**LAPORAN PENELITIAN**

**JARINGAN SYARAF TIRUAN UNTUK MEMPREDIKSI HARGA MATA UANG CRYPTOCURRENCY BITCOIN**



**DIAJUKAN OLEH :**

**RINI SOVIA 1005047601 (Ketua)**

**Musli Yanto 1007078901 (Anggota)**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**UNIVERSITAS PUTRA INDONESIA YPTK**

**PADANG**

**2018**

**Ringkasan**

The value of bitcoin currency is very volatile, hard to guess for every hour, so many of the bitcoin traders suffer losses because they are wrong in managing their bitcoin assets. Changes in the price of bitcoin itself are influenced by many things such as the closing of the bitcoin market in a country, the occurrence of hacker attacks on the bitcoin blockchain and the emergence of new coins that use technology similar to bitcoin. But when a stable market situation changes the price of bitcoin is purely influenced by market forces. By implementing an artificial neural network using backpropagation method, it will be able to predict the price of bitcoin by giving a form of predictive results that are strengthened with a fairly good value of accuracy. This research begins by determining prediction variables with target values that can be determined based on previous bitcoin prices. This artificial neural network process is able to conduct training and testing of data based on network patterns that have been formed, then the results of training and testing of the network will be analysed again, so that at the last stage the best network patterns will be used in the prediction process.

**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 2019

**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

**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

**DAFTAR PUSTAKA** xxii

**LAMPIRAN** xxiii

**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Perkembangan teknologi dari tahun ke tahun sangat pesat perkembangannya, tidak dipungkiri teknologi pada saat ini sudah merambah diberbagai aspek kehidupan. Teknologi sangat membantu pekerjaan manusia di berbagai bidang namun juga harus didorong dengan SDM yang mumpuni. Teknologi informasi adalah suatu teknologi yang digunakan untuk mengolah data meliputi ; memproses, mendapatkan, menyusun, menyimpan, memanipulasi data dengan berbagai cara untuk menghasilkan informasi yang berkualitas.

Jarigan Syaraf Tiruan adalah sistem komputasi dengan arsitektur dan operasinya berasal dari pengetahuan tentang sel saraf biologi di dalam otak. Jaringan Syaraf Tiruan dapat digambarkan sebagai model matematis dan komputasi untuk fungsi *aproksimasi nonlinier,* klasifikasi data, *cluster* dan *regresi non parametic* atau sebagai sebuah simulasi dari koleksi model saraf biologi (Citec Journal), Volume 2 Nomor 10, 1 November 2015, ISSN 2354-5771.

*Backpropagation* merupakan sebuah metode sistematik untuk pelatihan multilayer Jaringan Syaraf Tiruan. Metode ini memiliki dasar matematis yang kuat, obyektif, selain itu algoritma ini memiliki bentuk persamaan dan nilai koefisien dengan cara meminimalisasi jumlah kuadrat galat *error* melalui *training set*. Model jaringan ini banyak digunakan untuk diaplikasikan pada penyelesaian suatu masalah berkaitan dengan *identifikasi*, prediksi, pengenalan pola dan sebagainya. Pada latihan yang berulang–ulang, algoritma ini akan menghasilkan unjuk kerja yang lebih baik. (Journal Teknologi Informasi), Volume 10 Nomor 2, Oktober 2014, ISSN 1414-9999.

*Cryptocurrency* yaitu mata uang *virtual* yang digunakan sebagai mata uang alternatif yang bersifat *desentralisasi*. Artinya tidak ada satupun pihak yang mengatur baik dari pemerintah atau si pencetus *Cryptocurrency* itu sendiri. Oleh karena itu nilai *Cryptocurrency* tidak dapat dipengaruhi jika terjadi *inflasi* yang di sebabkan oleh bank *central*. Nilai *Cryptocurrency* di tentukan oleh peminat, penawaran dan permintaan di pasar, seperti halnya dengan emas ataupun logam mulia lainnya. Konsep dasar *bitcoin* yaitu membuat sistem *decentralized authority transaction* tanpa adanya pihak ketiga yang dapat melakukan *verifikasi* dengan menggunakan konsep *digital signature* pada setiap transaksi (Nakamoto, 2008). Koin elektronik merupakan sebuah nilai nominal yang dapat di transaksikan, dimana koin digital ini merupakan sebuah rangkaian *digital signature* yang saling terhubung (IJNS), Volume 4 No 4, 2015, ISSN 2302-5700.

Memprediksi adalah meramal secara khusus tentang apa yang akan terjadi pada observasi yang akan datang (Abruscato.1988) atau membuat perkiraan kejadian atau keadaan yang akan datang yang diharapkan akan terjadi (Carin.1992). Dengan memprediksi kita akan mendapatkan gambaran tentang sebuah hal atau suatu nilai pada masa yang akan datang. Disini penulis akan memprediksi harga dari suatu mata uang *Cryptocurrency* yaitu *bitcoin*. Penelitian prediksi harga *bitcoin* ini akan menggunakan Jaringan Syaraf Tiruan dengan metode *backpropagation*.

* 1. **Rumusan Masalah**

Berdarasarkan latar belakang diatas, maka dapat dirumuskan beberapa masalah:

1. Bagaimana merancang Jaringan Syaraf Tiruan dapat digunakan untuk melakukan proses prediksi harga *bitcoin*?
2. Bagaimana membangun Jaringan Syaraf Tiruan dengan menggunakan metode *backpropagation* untuk memprediksi harga *bitcoin*?
3. Bagaimana pengimplementasian sistem aplikasi Jaringan Syaraf Tiruan berbasis web dapat membantu para *trader bitcoin* untuk mendapatkan informasi mengenai harga *bitcoin* untuk satu jam kedepan?
   1. **Hipotesa**

Dari rumusan masalah diatas, dapat ditarik beberapa hipotesa :

1. Dengan merancang Jaringan Syaraf Tiruan diharapkan dapat melakukan proses prediksi harga *bitcoin*.
2. Dengan membangun Jaringan Syaraf Tiruan menggunakan metode *backpropagation* diharapkan dapat memprediksi harga *bitcoin*.
3. Dengan mengimplementasikan sistem aplikasi Jaringan Syaraf Tiruan berbasis web diharapkan mempermudah para *trader bitcoin* mendapatkan informasi prediksi harga *bitcoin* untuk satu jam kedepan.
   1. **Batasan Masalah**

Untuk menghindari meluasnya pokok pembahasan, maka dalam penelitian ini penulis membatasi permasalahan yang ada yaitu :

1. Data yang digunakan adalah data harga *bitcoin* pada www.bitcoin.co.id
2. Metode pengenalan pola yang digunakan adalah Jaringan Syaraf Tiruan dengan pola yang didapat dari data 12 jam kebelakang.
3. Metode yang digunakan dalam penelitian adalah metode *backpropagation.*
4. Sistem ini dibangun dengan menggunakan bahasa pemrograman PHP dan MySQL.
   1. **Tujuan Penelitian**

Dalam penelitian ini penulis memiliki beberapa tujuan yang ingin dicapai adapun tujuan tersebut:

1. Merancang sistem aplikasi Jaringan Syaraf Tiruan bertujuan untuk memprediksi harga *bitcoin* pada satu jam kedepan.
2. Memberikan refensi harga *bitcoin* untuk satu jam kedepan bagi para *trader bitcoin* melalui *website* yang akan dibuat.
3. Menganalisa tingkat akurasi prediksi harga *bitcoin* yang dihasilkan melalui perhitungan dengan Jaringan Syaraf Tiruan menggunakan metode *backpropagation.*
   1. **Manfaat Penelitian**
4. Membantu para investor supaya memiliki bahan pertimbangan dalam membeli atau menjual *bitcoin* mereka.
5. Membantu para *trader bitcoin* supaya mereka akan lebih cepat dalam menganalisa pergerakan harga *bitcoin*.

Membantu perusahaan-perusahaan yang bergerak dibidang *Cryptocurrency* sebagai sarana pertimbangan untuk pengelolaan *bitcoin* yang mereka miliki.

**BAB II**

**LANDASAN TEORI**

**2.1 Rekayasa Perangkat Lunak**

Rekayasa perangkat lunak adalah disiplin ilmu yang menangani perancangan, pembuatan dan pemeliharaan suatu perangkat lunak dengan memakai sistem atau prinsip aturan tertentu yang sistematis untuk menghasilkan sebuah perangkat lunak yang sesuai dengan kebutuhan nyata pemakai dengan tingkat fungsi dan efisiensi yang maksimal.

* + 1. **Definisi Perangkat Lunak**

Perangkat lunak (*software*) adalah sebuah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*). Sebuah program komputer tanpa terasosiasi dengan dokumentasinya maka belum dapat disebut perangkat lunak. Sistem berarti kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin disapai (Rosa A.S. dkk, 2013, p.2).

Perangkat lunak juga memunyai karakter diantaranya sebagai berikut (Rosa A.S. dkk 2013, p.2) :

1. Perangkat lunak dibangun dengan rekayasa (*software engineering*) bukan diproduksi secara manufaktur atau pabrikan.
2. Perangkat lunak tidak pernah usang (“*wear out*”) karena kecacatan dalam perangkat lunak dapat diperbaiki.
3. Barang produksi pabrikan biasanya komponen barunya aka terus diproduksi, sedangkan perangkat lunak dapat diperbaiki.

Rekayasa Perangkat Lunak lebih fokus pada bagaimana membuat perangkat lunak yang memenuhi kriteria sebagai berikut :

1. Dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan *(Maintainability)*.
2. Dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi (*Dependability* dan *Robust* ).
3. Efisien dari segi sumber yang digunakan.
4. Kemampuan untuk dipakai sesuai dengan kebutuhan *(usability).*
   * 1. **Sejarah Rekayasa Perangkat Lunak**

Perangkat lunak muncul akibat adanya krisis perangkat lunak (software crisis) yang terjadi pada tahun 1960. Krisis perangkat lunak merupakan akibat langsung dari lahirnya komputer generasi ke-3 yang canggih, ditandai dengan penggunaan *Integrated Circuit* (IC) untuk komputer. Performansi perangkat keras (*hardware*) yang meningkat, membuat adanya kebutuhan untuk memproduksi perangkat lunak yang lebih baik. Akibatnya perangkat lunak yang dihasilkan menjadi beberapa kali lebih besar dan kompleks. Pendekatan informal yang digunakan pada waktu itu dalam pengembangan perangkat lunak, menjadi tidak cukup efektif (secara *cost*, waktu dan kualitas). Biaya hardware mulai jatuh dan biaya perangkat lunak menjadi naik cepat. Karena itulah muncul pemikiran untuk menggunakan pendekatan *engineering* yang lebih pasti, efektif, standars dan terukur dalam pengembangan perangkat lunak (Richard Situmorang, 2014).

* + 1. **Tujuan Rekayasa Perangkat Lunak**

Secara umum, tujuan rekayasa perangkat lunak tidak berbeda dengan bidang rekayasa yang lain. Hal ini dapat kita lihat pada Gambar 2.1 di bawah ini :



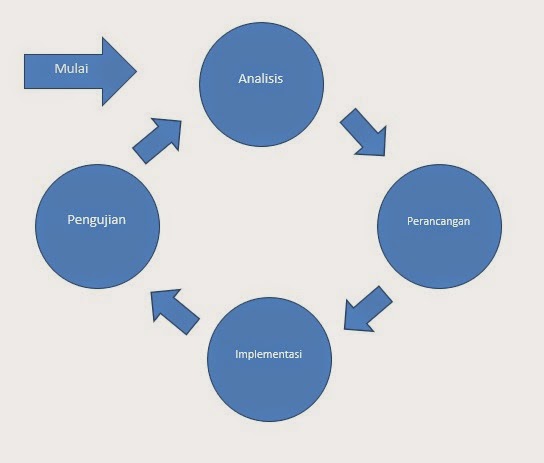
*Sumber : Cecep Lupi Hepyan, Erwin Gunadhi, dan Rina Kurniawati, 2012*

**Gambar 2.1 Tujuan Rekayasa Perangkat Lunak**

Dari Gambar di atas dapat diartikan bahwa bidang rekayasa akan selalu berusaha menghasilkan output yang kinerjanya tinggi, biaya rendah dan waktu penyelesaian yang tepat. Secara lebih khusus kita dapat menyatakan tujuan rekayasa perangkat lunak adalah (Cecep Lupi Hepyan, Erwin Gunadhi, dan Rina Kurniawati, 2012):

1. Memperoleh biaya produksi perangkat lunak yang rendah.
2. Menghasilkan perangkat lunak yang kinerjanya tinggi, handal dan tepat waktu.
3. Menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis platform.
4. Menghasilkan perangkat lunak yang biaya perawatannya rendah.
   * 1. **Tujuan Rekayasa Perangkat Lunak**

Proses-proses yang dilakukan dalam rekayasa perangkat lunak secara garis besar adalah sebagai berikut :



*Sumber : Rekayasa Perangat Lunak, Rosa A.S dkk,2013*

**Gambar 2.2 Tahapan Umum Rekayasa Perangkat Lunak**

Proses-proses pada gambar diatas dapat dilakukan berulang kali sampai perangkat lunak memenuhi kebutuhan pelanggan atau user. Untuk membangun perangkat lunak yang benar-benar baik maka diperlukan tahapan-tahapan rekayasa perangkat lunak. Perangkat lunak yang dibangun tanpa melalui tahapan rekayasa perangkat lunak yang baik maka akan seperti membuat baju tanpa pola dan rencana model baju yang baik (Rosa A.S. dkk, 2013, p.9).

Rekayasa perangkat lunak terdiri beberapa kegiatan yang harus dilakukan. Jika tahapan-tahapan tersebut tidak dilalui dengan baik, hampir bisa dipastikan perangkat lunak yang dihasilkan tidak akan mempunyai kualitas yang baik. Jadi tidak ada perangkat lunak yang bisa dihasilkan dengan baik hanya dengan satu malam seperti mitos pembuatan candi Prambanan araupun mitos pembuatan gunung Tangkuban Perahu (Rosa A.S. dkk, 2013, p.10).

Proses perangkat lunak (*Software Process*) adalah sekumpulan aktifitas yang memiliki tujuan untuk mengembangkan atau mengubah perangkat lunak. Secara umum proses perangkat lunak terdiri dari :

1. Pengumpulan Spesifikasi ( *Specification* )

Mengetahui apa saja yang harus dapat dikerjakan sistem perangkat lunak dan batasan pengembangan perangkat lunak.

1. Pengembangan (*Development*)

Pengembangan perangkat lunak untuk menghasilkan sistem perangkat lunak.

1. Validasi (*validation*)

Mamariksa apakah perangkat lunak sudah memnuhi keinginan pelanggan (*Costumer*)

1. Evalusi (*Evalution*)

Mengubah perangkat lunak untuk memenuhi perubahan kebutuhan pelanggan (*Custumer)* (Rosa A.S. dkk, 2013, p.11).

* + 1. ***Software Develompent life Cycle* (SLDC)**

Awal pengembanganya seorang programmer langsung melakukan pengodingan perangkat lunak tanpa menggunakan prosedur atau tahapan pengembangan perangkat lunak.

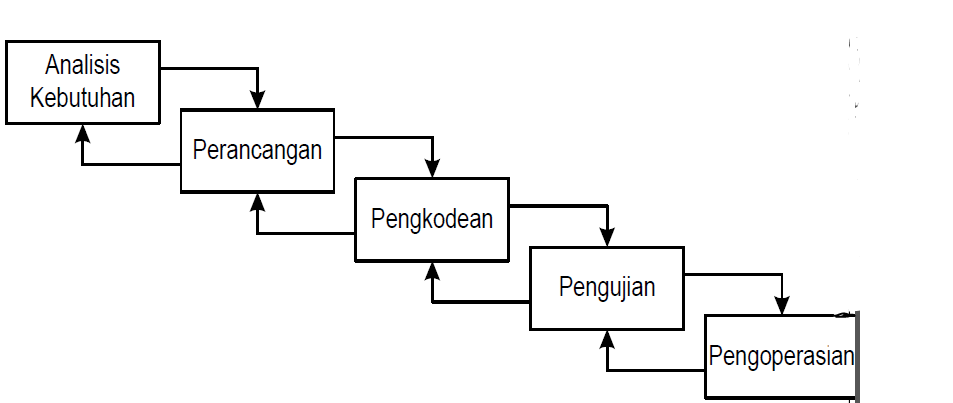
SLDC atau *Software Development Life Cycle* adalah proses pengembangan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik) (Rosa A.S. dkk, 2013, p.26).

* + 1. **Model SDLC**

SDLC memiliki bebeapa model dalam penerapan tahapan prosesnya. Beberapa model dasar SDLC antara lain :

* + - 1. ***Model waterfal***

Model SDLC air terjun *(waterwall)*sering juga disebut model sekuensial linier (*sequential linier* atau alur hidup klasik *(classic life style).* Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain pengkodean, pengujian, dan tahap pendukung *(support).*



*Sumber : Rekayasa Perangat Lunak, Rosa A.S dkk,2013*

**Gambar 2.3 Model Watefall**

1. Analisis kebutuhan Perangkat Lunak

Proses pengumpulan dilakukan dengan 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 focus pada desain pembuatan program perangkat lunak terlamasuk struktur daya, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengkodean. 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 di dokumentasikan.

1. Pembuatan kode Program

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

1. Pengujian

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

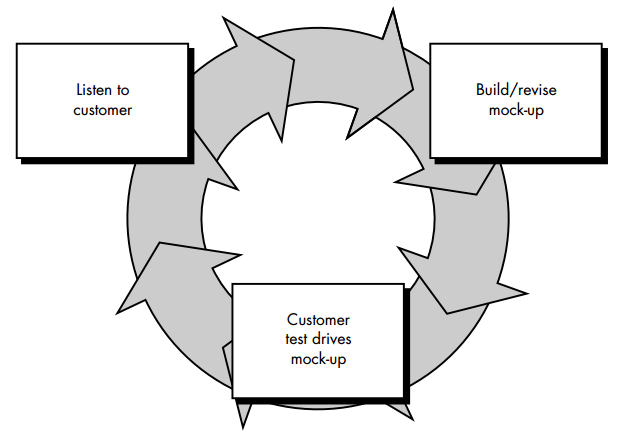
1. Pendukung *(support)* atau pemeliharaan *(maintenance)*

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke user. Perubahan bias terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

***2.1.6.2 Model Prototipe***

Model *Prototipe* dapat digunakan untuk menyambungkan ketidak pahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepadapengembang perangkat lunak(Rosa A.S. dkk, 2013,pp.31-32).

Model *Prototipe(prototyping model)* dimulai dari mengumplkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program *Prototipe* agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program *Prototipe* biasanya program yang belum jadi. Program ini biasanya menyediakan tampilan dengan simulasi alur perngkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Program *Prototipe* ini dievaluasi oleh pelanggan atau *user* sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau *user.*



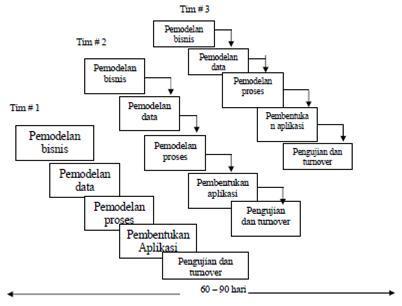
*Sumber : Rekayasa Perangat Lunak, Rosa A.S dkk,2013*

**Gambar 2.4 Model *Prototype***

*Mock-up* adalah suatu yang digunakan sebagai model desain yang digunakan untuk mengajar, demontrasi, evaluasi desain, promosi, atau keperluan lain. Sebuah *mock-up* disebut sebagai *Prototipe* perangkat lunak jika menyediakan atau mampu mendemonstrasikan sebagian besar fungsi sistem perangkat lunak dan memungkinkan penguji desain sistem perangkat lunak. iterasi terjadi pada pembuatan *Prototipe* sampai sesuai keinginan pelanggan (*customer*) atau *user.*

* + - 1. ***Model Rapid Application Development* (RAD)**

*Rapid Application Development* (RAD) adalah model proses pengembangan perangkat lunak yang bersifat *inkremental* terutama untuk waktu pengerjaan yang pendek. Model RAD adalah adaptasi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak. jika dibutuhkan perangkat lunak dibatasi dengan baik sehingga tim dapat menyelesaikan pembuatan perangkat lunak dengan waktu yang pendek. Model RAD membagi tim pengembang menjadi beberapa tim untuk mengerjakan beberapa kmponen masing-masing tim pengerjaan dapat dilakukan paralel. Model RAD dapat dilihat pada gambar 2.3 ( Rosa A.S. dkk, 2013, p.35).



*Sumber : Rekayasa Perangat Lunak, Rosa A.S dkk,2013*

**Gambar 2.5 Model RAD**

1. Pemodelan Bisnis

Pemodela yang dilakukan untuk memodelkan fungsi bisnis untuk mengetahui informasi apa yang terkait proses bisnis informasi apa saja yang harus dibuat, siapa yang harus membuat informasi itu, bagaimana alur informasi itu, proses apa saja yang terkain informsi itu.

1. Pemodelan data

Memodelkan daya apa saja yang dibutuhkan berdasarkan pemodelanbisnis dan mendefinisikan atribut-atributnya beserta relasinya dengan data-data yang lain.

1. Pemodelan proses

Mengimplemensikan fungsi bisnis yang sudah definisikan terkait dengan pendefinisian data.

