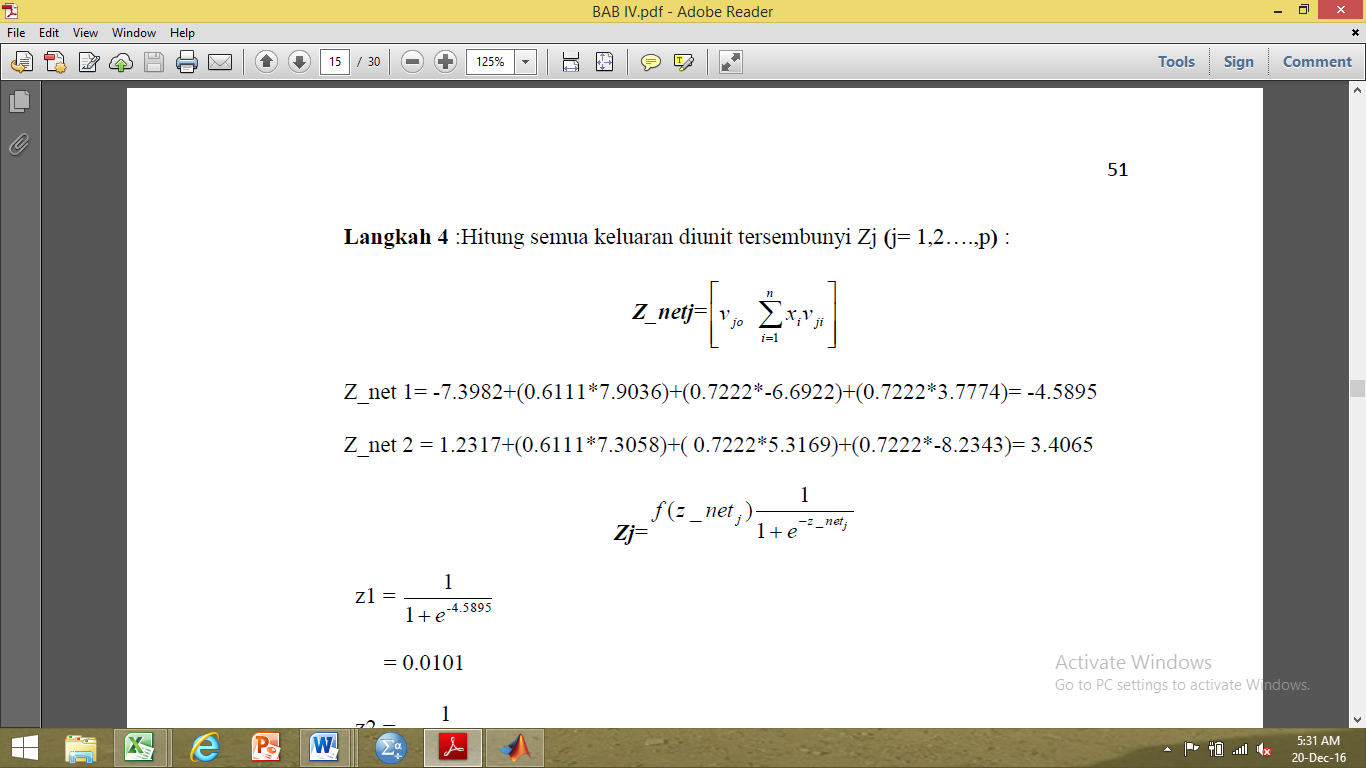
Langkah 2 : Untuk setiap pasang data pelatihan, lakukan 3-8.

Langkah 3 : Tiap unit masukan menerima sinyal dan meneruskannya   
keunit tersembunyi diatasnya.

Langkah 4 : Hitung semua keluaran diunit tersembunyi Zj (j= 1,2….,p).

Z\_net 1= + ( 0.8744\*96.3588) + (0.2178\*-362.878) + (0.7116\*24.1552) + (0.2373-60.0448) = - 0.2279

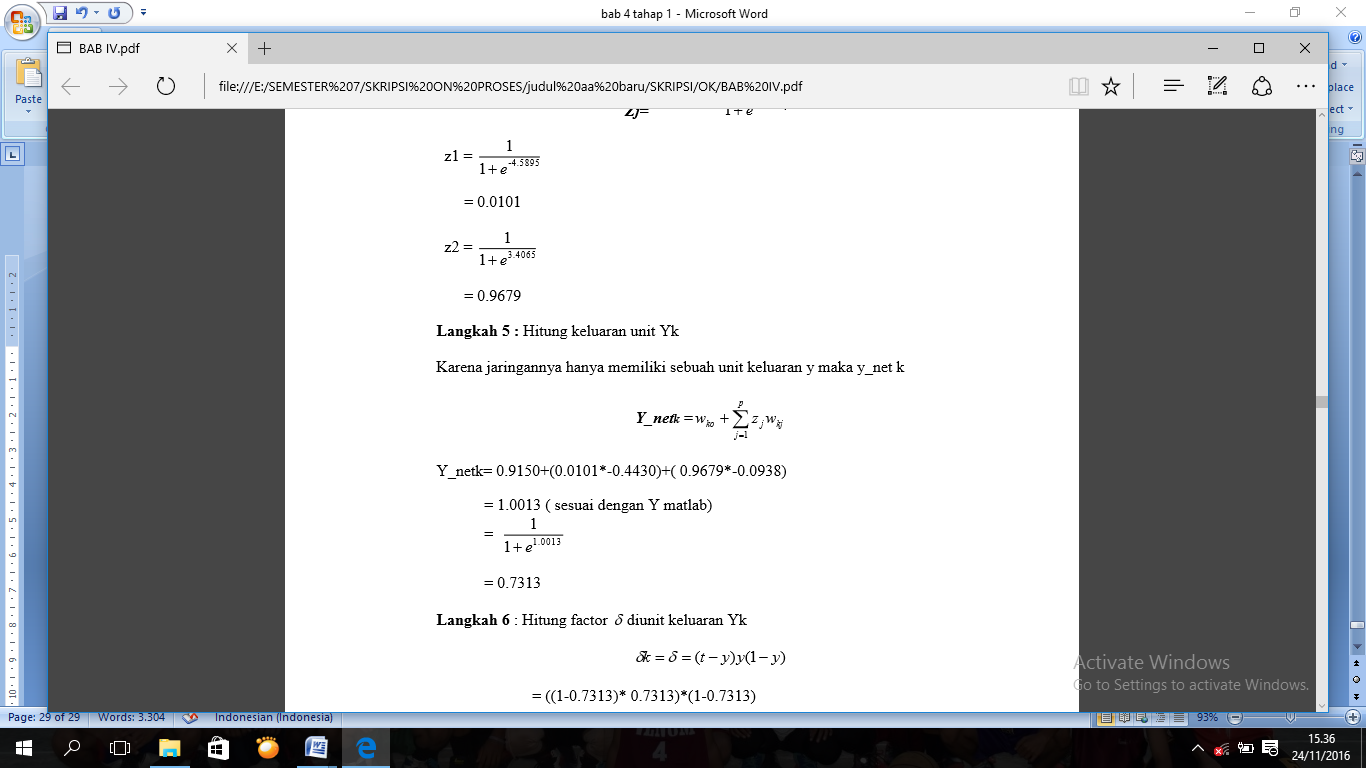
Z\_net 2= -147.375+( 0.8744\*97.0264) + (0.2178314.0554) + (0.7116-57.3593)+ (0.2373\*9.9249) = - 0.03568



Z1 = = 0.04432

Z2 = 0.4910

Langkah 5 : Hitung keluaran unit Yk

Karena jaringannya hanya memiliki sebuah unit keluaran y maka y\_net k

Y\_net = +( 5.3452 \*0.04432) + (-147.375\*0.4910)

= 0.1773

=

= 0.544229

Melakukan pencheckan terhadap error, iterasi akan berhenti bila |error|< 0.01.

*Error*  = (0.2872 – 0.544229) = -0.2570

Langkah 6 **:**hitung faktor diunit keluaran yk

= (0.2872 –0.544229) \* (0.544229 (1 – 0.544229))

= -0.1171

Suku perubahan bobot Wkj :

0.95 \* (-0.1171) \* 1 = -0.1112

0.95 \* (-0.1171) \* 0.0525= -0.0058

= 0.95 \* (-0.1171) \* 0.0026= -0.0002

Langkah 7 : Hitung penjumlahan kesalahan dari unit tersembunyi (

\_net1 = -0.1171\* 0.915=-0.1071

\_net2 = -0.1171\* -147.375 =17.2576

Faktor kesalahan di unit tersebunyi :

1 = 0.1071\* 0.04432\* (1 –0.04432) = 0.0045

2 = 17.2576\* 0.4910\* (1 –0.4910) = 4.3130

**Tabel 4.12 Perubahan bobot input ke hidden**

|  |  |  |
| --- | --- | --- |
|  | Z1 | Z2 |
| X1 | (0.95)\*  (0.0045)\*( 0.8744) =  0.00374 | (0.95)\*  (4.3130)\*( 0.8744) =  3.5827 |
| X2 | (0.95)\*  (0.0045)\*( 0.2179) =  0.0009 | (0.95)\*  (4.3130)\*( 0.2179) =  0.8928 |
| X3 | (0.95)\*  (0.0045)\*( 0.1445) =  0.0006 | (0.95)\*  (4.3130)\*( 0.1445) =  0.5920 |
| X4 | (0.95)\*  (0.0045)\*( 0.2373) =  0.0010 | (0.95)\*  (4.3130)\*( 0.2373) =  0.9723 |

Langkah 8 : Hitung semua perubahan bobot Perubahan bobot unit   
 keluaran :

+ (-0.1112) = 5.234

-147.375+ (-0.0058) = -143.3808

+ (-0.0002) = -0.685

**Tabel 4.13 Perubahan bias dan bias baru pada neuran hidden**

|  |  |  |  |
| --- | --- | --- | --- |
| **Perubahan Bias Neuron Hidden** | | **Bias Baru Pada Neuron Hidden Layer** | |
|  | 0.95 \* 0.0045 =  0.004275 |  | 0.004275+ 0.04432=  0.4859 |
|  | 0.95 \* 4.3130 =  4.0973 |  | 4.0973+ 0.4910=  4.5883 |

**Tabel 4.14 Perubahan bobot input baru**

|  |  |  |
| --- | --- | --- |
| **Baru** | **Z1** | **Z2** |
| X1 | 96.3588+ 0.00374) = 96.3625 | 97.0264+ (3.5827) = 100.6091 |
| X2 | -362.878+(0.0009) =  -363.8771 | 314.0554 + (0.8928) = 314.9482 |
| X3 | 24.1552+ (0.0006) =  24.1558 | -57.3593 + (0.5920) =-  -56.7673 |
| X4 | -60.0448+ (0.0010) =  -60.0438 | 9.9249 + (0.9723) = 10.8972 |

Dari hasil pengolahan dan perancangan menggunakan jaringan syaraf tiruan dengan metode backpropagation, maka dapat dilihat hubungan bahwa Jaringan Syaraf Tiruan bisa digunakan untuk memprediksi pergerakan *BI Rate* berdasarkan dari Kurs Dollar, Jumlah Uang Beredar, Inflansi, dan Ihsg sebagai input dan BI rate hari besok sebagai output nya.

*X = y \* (xmax-xmin) + xmin*

= 0.1773\* (13.615- 3.02) + 3.02 = 4.89

Pengolahan data secara manual ini selanjutnya akan diolah dengan menggunakan MATLAB untuk beberapa pola arsitektur jaringan. Hasil yang diperoleh dari proses pengolahan data menggunakan MATLAB selnjutnya akan dibandingkan dengan hasil penolahan data secara manual untuk melohat perbedaan hasilnya. Dalam menggunakan metode backpropagation, pola arsitektur yang dipakai sangat mempengaruhi dalam proses penentuan hasil. Setiap hasil yang diperoleh oleh suatu pola arsitektur memungkinkan berbeda dengan hasil yang didapatkan dengan menggunakan pola arsitektur yang lain.

**BAB V**

**IMPLEMENTASI DAN PENGUJIAN**

1. **Implementasi**

Implementasi adalah sebuah tindakan yang dilakukan untuk mengetahui bagaimana jika aplikasi yang telah dibangun dapat diimplementasikan ke dalam sebuah sistem, apakah aplikasi ini mampu memberikan manfaat yang baik bagi usernya. Implementasi juga dilakukan untuk mengetahui batasan sistem yang diperlukan dalam menjalankan aplikasi ini.

* + 1. **Sepesifikasi Sistem**

Dalam membangun aplikasi *Jaringan Syaraf Tiruan* ini dibutuhkan beberapa spesifikasi perangkat keras dan perangkat lunak untuk menunjang pembangunan sistem yang sesuai dengan kebutuhan.

* + - 1. **Perangkat Keras (*Hardware*)**

Perangkat keras yang digunakan dalam pembangunan aplikasi *Jaringan Syaraf Tiruan* ini adalah sebagai berikut :

1. Laptop Hp Pavilion G4.
2. *Processor* Intel(R) Core(TM) i5
3. Memory RAM 4 GB
4. *Harddisk 500 GB*
5. System Type 64-bit Operating System
   * + 1. **Perangkat Lunak (*Software*)**

Selain perangkat keras, perangkat lunak sangat diperlukan dalam perancangan sistem ini. Agar dapat berjalan dengan baik maka penulis memakai perangkat lunak yang benar–benar mampu atau diperlukan secara menyeluruh dalam perancangan sistem ini. Perangkat lunak yang digunakan pada saat implementasi aplikasi jaringan saraf tiruan ini adalah :

1. Sistem operasi Windows 7.
2. Notepad ++.
3. Web Server, seperti Mowes Portable II dan XAMPP.
4. Text editor, seperti, Photoshop, notepad, dll .
5. Web Browser, Seperti Google Crome dan Mozila Firefox.
   * 1. **Instalasi Software Pendukung**

Dalam membangun aplikasi *Jaringan Syaraf Tiruan* ini diperlukannya Software pendukung seperti *Mowes* *Portebel* sebagai media pendukung menjalankan hasil Php.

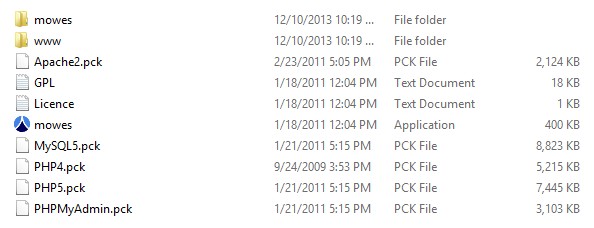
* + - 1. **Instalasi Mowes Portable II**

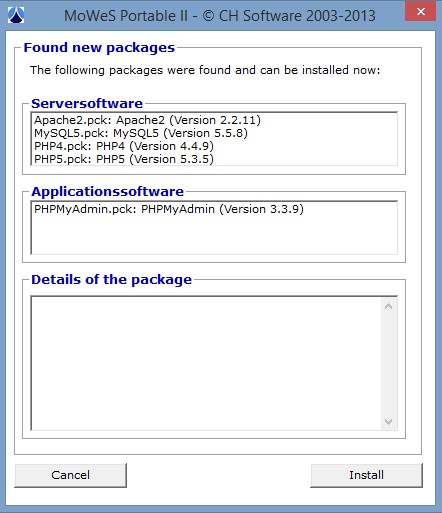
*Mowes Portable* merupakan sebuah paket untuk menjalankan *PHP*, apache dan *MySQL*. Dengan menggunakan mowes portable kita tidak perlu repot menginstall ketiga software tersebut secara terpisah.

Berikut cara menginstal *Mowes Portabel* :

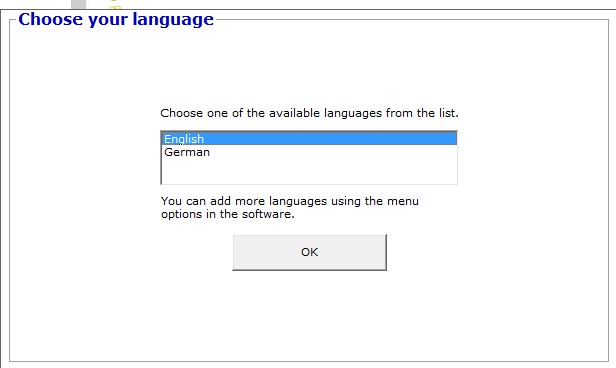
1. File mowes sebelum di instal.

Klik mowes.exe untuk memulai proses penginstalan, seperti gambar   
dibawah.

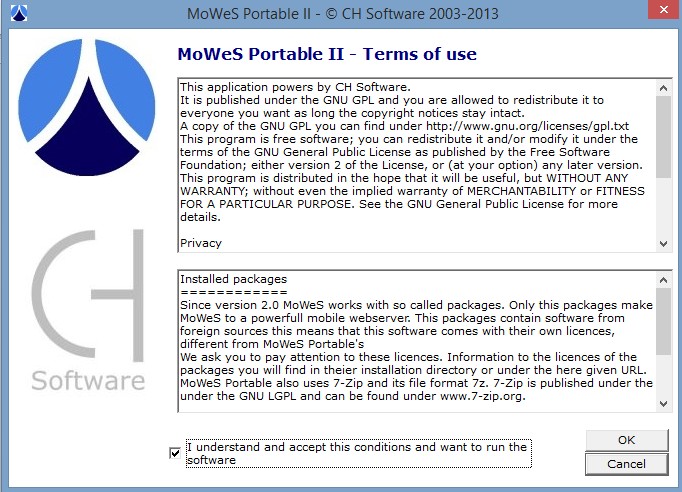




**Gambar 5.1 Tampilan Master Mowes**

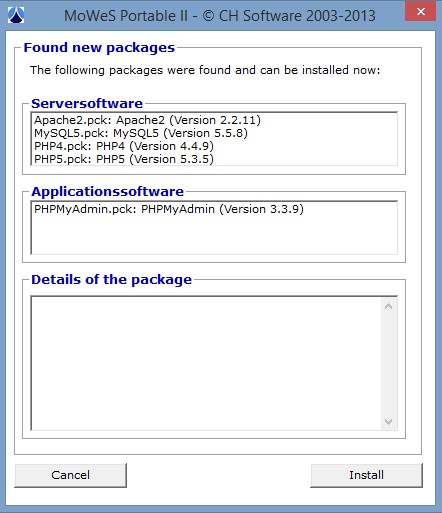
1. Sehingga akan tampil gambar.

**Gambar 5.2 Pemilihan Bahasa Penginstalan**

1. Pilih bahasa yang ingin di gunakan dalam penginstalan dan klik OK, maka akan tampil gambar dibawah.

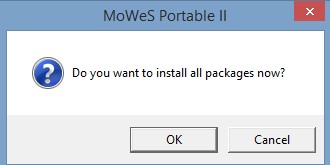
**Gambar 5.3 Halaman Persetujuan**

1. *checklist* “*I understan and accept this conditions and want to run the softwere*” kemudian klik OK, maka akan tampil gambar.