1. Pebuatan aplikasi

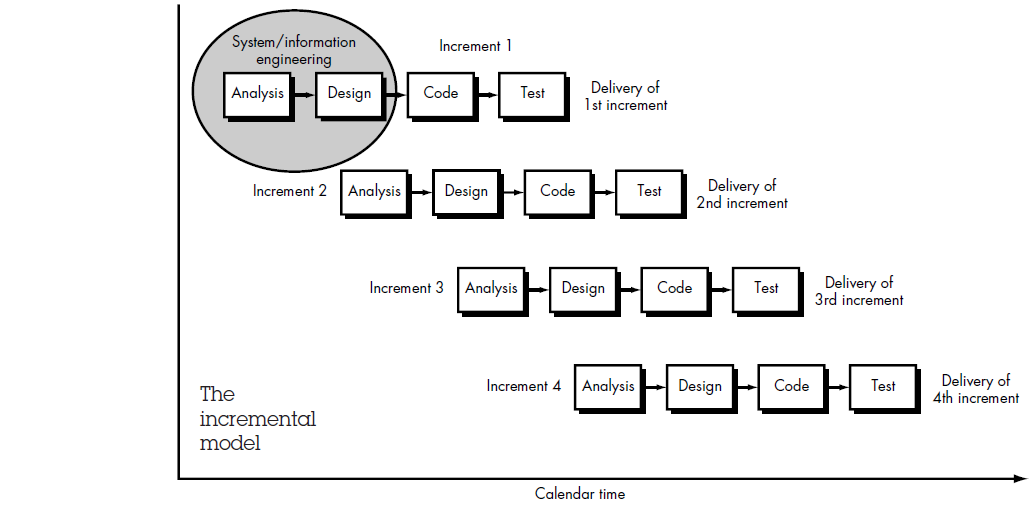
Mengimplementasikan pemodelan proses dan data menjadi program. Model RAD sangat menganjukan pemakaian komponen yang sudah ada jika dimungkinkan.

1. Pengujian dan pergantian

Menguji komponen-komponen yang dibuat. Jika sudah teruji maka tim pengembangang komponen dapat berakjak untuk mengembangkan komponen berikutnya.

* + - 1. ***Model Iteratif***

Model *Iteratif* (*iterative model*) mengkombinasikan proses-proses pada model air terjun dan iteratif pada model *Prototipe*. Model *inkremental* akan menghasilkan versi-versi perangkat lunak yang sudah mengalami penambahan fungsi untuk setiap pertambahannya.



*Sumber : Rekayasa Perangat Lunak, Rosa A.S dkk,2013*

**Gambar 2.6 Model *Iteratif***

* + - 1. **Model Spiral**

Model *Spiral* (*spiral model*) memasangkan iterative pada model *Prototipe* dengan kontrol dan aspek sistematis yang diambil dari model air terjun. Model *Spiral* menyediakan pengembangan dengan cara cepat dengan perangkat lunak memiliki versi yang terus bertambah fungsinya *(increment).*

Pada iterasi awal maka yang dihasilkan adalah *Prototipe* sedangkan pada iterasi akhir yang dihasilkan adalah perangkat lunak yang sudah lengkap. Model *Spiral* dibagi menjadi beberapa kerangka aktifitas atau disebut juga wilayah kerja (*task region*) antara lain ( Rosa A.S. dkk, 2013, pp.39-40):

1. Komunikasi dengan pelanggan *(customer communication)*

Aktifitas ini diperlukan untuk membangun komunikasi yang efektif antara pengembang (*developer*) dan pelanggan (*customer*).

1. Perencanaan (*planning*)

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

1. Analisis risiko (*risk analysis*)

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

1. Rekayasa (*engineering*)

Aktifitas ini diperlukan untuk membangun satu atau lebih representasi dari aplikasi perangkat lunak (dapat juga berupa *Protitipe*).

1. Kontruksi dan peluncuran (*contruction and release*)

Aktifitas ini dibutuhkan untuk mengkontruksi, menguji, melakukan instalasi, dan menyediakan dukungan terhadap *user* (misalnya dari segi dokumentasi dan pelaihan).

1. Evaluasi pelanggan (*customer evaluation*)

Aktifitas ini dibutuhkan untuk mendapatkan umpan balik berdasarkan evaluasi *representasi* perangkat lunak yang dihasilkan dari proses rekayasa dan diimplementasikan pada tahap *instalasi*.

* 1. ***Unified Modeling Languange*  (UML)**

*Unified Modelling Language* (UML) telah menjadi standar dalam industri untuk *visualisasi*, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan operation dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, atau VB. NET (JTI DINAMIK), Vol.XIV, No.1, Januari 2009, ISSN 0854-9524.

**2.2.1 Pengertian UML**

UML (*Unified Modelling Languange*) adalah suatu alat bantu yang sangat handal didunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atau visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme uang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain (Munawar, 2005)

UML dapat juga disebut sebagai sekumpulan pemodelan konvensi yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak dalam kaitannya dengan objek (Whitten, 2004, p430).

* + 1. **Sejarah UML**

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Bahasa pemrograman ini kurang berkembang dan dikembangkan lebih lanjut, namun dengan kemunculannya telah memberikan sumbangan yang besar pada *developer* pengembang bahasa pemrograman berorientasi objek selanjutnya. Pada tahun 1980-an berkembang bahasa pemrograman *smalltalk* yang kemudian diikuti dengan pekembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, dan CLOS. Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Metode yang pertama diperkenalkan oleh Sally Shlaer dan Stephen Mellor (Shlaer-Mellon, 1988) dan Pater Coad dan Edward Yourdon (Coad-Yourdon, 1991), diikuti oleh Grady Booch (Booch, 1991), James R. Rumbaugh, Michael R. Blaha, William Lorensen, Frederick Eddy, William Premerlani (Rumbaugh-Blaha-Premerlani-Eddy-Lorensen, 1991), dan masih banyak lagi.

Buku terkenal yang juga berkembang selanjutnya adalah karangan Ivan Jacobson (Jacobson, 1992) yang menerangkan perbedaan pendekatan yang focus pada *use case* dan proses pengembangan. Sekitar lima tahun kemudian muncul buku yang mengenai metedologi berorientasi objek yang diikuti dangan buku-bukuyang lainnya. Di dalamnya juga membahas mengenai konsep, definisi, notasi, terminologi, dan proses metodologi berorientasi objek.

Karena banyaknya metodologi-metodologi yang berkembang pesat saat itu, maka muncullah ide untuk membuat sebuah bahasa yang dapat dimenger semua orang. Usaha panyatuan ini banyak mengambil dari metodologi-metodologi yang bekembang saat itu. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep seperti konsep *Object Modelling Technique* (OTM) dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collosburators* (CRC) dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivan Jacobson, dan beberapa konsep lainnya dimana James R. Rumbaigh, Grady Booch, dan Ivan Jacobson bergabung dalam sebuah perusahaan yang bernama *Rational Software Corporation* menghasilkan bahasa yang disebut dengan *Unifield Modeling Languange* (UML)(Rosa A.S. dkk, 2013, pp.139-140):

* + 1. **Diagram UML**

Dalam *Unifield Modeling Languange*(UML) terdapat beberapa jenis diagram, diantaranya adalah :

* + - 1. ***Use-Case Diagram***

Merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Diagram ini berguna untuk menggambarkan interaksi antara pengguna dengan sebuah perangkat lunak. ada dua hal utama pada *use-case* yaitu(Rosa A.S. dkk, 2013, p.155):

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat sendiri, jadi walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use-case* merupakan fungsionalitas yang disediakan sistem sebaga unit-unit yang saling bertukar peran antar unit atau aktor.

Berikut adalah sismbol-simbol yang ada pada *use-case diagram* :

**Tabel 2.1 Simbol *Use-Case Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Actor* | Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
| 2 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
| 3 |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 4 |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara *eksplisit*. |
| 5 |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| 6 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

*Sumber : Rekayasa Perangat Lunak, Rosa A.S dkk,2013*

* + - 1. ***Class Diagram***

*Class 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.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kubutuhan sistem sehingga pembuat perangkat lunak atau programmer dapat membuat kelas-kelas didalam program perangkat lunak sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

1. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

1. Kelas yang diambil dari pendefinisian *Use-Case (controller)*

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *Use-case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

1. Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang atau membungkus data menjadi subuah sesatuan yang diambil maupun akan disimpan ke basis data.

Berikut adalah symbol-simbol yang ada pada *class diagram :*

**Tabel 2.2 Simbol *Class Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Generalization* | Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 2 |  | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | *Collaboration* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor |
| 5 |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 6 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 7 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya |

*Sumber : Rekayasa Perangat Lunak, Rosa A.S dkk,2013*

* + - 1. ***Activity Diagram***

Diagram activitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atauproses bisnis atau menu yang ada pada perangkat lunak. yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa A.S. dkk, 2013, p.161).

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan dari tampilan dari sistem / *userinterface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kusus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

**Tabel 2.3 Simbol *Activity Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| **1** |  | *Actifity* | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| **2** |  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| **3** |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| **4** |  | *Actifity Final Node* | Bagaimana objek dibentuk dan dihancurkan |
| **5** |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

*Sumber : Rekayasa Perangat Lunak, Rosa A.S dkk,2013*

* + - 1. ***Sequence Diagram***

Diagram sekuen menggambarkan kelakuan objek pada *Use-case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *Use-case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat scenario yang ada pada *Use-case.*

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *Use-case* yang memiliki proses sendiri atau yang penting semua *Use-case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *Use-case* yang didefinisikan maka diagram sekuen yang juga harus dibuat juga semakin banyak (Rosa A.S. dkk, 2013, p.165).

**Tabel 2.4 Simbol *Sequence Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *LifeLine* | Objek *entity*, antarmuka yang saling berinteraksi. |
| 2 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |
| 3 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |

*Sumber : Rekayasa Perangat Lunak, Rosa A.S dkk,2013*

* + - 1. ***Statechart Diagram***

*Statecart Diagram* atau dalam bahasa Indonesia disebut diagram mesin status atau sering juga disebut diagram status digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin sistem atau objek. Jika diagram sekuen dignakan untuk interaksi antar objek maka diagram status digunakan untuk interaksi di dalam sebuah objek. Perubahan tersebut digambarkan dalam suatu graf berarah. (Rosa A.S. dkk, 2013, p.163).

**Tabel 2.5 Simbol *Statechart Diagram***

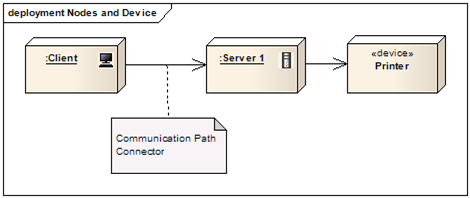
|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *State* | Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek. |
| 2 |  | *Initial Pseudo State* | Bagaimana objek dibentuk atau diawali |
| 3 |  | *Final State* | Bagaimana objek dibentuk dan dihancurkan |
| 4 |  | *Transition* | Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya |
| 5 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 6 |  | *Node* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

*Sumber : Rekayasa Perangat Lunak, Rosa A.S dkk,2013*

* + - 1. ***Deployment Diagram***

Diagram *deployment* atau *deployment diagram* menunjukan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut :

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device, node,* dan *hardware.*
2. Sistem *client/server* misalnya seperti gambar berikut :



*Sumber : Rekayasa Perangat Lunak, Rosa A.S dkk,2013*

**Gambar 2.7 Diagram *Deployment* Sistem *client/server***

1. Sistem terdistribusi murni.
2. Rekayasa ulang aplikasi.

Berikut adalah symbol-simbol yang ada pada diagram *deployment :*

**Tabel 2.6 Simbol *Deployment Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *komponen* | Pada deployment diagram, komponenkomponen yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka |
| 2 |  | *Node* | Node menggambarkan bagian-bagian hardware dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi. |
| 3 |  | *Association* | Sebuah association digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara element-elemen hardware. |

*Sumber : Rekayasa Perangat Lunak, Rosa A.S dkk,2013*

* 1. **Kecerdasan Buatan / *Artificial Intelligence***

Kecerdasan Buatan (*Artificial Intelligence atau AI*) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia.

* + 1. **Definisi Kecerdasan Buatan**

Kecerdasan buatan (*Artificial Intelligence*) merupakan salah satu 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 oleh manusia.

Menurut Jhon McCarthy, 1956, AI : untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendisain mesin agar dapat menirukan perilaku manusia cerdas berarti memiliki pengetahuan ditambah pengalaman, penalaran (bagaimana membuat keputusan dan mengambil tindakan), moral yang baik.

Manusia cerdas pandai dalam menyelesaikan permasalahan karena manusia mempunyai pengetahuan dan pengalaman. Pengalaman di peroleh dari belajar, semakin banyak bekal pengetahuan yang dimiliki tentu akan lebih mampu menyelesaikan suatu permasalahan. Tapi bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengalaman dan pengetahuan.

Demikian juga agar mesil bisa cerdas (bertindak seperti dan sebaik manusia) makan harus diberi bekal pengetahuan, sehingga mempunyai kemampuan untuk menalar. Untuk membuat aplikasi kecerdasan buatan ada 2 bagian utaman yang sangat dibutuhkan (jurnal SAINTIKOM, Vol.5, No.2, 2008) :

1. Basis pengetahuan (knowladge base), bersifat fakta-fakta, teori, pemikiran dan hubungan antar satu dengan yang lainnya.
2. Motor inferensi (Inteference engine), kemampuan menarik kesimpulan berdasarkan pengetahuan dan pengalaman.

Penerapan konsep kecerdasan buatan pada komputer adalah sebagai berikut :



*Sumber : Kecerdasan Buatan, Muhammad Dahria, 2008*

**Gambar 2.8 Penerapan Konsep Kecerdasan Buatan di Komputer**

* + 1. **Sejarah Kecerdasan Buatan**

Selama bertahun-tahun para filosuf berusaha mempelajari kecerdasan yang dimiliki manusia. Dari pemikiran tersebut lahirlah **AI** sebagai cabang ilmu yang berusaha mempelajari dan meniru kecerdasan manusia. Secara lengkap sejarah perkembangan **AI** dapat dikemukakan sebagai berikut:

**Abad ke-17 sampai Abad ke-19**

Abad ini merupakan titik awal perkembangna Kecerdasan Buatan. Hal ini ditandai oleh penemuan-penemuan berikut :

**Rane Descartes** mengemukakan bahwa semua tidak ada yang pasti kecuali kenyataan bahwa seseorang bisa berfikir.

**Blaise Pascal** berhasil menciptakan mesin penghitungan digital mekanis pertama pada tahun 1642.

**Charles Babbage** dan **Ada Lovenace**  berhasil membuat mesin penghitung mekanis yang dapat deprogram.

**Bertrand Russell & North Whitehead** menerbitkan buku Principa Matematika, yang merombak logika formal.

**Pada 1950-1970**

Tahun-tahun ini merupakan tahun pembuka bagi Kecerdasan Buatan, dimana para ilmuan dan peneliti mulai memikirkan cara agar mesin dapat melakukan pekerjaannya seperti yang dilakukan oleh manusia. Hal ini ditandai oleh beberapa penemuan-penemuan berikut. Pada Februari 1951, *University Of Manchester* telah berhasil mengembangkan komputer elektronik pertama di Dunia diberi nama “Ferranti Mark I” (T. Sutojo. dkk , 2011, p.3-7).

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

Jika dibandingkan dengan kecerdasan alami (kecerdasan yang dimiliki oleh manusia), kecerdasan buatan memiliki keuntungan, antara lain (T. Sutojo. dkk , 2011, p.10-11) :

1. Kecerdasan Buatan lebih bersifat permanen. Kecerdasan alami bisa berubah karena sifat manusia pelupa. Kecerdasan buatan tidak berubah selama sistem komputer dan program tidak merubahnya.
2. Kecerdasan Buatan lebih mudah diduplikasikan dan disebarkan, pengetahuan yang diletakan pada suatu sistem computer, pengetahuan tersebut dapat disalin dari komputer tersebut dan dapat dipindahkan dengan mudah ke komputer lain.
3. Kecerdasan Buatan lebih murah dibandingkan Kecerdasan Alami karena kecerdasan alami akan memakan waktu sangat lama dalam pelaksanaannya.
4. Kecerdasan Buatan bersifat konsisten sedangakan kecerdasan alami lebih cenderung berubah-ubah.
5. Kecerdasan Buatan dapat didokumentasikan dengan cara melacak setiap aktifitas dari system tersebut.
6. Kecerdasan Buatan dapat mengerjakan pekerjaan lebih cepat disbanding dengan Kcerdasan Alami.
7. Kecerdasan Buatan dapat mengerjakan pekerjaan lebih baik dibanding Kecerdasan Alami.

Sementara itu, kecerdasan alami memiliki keuntungan, antara lain :

1. Kratif
2. Kecerdasan Alami memungkinkan seseorang menggunakan pengalaman secara langung.
3. Pemikiran manusia dapat digunakan secara luas, sedangkan Kecerdasan Buatan sangat terbatas.
   * 1. **Komputasi Kecerdasan Buatan dan Pemrograman *Konvensional***

Pada umumnya pemrograman *Konvensional* hanya diperuntukkan sebagai alat hitung, sedangkan Kecerdasan Buatan digunakan untuk menyelesaikan suatu permasalahan. Oleh karena itu, ada beberapa perbedaaan yang mendasar antara Kecerdasan Buatan dengan pemrograman *Konvnsional.* Tabel 2.7 menunjukan perbedaan komputasi kecerdasan buatan dengan komputasi pemrograman *Konvensional* (T. Sutojo. dkk , 2011, p.12).

**Tabel 2.7 Kecerdasan Buatan VS Perograman *Konvensional***

|  |  |  |
| --- | --- | --- |
| **Dimensi** | **Kecerdasan Buatan** | **Pemrograman *Konvensional*** |
| Pemrosesan | Pengetahuan diperoleh dari mekanisme pemrosesan (inferensi) | Digabung dalam satu program *sekuensial* |
| Eksekusi | Eksekusi dilakukan secara *heuristik* dan logis | Secara algoritma (*step-by-step*) |
| Sifat *input* | Bisa tidak lengkap | Harus lengkap |
| Manipulasi | Efektif pada basis pengetahuan yang besar | Efektif pada database yang besar |
| Keterangan | Disediakan | Biasanya tidak didasarkan |
| Fokus | Pengetahuan | Data & informasi |
| Struktur | kontrol dipisahkan dari pengetahuan | Kontrol terintegrasi dengan informasi |
| Sifat *output* | *Kuantitatif* | *Kualitatif* |
| Perubahan | Perubahan pada kaidah dapat dilakukan dengan kaidah yang sedikit | Pada program merepotkan |
| Pemeliharaan &*update* | *Relatif* mudah | Sulit |
| Kemampuaan menalar | Ya | tidak |

*Sumber : Kecerdasan Buatan, T. Sutojo. dkk, 2011*

* 1. **Jaringan Syaraf Tiruan**

Jaringan Saraf Tiruan (JST) adalah paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia. Elemen kunci dari paradigma ini adalah struktur dari sistem pengolahan informasi yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (neuron), bekerja bekerja serentak untuk menyelesaikan masalah tertentu (T. Sutojo. dkk , 2011, p.283)

* + 1. **Model Matematika**

JST dibentuk sebagai generalisasi model matematika dari jaringan saraf biologi, dengan asumsi bahwa (Jong Jek Siang, 2009, p. 2) :

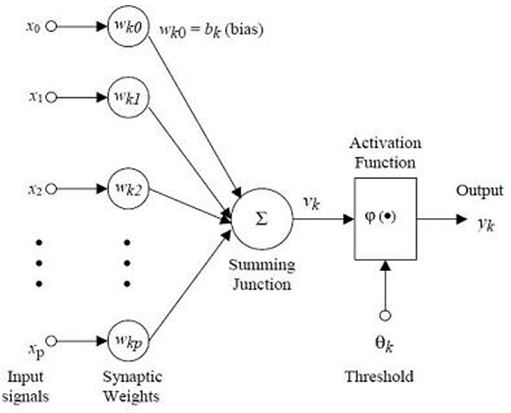
1. Peprosesan informasi terjadi pada banyak elemen sederhana (*neuron*)*.*
2. Sinyal dikirimkan diantara *neuron-neuron*melalui penghubung-penghubung.
3. Penghubung antar *neuron* memiliki *bobot* yang akan memperkuat atau memperlemah sinyal.
4. Untuk menentukan *output,* setiap *neuron* menggunakan fungsi aktivasi (biasanya bukan fungsi *linier*) yang dikenakan pada jumlah *input* yang diterima. Besarnya *output* ini selanjutnya dibandingkan dengan suatu batas ambang.

Terdapat tiga komponen dasar penting ketika kita membuat sebuah model fungsional dari *neuron* biologis. Pertama, sinapsis *neuron* dimodelkan sebagai *bobot.* Kekuatan hubungan antara masukan dan *neuron* ditentukan oleh nilai *bobot*. Nilai *bobot* negatif mencerminkan koneksi hambat, sedangkan nilai-nilai positif menandakan koneksi rangsa sel. Komponen kedua adalah penjumlahan semua masukan yang dimodifikasi oleh masing-masing *bobot*. Kegiatan ini disebut sebagai kombinasi *linier*. Komponen ketiga bertindak sebagai fungsi control aktivasi *amplitude output* dari neuron (T. Sutojo. dkk , 2011, p.290).

Berdasarkan model matematis tersebut, baik tidaknya suatu model JST ditentukan oleh hal-hal berikut (Jong Jek Siang, 2009, pp.3-4) :

1. Pola hubungan antar *neuron* (disebut arsitektur jaringan).
2. Metode untuk menentukan *bobot* penghubung (disebut metode *training/ learning*/ algoritma).
3. Fungsi aktivasi.

Secara sistematis, proses ini dijelaskan dalam gambar dibawah ini :



*Sumber : Kecerdasan Buatan, T. Sutojo. dkk , 2011.*

**Gambar 2.9 Model Matematis JST**

*y* menerima input dari *neuron x1,x2* dan *x3* dengan bobot hubungan masing-masing adalah *w1, w2* dan *w3*. Ketiga impuls neuron yang ada dijumlahkan.

*x1w1 + x2w2 + x3w3*

Besarnya *impuls* yang diterima oleh *y* mengikuti fungsi aktivasi *y* = *f(net).* Apabila nilai fungsi aktivasi cukup kuat,maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk merubah *bobot*.

* + 1. **Arsitektur Jaringan Syaraf Tiruan**

Jaringan Saraf Tiruan memiliki beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi [1]. Arsiterktur Jaringan Saraf Tiruan itu antara lain (Jurnal Rekrusif, Samuel, Ernawati, Diah. Vol.3 No.2 November 2015, ISSN 2303-0755. p.84) :

1. Jaringan Layar Tunggal *(single layer network)*

Jaringan dengan lapisan tunggal terdiri dari1 layar *input*dan 1 layar *output*. Setiap *neuron/*unit yang terdapat dilapisan/*layerinput* selalu terhubung dengan setiap *neural* yang terdapat pada layar *output.* Contoh algoritma Jaringan Saraf Tiruan yang menggunakan metode ini yaitu :ADALINE, hofield, Perceptron. Arsitektur jaringan layar tunggal dapat dilihat pada gambar berikut (T. Sutojo. dkk , 2011, p.292) :



*Sumber : Implementation of Signature Recognitionby Using Backpropagation, Rini Sovia, Musli Yanto, and Widya Nursanty. Vol 1, No 1, 2016.*

**Gambar 2*.*10 Jaringan Satu Layar**

1. Jaringan Layar Jamak *(multi layer network)*

Jaringan dengan lapisan jamak memiliki cir khas tertentu yaitu memiliki 3 jenis *layer*, yakni *input, layer output*, layar tersembunyi (*hidden layer*). Contoh algoritma Jaringan Saraf Tiruan menggunakan metode ini yaitu : MADALINE, *Backpropagation, Neocognitron*. Untuk lebih memahami jaringan layar jamak, dapat dilihat pada gambar berikut ini (T. Sutojo. dkk , 2011, p.293) :



*Sumber : Implementation of Signature Recognitionby Using Backpropagation, Rini Sovia, Musli Yanto, and Widya Nursanty. Vol 1, No 1, 2016.*

**Gambar 2*.*11 jaringan Multi Layar**

1. Jaringan dengan lapisan kompetitif

Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif. Contoh algoritma yang menggunakan metode ini adalah LVQ. Arsitektur jaringan kompetitif dapat diperhatikan pada gambar dibawah ini (T. Sutojo. dkk , 2011, p.294) :



*Sumber : Implementation of Signature Recognitionby Using Backpropagation, Rini Sovia, Musli Yanto, and Widya Nursanty. Vol 1, No 1, 2016.*

**Gambar 2*.*12 Jaringan Kompetitif**

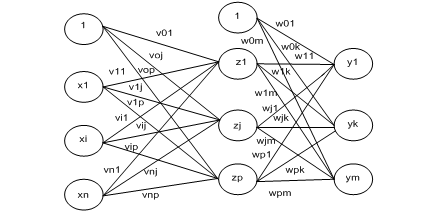
* 1. ***Backpropagation***

*Backpropagation* adalah salah satu metode jaringan syaraf tiruan dengan proses belajar terawai.(Jurnal Ultimatics, Sari. Vol.3 No.2 Desember 2011, ISSN 2085-4552. p.23).

* + 1. **Arsitektur *Backpropagation***

*Backpropagation* memiliki beberapa unit yang ada dalam satu atau lebih layar tersembunyi. Gambar 2.11 adalah arsitektur *backpropagation*dengan n buah masukan (ditambah sebagai bias), sebuah layar tersembunyi yang terdiri dari p unit (ditambah sebuah bias), serta m buah unit keluaran.

merupakan *bobot* garis dari unit masukan ke unit layar tersembunyi ( merupakan *bobot* garis yang menghubungkan bias di unit masukan ke unit layar tersembunyi ). merupakan *bobot* dari unit layar tersembunyi ke unit keluaran ( merupakan bobot dari bias di layar tersembunyi ke unit keluaran ).



*Sumber : Jaringan Saraf Tiruan & Pemrogramannya menggunakan MATLAB, Jong Jek Siang, 2009.*

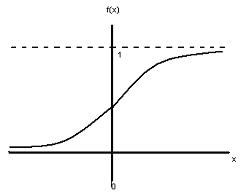
**Gambar 2.13 Arsitektur *Backpropagation***

* + 1. **Fungsi *Activasi***

Dalam *Backpropagation,*  fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu : *continu, terdiferensial* dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering dipakai adalah :

dengan aturan

Grafis fungsi tampak pada gambar dibawah ini :

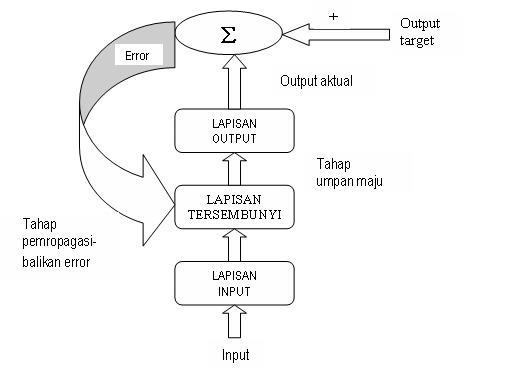


*Sumber : Jaringan Saraf Tiruan & Pemrogramannya menggunakan MATLAB, Jong Jek Siang, 2009.*

**Gambar 2.14 Fungsi Aktivasi *Sigmoid Function***

* + 1. **Pelatihan Algoritma *Backpropagation***

Seperti telah disebut diatas, bahwa pelatihan algoritma *Backpropagation* terdiri dari 3 fase, dimana ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi.



*Sumber : Jaringan Saraf Tiruan & Pemrogramannya menggunakan MATLAB, Jong Jek Siang, 2009.*

**Gambar 2.15 Alur Kerja Jaringan *Propagasi* Balik**

Agar dapat digunakan untuk suatu palkasi, Jaringan Saraf Tiruan perlu ‘belajar’ terlebih dahulu. Caranya, pada jaringan dimasukan sekumpulan cotoh pelatihan yangdisebut set pelatihan. Set pelatihan ini digambarkan dengan sebuah fektor *feature* yang disebut *vector input* yang diasosiasikan dengan sebuat output menjadi target pelatihannya (Diyah Puspitaningrum, 2006).umumnya kondisi penghentian yang sering dipakai adalah jumlah *iterasi* atau kesalahan. *Iterasi* akan dihentikan jika jumlah *iterasi* yang dilakukan sudah melebihi jumlah maksimum *iterasi* yang ditentukan atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan. Berikut adalah langkah-langkah pelatihan pada algoritma *Backpropagation* dengan satu layar tersembunyi (dengan fungsi aktivasi *siqmoid biner*) (Jong Jek Siang, 2009).

Langkah 0 : inisialisasi bobot dengan bilangan acak kecil.

Langkah 1 : jikakondisi penghentian belum terpenuhi, lakukan langkah 2-9

Langkah 2 : untuk setiap pasang pada data pelatihan, lakukan langkah 3-8

Langkah 3 : tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi diatasnya.

Langkah 4 : hitung semua keluaran di unit tersembunyi *( J = 1,2,…,p)*

] (I)

(II)

Langkah 5 : hitung semua keluaran jaringan ke unit (k= 1,2,…,m)

] (III)

(IV)

Fase II : Propagasi mundur

Langkah 6 : hitung faktor unit keluaran berdasarkan kesalahan di setiap unit keluaran *(k= 1,2,…,m)*

) (1-) (V)

merupakan unit kesalahan yang akan dipakai dalam perubahan *bobot* layar dibawahnya

Hitung suku perubahan *bobot* (yang akan dipakai nanti untuk merubah *bobot* dengan laju tercepat

= (VI)

Langkah 7 : hitung factor unit tersembunyi berdasarkan kesalahan di setiap unit

(VII)

Factor tersembunyi :

(VII)

Hitung suku perubahan bobot (yang akan dipakai nanti untuk merubah bobot

= (IX)

Fase III : Perubahan *Bobot*

Langkah 8 : hitung semua perubahan *bobot*

Perubahan *bobot* garis yang menuju ke unit keluaran :

= (X)

Perubahan *bobot* garis yang menuju ke unit tersembunyi :

= (XI)

Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, hanya *propagasi* maju (langkah 4 dan 5 yang dipakai untuk menentukan keluaran jaringan).

Jika fungsi aktivasi yang dipakai bukan *siqmoid biner,* maka langkah 4 dan 5 harus disesuaikan. Demikian juga turunannya pada pola 6 dan 7.

* + 1. **Optimalisai Arsitektur *Backpropagation***

Masalah utam yang dihadapi dalam *Backpropagation* adalah lamanya iterasi yang harus dilakukan. *Backpropagation*tidak dapat memberikan kepasrtian tentang beberapa *epoch* yang harus dilalui untuk mencapai kondisi yang diinginkan. Oleh karena itu orang berusaha meneliti bagaimana parameter-parameter jaringan dibuat sehingga menghasilkan jumlah iterasi yang relative lebih sedikit. (Jong Jek Siang, 2009).

* + - 1. ***Pemilihan Bobot dan Bias Awal***

*Bobot* awal akan mempengaruhi apakah jaringan mencapai titik minimum local atau global, dan seberapa cepat konvergensinya.

**Nyuyen** dan **Window (1990)** mengusulkan cara membuat inisialisasi *bobot* dan bias ke unit tersembunyi sehingga menghasilkan iterasi lebih cepat.

Misal n = jumlah unit masukan

P = jumlah unit tersembunyi

= faktor skala = 0.7

Algoritma inisialisasi Nyuyen Windrow adalah sebagai berikut :

1. Inisialisasi semua bobot ( (lama)) dengan bilangan acak dalam *interval* [-*0,5, 0,5*]
2. Hitung
3. Bobot yang dipakai sebagai berikut = =
4. Bias yang dipakai sebagai inisialisasi = = bilangan acak antara – dan
   * + 1. ***Jumlah Unit Tersembunyi***

Hasil teoritis yang didapat menunjukan bahwa jaringan dengan sebuah layar tersembunyi sudah cukup bagi *Backpropagation* untuk mengenali sembarang perkawanan antara masukan dan target dengan tingkat ketelitian yang ditentukan. Akan tetapi penambahan jumlah layar tersembunyi kadangkala membuat penelitian lebih mudah. Jika jaringan memiliki lebih dari satu layar tersembunyi, maka algoritma pelatihan yang dijabarkan sebelumnya perlu direvisi. Dalam propagasi maju, keluaran harus dihitung untuk tiap layar, dimulai dari layar tersembunyi paling bawah. Sebaliknya, dalam propagasi mundur, faktor perlu dihitung untuk tiap layar tersembunyi, dimulai dari layar keluaran (Jong Jek Siang, 2009).

* + - 1. ***Jumlah Pola Pelatihan***

Tidak kepastian tentang beberapa banyak pola yang diperlukan agar jaringan dapat dilatih dengan sempurna. Jumlah pola yang dibutuhkan dipengaruhi oleh banyaknya *bobot* dalam jaringan serta tingkat akurasi yang diharapakan. Aturan kasarnya dapat dibentuk berdasarkan rumsan :

Jumlah pola = *jumlah bobot / tingakt akurasi*

Unutk jaringan dengan *80 bobot* dan tingkat akurasi *0,1* maka *800* pola masukan diharapkan akan mampu mngenali dengan benar *90%* pola diantaranya (Jong Jek Sing, 2009).

* + - 1. ***Lama Iterasi***

Tujuan utama penggunaan *Backpropagation* adalah mendapatkan keseimbangan antara pengenalan pola penelitian secara benar dan *respon* yang baik untuk pola lain yang sejenis (disebut data pengujian). Jaringan dapat dilatih terus menerus hingga pola pelatihan dikenali dengan benar. Akan tetapi hal itu tidak menjamin jaringan akan mampu mengenali pola pengujian dengan tepat. Jadi tidaklah bermanfaat untuk meneruskan iterasi hingga semua kesalahan pola pelatihan = 0 (Jong Jek Sing, 2009).

* 1. **Prediksi Dengan Algoritma *Backpropagation***

Ketika prediksi sama dengan ramalan atau perkiraan. Menurut kamus bahasa Indonesia besar, prediksi adalah hasil dari kegiatan memprediksi atau meramal atau memperkirakan. Prediksi bisa berdasarkan metode ilmiah ataupun bubjektif belaka.

Prediksi sendiri sebenarnya mempertimbangkan nilai-nilai yang belum terlihat pada masa akan datang berdasrkan pola-pola yang terjadi sebelumnya. Prediksi merupakan proses memperkirakan akan sesuatu dengan mengkoreksi aksi sebelumnya, untuk meramalkan suatu kondisi dbutuhkan perhitungan yang tepat guna menjawab permasalahan tesebut. Perhitungan tersebut dapat menggunakan Jaringan Saraf Tiruan *Backpropagation*.

Langkah-langkah membangun struktur jaringan untuk peramalan sebagai berikut (maru’ao, 2010):

1. Pembagian Data

Langkah pertama adalah pembagian data. Data dibagi menjadi data pelatihan dan pengujian. Beberapa komposisi data pelatihan dan pengujian yang sering digunakan adalah sebagai berikut :

1. *80%* untuk data peatihan dan *20%* untuk data pengujian
2. *70%* untuk data pelatihan dan *30%* untuk data pengujian
3. *2/3* untuk data pelatihan dan *1/3* untuk data pengujian
4. *50%* untuk data pelatihan dan *50%* untuk data pengujian
5. *60%* untuk data pelatihan dan *40%* untuk data pengujian

Aspek pembagian data harus ditekankan agar jaringan mendapat data pelatihan yang secukupnya dan data pengujian dapat menguji prestasi pelatihan yang dilakukan berdasarkan nilai MSE data pelatihan dan pengujian.

1. *Preprocessing* / *Normalisasi*

sebelum dignakan untuk proses pelatihan, perlu dilakukan peaksanaan terhadap harga-harga *input* dan *target* sedemikian hingga data-data *input* dan *target* tersebut masuk dalam satu *range* tertentu disebut *preprocessing* atau normalisasi data. Runtun data masukan dan target dinormalisasi dangan membawa data ke bentuk normal yangmemiliki *mean =0* dan *devisi* standar *= 1*.

1. Perancangan struktur jaringan yang optimum

Langkah selanjutnya adalah penentuan jumlah lapisan masukan (*Input)*, lapisan tersembunyi, dan jumlaj lapisan keluaran yang akan digunakan dalam jaringan. Pengguaan jaringan dengan dua atau lebih lapisan tersembunyi dalam masalah peramalan kebanyakan tidak akan memberikan pengaruh yang sangat besar tehadap prestasi jaringan untuk melakukan peramalan. Selain itu memperlambat proses pelatihan yang disebabkan bertambahnya unit.

1. Pemilihan *koefisen* pemahaman (*learning rate* ) dan *Momentum*

Pemilihan *koefisien* pemahaman dan *momentum* mempunyai peranan yang penting untuk struktur jaringan yang akan dibangun dan digunakan dalam peramalan, hasil keputusan yang Pkurang memuaskan dapat diperbaiki dengan pengunaan *koefisien* pemahaman dan momentum secara *trial and error* untuk mendapatkan nilai *bobot* yang paling *optimum* agar MSE jaringan dapat diperbaiki.

1. *Postprocessing* / *Denormalisai*

Setelah proses pelatihan selesai, harga-harga ternormalisasi *output* jaringan harus dikembalikan (dinormalisasi) ke harga aslinya untuk mendapatkan nilai *output* pada *range*  yang sebenarnya.

* + 1. ***Bitcoin***

Bitcoin merupakan *pionir* dalam *Cryptocurrency* sekaligus implementasi teknologi *blockchain* yang pertama. Beberapa konsep yang digunakan adalah adanya sebuah basis data bernama *blockchain* yaitu berupa “buku besar” (*ledger*) yang dapat dilihat oleh semua orang, sehingga semua orang dapat melihat dan memvalidasi transaksi keuangan yang dilakukan di dalam *blockchain* tersebut. Buku besar ini mencatat seluruh transaksi yang terjadi, sehingga alur transaksi dapat dilihat dengan mudah (Dimaz A.W, 2017, p.35).

*Bitcoin* adalah uang digital yang tidak diatur oleh bank central manapun dan negara manapun. Bitcoin adalah uang yang tidak berbentuk hanya bisa dilihat dari saldo yang terhubung dengan *server* yang menyimpan data uang digital tersebut. Meskipun informasi harta *bitcoin* disimpan di *server*, tidak ada orang lain yang bisa membelanjakan *bitcoin* tanpa sepengetahuan pemilik karena setiap pemilik memiliki kunci rahasia masing-masing (Dimaz A.W, 2017, p.36).

* + - 1. **Sejarah *Bitcoin***

Perjalanan *bitocin* dimulai dari terciptanya *Genesis Block*, yakni blok bernomor 0 yang dibuat pada tanggal 3 januari 2009 oleh Satoshi Nakamoto. Sejak *bitcoin* dikembangkan berbagai perbaikan diterapkan berdasarkan saran dan masukan dari anggota *Cypherpunk* yang terpukau dengan ide mata uang virtual ini. *Bitcoin* yang semula tidak bernilai, lambat laun mendapatkan tempat di berbagai komunitas sebagai sebuah komoditas virtual yang dijalankan oleh sistem independen. Harga *bitcoin* ikut terkeret naik, berhubung semakin banyak pula pengguna yang terlibat. Kini setelah 8 tahun sejak *Genesis Block*, *bitcoin* menjadi mata uang virtual dengan valuasi pasar terbesar di dunia (Dimaz A.W, 2017, p.31).

* + - 1. **Jaringan *Bitcoin***

Terdapat 2 jenis jaringan di dalam sistem *bitcoin* yang memiliki cara kerja sistem yang sama, namun tidak saling berhubungan. *Mainnet* merupakan jaringan utama dimana transaksi-transaksi *bitcoin* dilakukan. Sementara *testnet* merupakan jaringan yang digunakan untuk ujicoba. Saat ini tesnet merupakan versi ketiga yang disebut **Testnet3** (Dimaz A.W, 2017, p.39)..

Perbedaan lain yang mencolok di antara 2 jenis jaringan adalah awalan kalimat yang digunakan. Mainnet menggunakan awalan 1, sementara tesnet3 menggunakan awalan 2 atau 3. Selain itu *bitcoin* di dalam testnet3 dapat diperoleh secara gratis dari berbagai *faucet* yang menyediakan, misalnya yang disediakan oleh Coinfaucet.eu. meskipun memili mekanisme yang sama dengan *bitcoin*, test *bitcoin* tidak disarankan untuk diperjualbelikan karena setiap saat dapat dihapus (Dimaz A.W, 2017, p.39).

*Bitcoin* juga memiliki sebuah jaringan lokal yang disebut dengan *regtest* (*regression test mode*). *Regtest* merupakan fasilitas dalam *bitcoin core* dimana pengembang dapat membuat jaringan *regtest* lokal sendiri untuk mengujicoba sistem tanpa resiko merusak sistem jaringan utama *mainnet* maupun mengganggu jaringan *tesnet*. *Regtest* memiliki *fleksibilitas* dalam hal membuat blok baru dan kemudahannya untuk di-*reset* ulangapabila terjadi hal-hal yang diluar kendali (Dimaz A.W, 2017, p.40).

* 1. **WEB**

*Word Wide Web* (WWW), lebih dikenal dengan web, merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet. Web pada awalnya adalah ruang informasi dalam internet, dengan menggunakan teknologi *hyperteks,* pemakai dituntun untuk menemukan informasi dengan mengikuti link yang disediakan dalam dokumen web yang ditampilkan dalam *browser* web (Betha Sidik, 2014, p.3).

* + 1. **Defenisi Website**

*Website* atau situs dapat diartikan sebagai sekumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video, dan atau gabungan dari semuanya, baik yang bersifat statis maupun yang bersifat dinamis yang membentuk suati rangkaian bangunan yang saling terkair dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*Ihyperlink*). Bersifat dinamis apabila isi informasi website selaluberubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna (Jurnal STT-Garut), Vol.09, No.40, 2012, ISSN : 2302-7339.

Sebuah website biasanya ditempatkan setidaknya pada sebuah server web yang dapat diakses melalui jaringan seperti internet, ataupun Local Area Network (LAN) melalui alamat internet yang dikenali sebagai URL (Uniform Resource Locator). Gabungan atas semua situs yang dapat diakses publik di internet disebut pula sebagai World Wide Web atau lebih dikenal dengan singkatan WWW (Eka P.W.M, S.kom, M.kom, 2015 : 1-2).