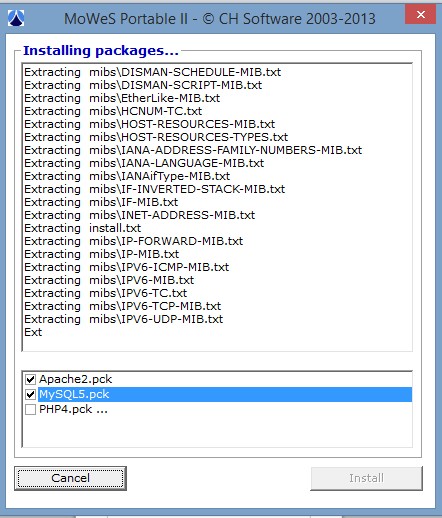


**Gambar 5.4 Halaman *List* *Packed* *Mowes***

1. Klik “instal” dan akan muncul tampilan seperti gambar.



**Gambar 5.5 Persetujuan *List* *Packed* *Mowes***

1. Klik OK dan akan muncul tampilan seperti gambar.

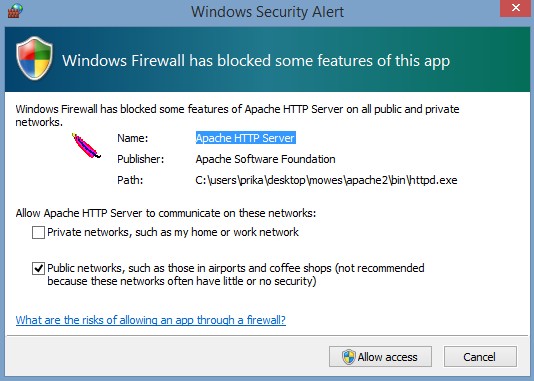
**Gambar 5.6 Proses *Instal* *Mowes***

1. Tunggu proses instal, jika muncul gambar dibawah klik “*Alow* *access*”.



**Gambar 5.7 *Windows* *security* *alert* *MySQL***

1. Setelah proses diatas, akan muncul gambar seperti di bawah dan klik “*Allow* *access*”.

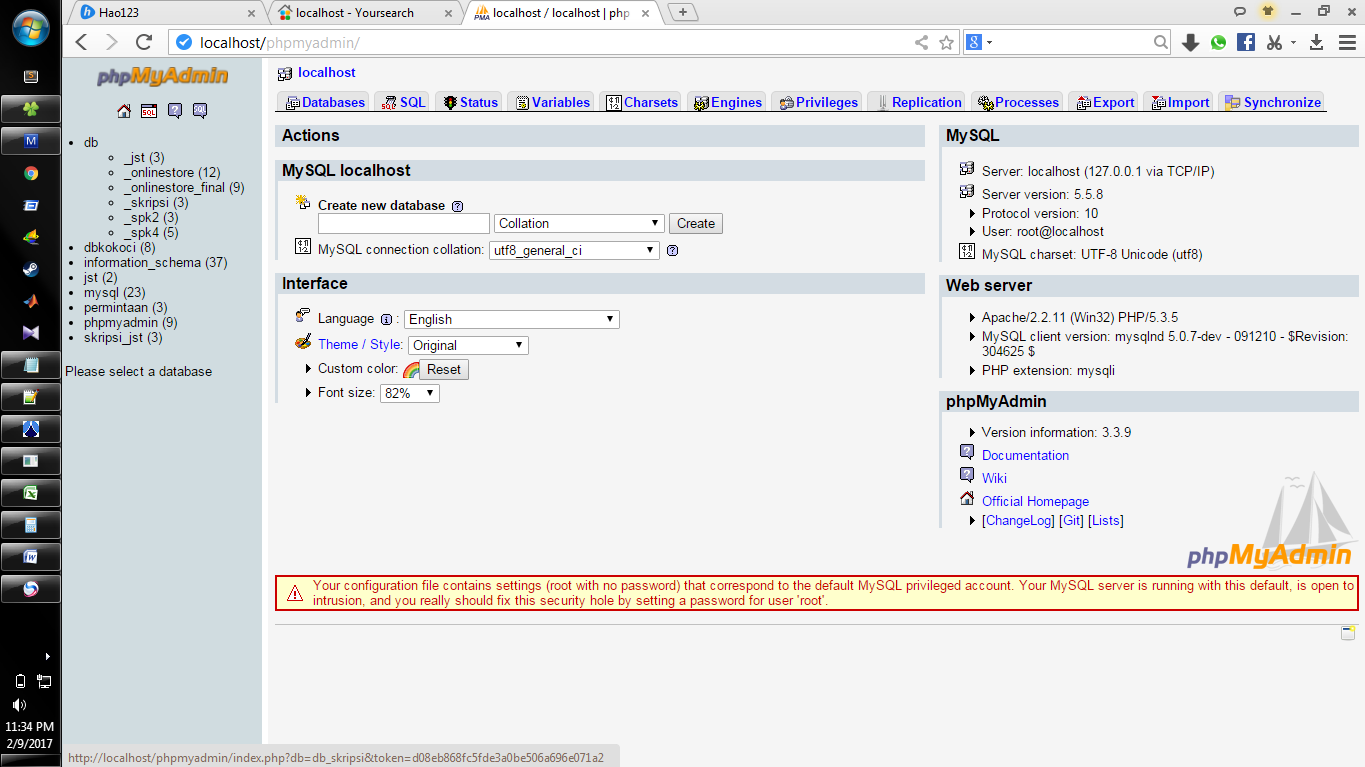


**Gambar 5.8 *Windows* *security* *apache* *HTTP* *Server***

1. Proses penginstalan selesai sehingga tampil gambar.

**Gambar 5.9 Mowes portable II sedang *running***

Setelah selesai akan muncul seperti pada gambar diatas maka bukalah mozillanya dan ketikan “*localhost*/*phpmyadmin*” dan akan tampil *Database phpMyAdmin* seperti gambar berikut:



**Gambar 5.10 Halaman Database PhpMyAdmin**

* 1. **Pengujian**

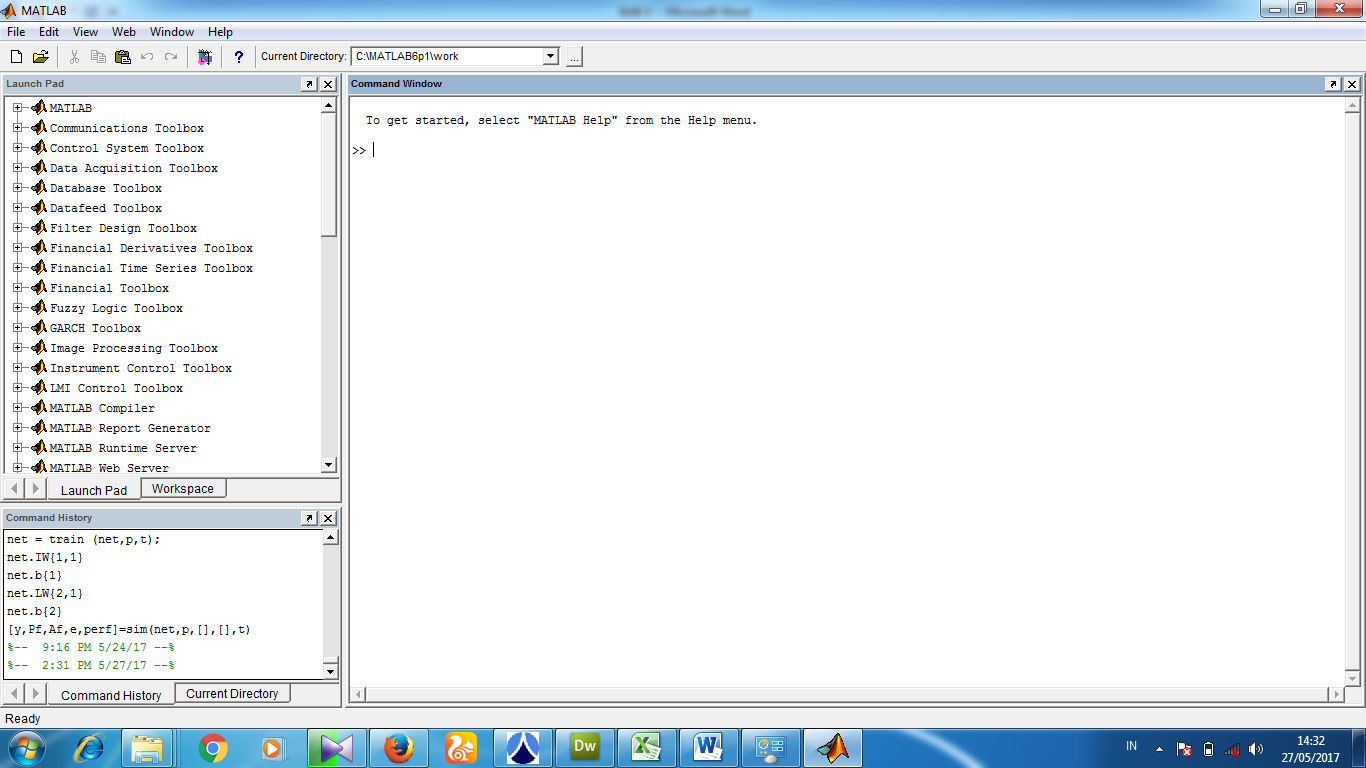
Pengujian dan implementasi sistem bertujuan untuk melihat apakah sistem yang dirancang sudah sesuai dengan apa yang diinginkan atau belum, setelah dilakukannya pengujian dan implementasi, kualitas sebuah *system* akan terlihat. Tampilan program yaitu merupakan sub bab yang menjelaskan tentang proses dimulainya sampai program ini selesai dieksekusi, *point-point* pada sub bab ini akan menjelaskan tentang bagaimana sebuah *form* dijalankan dan apa saja fungsi yang terdapat pada *form* tersebut.

* + 1. **Proses Pembuktian dengan Matlab**

Matlab merupakan aplikasi pendukung untuk membantu mendapatkan nilai *Hidden* dan nilai *Bias* dari data dengan metode *backpropagation.* Berikut merupakan langkah-langkah pengujian dengan matlab :

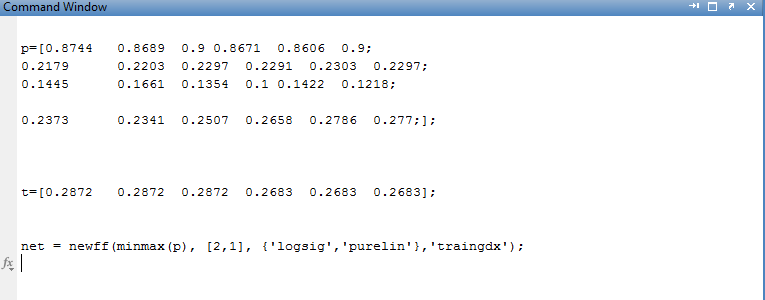
1. Tampilan Awal Matlab.

Tamplan ini merupakan tampilan awal dari aplikasi matlab, dimana nantinya akan memulai proses pengujian data.



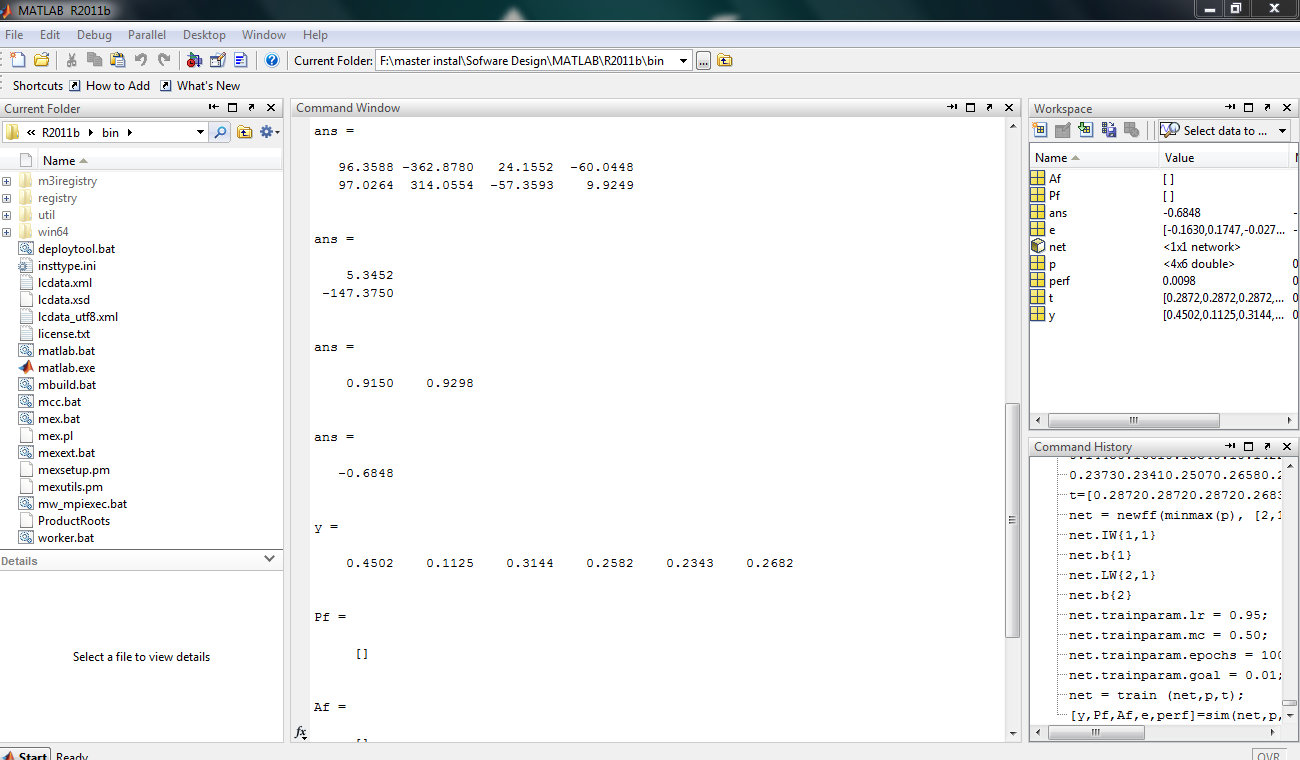
**Gambar 5.11 Tampilan Awal Matlab**

1. Selanjutnya masukan nilai Input dan Target pada halaman *Command* *Window* yang ada di Matlab dengan rumus untuk *input* >> *p=[ ;];* dan *Target >> t=[ ;];*. Seperti gambar dibawah ini :

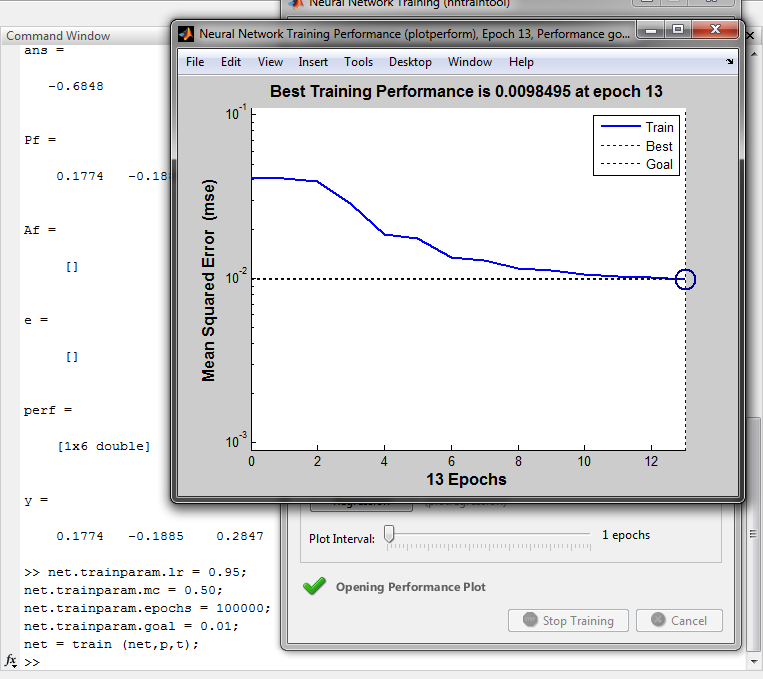


**Gambar 5.12 Tampilan Masukan Input Dan Target**

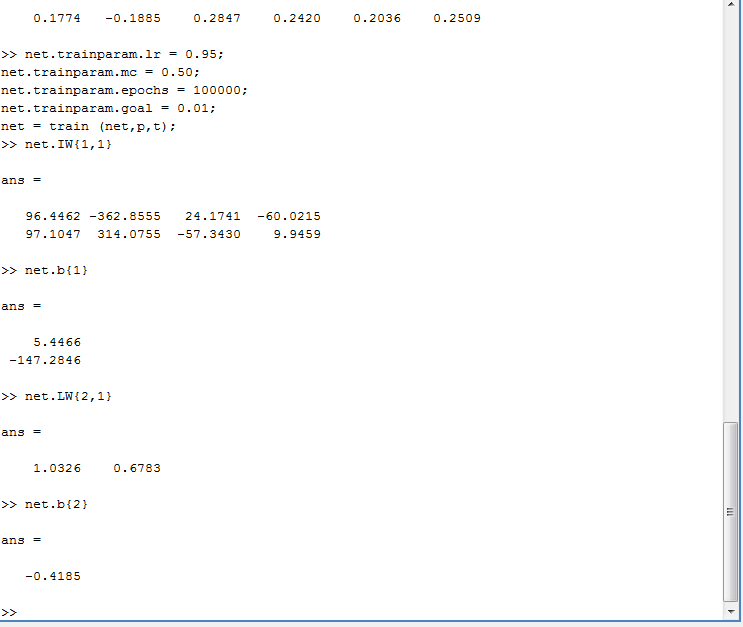
1. Setelah masukan nilai input dan target, selanjutnya masukan beberapa *codding* untuk mendapatkan nilai dari *bias input* ke *hidden* dan *nilai* *bias* dari *hidden* ke *output*, seperti gambar dibawah ini :



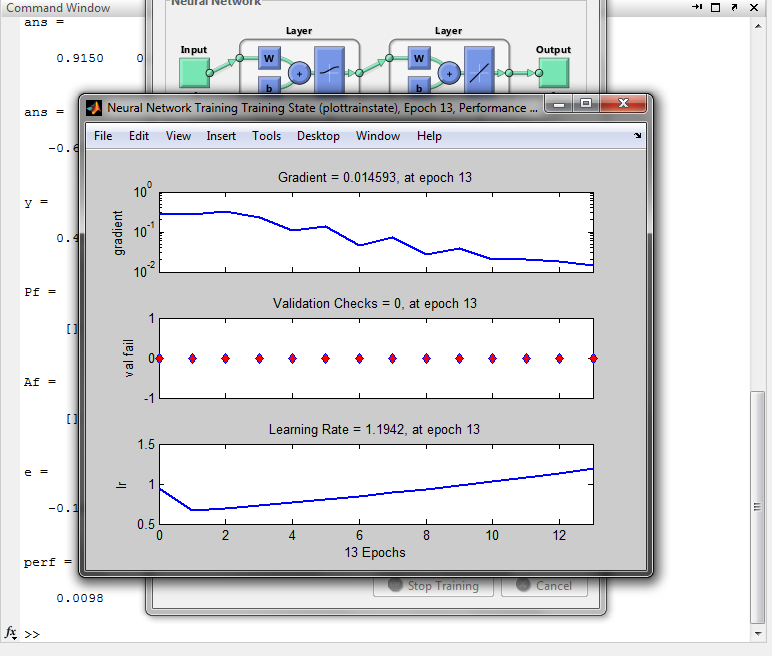
**Gambar 5.13 Tampilan Mencari Nilai Bias**

1. Setelah mendapatkan nilai bias selanjutnya lakukan pengujian dengan memasukan *codding* dengan nilai *net*.*trainparam*.*epochs* = *100*,*000;* *net.trainparam.goal = 0.01;* dan setelah dimasukan lakukan pengujian dengan masukan *codding net = train (net,p,t);*. Setelah mendapatkan nilai maksimal *Epoch* yaitu di angka *Epoch 13/100000, MSE 0.0098496 , Gradient 0.014593.* seperti gambar dibawah ini :

**Gambar 5.14 Tampilan Pengujian *Epoch***

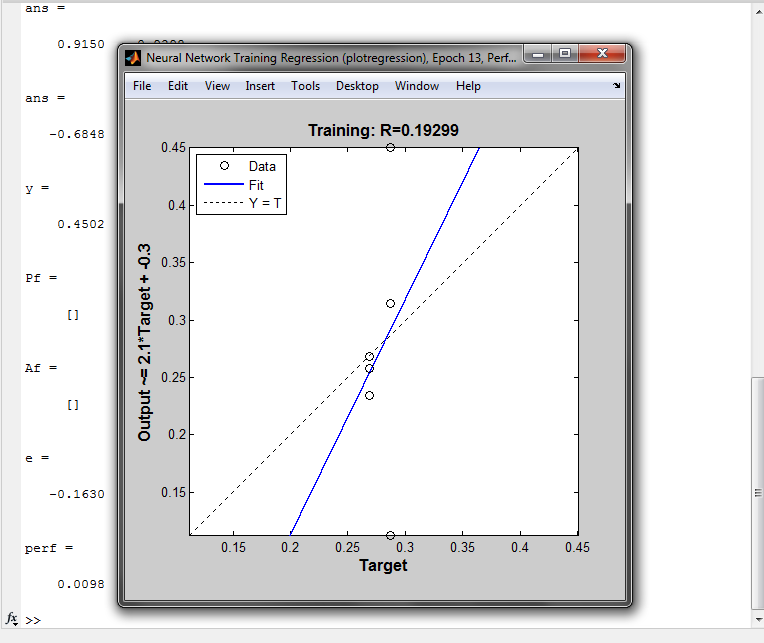
1. Setelah melakukan pengujian untuk mendaptkan *epoch*, perlu melakukan lagi pencarian nilai *bias* agar mendapatkan nilai *bias* akurat , seperti gambar dibawah ini :