* + 1. **Sejarah Website**

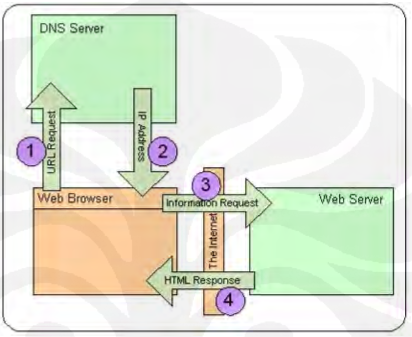
Penemu *website* adalah Sir Timothy John ¨Tim¨ Berners-Lee, sedangkan *website* yang tersambung dengan jaringan, pertamakali muncul pada tahun 1991. Maksud dari Tim ketika membuat *website* adalah untuk mempermudah tukar menukar dan memperbarui informasi kepada sesama peneliti di tempat dia bekerja. Pada tanggal 30 April 1993, CERN (tempat dimana Tim bekerja) menginformasikan bahwa WWW dapat digunakan secara gratis oleh semua orang (Betha Sidik, 2014, p.5).

Sebuah website dibuat di dalam sebuah sistem komputer yang dikenal dengan *server* web, juga disebut HTTP Server, dan pengertian ini juga bisa menunjuk pada software yang dipakai untuk menjalankan sistem ini, yang kemudian menerima lalu mengirimkan halaman-halaman yang diperlukan untuk merespon permintaan dari pengguna. *Apache* adalah piranti lunak yang biasa digunakan dalam sebuah *webserver*, kemudian setelah itu adalah *Microsoft Internet Information Services* (IIS) (Betha Sidik, 2014, p.5).

* + 1. **cara kerja website**

Sebuah halaman web diakses dengan menggunakan web *browser* denganmenuliskan URL atau mengikuti *link*yang menuju kepadanya. *Uniform Resource Locator* (URL) akan menunjukan lokasi dokumen yang dikelola oleh sebuah web *server*. URL diubah menjadi alamat IP *server*yang bersangkutan. *Browser* kemudian mengirim request *Hypertext transfer Protocol* (HTTP) ke web *server* dan web *server* akan menjawab dokumen yang diminta dalam format *hyperttext markup language* (HTML). HTTP adalah suatu protokol yang menentukan aturan yang perlu diikuti oleh web *browser* dalam meminta atau mengambil suatu dokumen. Protokol ini merupakan protokol standart yang digunakan untuk mengakses dokumen HTML (Betha Sidik, 2014, p.6).

Dalam komunikasi jaringan komputer diatur oleh protokol yang memungkinkan beragam jaringan komputer untuk berkomunikasi. Protokol ini secara resmi dikenal dengan *Transmission Control Protocol* (TCP/IP) yang merupakan cara untuk mempaketkan sinyal elektronik sehingga data tersebut dapat dikirim ke komputer lain. Berikut gambaran cara kerja web yang di akses :



*Sumber : Implementasi ontologi..., Nuriana Ayuningtyas, FT UI, 2009*

**Gambar 2.16 Cara Kerja Web**

**BAB III**

**METODOLOGI PENELITIAN**

* 1. **Kerangka Penelitian**

Kerangka penilitian akan mempermudah penulis untuk melakukan penelitian maupun dalam menyusun laporan penelitian nantinya. Berikut adalah bentuk gambaran kerangka penelitian yang akan dilakukan penulis :

**Penelitian Pendahuluan**

**Analisa dan Perancangan Sistem Sistem**

**Pengimplemantasian Sistem**

**Pengujian Sistem**

**Pengumpulan Data**

**Gambar 3.1 Kerangka Penelitian**

* 1. **Tahapan Penelitian**

Penelitian terhadap Jaringan Syaraf Tiruan untuk memprediksi harga *bitcoin* menggunakan metode *Backpropagation* dilakukan beberapa tahap penelitian. Tahapan tersebut akan di jabarkan pada bagian sub-bab ini.

* + 1. **Penelitian Pendahuluan**

Penelitian pendahuluan atau sering disebut dengan *Study* penelitian merupakan tahapan awal dalam metodelogi penelitian. Pada tahap ini dilakukan penelitian dengan cara menganalisa terlebih dahulu masalah-masalah yang akan dikembangkan. Dengan adanya penelitian ini diharapkan dapat membantu melakukan prediksi dalam pergerakan harga *bitcoin*.

* + 1. **Pengumpulan Data**

Untuk mendapatkan data yang optimal diperlukan diperlukan waktu, tempat dan metode penelitian dalam sebuah penelitian, yaitu dapat dijelaskan sebagai berikut :

* + - 1. ***Waktu Penelitian***

Waktu yang digunakan untuk penelitian kurang lebih selama 5 bulan mulai bulan september 2017 sampai january 2018, untuk lebih terperininya dapat dilihat pada tabel berikut :

Tabel 3.1 Waktu Penelitian

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **KEGIANTAN** | **September** | **Oktober** | **November** | **Januari** | **Februari** | **Maret** | **April** | **Mei** |
| 1. | Perancangan Penelitian |  |  |  |  |  |  |  |  |
| 2. | Studi Pustaka Dan Pengumpulan Data |  |  |  |  |  |  |  |  |
| 3. | Analisa Data dan Sistem |  |  |  |  |  |  |  |  |
| 4. | Perancangan Aplikasi |  |  |  |  |  |  |  |  |
| 5. | Pembuatan Aplikasi |  |  |  |  |  |  |  |  |
| 6. | Uji Coba |  |  |  |  |  |  |  |  |
| 7. | Implementasi |  |  |  |  |  |  |  |  |
| 8. | Bimbingan Skripsi |  |  |  |  |  |  |  |  |
| 9. | Penyusunan Skripsi |  |  |  |  |  |  |  |  |

* + - 1. ***Tempat Penelitian***

Profil perusahaan :

Nama : PT. Bit Coin Indonesia

Website : [www.indodax.co.id](http://www.indodax.co.id)

Alamat : Jl. Nakula No. 88 B, Seminyak, Badung, Bali 80361

Telepon : 0361 – 847 5706

* + - 1. ***Metode Penelitian***

Untuk mendapatkan hasil yang efektif maka diperlukan data yang valid dan informasi yang sesuai dengan masalah-masalah yang dihadapi.

Adapun metode pengumpulan data antara lain :

1. Penelitian Lapangan (*Field Research*)

Penelitian ini dilakukan dengan melihat harga dan pergerakan dari pasar saham *bitcoin* indonesia secara online melalui website www.indodax.com serta mewawancarai beberapa pemain saham *bitcoin* yang berada di padang.

1. Penelitian kepustakaan (*Library Research*)

Penelitian ini dilakukan untuk memperoleh data skunder dengan membaca buku-buku, literature-literatur dan *browsing* Internet agar dapat memperoleh data dan informasi yang dapat membantu dalam perancangan sistem yang akan dibuat dalam laporan ini.

1. Penelitian Laboratorium (*Laboratory Research*)

Merupakan penelitian yang dilakukan secara komputerisasi untuk menguji kebenaran yang telah diselidiiki:

1. *Hardware* yang digunakan :
2. *Laptop Asus A450L dengan Processor Intel Core i5 2.3GHz*
3. *Printer Cannon*
4. *RAM 8GB*
5. *Media penyimpanan eksternal (Flashdisk).*
6. *Software* pendukung yang digunakan :
7. *Sistem Operasi Windows 8.1 pro*
8. *Microsft Office 2013 Profesional Plus.*
9. *Google Chrome 61.0.3163.100(64-bit)*
10. *Relational Rose* *2003*
    * 1. **Analisa dan Perancangan**
         1. ***Penganalisaan Data***

Setelah data berhasil dikumpulkan selanjutnya melakukan analisa terhadap data tersebut. Analisa data adalah usaha untuk mengolah data sehingga didapat suatu kesimpulan yang diharapkan sesuai dengan tujuan penelitian.

Pada tahap ini akan dilakukan evaluasi kinerja, identifikasi terhadap masalah yang ada, rancangan sistem dan langkah-langkah yang dibutuhkan untuk perancangan sistem yang diinginkan sampai pada analisa yang diharapkan.

* + - 1. ***Penganalisaan Sistem***

Analisa sistem merupakan suatu penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Analisa ini dilakukan untuk mengetahui apa saja kebutuhan yang dibutuhkan sehingga nantinya sistem akan menjadi lebih efektif dan efisien dalam pengimplementasiannya.

* + - 1. ***Perancangan Sistem***

Pada tahap ini dilakukan pengumpulan fakta-fakta yang mendukung perancangan sistem. Pada proses perancangan Aplikasi *Jaringan Saraf Tiruan* ini, penulis menggunakan UML sebagai alat bantu untuk menjelaskan dan menggambarkan alur kerja sistem. Diantaranya diagram yang digunakan yaitu :

1. *Use Case Diagram*

Digunakan untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. *Use Case* mendeskripsikan interaksi tripikal antara para pengguna sistem dengan sistem itu sendiri, dengan sebuah narasi tentang bagaimana sistem tersebut digunakan. Yang ditentukan adalah “apa” yang dibuat sistem, dan bukan “bagaimana”.

1. *Class Diagram*

Digunakan untuk menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode atau fungsi). *Class Diagram* mendeskripsikan jenis-jenis *objek* dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class Diagram* juga menunjukan properti dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan *objek-objek* tersebut.

1. *Statechart Diagram*

Digunakan untuk menggambarkan transisi dari perubahan keadaan (dari suatu *state* ke *state* lainnya) suatu *objek* pada pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *Statechart Diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram* ).

1. *Activity Diagram*

Digunakan untuk menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang., bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berekhir. *Activity Diagram* adalah teknik untuk menggambarkan logika *procedural,* dan jalur kerja.

1. *Sequence Diagram*

Digunakan untuk menggambarkan interaksi antar *objek* di dalam dan disekitar sistem (termasuk pengguna, *display,* dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Sebuah *Sequence Diagram,* secara khusus menjabarkan *behavior* sebuah scenario tunggal. *Sequence Diagram* terdiri atas dimensi *vertical* (waktu) dan dimensi *horizontal* (*objek-objek* yang terkait).

1. *Collaboration Diagram*

*Collaboration Diagram* juga menggambarkan interaksi antar objek seperti *Sequence Diagram.* Tetapi lebih menekankan pada peran masing-masing *objek* yang bukan pada waktu penyampaian *message.*

* + 1. **Pengimplementasian Sistem**

Merupakan tahap penelitian yang dilakukan untuk mempraktekan langsung hasil dari analisa yang bertujan untuk meguji kebenaran proses yang dilakukan secara manual dan dengan program. Penulis mengunakan Program MATLAB sebagai bantuan untuk menentukan nilai *bobot* dan *bias*, dan juga sebagai sarana pembanding antara hasil manual dengan sistem.

* + 1. **Pengujian Sistem**

Pada tahap ini adalah tahap untuk mempraktekan langsung hasil dari analisa yang bertujuan untuk menguji kebenaran sistem yang dirancang apakah sudah berjalan dengan benar dan sesuai dengan perumusan masalah yang sudah ditentukan sebelumnya.

Tahap ini juga menjelaskan bagaimana cara menggunakan aplikasi *Jaringan Syaraf Tiruan* dengan metode *Backpropogation* dalam memprediksi harga *bitcoin*. Untuk pengujian sistemnya menggunakan metode *BlackBox,*  seperti membuka halaman utama, register, login, memasukan data yang akan diolah lalu melihat hasil prediksi. Secara garis besar nantinya sistem akan meminta login kepada pengguna sebelum mengakses proses prediksi jika pengguna tidak mempunyai username dan password maka pengguna harus registrasi lebih dulu. setelah login sukses jika pengguna mengakses bagian prediksi maka sistem akan meminta data yang diperlukan untuk proses prediksi. Setelah data lengkap dan pengguna melakukan proses maka hasil prediksi akan otomatis dikeluarkan oleh sistem.

**BAB IV**

**ANALISA DAN PERANCANGAN**

* 1. **Pendahuluan**

Metode yang baik dalam peramalan secara matematis ada metode yang memili tingkat ketelitian yang tingi, semakin tinggi tingkat ketelitian 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 prediksi. Untuk serangkaian data yang akan ditentukan nilainya, berbeda tingkat penyimpangannya yang secara terakumulasi membentuk hasil prediksi tersebut. Untuk mendapatkan analisa, maka hasil-hasilnya ditampilkan dalam bentuk grafik.

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

Analisa sistem merupakan suatu uraian dari sistem informasi yang utuh menjadi suatu bagian yang terbentuk dari komponen yang berguna untuk mengidentifikasi dan mengevaluasi peramalan prediksi, hambat yang terjadi sehingga dapat diusulkan suatu perbaikan dalam mengembangkan sistem dengan menggunakan perangkat lunak (*software*) Jaringan Syaraf Tiruanmenggunakan metode *Backpropagation* dalam memprediksi pergerakan harga *bitcoin*.

Data-data atribut yang digunakan adalah sebagai berikut :

1. Pergerakan Grafik *Open*.
2. Pergerakan Grafik *High*.
3. Pergerakan Grafik *Low*.
4. Jumlah *Bitcoin*.
5. Permintaan *Bitcoin*.

Nantinya pergerakan grafik open disimbolkan pada x1, pergerakan grafik high disimbolkan dengan x2, pergerakan grafik low dengan x3, jumlah *bitcoin* disimbolkan dengan x4 dan permintaan *bitcoin* disimbolkan dengan x5.

Data-data yang diperlukan dalam perancangan aplikasi Jaringan Syaraf Tiruanadalah hasil penelitian murni dengan mengambil variabel dari nilai-nilai yang bisa dijadikan prediktor untuk memprediksi permintaan produk antara lain meliputi Pergerakan grafik *open*, pergerakan grafik *high*, pergerakan grafik *low*, jumlah *bitcoin* dan permintaan *bitcoin*. Selanjutnya data dibagi dua yaitu, data pada tanggal 26 april 2018 yang digunakan sebagai pelatihan dan data pada tanggal 27 mei yang digunakan sebagai pengujian.

**Tabel 4.1 Data *Bitcoin* 26 april 2018**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| waktu | open | high | low | close | volume | permintaan |
| 0:00 | 125210000 | 126459000 | 124850000 | 126456000 | 5,06 | Naik |
| 1:00 | 126965000 | 126965000 | 125303000 | 125975000 | 6,5 | Turun |
| 2:00 | 124977000 | 126291000 | 124977000 | 125878000 | 2,74 | Turun |
| 3:00 | 125761000 | 127500000 | 125761000 | 126989000 | 2,76 | Naik |
| 4:00 | 127745000 | 128498000 | 127744000 | 128074000 | 0,87 | Naik |
| 5:00 | 128990000 | 128999000 | 127861000 | 128236000 | 4,6 | Naik |
| 6:00 | 126000000 | 126118000 | 124872000 | 124872000 | 9,12 | Turun |
| 7:00 | 124792000 | 124792000 | 122503000 | 122503000 | 22,34 | Turun |
| 8:00 | 123499000 | 124488000 | 122508000 | 123986000 | 13,77 | Naik |
| 9:00 | 122747000 | 124152000 | 122503000 | 123989000 | 6,87 | Naik |
| 10:00 | 124250000 | 124273000 | 124000000 | 124247000 | 5,32 | Naik |
| 11:00 | 123791000 | 124200000 | 123000000 | 124000000 | 5,19 | Turun |

**Tabel 4.2 Data *Bitcoin* 27 mei 2018**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| waktu | open | high | low | close | volume | permintaan |
| 0:00 | 108360000 | 108495000 | 108359000 | 108494000 | 0,75 | Naik |
| 1:00 | 108506000 | 108608000 | 108506000 | 108608000 | 0,72 | Naik |
| 2:00 | 108578000 | 108643000 | 108578000 | 108643000 | 0,38 | Naik |
| 3:00 | 108506000 | 108638000 | 108506000 | 108601000 | 0,43 | Turun |
| 4:00 | 108566000 | 108888000 | 108262000 | 108680000 | 3,11 | Naik |
| 5:00 | 107805000 | 107807000 | 106999000 | 107432000 | 11,24 | Turun |
| 6:00 | 106095000 | 106499000 | 106095000 | 106478000 | 3,45 | Turun |
| 7:00 | 106284000 | 106314000 | 105025000 | 105050000 | 14,18 | Turun |
| 8:00 | 105101000 | 105803000 | 105101000 | 105600000 | 5,17 | Naik |
| 9:00 | 105897000 | 106286000 | 105820000 | 106284000 | 1,98 | Naik |
| 10:00 | 105980000 | 106197000 | 105599000 | 105991000 | 3,65 | Turun |
| 11:00 | 105898000 | 106086000 | 105717000 | 105988000 | 1,98 | Turun |

* + 1. **Proses Analisa Sistem**

Proses analisa sistem dilakukan berdasarkan data yang diperoleh dari pergerakan harga pada pasar indonesia. Pada tahap ini data yang telah ada akan dibagi menjadi data pelatihan dan data pengujian.

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

1. **Data Pelatihan**

Berikut merupakan data yang akan dijadikan sebagai pelatihan dalam mengimplementasikan Jaringan Syaraf Tiruan untuk memprediksi Pergerakan harga *bitcoin*. Data yang digunakan adalah data pada tanggal 26 april 2018 pada pukul 00:00– 11.00:

**Tabel 4.3 Data Input Pelatihan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x1 | x2 | x3 | x4 | x5 | T |
| 125210000 | 126459000 | 124850000 | 5,06 | 1 | 126456000 |
| 126965000 | 126965000 | 125303000 | 6,5 | -1 | 125975000 |
| 124977000 | 126291000 | 124977000 | 2,74 | -1 | 125878000 |
| 125761000 | 127500000 | 125761000 | 2,76 | 1 | 126989000 |
| 127745000 | 128498000 | 127744000 | 0,87 | 1 | 128074000 |
| 128990000 | 128999000 | 127861000 | 4,6 | 1 | 128236000 |
| 126000000 | 126118000 | 124872000 | 9,12 | -1 | 124872000 |
| 124792000 | 124792000 | 122503000 | 22,34 | -1 | 122503000 |
| 123499000 | 124488000 | 122508000 | 13,77 | 1 | 123986000 |
| 122747000 | 124152000 | 122503000 | 6,87 | 1 | 123989000 |
| 124250000 | 124273000 | 124000000 | 5,32 | 1 | 124247000 |
| 123791000 | 124200000 | 123000000 | 5,19 | -1 | 124000000 |

**b. Data Pengujian**

Berikut merupakan data yang akan dijadikan sebagai pelatihan dalam mengimplementasikan Jaringan Syaraf Tiruan untuk memprediksi pergerakan harga *bitcoin*.Data yang digunakan adalah data pada tanggal 26 mei 2018 pada pukul 00:00 – 11.00:

**Tabel 4.4 Data Input Pengujian**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x1 | x2 | x3 | x4 | x5 | T |
| 108360000 | 108495000 | 108359000 | 0,75 | 1 | 108494000 |
| 108506000 | 108608000 | 108506000 | 0,72 | 1 | 108608000 |
| 108578000 | 108643000 | 108578000 | 0,38 | 1 | 108643000 |
| 108506000 | 108638000 | 108506000 | 0,43 | -1 | 108601000 |
| 108566000 | 108888000 | 108262000 | 3,11 | 1 | 108680000 |
| 107805000 | 107807000 | 106999000 | 11,24 | -1 | 107432000 |
| 106095000 | 106499000 | 106095000 | 3,45 | -1 | 106478000 |
| 106284000 | 106314000 | 105025000 | 14,18 | -1 | 105050000 |
| 105101000 | 105803000 | 105101000 | 5,17 | 1 | 105600000 |
| 105897000 | 106286000 | 105820000 | 1,98 | 1 | 106284000 |
| 105980000 | 106197000 | 105599000 | 3,65 | -1 | 105991000 |
| 105898000 | 106086000 | 105717000 | 1,98 | -1 | 105988000 |

Keterangan :

1. Pergerakan Grafik *Open*, disimpan pada variabel X1
2. Pergerakan Grafik *High*, disimpan pada variabel X2
3. Pergerakan Grafik *Low*, disimpan pada variabel X3
4. Jumlah *Bitcoin*, disimpan pada variabel X4
5. Permintaan *Bitcoin*, disimpan pada variabel X5

*Representasi variable permintaan :*

1. jika permintaan *bitcoin* turun, maka dikategorikan rendah, dengan singkatan = -1
2. jika permintaan *bitcoin* tetap, maka dikategorikan sedang, dengan singkatan = 0
3. jika permintaan *bitcoin* naik, maka dikategorikan tinggi, dengan singkatan = 1
   * + 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), (0.8)] 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 0.8pencarian dibawah ini menunjukan hasil *transformasi* untuk data pelatihan yang nantinya dipakai sebagai data pelatihan *backpropagation*.