**Gambar 5.15 Tampilan Pencarian nilai bias ke-dua**

1. Selanjunya lihat hasil dari *tranning state plottrainstate* seperti gambar dibawah ini :

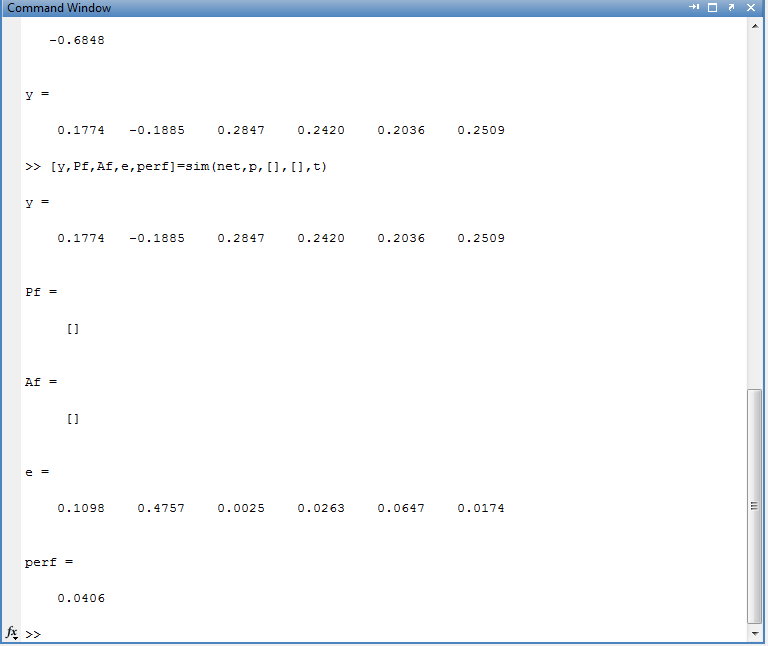
**Gambar 5.16 Tampilan *Tranning State PlottrainState***

1. Selanjutnya menampilkan tampilan dari *top performance tranning regression* sepert gambar di bawah ini :



**Gambar 5.17 Tampilan *Training Regression***

1. Selanjutnya di buktikan dengan pencarian manual dan dibandingkan dengan nilai yang di dapat di matlab nilai *Y* dan *Error-*nya dengan memakai *1 epoch*. seperti gambar dibawah ini.



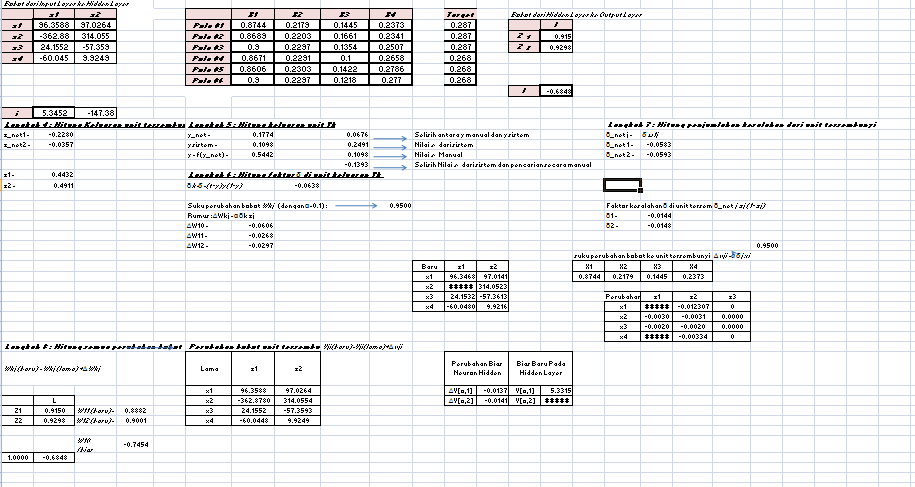
**Gambar 5.18 Tampilan Nilai Y dan E dalam Matlab**

* + 1. **Pengujian Manual**

Pada proses pengujian manual ini menggunakan *sofrware* *Microsoft Exel*-2007. Proses ini adalah proses pencarian nilai manual perancangan algoritma *Backpropagation.* Pada tahap perancangan ini hasil analisi yang telah didapatkan akan diterapkan pada web yang telah dirancang. Adapun proses pencarian adalah :

1. Hitung semua keluaran di unit tersembunyi *Zj*
2. Hitung keluaran unit *Yk*
3. hitung faktor diunit keluaran *yk*
4. Hitung penjumlahan kesalahan dari unit tersembunyi (

Proses perhitungan dapat digambarkan pada gambar dibawah ini :



**Gambar 5.19 Tampilan proses pencarian manual**

* + 1. **Pengujian Aplikasi**

Pada tahap ini penulis melakukan pengujian terhadap aplikasi yang telah dirancang dan memastikan aplikasi bekerja sesuai dengan yang direncanakan dengan Metode *BlackBox*, seperti pada tabel berikut :

**Tabel 5.1 Pengujian perangkat lunak**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Fungsi** | **Cara Pengujian** | **Hasil Yang Diharapkan** | **Hasil** |
| 1 | Login | Masuk dengan username dan password terlebih dahulu (admin) | Data diverifikasi dengan data yang ada pada database dan akan menampilkan pesan sukses. | OK |
| 2 | Halaman  Home | Membuka halaman  utama setelah berhasil login ke dalam website | Menampilkan halaman  utama dan menyediakan fasilitas yang ada pada website | OK |
| 3 | Olah Data | Membuka halaman data dengan mengklik menu olah data . (admin) | Menampilkan data yang telah tersimpan pada database | OK |
| 4 | Tambah Data | Melakukan penambahan dengan klik form new data barang (admin) | Data yang di inputkan tersimpan pada database | OK |
| 5 | Edit Data | Melakukan pengubahan Data. (admin) | Data berubah sesuai dengan yang diinputkan. | OK |
| 6 | Hapus data | Melakukan penghapusan data barang (admin) | Data barang yang dipilih terhapus pada database | OK |
| 7 | Kelola Data | Melakukan pengelolaan data yang sudah di inputkan (admin) | data siap di kelola dan menampilkan form kelola | OK |
| 8 | Halaman prediksi | Melihat hasil dari kelola data | Menampilkan hasil dari kelola data dengan menampilkan data yang sudah di kelola | OK |
| 9 | Halaman Proses Prediksi | Melihat proses prediksi data | Menampilkan hasil prediksi BI Rate | OK |
| 10 | Halaman Cetak Hasil Prediksi | Melihat Form Untuk Cetak Prediksi | Menampilkan Form Untuk Cetak Prediksi | OK |
| 11 | Logout | Destroyed session Login | Keluar dari sistem. | OK |

* + 1. **Pengujian Interface**

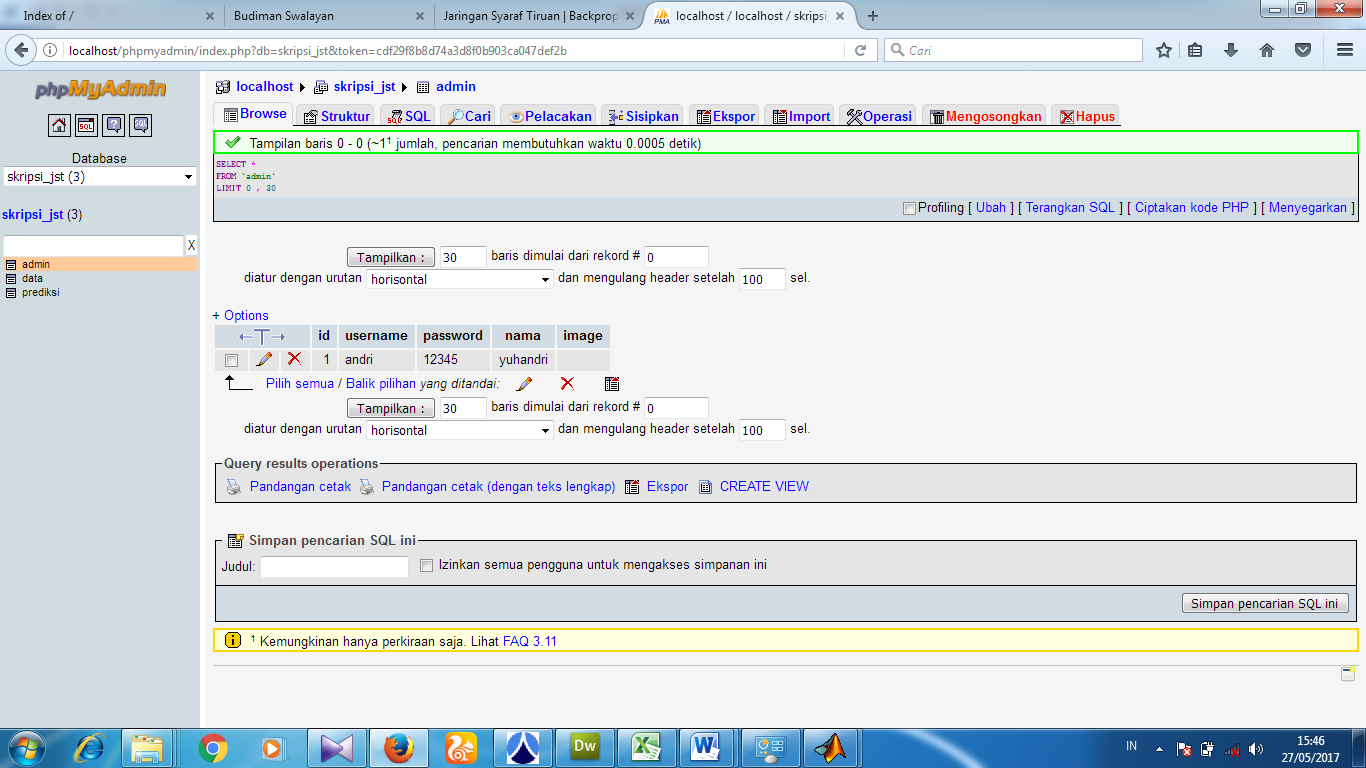
Pada pengujian interface ini, dilihat apakah interface sistem sudah memenuhi syarat sebagai interface yang *userfriendly* bagi pengguna nantinya. Pengujian interface dilakukan agar sistem berbasis website mudah digunakan oleh pengguna nantinya.

* + - 1. **Tampilan Perancangan Database**

Tampilan perancangan database disini sebagai media penyimpanan untuk data yang akan kita proses di aplikasi yang akan dijalankan.

1. *Tampilan Tabel Admin*

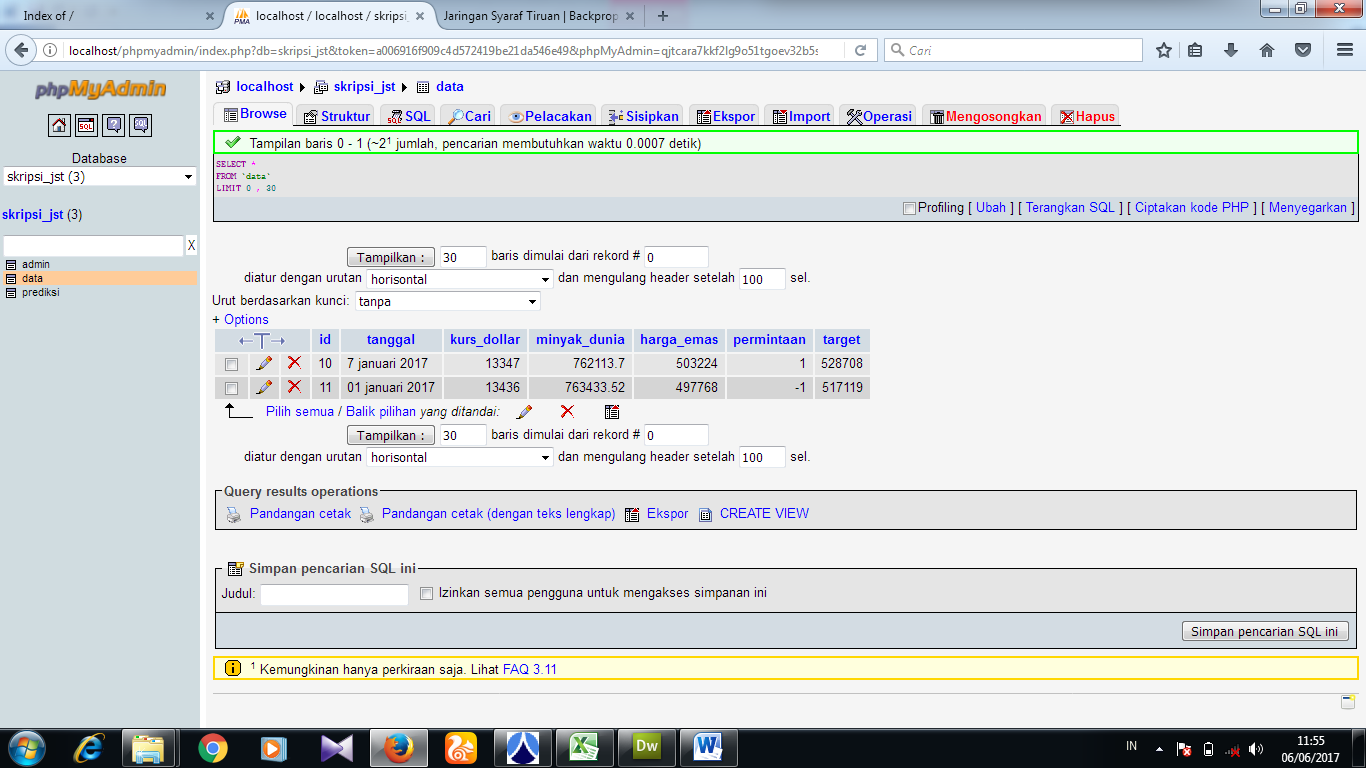
Tampilan *table manager* berfungsi sebagai table yang menyimpan secara keseluruhan biodata yang sudah di inputkan oleh *admin*.



**Gambar 5.20 Halaman Tabel *Manager Database***

1. *Tampilan Tabel Data*

Tabel barang berfungsi sebagaimediapenyimpanan semua data yang sudah di inputkan oleh *admin*.



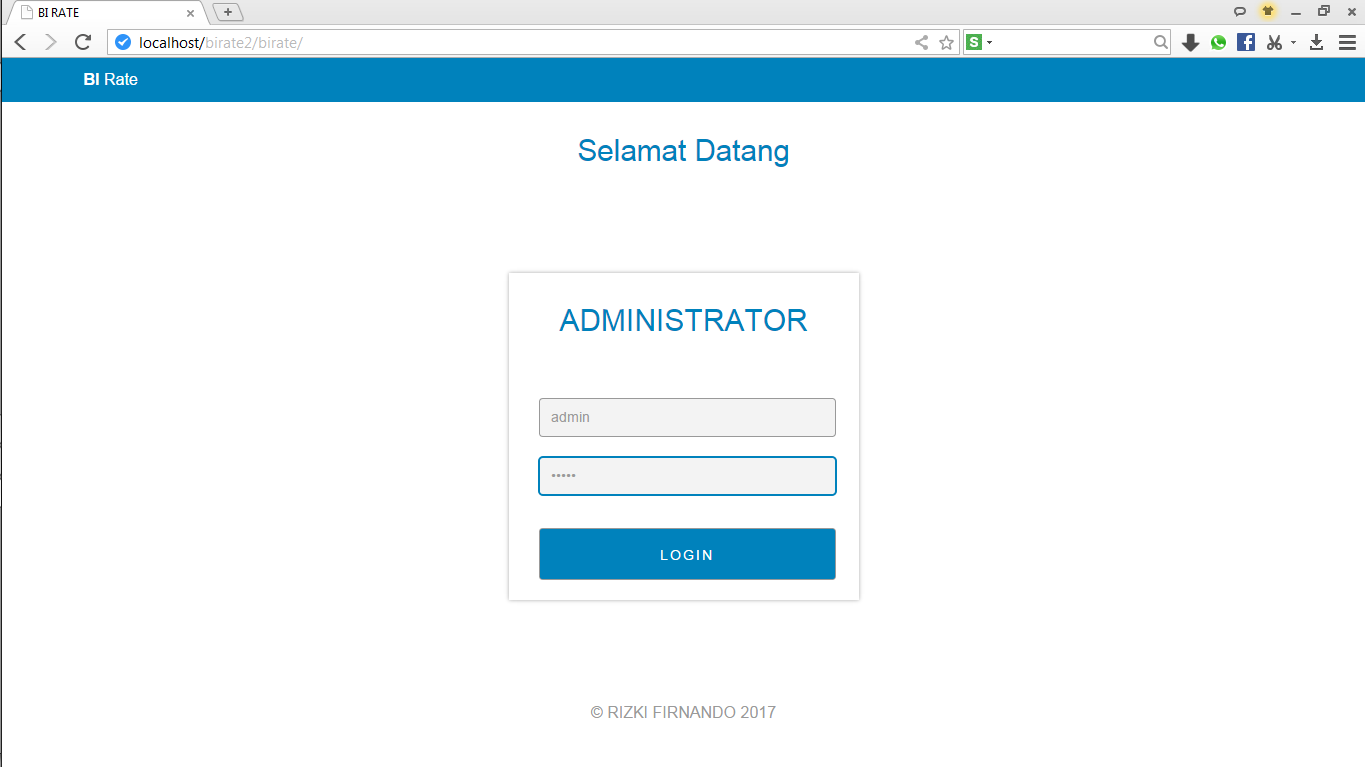
**Gambar 5.21 Halaman Tabel *Data Database***

* + - 1. **Tampilan Berbasis Website**

Tampilan berbasis *Website* ini merupakan aplikasi yang offline dimana nantinya membukanya dengan menjalankan *Mowes Potrteble* lalu akan tampil secara otomatis *Localhost* pada *Mozilla Firefox*. Dengan begitu admin dapat mengakses sistem ini. Aplikasi berbasis website dapat digunakan oleh admin, dimana admin dapat menjalankan semua fungsi masing-masing seperti mengakses menu-menu tertentu sesuai dengan ketentuan.

* + - * 1. *Tampilan Form Login*

*Form login* berfungsi untuk administrator masuk ke dalam sistem, untuk dapat login maka admin harus memasukkan *username* dan *password* yang benar pada *form login* seperti pada gambar berikut :

****

**Gambar 5.22 Tampilan *Form Login***

* + - * 1. *Tampilan Home*

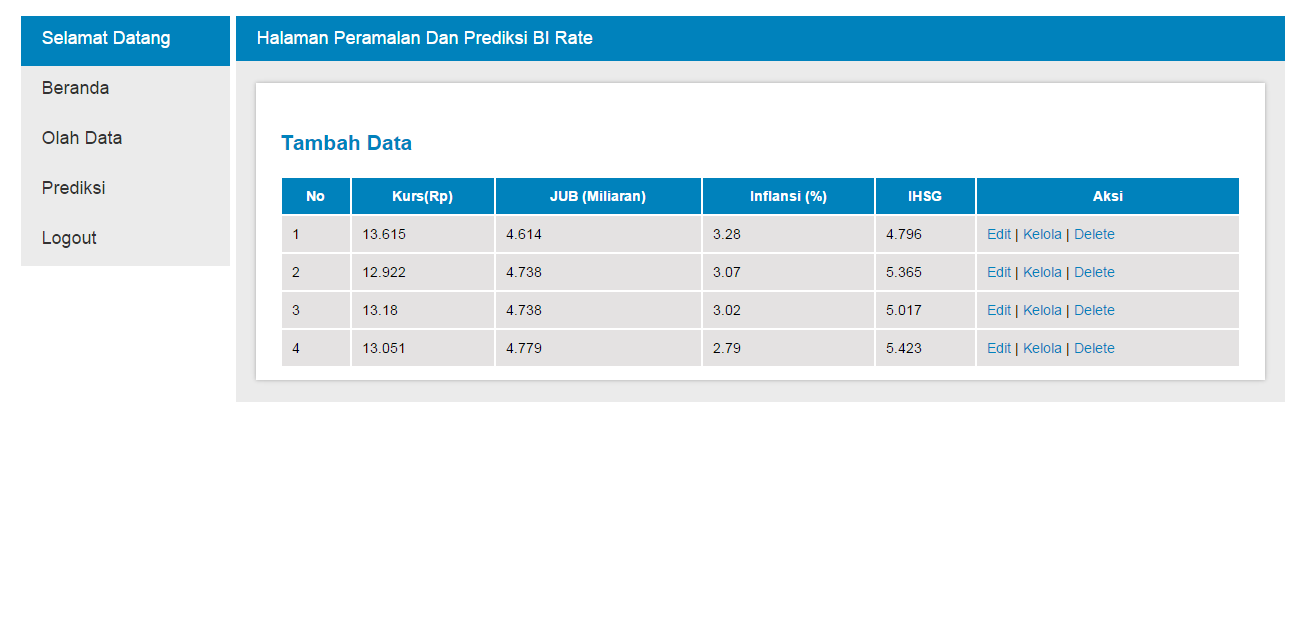
Tampilan *Home* merupakan tampilan awal pada *form menu* utama. Pada tampilan menu utama atau *home* ini terdapat beberapa menu yang dapat diakses oleh Admin Tampilan *form menu* utama dapat dilihat pada Gambar dibawah ini :