**Tabel 4.5 Proses Transformasi Data Pelatihan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x1 | x2 | x3 | x4 | x5 | T |
| 125210000 | 126459000 | 124850000 | 5,06 | 1 | 126456000 |
| 126965000 | 126965000 | 125303000 | 6,5 | -1 | 125975000 |
| 124977000 | 126291000 | 124977000 | 2,74 | -1 | 125878000 |
| 125761000 | 127500000 | 125761000 | 2,76 | 1 | 126989000 |
| 127745000 | 128498000 | 127744000 | 0,87 | 1 | 128074000 |
| 128990000 | 128999000 | 127861000 | 4,6 | 1 | 128236000 |
| 126000000 | 126118000 | 124872000 | 9,12 | -1 | 124872000 |
| 124792000 | 124792000 | 122503000 | 22,34 | -1 | 122503000 |
| 123499000 | 124488000 | 122508000 | 13,77 | 1 | 123986000 |
| 122747000 | 124152000 | 122503000 | 6,87 | 1 | 123989000 |
| 124250000 | 124273000 | 124000000 | 5,32 | 1 | 124247000 |
| 123791000 | 124200000 | 123000000 | 5,19 | -1 | 124000000 |

a : data minimum = 122503000

b : data maksimum = 128999000

X1 = 0,8 \* (125210000–122503000)/( 128999000 –122503000) + 0,1 = 0,4334

0,8 \* (126965000–122503000)/( 128999000 –122503000) + 0,1 = 0,6495

0,8 \* (124977000–122503000)/( 128999000 –122503000) + 0,1 = 0,4047

0,8 \* (125761000–122503000)/( 128999000 –122503000) + 0,1 = 0,5012

0,8 \* (127745000–122503000)/( 128999000 –122503000) + 0,1 = 0,7456

0,8 \* (128990000–122503000)/( 128999000 –122503000) + 0,1 = 0,8989

0,8 \* (126000000–122503000)/( 128999000 –122503000) + 0,1 = 0,5307

0,8 \* (124792000–122503000)/( 128999000 –122503000) + 0,1 = 0,3819

0,8 \* (123499000–122503000)/( 128999000 –122503000) + 0,1 = 0,2227

0,8 \* (122747000–122503000)/( 128999000 –122503000) + 0,1 = 0,1300

0,8 \* (124250000–122503000)/( 128999000 –122503000) + 0,1 = 0,3151

0,8 \* (123791000–122503000)/( 128999000 –122503000) + 0,1 = 0,2586

X2 = 0,8 \* (126459000–122503000)/( 128999000 –122503000) + 0,1 = 0,5872

0,8 \* (126965000–122503000)/( 128999000 –122503000) + 0,1 = 0,6495

0,8 \* (126291000–122503000)/( 128999000 –122503000) + 0,1 = 0,5665

0,8 \* (127500000–122503000)/( 128999000 –122503000) + 0,1 = 0,7154

0,8 \* (128498000–122503000)/( 128999000 –122503000) + 0,1 = 0,8383

0,8 \* (128999000–122503000)/( 128999000 –122503000) + 0,1 = 0,9000

0,8 \* (126118000–122503000)/( 128999000 –122503000) + 0,1 = 0,5452

0,8 \* (124792000–122503000)/( 128999000 –122503000) + 0,1 = 0,3819

0,8 \* (124488000–122503000)/( 128999000 –122503000) + 0,1 = 0,3445

0,8 \* (124152000–122503000)/( 128999000 –122503000) + 0,1 = 0,3031

0,8 \* (124273000–122503000)/( 128999000 –122503000) + 0,1 = 0,3180

0,8 \* (124200000–122503000)/( 128999000 –122503000) + 0,1 = 0,3090

X3 = 0,8 \* (124850000–122503000)/( 128999000 –122503000) + 0,1 = 0,3890

0,8 \* (125303000–122503000)/( 128999000 –122503000) + 0,1 = 0,4448

0,8 \* (124977000–122503000)/( 128999000 –122503000) + 0,1 = 0,4047

0,8 \* (125761000–122503000)/( 128999000 –122503000) + 0,1 = 0,5012

0,8 \* (127744000–122503000)/( 128999000 –122503000) + 0,1 = 0,7454

0,8 \* (127861000–122503000)/( 128999000 –122503000) + 0,1 = 0,7599

0,8 \* (124872000–122503000)/( 128999000 –122503000) + 0,1 = 0,3917

0,8 \* (122503000–122503000)/( 128999000 –122503000) + 0,1 = 0,1000

0,8 \* (122508000–122503000)/( 128999000 –122503000) + 0,1 = 0,1006

0,8 \* (122503000–122503000)/( 128999000 –122503000) + 0,1 = 0,1000

0,8 \* (124000000–122503000)/( 128999000 –122503000) + 0,1 = 0,2844

0,8 \* (123000000–122503000)/( 128999000 –122503000) + 0,1 = 0,1612

a : data minimum = 0,87

b : data maksimum = 22,34

X4 0,8 \* (5,06–0,87)/( 22,34 –0,87) + 0,1 = 0,2561

0,8 \* (6,5–0,87)/( 22,34 –0,87) + 0,1 = 0,3098

0,8 \* (2,74–0,87)/( 22,34 –0,87) + 0,1 = 0,1697

0,8 \* (2,76–0,87)/( 22,34 –0,87) + 0,1 = 0,1704

0,8 \* (0,87–0,87)/( 22,34 –0,87) + 0,1 = 0,1000

0,8 \* (4,6–0,87)/( 22,34 –0,87) + 0,1 = 0,2390

0,8 \* (9,12–0,87)/( 22,34 –0,87) + 0,1 = 0,4074

0,8 \* (22,34–0,87)/( 22,34 –0,87) + 0,1 = 0,9000

0,8 \* (13,77–0,87)/( 22,34 –0,87) + 0,1 = 0,5807

0,8 \* (6,87–0,87)/( 22,34 –0,87) + 0,1 = 0,3236

0,8 \* (5,32–0,87)/( 22,34 –0,87) + 0,1 = 0,2658

0,8 \* (5,19–0,87)/( 22,34 –0,87) + 0,1 = 0,2610

**Tabel 4.6 Keterangan Nilai Permintaan**

|  |  |  |
| --- | --- | --- |
| **No** | **Nilai** | **Keterangan** |
| 1. | 1 | Naik |
| 2. | -1 | Turun |
| 3. | -1 | Turun |
| 4. | 1 | Naik |
| 5. | 1 | Naik |
| 6. | 1 | Naik |
| 7. | -1 | Turun |
| 8 | -1 | Turun |
| 9 | 1 | Naik |
| 10 | 1 | Naik |
| 11 | 1 | Naik |
| 12 | -1 | Turun |

X5 0,8 \* (1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,9000

0,8 \* (-1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,1000

0,8 \* (-1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,1000

0,8 \* (1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,9000

0,8 \* (1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,9000

0,8 \* (1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,9000

0,8 \* (-1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,1000

0,8 \* (-1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,1000

0,8 \* (1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,9000

0,8 \* (1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,9000

0,8 \* (1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,9000

0,8 \* (-1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,1000

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

T = 0,8 \* (126456000–122503000)/( 128999000 –122503000) + 0,1 = 0,5868

0,8 \* (125975000–122503000)/( 128999000 –122503000) + 0,1 = 0,5276

0,8 \* (125878000–122503000)/( 128999000 –122503000) + 0,1 = 0,5156

0,8 \* (126989000–122503000)/( 128999000 –122503000) + 0,1 = 0,6525

0,8 \* (128074000–122503000)/( 128999000 –122503000) + 0,1 = 0,7861

0,8 \* (128236000–122503000)/( 128999000 –122503000) + 0,1 = 0,8060

0,8 \* (124872000–122503000)/( 128999000 –122503000) + 0,1 = 0,3917

0,8 \* (122503000–122503000)/( 128999000 –122503000) + 0,1 = 0,1000

0,8 \* (123986000–122503000)/( 128999000 –122503000) + 0,1 = 0,2826

0,8 \* (123989000–122503000)/( 128999000 –122503000) + 0,1 = 0,2830

0,8 \* (124247000–122503000)/( 128999000 –122503000) + 0,1 = 0,3148

0,8 \* (124000000–122503000)/( 128999000 –122503000) + 0,1 = 0,2844

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

**Tabel 4.7 Hasil Transformasi Data Input Pelatihan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X1 | X2 | X3 | X4 | X5 | T |
| 0,4334 | 0,5872 | 0,3890 | 0,2561 | 0,9000 | 0,5868 |
| 0,6495 | 0,6495 | 0,4448 | 0,3098 | 0,1000 | 0,5276 |
| 0,4047 | 0,5665 | 0,4047 | 0,1697 | 0,1000 | 0,5156 |
| 0,5012 | 0,7154 | 0,5012 | 0,1704 | 0,9000 | 0,6525 |
| 0,7456 | 0,8383 | 0,7454 | 0,1000 | 0,9000 | 0,7861 |
| 0,8989 | 0,9000 | 0,7599 | 0,2390 | 0,9000 | 0,8060 |
| 0,5307 | 0,5452 | 0,3917 | 0,4074 | 0,1000 | 0,3917 |
| 0,3819 | 0,3819 | 0,1000 | 0,9000 | 0,1000 | 0,1000 |
| 0,2227 | 0,3445 | 0,1006 | 0,5807 | 0,9000 | 0,2826 |
| 0,1300 | 0,3031 | 0,1000 | 0,3236 | 0,9000 | 0,2830 |
| 0,3151 | 0,3180 | 0,2844 | 0,2658 | 0,9000 | 0,3148 |
| 0,2586 | 0,3090 | 0,1612 | 0,2610 | 0,1000 | 0,2844 |

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**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x1 | x2 | x3 | x4 | x5 | T |
| 108360000 | 108495000 | 108359000 | 0,75 | 1 | 108494000 |
| 108506000 | 108608000 | 108506000 | 0,72 | 1 | 108608000 |
| 108578000 | 108643000 | 108578000 | 0,38 | 1 | 108643000 |
| 108506000 | 108638000 | 108506000 | 0,43 | -1 | 108601000 |
| 108566000 | 108888000 | 108262000 | 3,11 | 1 | 108680000 |
| 107805000 | 107807000 | 106999000 | 11,24 | -1 | 107432000 |
| 106095000 | 106499000 | 106095000 | 3,45 | -1 | 106478000 |
| 106284000 | 106314000 | 105025000 | 14,18 | -1 | 105050000 |
| 105101000 | 105803000 | 105101000 | 5,17 | 1 | 105600000 |
| 105897000 | 106286000 | 105820000 | 1,98 | 1 | 106284000 |
| 105980000 | 106197000 | 105599000 | 3,65 | -1 | 105991000 |
| 105898000 | 106086000 | 105717000 | 1,98 | -1 | 105988000 |

a : data minimum = 105025000

b : data maksimum = 108888000

X1 = 0,8 \* (108360000–105025000)/( 108888000 –105025000) + 0,1 = 0,7907

0,8 \* (108506000–105025000)/( 108888000 –105025000) + 0,1 = 0,8209

0,8 \* (108578000–105025000)/( 108888000 –105025000) + 0,1 = 0,8358

0,8 \* (108506000–105025000)/( 108888000 –105025000) + 0,1 = 0,8209

0,8 \* (108566000–105025000)/( 108888000 –105025000) + 0,1 = 0,8333

0,8 \* (107805000–105025000)/( 108888000 –105025000) + 0,1 = 0,6757

0,8 \* (106095000–105025000)/( 108888000 –105025000) + 0,1 = 0,3216

0,8 \* (106284000–105025000)/( 108888000 –105025000) + 0,1 = 0,3607

0,8 \* (105101000–105025000)/( 108888000 –105025000) + 0,1 = 0,1157

0,8 \* (105897000–105025000)/( 108888000 –105025000) + 0,1 = 0,2806

0,8 \* (105980000–105025000)/( 108888000 –105025000) + 0,1 = 0,2978

0,8 \* (105898000–105025000)/( 108888000 –105025000) + 0,1 = 0,2808

X2 = 0,8 \* (108495000–105025000)/( 108888000 –105025000) + 0,1 = 0,8186

0,8 \* (108608000–105025000)/( 108888000 –105025000) + 0,1 = 0,8420

0,8 \* (108643000–105025000)/( 108888000 –105025000) + 0,1 = 0,8493

0,8 \* (108638000–105025000)/( 108888000 –105025000) + 0,1 = 0,8482

0,8 \* (108888000–105025000)/( 108888000 –105025000) + 0,1 = 0,9000

0,8 \* (107807000–105025000)/( 108888000 –105025000) + 0,1 = 0,6761

0,8 \* (106499000–105025000)/( 108888000 –105025000) + 0,1 = 0,4053

0,8 \* (106314000–105025000)/( 108888000 –105025000) + 0,1 = 0,3669

0,8 \* (105803000–105025000)/( 108888000 –105025000) + 0,1 = 0,2611

0,8 \* (106286000–105025000)/( 108888000 –105025000) + 0,1 = 0,3611

0,8 \* (106197000–105025000)/( 108888000 –105025000) + 0,1 = 0,3427

0,8 \* (106086000–105025000)/( 108888000 –105025000) + 0,1 = 0,3197

X3 = 0,8 \* (108359000–105025000)/( 108888000 –105025000) + 0,1 = 0,7904

0,8 \* (108506000–105025000)/( 108888000 –105025000) + 0,1 = 0,8209

0,8 \* (108578000–105025000)/( 108888000 –105025000) + 0,1 = 0,8358

0,8 \* (108506000–105025000)/( 108888000 –105025000) + 0,1 = 0,8209

0,8 \* (108262000–105025000)/( 108888000 –105025000) + 0,1 = 0,7704

0,8 \* (106999000–105025000)/( 108888000 –105025000) + 0,1 = 0,5088

0,8 \* (106095000–105025000)/( 108888000 –105025000) + 0,1 = 0,3216

0,8 \* (105025000–105025000)/( 108888000 –105025000) + 0,1 = 0,1000

0,8 \* (105101000–105025000)/( 108888000 –105025000) + 0,1 = 0,1157

0,8 \* (105025000–105025000)/( 108888000 –105025000) + 0,1 =0,2646

0,8 \* (105599000–105025000)/( 108888000 –105025000) + 0,1 = 0,2189

0,8 \* (105717000–105025000)/( 108888000 –105025000) + 0,1 = 0,2433

a : data minimum = 0,38

b : data maksimum = 14,18

X4 0,8 \* (0,75–0,38)/( 14,18 –0,38) + 0,1 = 0,1214

0,8 \* (0,72–0,38)/( 14,18 –0,38) + 0,1 = 0,1197

0,8 \* (0,38–0,38)/( 14,18 –0,38) + 0,1 = 0,1000

0,8 \* (0,43–0,38)/( 14,18 –0,38) + 0,1 = 0,1029

0,8 \* (3,11–0,38)/( 14,18 –0,38) + 0,1 = 0,2583

0,8 \* (11,24–0,38)/( 14,18 –0,38) + 0,1 = 0,7296

0,8 \* (3,45–0,38)/( 14,18 –0,38) + 0,1 = 0,2780

0,8 \* (14,18–0,38)/( 14,18 –0,38) + 0,1 = 0,9000

0,8 \* (5,17–0,38)/( 14,18 –0,38) + 0,1 = 0,3777

0,8 \* (1,98–0,38)/( 14,18 –0,38) + 0,1 = 0,1928

0,8 \* (3,65–0,38)/( 14,18 –0,38) + 0,1 = 0,2896

0,8 \* (1,98–0,38)/( 14,18 –0,38) + 0,1 = 0,1928

**Tabel 4.9 Keterangan Nilai Permintaan**

|  |  |  |
| --- | --- | --- |
| **No** | **Nilai** | **Keterangan** |
| 1. | 1 | Naik |
| 2. | 1 | Naik |
| 3. | 1 | Naik |
| 4. | -1 | Turun |
| 5. | 1 | Naik |
| 6. | -1 | Turun |
| 7. | -1 | Turun |
| 8 | -1 | Turun |
| 9 | 1 | Naik |
| 10 | 1 | Naik |
| 11 | -1 | Turun |
| 12 | -1 | Turun |

X5 0,8 \* (1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,9000

0,8 \* (1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,9000

0,8 \* (1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,9000

0,8 \* (-1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,1000

0,8 \* (1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,9000

0,8 \* (-1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,1000

0,8 \* (-1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,1000

0,8 \* (-1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,1000

0,8 \* (1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,9000

0,8 \* (1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,9000

0,8 \* (-1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,1000

0,8 \* (-1 + 1 ) / ( 1 + 1 ) + 0,1 = 0,1000

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

T = 0,8 \* (108494000 –105025000)/( 108888000 –105025000) + 0,1 = 0,8184

0,8 \* (108608000 –105025000)/( 108888000 –105025000) + 0,1 = 0,8420

0,8 \* (108643000 –105025000)/( 108888000 –105025000) + 0,1 = 0,8493

0,8 \* (108601000 –105025000)/( 108888000 –105025000) + 0,1 = 0,8406

0,8 \* (108680000 –105025000)/( 108888000 –105025000) + 0,1 = 0,8569

0,8 \* (107432000 –105025000)/( 108888000 –105025000) + 0,1 = 0,5985

0,8 \* (106478000 –105025000)/( 108888000 –105025000) + 0,1 = 0,4009

0,8 \* (105050000 –105025000)/( 108888000 –105025000) + 0,1 = 0,1052

0,8 \* (105600000 –105025000)/( 108888000 –105025000) + 0,1 = 0,2191

0,8 \* (106284000 –105025000)/( 108888000 –105025000) + 0,1 = 0,3607

0,8 \* (105991000 –105025000)/( 108888000 –105025000) + 0,1 = 0,3001

0,8 \* (105988000 –105025000)/( 108888000 –105025000) + 0,1 = 0,2994

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 4.10 Hasil Transformasi Data Pengujian**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X1 | X2 | X3 | X4 | X5 | T |
| 0,7907 | 0,8186 | 0,7904 | 0,1214 | 0,9000 | 0,8184 |
| 0,8209 | 0,8420 | 0,8209 | 0,1197 | 0,9000 | 0,8420 |
| 0,8358 | 0,8493 | 0,8358 | 0,1000 | 0,9000 | 0,8493 |
| 0,8209 | 0,8482 | 0,8209 | 0,1029 | 0,1000 | 0,8406 |
| 0,8333 | 0,9000 | 0,7704 | 0,2583 | 0,9000 | 0,8569 |
| 0,6757 | 0,6761 | 0,5088 | 0,7296 | 0,1000 | 0,5985 |
| 0,3216 | 0,4053 | 0,3216 | 0,2780 | 0,1000 | 0,4009 |
| 0,3607 | 0,3669 | 0,1000 | 0,9000 | 0,1000 | 0,1052 |
| 0,1157 | 0,2611 | 0,1157 | 0,3777 | 0,9000 | 0,2191 |
| 0,2806 | 0,3611 | 0,2646 | 0,1928 | 0,9000 | 0,3607 |
| 0,2978 | 0,3427 | 0,2189 | 0,2896 | 0,1000 | 0,3001 |
| 0,2808 | 0,3197 | 0,2433 | 0,1928 | 0,1000 | 0,2994 |

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

Pada penelitian ini arsitektur Jaringan Syaraf Tiruan yang digunakan untuk prediksi jumlah permintaan produk adalah Jaringan Syaraf Tiruan *Backpropagation* yang terdiri dari :

1. Lapisan *Input*

Dengan 5 simpul masing-masing untuk pergerakan grafik open (*x1*), pergerakan grafik high (*x2*), pergerakan grafik low (*x3*), volume *bitcoin* (*x4*), Permintaan *bitcoin* (*x5*).

1. Lapisan Output

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

1. Lapisan Tersembunyi

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

Z3

Z2

Y

X3

V14

V24

V15

V25

V12

V22

V13

V23

W11

W12

V11

V21

**Harga Bitoin**

***Input Layer Hidden Layer Output Layer***

**Gambar 4.1 Arsitektur Jaringan Syaraf Tiruan Untuk Harga *Bitcoin***

Keterangan :

*X* = Masukan (Input)

*J* = *1* s/d n( *n=5*)

*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 Mengetahui harga *Bitcoin***

Pola proses pelatihan ini, penulis akan mencoba melakukan pelatihan dengan menggunakan beberapa pola 5-2-1. Pola ini adalah pola dengan akurasi terbaik, dimana :

1. Jumlah *neuron* pada input layer sebanyak 5 *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
   * + 1. **Perancangan Algoritma *Backpropagation***

Pada tahap perancangan ini hasil analisa 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 mengatur 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 *actual output* 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 kembali lagi ke tahapan ke dua yaitu tahap *activation.*

Berikut ini diberikan contoh perhitungan pelatihan atau prediksi dengan Jaringan Syaraf Tiruan untuk menentuknan prediksi pergerakan harga *bitcoin*, sebagai contoh pelatihan menggunakan 4 buah variabel input *vector x,*yaitu :

X1 = Pergerakan Grafik *Open*

X2 = Pergerakan Grafik *High*

X3 = Pergerkan Grafik *Low*

X4 = Jumlah *Bitcoin*

X5 = Permintaan *Bitcoin*

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 4.11 Data Input**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X1 | X2 | X3 | X4 | X5 | T |
| 0,4334 | 0,5872 | 0,3890 | 0,2561 | 0,9000 | 0,5868 |
| 0,6495 | 0,6495 | 0,4448 | 0,3098 | 0,1000 | 0,5276 |
| 0,4047 | 0,5665 | 0,4047 | 0,1697 | 0,1000 | 0,5156 |
| 0,5012 | 0,7154 | 0,5012 | 0,1704 | 0,9000 | 0,6525 |
| 0,7456 | 0,8383 | 0,7454 | 0,1000 | 0,9000 | 0,7861 |
| 0,8989 | 0,9000 | 0,7599 | 0,2390 | 0,9000 | 0,8060 |
| 0,5307 | 0,5452 | 0,3917 | 0,4074 | 0,1000 | 0,3917 |
| 0,3819 | 0,3819 | 0,1000 | 0,9000 | 0,1000 | 0,1000 |
| 0,2227 | 0,3445 | 0,1006 | 0,5807 | 0,9000 | 0,2826 |
| 0,1300 | 0,3031 | 0,1000 | 0,3236 | 0,9000 | 0,2830 |
| 0,3151 | 0,3180 | 0,2844 | 0,2658 | 0,9000 | 0,3148 |
| 0,2586 | 0,3090 | 0,1612 | 0,2610 | 0,1000 | 0,2844 |

Arsitektur jairngan yang akan dipilih misalkan terdiri dari jumlah *input layer* terdiri dari 5 *neuron* yang variabelnya adalah inputan dari data-data yang akan di gunakan untuk memprediksi dan harga *bitcoin* sebagai target / *output* (T).