**Gambar 5.23 Halaman *Home Website***

* + - * 1. *Tampilan Olah Data*

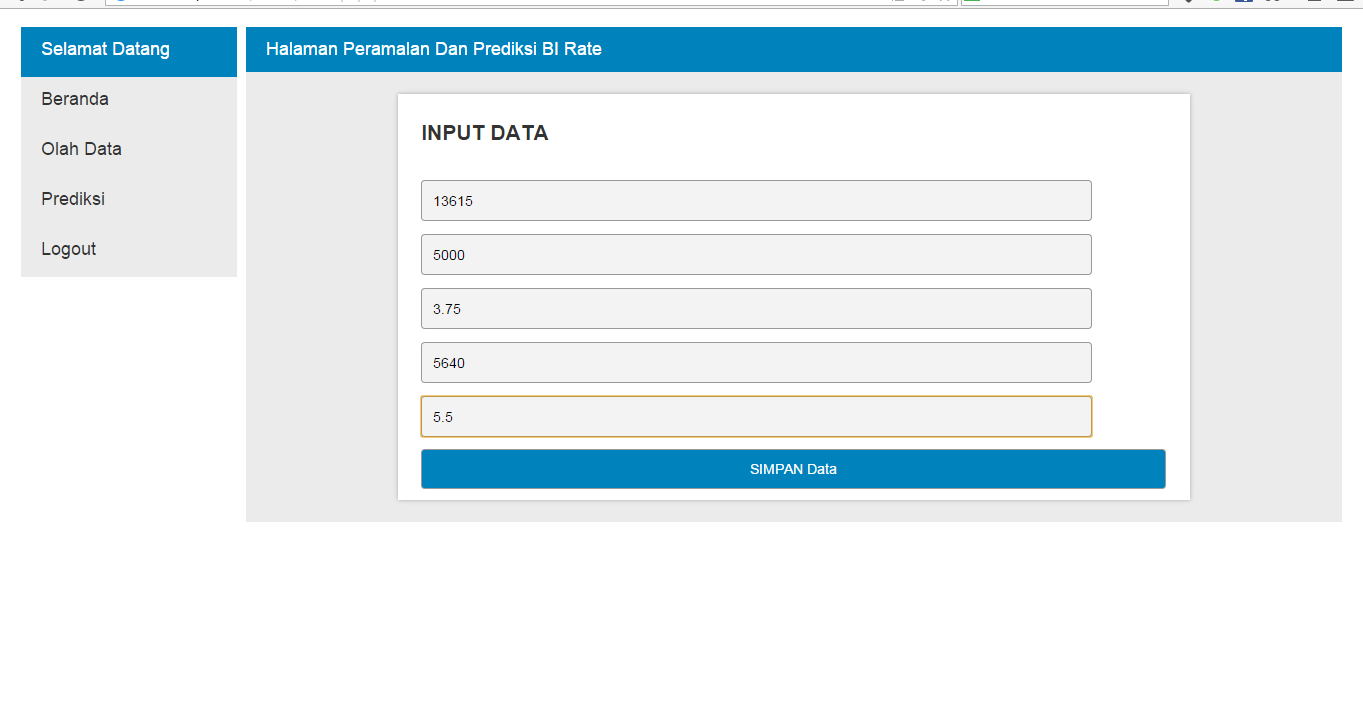
Pada tampilan ini adalah semua data yang ada di system dan admin dapat menambah data pada sistem. Seperti gambar berikut :



**Gambar 5.24 Tampilan Olah Data**

* + - * 1. *Tampilan Tambah Data*

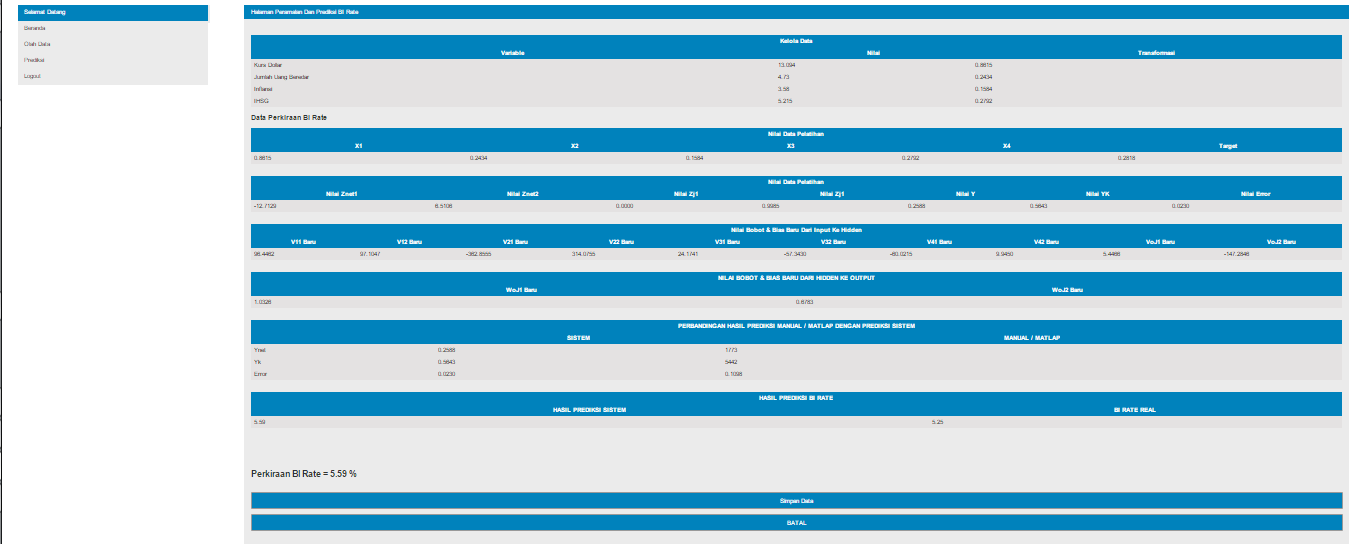
Tampilan input new data merupakan halaman untuk admin mnambahkan Data untuk kebutuhan memprediksi.



**Gambar 5.25 Tampilan Form Input New Data**

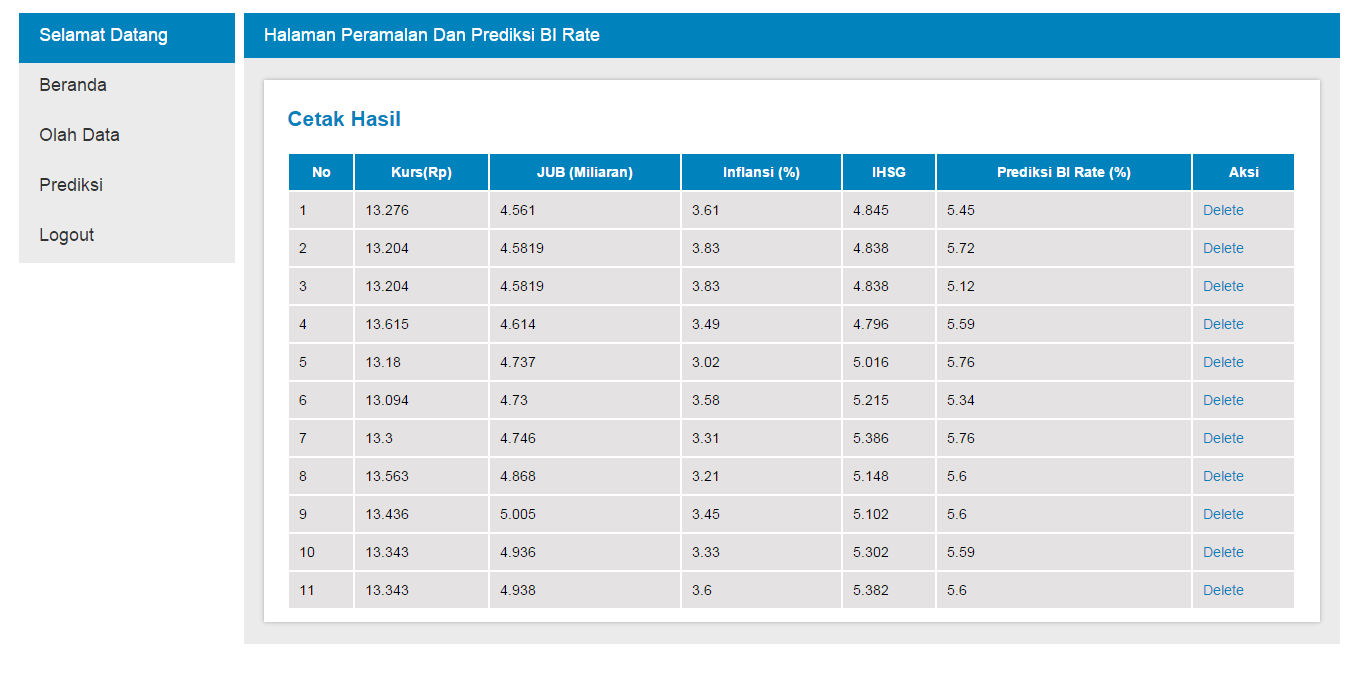
* + - * 1. *Tampilan Kelola Data dan Proses Prediksi*

Tampilan ini berisi tentang halaman dimana admin bisa melihat data yang sudah ada untuk di prediksi.