Untuk membentuk Jaringan Syaraf Tiruan, terlebih dahulu dilakukan inisialisasi bobot awal, bobot awal yang menghubungkan simpul-simpul pada lapisan *input* dan lapisan tersembunyi untuk arsitektur di atas adalah v = (v11,v21,v31,v12,v22,v32,v13,v23,v33,v14,v24,v34,v15,v25,v35) dan bobot bias di pilih secara acak, demikian pula bobot awal yang menghubungkan simpul-simpul pada lapisan tersembunyi dan lapisan *output* (w11, w22, w31) juga dipilih secara acak.

**Tabel 4.12 Bobot awal dan bias *input* ke *hidden* dari matlab**

|  |  |  |
| --- | --- | --- |
|  | **Z1** | **Z2** |
| **X1** | 3,6723 | 4,0125 |
| **X2** | -5,6066 | 5,2654 |
| **X3** | 1,7995 | -4,6369 |
| **X4** | -2,4841 | 0,4454 |
| **X5** | 5,1308 | 4,4182 |
| **Bias** | -3,8300 | -2,4535 |

**Tabel 4.13 Bobot awal dan *hidden* ke *output* dari matlab**

|  |  |
| --- | --- |
|  | **Y** |
| **Z1** | -0,6848 |
| **Z2** | 0,9412 |
| **BIAS** | 0,9143 |

Diketahui

X1 = 0,4334 X2 = 0,5872 X3 = 0,3890 X4 = 0,2561 X5 = 0,9000

V11 = 3,6723 V21 = 4,0125

V12 = -5,6066 V22 = 5,2654

V13 = 1,7995 V23 = -4,6369

V14 = -2,4841 V24 = 0,4454

V15 = 5,1308 V25 = 4,4182

W11 = -3,8300 W12 = -2,4535

-0,6848

0,9412

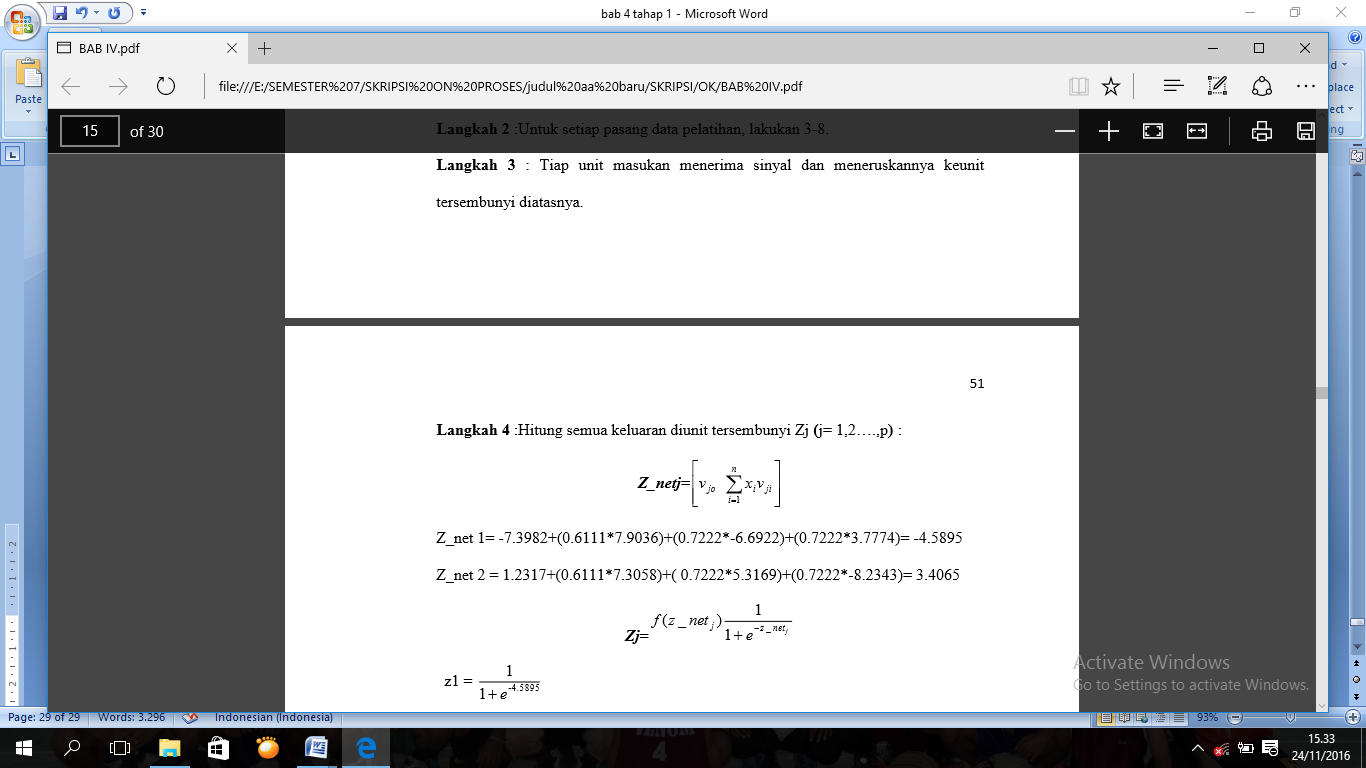
0,9143

Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil.

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

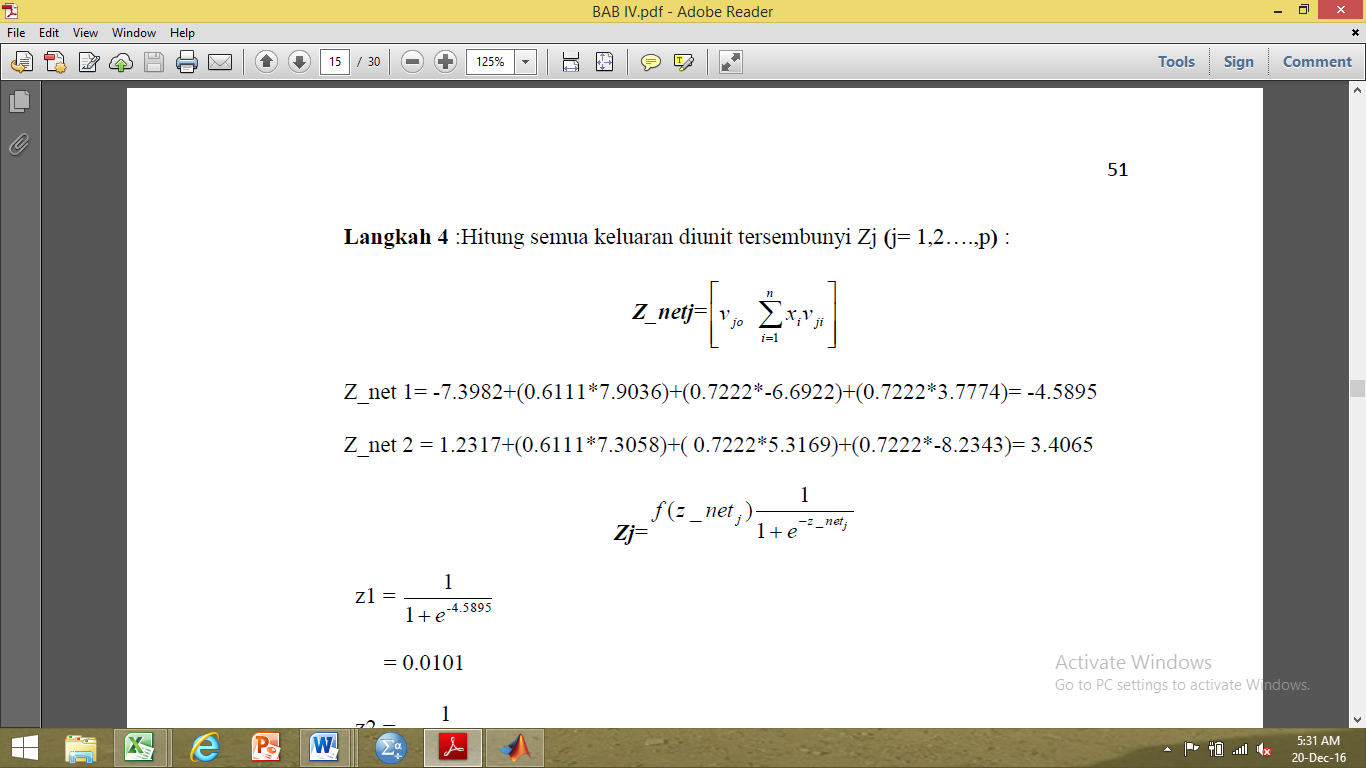
Langkah 2 : Untuk setiap pasang data pelatihan, lakukan 3-8.

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\_net1 = + (0,4334\*3,6723) + (0,5872\*-5,6066) + (0,3890\*1,7995)+ (0,2561\*-2,4841) + (0,9000\*5,1308) = -0,8491

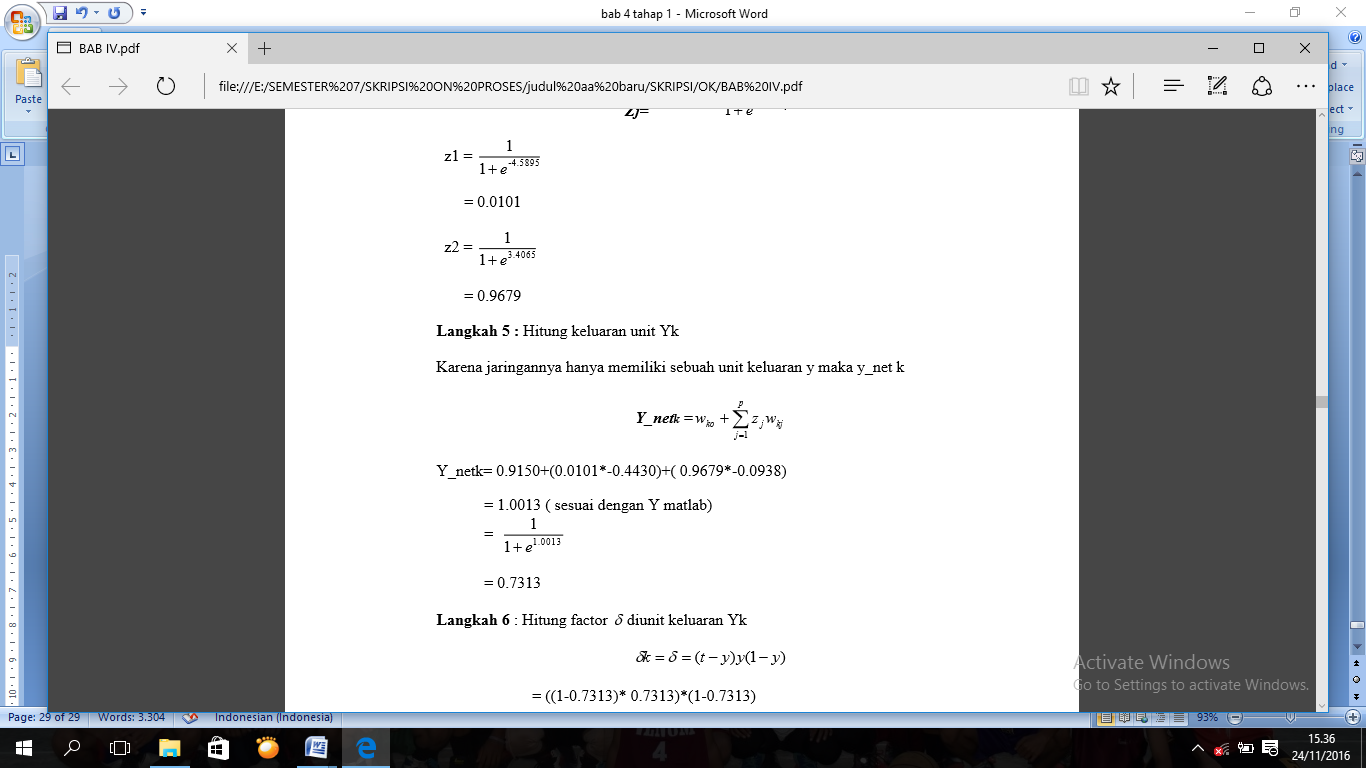
Z\_net2 = + (0,4334\*4,0125) + (0,5872\*5,2654) + (0,3890\*-4,6369)+ (0,2561\*0,4454) + (0,9000\*4,4182) = 4,6641



Z1 = = 0,2996

Z2 = = 0,9907

Langkah 5 : Hitung keluaran unit Yk

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

= + (-0,6848\*0,2996) + (0,9412\*0,9907) = 1,6415

y = = 0,8377

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

*Error*  = (0,5868 – 0,8377) = -0,2509

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

= (0,5868–0,8377) \* 0,8377 (1 – 0,8377)

= -0,0341

Suku perubahan bobot Wkj :

0,95 \* (0,0341) \* 1 = -0,0324

0,95 \* (0,0341) \* 0,2996= -0,0097

= 0,95 \* (0,0341) \* 0,0324= -0,0321

Langkah 7 : Hitung penjumlahan kesalahan dari unit tersembunyi (

\_net1 = -0,0341\* -0,6848 = 0,0234

\_net2 = -0,0341 \* 0,9143 = -0,0321

Faktor kesalahan di unit tersebunyi :

1 = 0,0234 \* -0,0097\* (1 –(-0,0097)) = 0,0049

2 = -0,0321 \* -0,0321 \* (1 – (-0,0321)) = -0,0003

**Tabel 4.14 Perubahan bobot input ke hidden**

|  |  |  |
| --- | --- | --- |
|  | Z1 | Z2 |
| X1 | (0.95)\*  (0,0049)\*( 0,4334) = 0,0020 | (0.95)\*  (-0,0003)\*( 0,4334) =  -0,0001 |
| X2 | (0.95)\*  (0,0049)\*( 0,5872) = 0,0027 | (0.95)\*  (-0,0003)\*( 0,5872) =  -0,0002 |
| X3 | (0.95)\*  (0,0049)\*( 0,3890) = 0,0018 | (0.95)\*  (-0,0003)\*( 0,3890) =  -0,0001 |
| X4 | (0. 95)\*  (0,0049)\*( 0,2561) = 0,0012 | (0. 95)\*  (-0,0003)\*( 0,2561) =  -0,0001 |
| X5 | (0. 95)\*  (0,0049)\*( 0,9000) = 0,0042 | (0. 95)\*  (-0,0003)\*( 0,9000) =  -0,0003 |

Langkah 8 : Hitung semua perubahan bobot Baru unit   
 keluaran :

-0,0097 + (-0,6848) = -0,6945

-0,0321 + (0,9412) = 0,9091

-0,0324 + (0,9143) = 0,8819

**Tabel 4.15 Perubahan bias dan bias baru pada neuran hidden**

|  |  |  |  |
| --- | --- | --- | --- |
| **Perubahan Bias Neuron Hidden** | | **Bias Baru Pada Neuron Hidden Layer** | |
|  | 0.95 \* 0,0049 =  0,0047 |  | 0,0047 + -3,6723 = 3,8253 |
|  | 0.95 \* -0,0003 =  -0,0003 |  | -0,0003 + -2,4535 = 2,4538 |

**Tabel 4.16 Perubahan bobot input baru**

|  |  |  |
| --- | --- | --- |
| **BARU** | **Z1** | **Z2** |
| **X1** | 3,6723 + 0,0020= 3,6743 | 4,0125 + (-0,0001) = 4,0124 |
| **X2** | -5,6066 + 0,0027= -5,6039 | 5,2654 + (-0,0002) = 5,2652 |
| **X3** | 1,7995 + 0,0018 = 1,8013 | -4,6369 +(-0,0001)= -4,6370 |
| **X4** | -2,4841+ 0,0012= -2,4829 | 0,4454 + (-0,0001) = 0,4453 |
| **X5** | 5,1308 + 0,0042= 5,1350 | 4,4182 + (-0,0003)= 4,4179 |

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 harga *bitcoin* berdasarkan pergerakan grafik open, pergerakan grafik high, pergerakan grafik low, volume *bitcoin*, permintaan *bitcoin* dan harga *bitcoin* satu jam kedepan sebagai output nya.

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

= 0,8377\* (128999000 - 122503000) + 122503000 = 127944699,2

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 selanjutnya akan dibandingkan dengan hasil penolahan data secara manual untuk melihat 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.

* 1. **Perancangan Sistem** 
     1. **Perancangan Sistem Dengan Menggunakan UML**

Aplikasi ini dirancang menggunakan alat bantu berupa UML (*Unified Modelling Language*) agar mempermudah memindahkan konsep sistem yang dirancang ke dalam bentuk program, dimana perancangannya digambarkan dalam bentuk diagram-diagram berikut :

* + - 1. ***Use Case Diagram***

*Use case diagram* adalah abtraksi dari interaksi antara sistem dengan *actor*. *Use case* diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case* digunakan untuk mengetahui setiap fungsi dari sistem dan siapa saja yang dapat mengakses atau menggunakan fungsi tersebut.

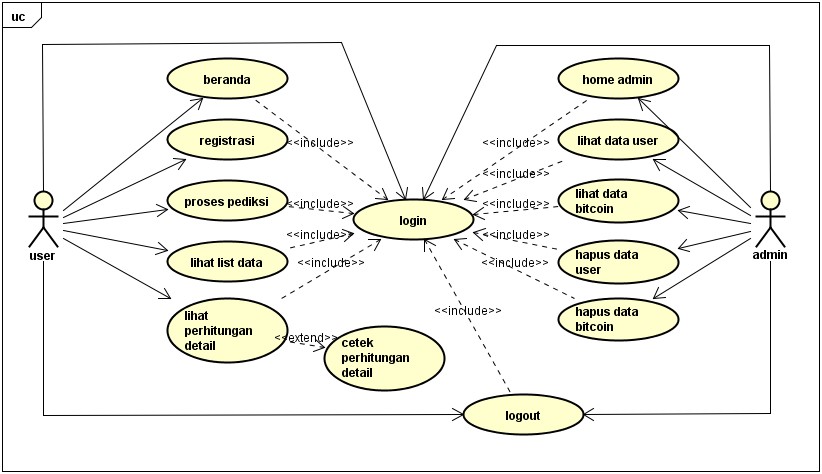
1. ***Definisi Actor***

# Tabel 4.17 definisi *Actor*

|  |  |  |
| --- | --- | --- |
| **No** | **Aktor** | **Deskripsi** |
| 1 | *Admin* | Pada sistem ini *admin* merupakan aktor yang dapat memanagemen data, seperti menghapus data-data yang bermasalah atau data-data yang sudah tidak *relevan* |
| 2 | *User* | Pada sistem ini *user* merupakan aktor yang dapat mengakses fitur-fitur yang berkaitan dengan user seperti, prediksi *bitcoin,* lihat list data, lihat perhitungan detail, dan cetak perhitungan. |

Berikut ini merupakan *Use Case* *Diagram* dari Jaringan Syaraf Tiruan yang akan dibangun:

1. ***Use Case Diagram***



Gambar 4.2 *Use Case Diagram*

Dalam *Use case Diagram* tersebut menggambarkan dimana *Admin* nantinya hanya dapat memanagemen data seperti menghapus data bictoin dan menghapus data *user* yang bermasalah, sedangkan *user* dapat melakukan proses prediksi, melihat list semua data dan melihat detail dari perhitungan.