**  
Gambar 5.26 Tampilan Form Kelola Data**

* + - * 1. *Tampilan Data Hasil Prediksi*

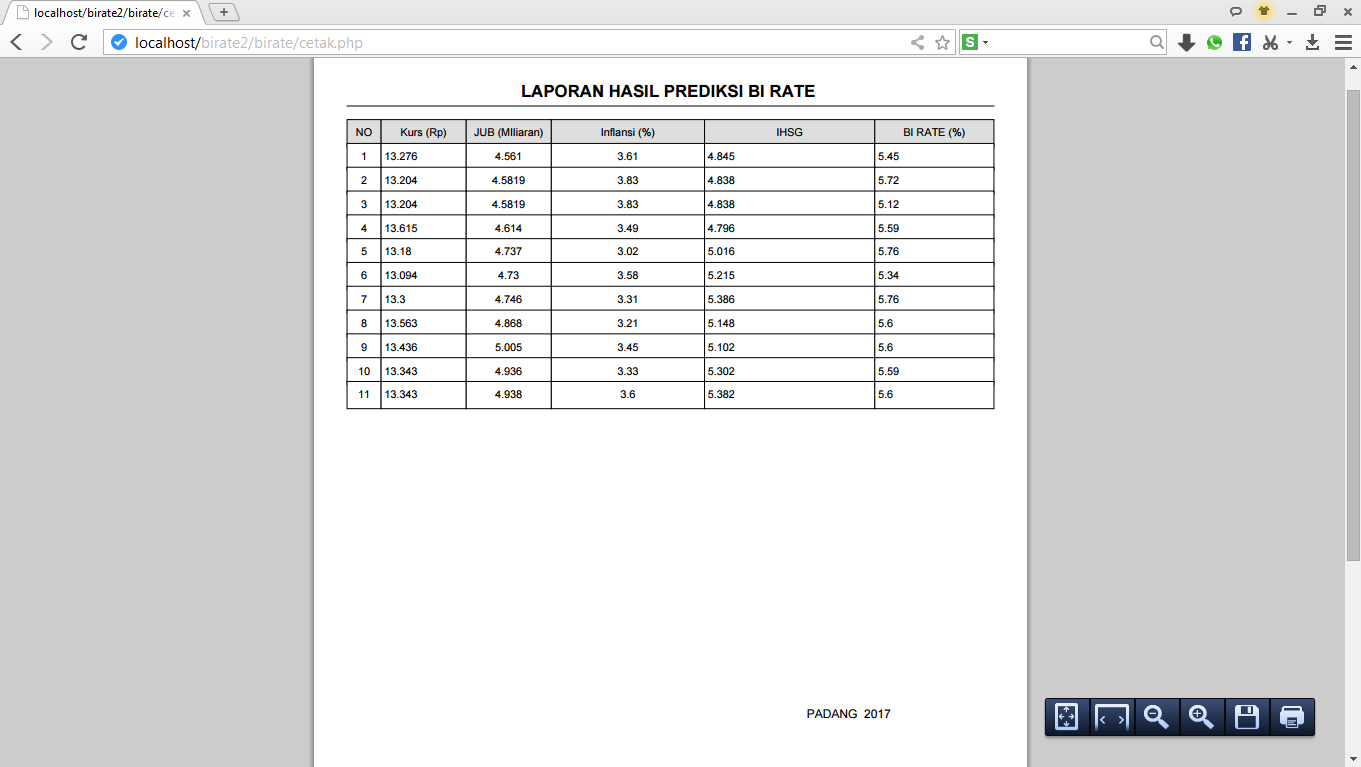
Tampilan data prediksi merupakan halaman data yang sudah ada hasil prediksi *BI Rate*.



**Gambar 5.27 Tampilan Form Data Hasil Prediksi**

* + - * 1. *Tampilan Cetak Prediksi*

Tampilan proses cetak data prediksi merupakan halaman yang menampilkan Hasil prediksi harga emas secara keseluruhan.



**Gambar 5.26 Tampilan Form Cetak Prediksi**

**BAB VI**

**PENUTUP**

Berdasarkan hasil pembangunan aplikasi yang telah penulis lakukan, penulis membuat suatu kesimpulan dan mengajukan beberapa saran yang berhungungan dengan pembahasan yang telah dikemukakan di bab-bab sebelumnya.

**6.1 Kesimpulan**

Penelitian ini melakukan pengujian dengan mengimplementasikan software Matlab R2011B terhadap data data yang digunakan untuk memprediksi *BI Rate* , maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Dengan merancang dan membangun aplikasi Jaringan Syaraf Tiruan mampu melakukan prediksi dan peramalan BI Rate.
2. Dengan menerapkan algoritma *Backpropagation* mampu melakukan prediksi dan peramalan BI Rate dengan lebih cepat, akurat dan meminimalisir kesalahan dalam memprediksi BI Rate dikarenakan prediksi BI Rate dilakukan menggunakan sistem yang didasarkan pada beberapa indikator yaitu Kurs Dollar, Jumlah Uang Beredar, Inflansi , dan IHSG. .
3. Dengan merancang Aplikasi Jaringan Syaraf Tiruan dapat membantu instansi pemerintah, swasta maupun individu tertentu dalam membuat kebijakan yang berhubungan dengan BI Rate.

**6.2 Keterbatasan Sistem**

Sistem Aplikasi yang dirancang untuk memprediksi BI Rate , dalam perancangan dan pembangunan aplikasi ini, masih terdapat beberapa keterbatasan, adapun keterbatasannya adalah sebagai berikut :

1. Aplikasi Jaringan Syaraf Tiruan dengan metode *Backpropagation* ini hanya di implementasikan untuk memprediksi BI Rate dengan menggunakan data yang sudah ada dalam sistem.
2. Aplikasi Jaringan Syaraf Tiruan dengan metode *Backpropagation* ini hanya menggunakan 4-2-1 dimana 4 adalah inputnya yang terdiri Kurs Dollar, Jumlah Uang Beredar, Inflansi , IHSG, dan Permintaan, 2 merupakan hidden layernya dan 1 adalah hasil outputnya.
3. Aplikasi Jaringan Syaraf Tiruan dengan metode *Backpropagation* ini dalam proses prediksi BI Rate hanya dapat memproses per item saja. Dan sistem akan menampilkan seluruh hasil prediksi BI Rate dari data input yg telah di olah.

**6.3 Saran**

Dari penelitian yang telah dilakukan dan sebagai akhir dari penulisan ini, adapun kekurangan dari sistem ini yaitu masih terdapatnya beberapa fitur yang masih sederhana dan juga keterbatasan pengetahuan dari penulis terhadap pembangunan sistem. Maka dari itu terdapat beberapa saran yaitu :

1. Dari kajian diatas penulis mengharapkan adanya penelitian lebih lanjut, karena adanya beberapa fitur yang masih sederhana dan belum sempurna yang belum dibahas karena keterbatasan penulis dalam melakukan penelitian ini
2. Pembuatan aplikasi Jaringan Syaraf Tiruan untuk memprediksi dan meramalakan BI Rate ini masih sangat sederhana dan jauh dari kesempurnaan sehingga perlu adanya pengembangan sistem dan tampilan yang lebih menarik.
3. Dalam pengembangan system aplikasi Jaringan Syaraf Tiruan ini perlu adanya penambahan beberapa fitur yang bisa membuat system ini lebih baik dari sebelumnya.

DAFTAR PUSTAKA

Alexander F. K. Sibero. 2013.”Web programming Power Pack”.Yogyakarta : Diterbitkan MediaKom.

Amalia Ayu Azami ,Agus Maman Abadi (2012), “Prediksi Suku Bunga Bank Indonesia (BI Rate) Menggunakan Metode *Neuro Fuzzy*”.

Asrul Huda (2014). “Penerapan Jaringan Syaraf Tiruan Dalam Memprediksi Harga   
 Saham Pada Pasar Modal Indonesia”. “Jurnal Teknologi Informasi &  
 Pendidikan”Vol. 7 No. 1. 2086 – 4981.

A.S, Rosa dan M. Shalahuddin .2013.”Rekayasa PerangkatLunak, Terstruktur dan Berorientasi Objek”.Bandung :Informatika Bandung.

Bank Indonesia. (2016) Defenisi BI *Rate.* Diakses dari <http://www.bi.go.id>

Bank Indonesia. (2016) Data Kurs Rupiah ,Inflansi , Jumlah Uang Beredar , dan Suku Bungan BI Diakses dari <http://www.bi.go.id>

Bank Indonesia. (2016) Data IHSG.Diakses dari <http://www.fbs.id>

Tanjung Dahriani Hakim, (2015) “Jaringan Saraf Tiruan dengan *Backpropagation* untuk Memprediksi Penyakit Asma”.

Febrina Mira, Faula Arina, Ratna Ekawati (2013) “Peramalan Jumlah Permintaan Produksi Menggunakan Metode Jaringan Syaraf Tiruan (JST) *BackPropagation*”.

Kiki dan Kusumadewi, (2004) “Jaringan Syaraf Tiruan Dengan Metode *Backpropagation* Untuk Mendeteksi Gangguan Psikolgi”.

Mandala, Eka Praja Wiyata .2015.“Web Programing, Project 1 e.p.w.m. forum”.Yogyakarta :Penerbit CV. Andi Offset.

Nurmila Nazla, Aris Sugiharto, dan, Eko Adi Sarwoko (2007) “Algoritma *Backpropagation Neural Network* Untuk Pengenalan Pola Karakter Huruf Jawa”,

Puspitaningrum, Diyah. 2006. “Pengantar Jaringan Syaraf Tiruan”. Yogyakarta : Penerbit CV. Andi Offset.

Rizky Soetam (2011). “Konsep Dasar Rekayasa Perangkat Lunak” , Jakarta : Penerbit Prestasi Pustaka Publisher.

Simarmata, Janner. 2010.”Rekayasa Perangkat Lunak”.Yogyakarta : Penerbit CV. Andi Offset.

Sovia, Rini dan Jimmy Febio (2011). “Membangun Aplikasi E-Library menggunakan HTML, PHP SCRIPT dan MYSQL Database”. ”Jurnal Teknologi Informasi & Pendidikan”. Volume 3 No.1, 2086-4981.

Siang, Jong Jek. 2009,“ *Jaringan Saraf Tiruan & Pemrogramannya menggunakan MATLAB*”. Penerbit :Andi Publishing.

Sutojo,T. S.Si.,M.Kom., Edy Mulyanto, S.Si.,M.Kom., Dr. Vincent Suhartono (UDINUS),2011,“*KecerdasanBuatan*”. Penerbit :Andi Publishing