1. ***Skenario Use Case Diagram***
2. ***Login User***

Dibawah ini adalah *Use Case Skenario Login*  yang aktornya yaitu *user* pada *Use Case Diagram.*

**Tabel 4.18 Skenario *Use Case Diagram Login User***

|  |  |
| --- | --- |
| **Identifikasi** | |
| Nama | *Login* |
| Tujuan | Masuk kedalam sistem sebagai *user* |
| Deskripsi | Proses *login* merupakan merupakan proses untuk masuk ke sistem sebagai *user* |
| Aktor | *user* |
| Use Case yang berkaitan | - |
| **Skenario Utama** | |
| Kondisi Awal | Form *login* ditampilkan |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Mengisi Form Form *Login* | 1. Mengidentifikasi data *login* dengan data *user* di *database* |
|  | 1. Bila cocok sistem akan menampilkan halaman menu utama untuk *user* |
| **Skenario Alternatif** | |
| **Aksi Aktor** | **Reaksi Sistem** |
|  | 1. Menampilkan pesan bahwa Username dan *password* tidak benar |
| 1. Mengisi kembali Form *login* | 1. Mengautentikasi data *login* dengan data *user* pada *database* |
|  | 1. Bila cocok sistem akan menampilkan halaman utama untuk *user* |
| Kondisi Awal | *User* dapat melakukan kegiatan pada sistem prediksi sesuai dengan kewenangan sebagai *User* |

1. **Proses Prediksi Oleh User**

Di bawah ini adalah *Use Case Skenario* proses prediksi yang aktornya yaitu *user* pada *Use Case Diagram.*

**Tabel 4.19 Skrenario *Use Case Diagram* Proses Prediksi**

|  |  |
| --- | --- |
| **Identifikasi** | |
| Nama | Proses Prediksi |
| Tujuan | User malakukan proses predikisi |
| Deskripsi | User melakukan proses prediksi dengan cara mamasukan data-data yang dibutuhkan untuk proses |
| Aktor | User |
| Use Case yang berkaitan | - |
| **Skenario Utama** | |
| Kondisi Awal | Form prediksi ditampilkan |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Mengisi data yang diperlukan untuk prediksi (open, high, low, close, volume, warna candle) | 1. Mamproses data yang sudah di inputkan |
|  | 1. Menampilkan hasil prediksi harga *bitcoin* yang sudah di inputkan |
| **Skenario Alternatif** | |
| **Aksi Aktor** | **Reaksi Sistem** |
|  | 1. Menampilkan data tidak lengkap untuk diprediksi |
| 1. Mengisi kembali Form prediksidengan lengkap | 1. Bila data lengkap sistem akan menampilkan hasil prediksi harga bitoin |
| Kondisi Awal | User dapat melakukan kegiatan yang diperlukan untuk memprediksi harga |

1. **Cetak Perhitungan Detail**

Dibawah ini adalah *Use Case Skenario* cetak perhitungan detail yang aktornya yaitu user pada *Use Case Diagram*

**Tabel 4.20 Skenario *Use Case Diagram* cetak perhitungan detail**

|  |  |
| --- | --- |
| **Identifikasi** | |
| Nama | Cetak Perhitungan Detail |
| Tujuan | Mencetak perhitungan detail dari proses prediksi |
| Deskripsi | Proses Cetak perhitungan detail merupakan proses untuk mencetak semua detail dari perhitungan prediksi secara matematis |
| Aktor | User |
| Use Case yang berkaitan | - |
| **Skenario** | |
| Kondisi Awal | Form list data ditampilkan |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Memilih data yang ingin dilihat secara detail | 1. Menampilkan perhitungan detail yang dipilih |
| 3. Melakukan percetakan detail data | 4. sistem akan melakukan percetakan hasil prediksi |

1. ***Login Admin***

Dibawah ini adalah *Use Case Skenario Login*  yang aktornya yaitu *Admin* pada *Use Case Diagram.*

**Tabel 4.21 Skenario *Use Case Diagram Login Admin***

|  |  |
| --- | --- |
| **Identifikasi** | |
| Nama | *Login* |
| Tujuan | Masuk kedalam sistem sebagai *Admin* |
| Deskripsi | Proses *login* merupakan merupakan proses untuk masuk ke sistem sebagai *Admin* |
| Aktor | *Admin* |
| Use Case yang berkaitan | - |
| **Skenario Utama** | |
| Kondisi Awal | Form *login* ditampilkan |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Mengisi Form Form *Login* | 1. Mengidentifikasi data *login* dengan data *admin* di *database* |
|  | 1. Bila cocok sistem akan menampilkan halaman menu utama untuk *admin* |
| **Skenario Alternatif** | |
| **Aksi Aktor** | **Reaksi Sistem** |
|  | 1. Menampilkan pesan bahwa *Username* dan *password* tidak benar |
| 1. Mengisi kembali Form *login* | 1. Mengautentikasi data *login* dengan data *user* pada *database* |
|  | 1. Bila cocok sistem akan menampilkan halaman utama untuk *admin* |
| Kondisi Awal | *Admin* dapat melakukan kegiatan pada sistem prediksi sesuai dengan kewenangan sebagai *Admin* |

1. **Hapus Data User Oleh Admin**

Dibawah ini adalah *Use Case Skenario* hapus data user yang aktornya yaitu admin pada *Use Case Diagram*

**Tabel 4.22 Skenario *Use Case* Hapus Data User**

|  |  |
| --- | --- |
| **Identifikasi** | |
| Nama | Hapus data *user* |
| Tujuan | Menghapus data *user* |
| Deskripsi | Menghapus data-data *user* yang bermasalah |
| Aktor | *Admin* |
| Use Case yang berkaitan | - |
| **Skenario** | |
| Kondisi Awal | Form list data *user* ditampilkan |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Memilih data yang ingin dihapus |  |
| 2. Mengklik hapus pada data yang dipilih | 4. Menghapus data yang dipilih |
|  | 5. Menampilkan list data *user* |

1. **Hapus data bitcoin oleh admin**

Dibawah ini adalah *Use Case Skenario* hapus data bitcoin yang aktornya yaitu admin pada *Use Case Diagram*

**Tabel 4.23 Skenario *Use Case Diagram Hapus Data Bitcoin***

|  |  |
| --- | --- |
| **Identifikasi** | |
| Nama | Hapus data *bitcoin* |
| Tujuan | Menghapus data *bitcoin* |
| Deskripsi | Menghapus data-data *bitcoin* yang bermasalah atau yang sudah tidak *relevan* |
| Aktor | *Admin* |
| Use Case yang berkaitan | - |
| **Skenario** | |
| Kondisi Awal | Form list data *bitcoin* ditampilkan |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Memilih data yang ingin dihapus |  |
| 2. Mengklik hapus pada data yang dipilih | 4. Menghapus data yang dipilih |
|  | 5. Menampilkan list data *bitcoin* |

* + - 1. ***Class Diagram***

*Class Diagram* menggambarkan *Class*, *fitur* dan hubungan-hubungan statis diantara sistem. Dalam *Class Diagram* terdapat *atribute* dan *metode* yang memiliki peran masing-masing.

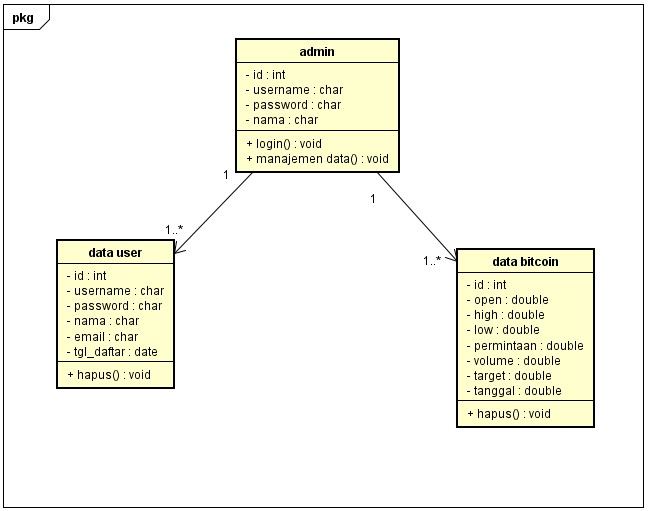
1. Definisi *Class Diagram Admin*

Defenisi *Class Diagram Admin* dapat dijelaskan pada tabel dibawah ini yang menggambarkan *Class*, fitur dan hubungan-hubungan yang terjadi.

**Tabel 4.24 Definisi *Class Diagram Admin***

|  |  |  |
| --- | --- | --- |
| **No** | **Class** | **Deskripsi** |
| 1 | Admin | Berisikan data-data *admin* seperti username, password, nama *admin* yang bisa mengakses sistem. |
| 2 | Data | Berisikan data user dan data bitcoin untuk proses managemen data nantinya. |

Berikut ini merupakan *Class Diagram Admin* dari sistem yang akan dibangun:



Gambar 4.3 *Class Diagram Admin*

Pada gambar diatas merupakan class diagram dari admin pada sistem jaringan syaraf tiruan, dimana admin disini mempunyai wewenang atas management data.

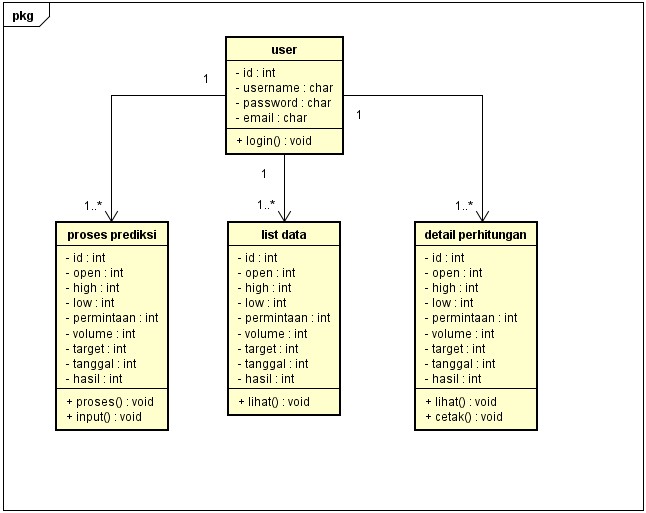
1. Definisi *Class Diagram User*

Defenisi *Class Diagram User* dapat dijelaskan pada tabel dibawah ini yang menggambarkan *Class*, fitur dan hubungan-hubungan yang terjadi.

**Tabel 4.25 Definisi *Class Diagram User***

|  |  |  |
| --- | --- | --- |
| **No** | **Class** | **Deskripsi** |
| 1 | User | Berisikan data-data *User* seperti username, password, nama, email yang bisa mengakses sistem. |
| 2 | Data | Berisi data input sebagai kelengkapan data untuk prediksi. |

Berikut ini merupakan *Class Diagram User* dari sistem yang akan dibangun:



Gambar 4.4 *Class Diagram User*

Pada gambar *class* diagram di atas *User* dalam sistem ini adalah seorang pengguna yang dapat melakukan pemrosesan untuk melakukan prediksi maka data *user* akan disimpan pada tabel data. Setiap *Class* dihubungkan denga garis panah, setiap garis memiliki sebuah angka disetiap ujungnya, angka pada ujung garis ini dinamakan *multiplicity* yang mana menunjukan jumlah kejadian minimun dan maksimum dari suatu *Class* untuk satu kejadian dari *Class* yang terkait

1. Perancangan Database

Perancangan *Database* dilakukan setelah pemodelan sistem selesai dibuat, dengan menggunakan *MySql* sebagai *Database Server,* dalam hal ini dilakukan perancangan terhadap *field-field* yang akan digunakan pada setiap *database*.

1. Tabel *Admin*

Tabel *Admin* menampung data-data *admin (Manager)* yang terdiri dari lima *field* dengan struktur *field* seperti berikut :

Tabel 4.26 *Admin*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Filed** | **Type** | **Ket** |
| 1 | id | int(11) | Primary key |
| 2 | username | varchar(25) |  |
| 3 | password | varchar(25) |  |
| 4 | nama | varchar(25) |  |

1. Tabel Data

Tabel data menampung data-data lengkap yang berhubungan dengan grafik open, grafik high, grafik low, volume, permintaan dan harga yang akan diprediksi. Terdiri dari *field-field* dengan struktur seperti berikut :

Tabel 4.27 Data Input

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Filed** | **Type** | **Ket** |
| 1 | id | int(11) | Primary key |
| 2 | open | double() |  |
| 3 | high | double() |  |
| 4 | low | double() |  |
| 5 | volume | double() |  |
| 6 | permintaan | double() |  |
| 7 | Target | double() |  |
| 8 | Tanggal | datetime |  |

1. Tabel User

Tabel *User* menampung data-data *User (Manager)* yang terdiri dari lima *field* dengan struktur *field* seperti berikut :

Tabel 4.28 *User*

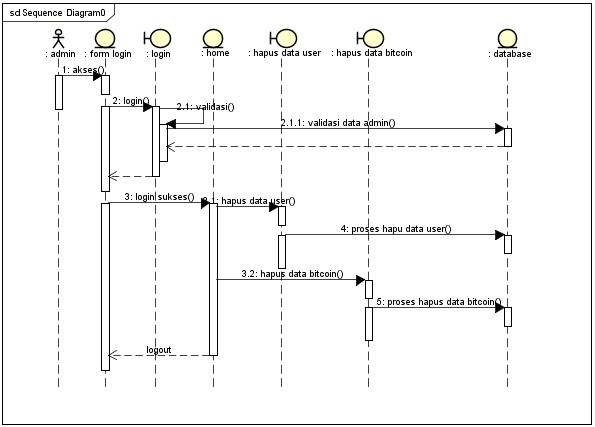
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Filed** | **Type** | **Ket** |
| 1 | id | int(11) | Primary key |
| 2 | username | varchar(25) |  |
| 3 | password | varchar(25) |  |
| 4 | nama | varchar(20) |  |
| 5 | email | varchar(20) |  |
| 6 | Tgl daftar | date |  |

* + - 1. ***Sequence Diagram***

*Sequence Diagram* adalah gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan suatu system pada web yang sesuai dengan *Use Case Diagram.*

1. *Sequence Diagram Admin*

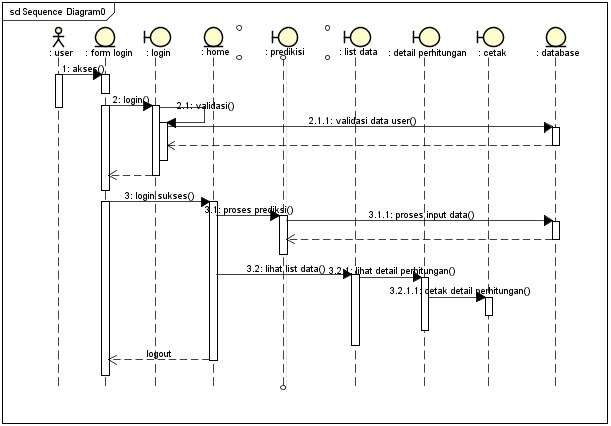
Diagram ini menjelaskan urutan langkah-langkah yang dilakukan oleh.



Gambar 4.5 *Sequence Diagram Admin*

1. *Sequence Diagram User*

Diagram ini menjelaskan urutan langkah-langkah yang dilakukan oleh *user*.



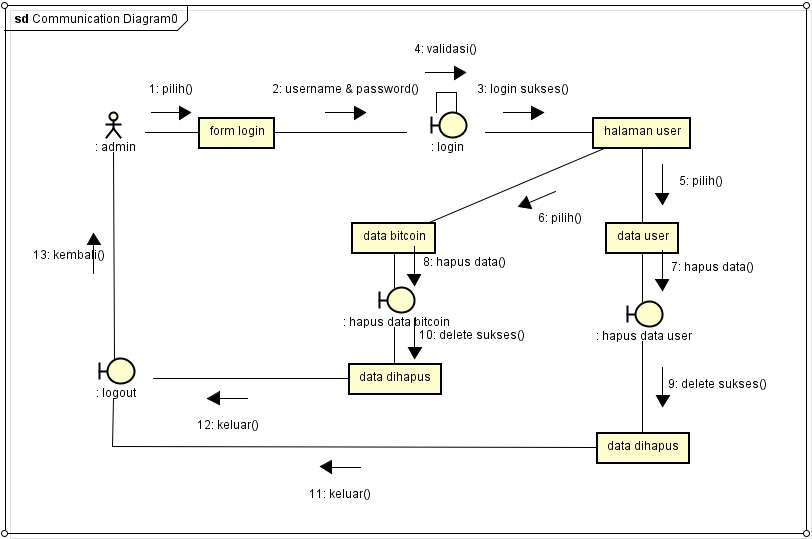
Gambar 4.6 *Sequence Diagram User*

* + - 1. ***Collaboration Diagram***

*Colaboration Diagram* adalah cara *alternative* untuk mengetahui tahap-tahap terjadinya suatu aktifitas pada web. Perbedaan antara *Collaburation* dengan *Sequence* Diagram adalah *Collaburation Diagram* memperlihatkan bagaimana hubungan antara beberapa objek, sedangkan *Sequence Diagram* memperlihatkan bagaimana urutan kejadian.

1. *Collaboration Diagram Admin*

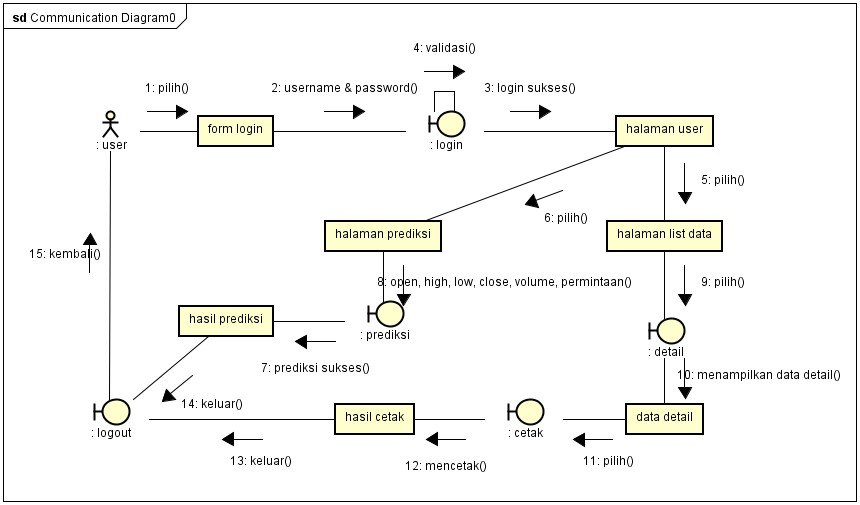
Diagram ini menjelaskan proses *admin* ketika mengakses sistem mulai dari login, hapus data user, hapus data bitcoin sampai logout.



Gambar 4.7 *Collaboration Diagram Admin*

1. *Collaboration Diagram User*

Diagram ini menjelaskan proses *admin* ketika mengakses sistem mulai dari login, prediksi, lihat list data, cetak detail data sampai ke logout.



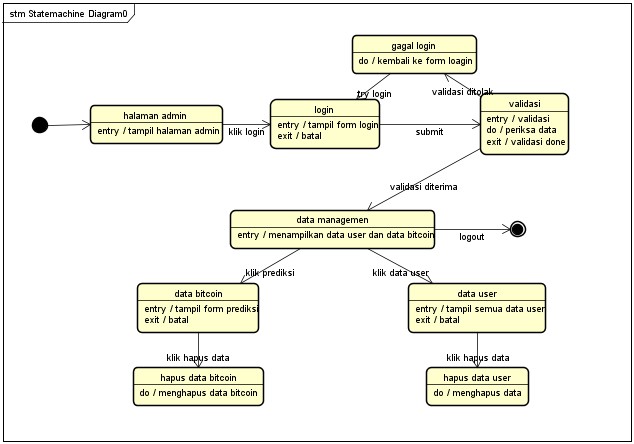
Gambar 4.8 *Collaboration Diagram User*

* + - 1. ***State Chart Diagram***

*State Chart Diagram* menggambarkan tentang kelakuan (*behavior*) sistem yang akan dibangun. *State Chart Diagram admin* akanmelakukan prediksi pada data yang ada untuk diprediksi dan menampilkan halaman Home untuk melakukan aktivitas *admin.*

1. *Statechart Diagram Admin*

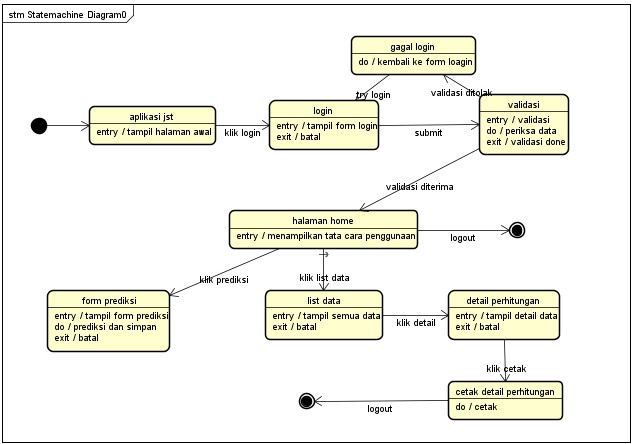
Berikut adalah *State Chart* untuk menggambarkan bagaimana *Admin* mengakses sistem.



Gambar 4.9 *State Chart Diagram Admin*

1. *Statechart Diagram User*

Berikut adalah *State Chart* untuk menggambarkan bagaimana *User* mengakses system



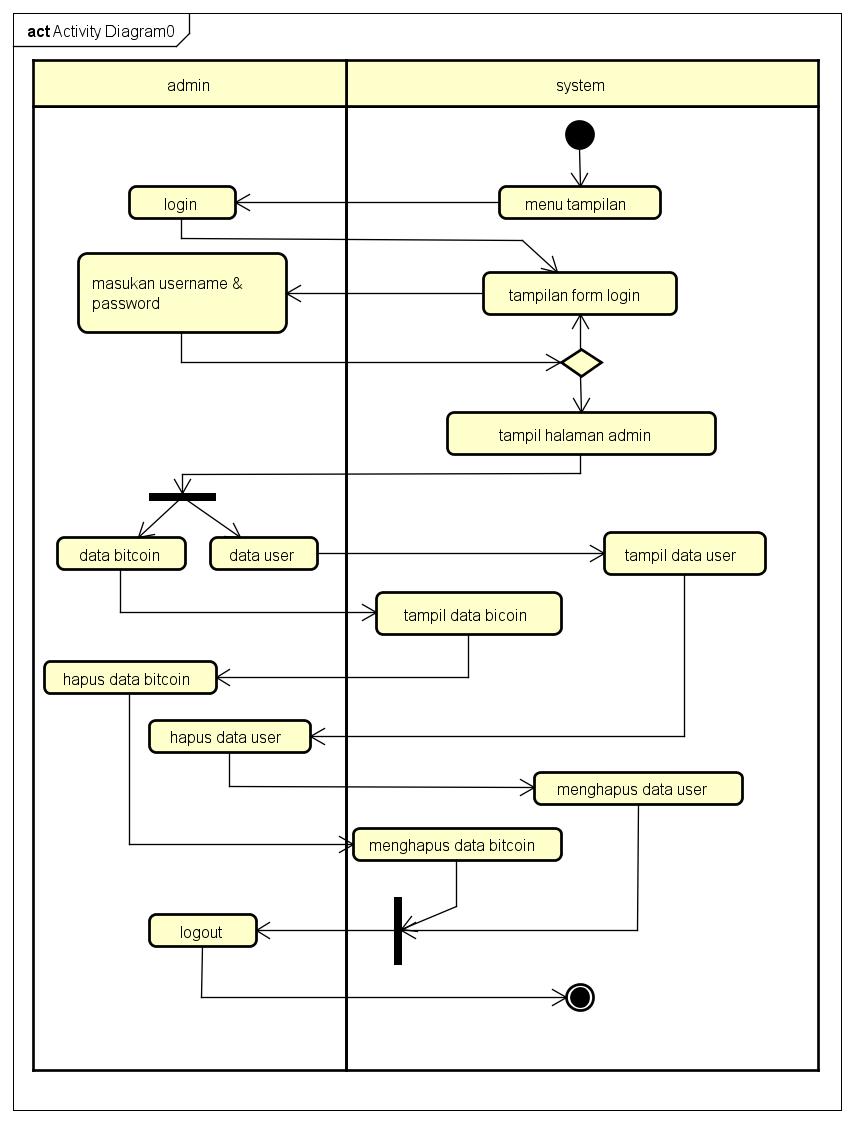
Gambar 4.10 *State Chart Diagram User*

* + - 1. ***Activity Diagram***

*Activity diagram* menggambarkan aliran aktivitas dalam sistem yang dirancang dimana aliran berawal, decision yang mungkin terjadi dan bagaimana aliran berakhir. Dan juga merupakan sebuah diagram dimana dalam diagram tersebut terdapat semua aktivitas yang dapat dilakukan pada web prediksi permintaan pasokan barang. Dalam *Activity diagram* digambarkan aktifitas apa saja yang dapat dilakukan dari setiap actor yang ada. Adapun *Activity diagram* yang menggambark anaktivitas apa saja yang dilakukan pada Prediksi jumlah pasokan barang adalah sebagai berikut :

1. *Activity Diagram Admin*

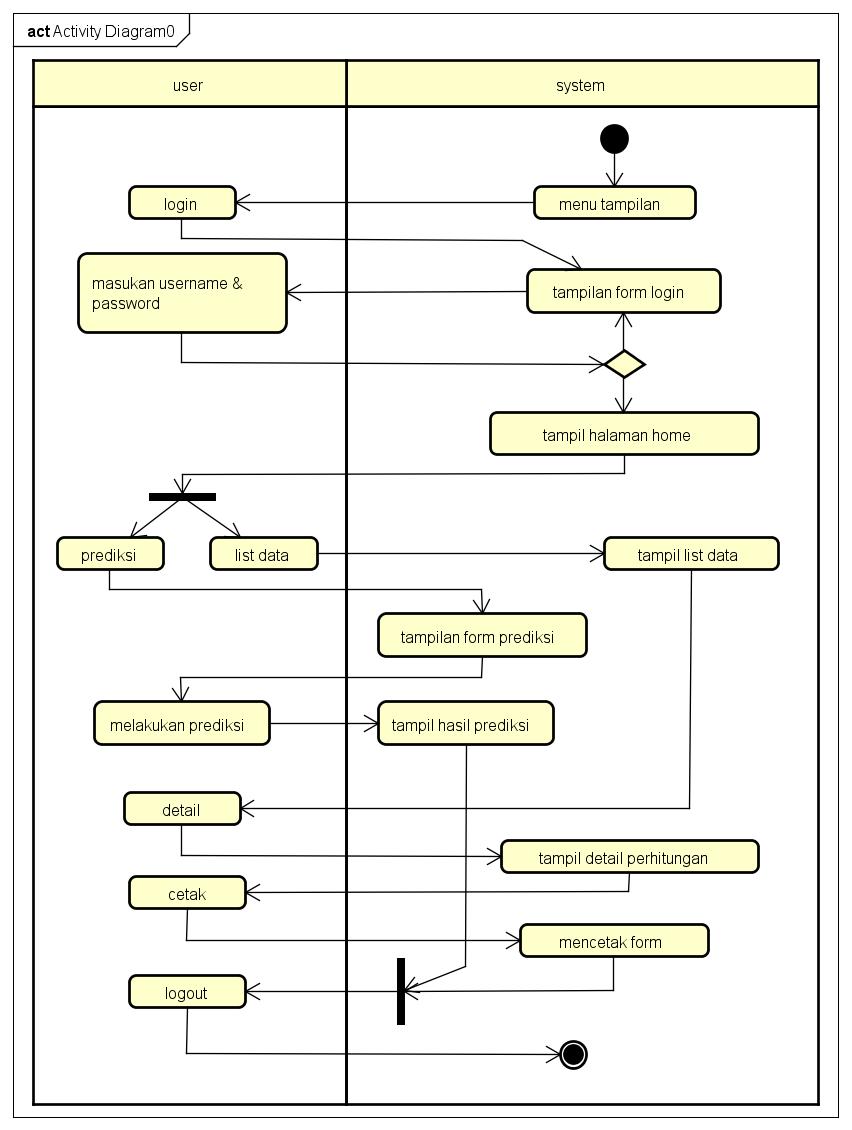
*Activity diagram* *Admin* menggambarkan segala aktifitas yang dilakukan *admin* secara garis besar dalam managemen data.



**Gambar 4.11 *Activity Diagram Admin***

1. *Activity Diagram User*

*Activity diagram* *User* menggambarkan segala aktifitas yang dilakukan *user* secara garis besar dalam proses prediksi.



**Gambar 4.12 *Activity Diagram User***

* + - 1. ***Deployment Diagram***

*Deployment Diagram* menggambarkan secara lengkap bagaimana komponen di *Deploy* dalam infrastruktur *system*, dimana komponen akan terletak , bagaimana kemampuan jaringan pada kondisi tertentu, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal. Hubungan antar node (misalnya TCP/IP) *requiretment* dapat juga didefenisikan dalam diagram ini.

Google Chrome

Mozilla Firefox

Xampp

Mysql Server

PHP,CSS

**Gambar 4.13 *Deploymant Diagram***

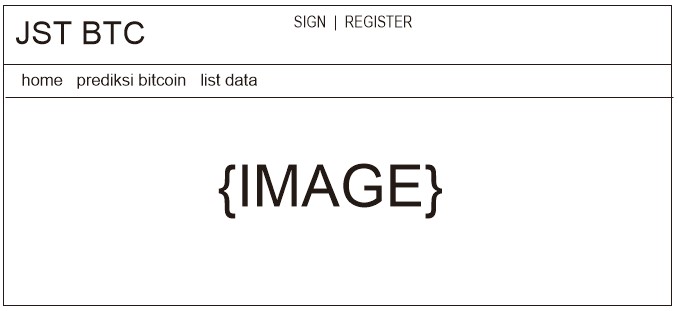
*Deployment Diagram* merupakan diagram yang bertujuan agar *admin* tidak mengalami kesulitan-kesulitan dalam memprediksi harga *bitcoin*. Pada *server admin*, *webserver* dan *database* yang digunakan adalah *XAMPP* dan *MySql Server*. Sementara *customer* dapat melakukan akses dengan menggunakan *browser* seperti *Google Chrome* dan *Mozilla Firefox*.

* + 1. **Desain *Interface* atau Antar Muka**

Disain *interface* atau perancangan antarmuka adalah bentuk dari rancangan tampilan sementara dari pembuatan Jaringan Syaraf Tiruan. Perancangan ini dibuat untuk memberikan penjelasan tentang tampilan yang dihadapkan pada *actor* pada saat menggunakan sistem, Sehingga dapat mempermudah dalam mengimplementasikan sistem serta akan memudahkan pembangunan aplikasi yang memenuhi prinsip perancangan antarmuka yang baik.

1. Desain Halaman Awal

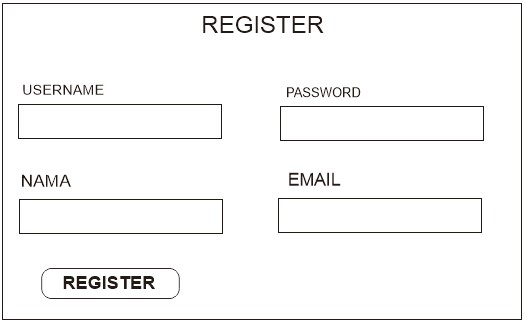
Halaman ini berisikan tentang halaman awal dari web Jaringan Syaraf Tiruan – *Backpropagation* prediksi *bitcoin*. Desain halaman ini dapat digambarkan sebagai berikut.



**Gambar 4.14 Perancangan Antarmuka Halaman Awal**

1. Desain Halaman *Register*

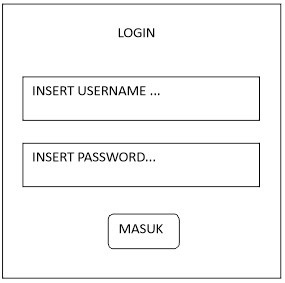
Untuk menjadi seorang *user* aktor harus melakukan registrasi dulu, sehingga mendapatkan username dan password untuk login ke halaman prediksi nantinya.



**Gambar 4.15 Perancangan Antarmuka Register**

1. Desain Halaman *Login*

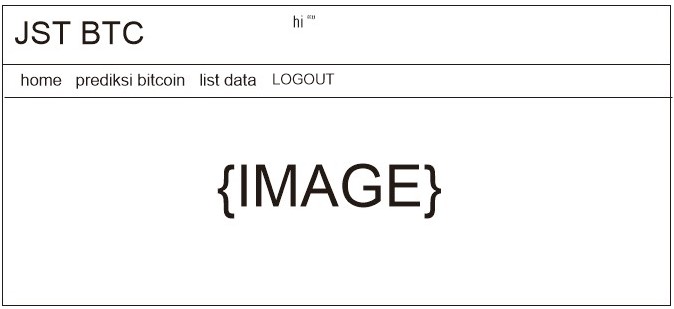
Untuk melakukan pengelolaan data, *user* harus melakukan *login* terlebih dahulu ke dalam sistem. Untuk dapat masuk kedalam halaman *user*, pengguna harus memiliki username dan password yang akan diinputkan.



**Gambar 4.16 Perancangan Antarmuka *Login* *User***

1. Desain halaman *Home*

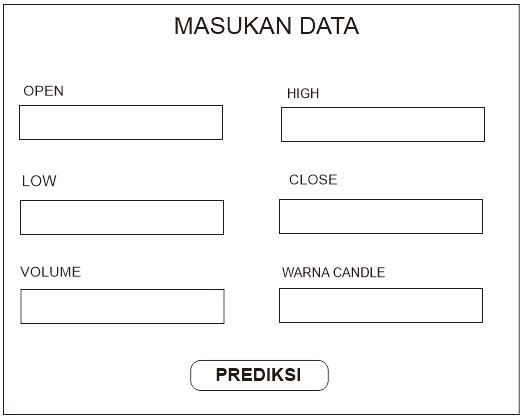
Halaman ini berisikan halaman awal dari web Jaringan Syaraf Tiruan setelah kita sudah Login. Desain halaman ini dapat digambarkan sebagai berikut :



**Gambar 4.17 Perancangan Halaman *Home***

1. Desain Halaman Predikisi

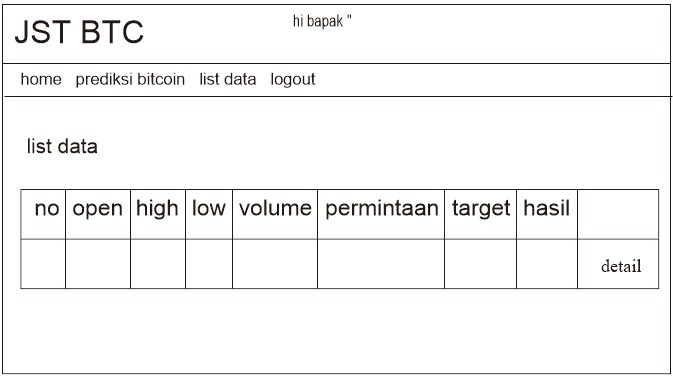
Halaman ini menampilkan form predikis dimana nanti user menginputkan data yang diperlukan untuk prediksi.



**Gambar 4.18 Perancangan Halaman Prediksi**

1. Desain Halaman List Data

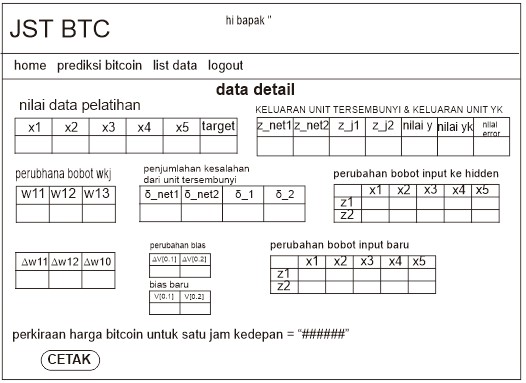
Halaman ini menampilkan seluruh data yang diprediksi oleh user lain.



**Gambar 4.19 Perancangan Halaman *List* Data**

1. Desain Halaman Detail

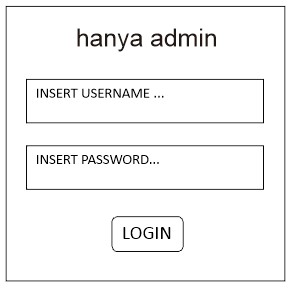
Halaman ini menampilkan detail perhitungan dari proses prediksi.



**Gambar 4.20 Perancangan Halaman Data Detail**

1. Halaman login admin

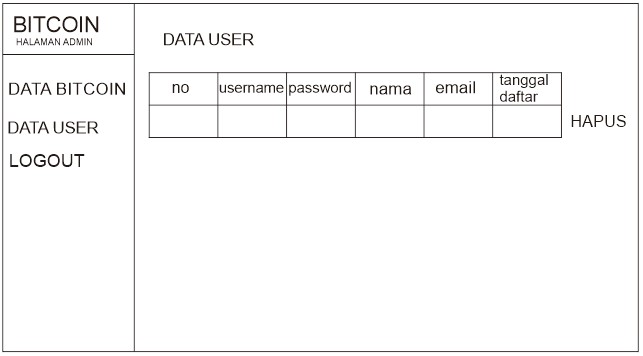
Halaman ini menampil form login untuk admin, dimana nantinya admin akan masuk ke dalam sistem untuk memanagemen data.



**Gambar 4.21 Perancangan Halaman *Login Admin***

1. Halaman data *user* pada *admin*

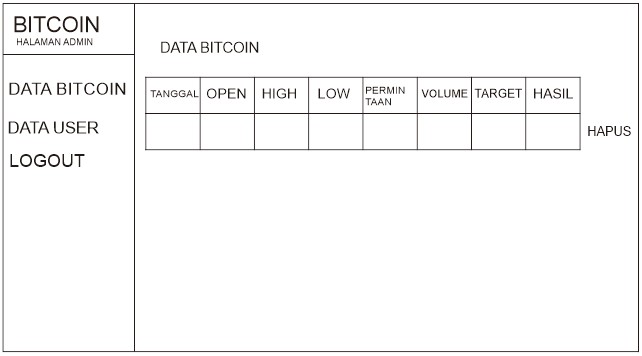
Halaman ini berisikan data-data keseluruham user dimana nantinya *admin* akan menghapus data-data yang bermasalah.



**Gambar 4.22 Perancangan Halaman Data *User***

1. Halaman Data *Bitcoin* Pada *Admin*

Halaman ini berisikan keseluruham data-data *bitcoin* dimana nantinya *admin* akan menghapus data-data yang bermasalah.



**Gambar 4.23 Perancangan Halaman Data *Bitcoin***

**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.

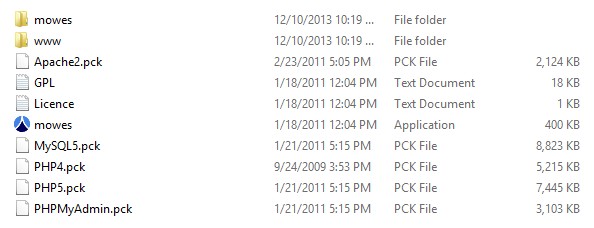
* + - 1. **Instalasi Mowes Portable II**

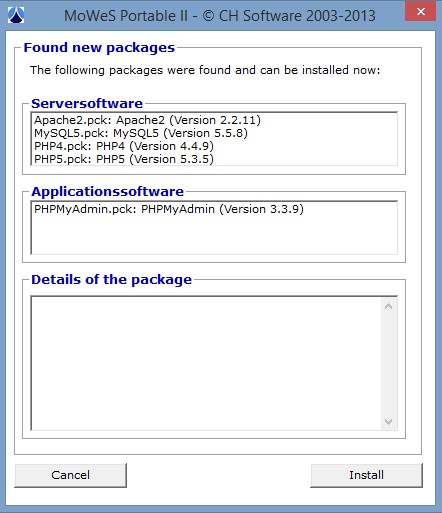
*Mowes Portable* merupakan sebuah paket untuk menjalankan *PHP*, apache dan *MySQL*. Dengan menggunakan mowes portable kita tidak perlu repot menginstall ketiga software tersebut secara terpisah.

Berikut cara menginstal *Mowes Portabel* :

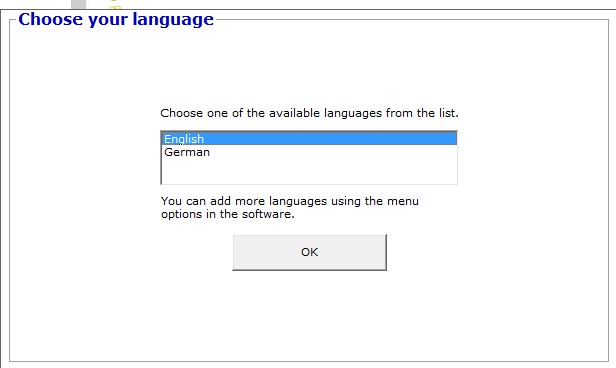
1. File mowes sebelum di instal.

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

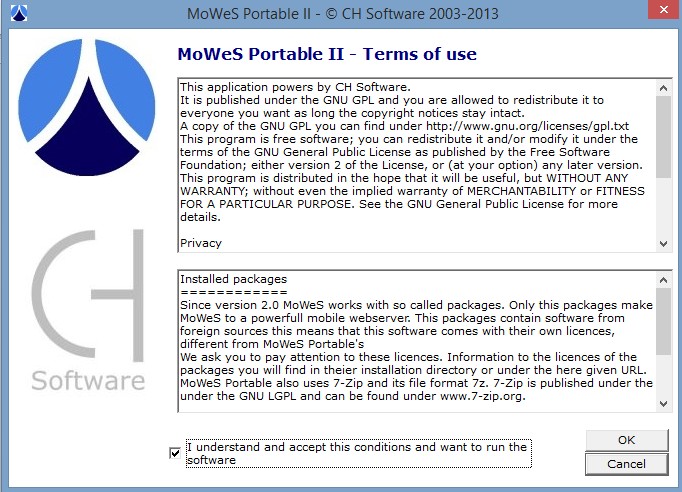




**Gambar 5.1 Tampilan Master Mowes**

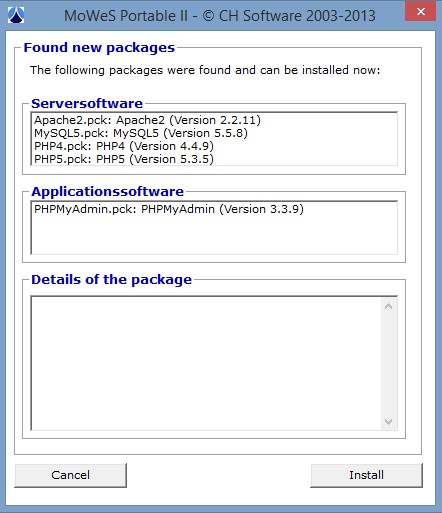
1. Sehingga akan tampil gambar.

**Gambar 5.2 Pemilihan Bahasa Penginstalan**

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

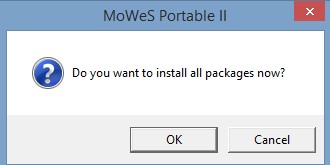
**Gambar 5.3 Halaman Persetujuan**

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

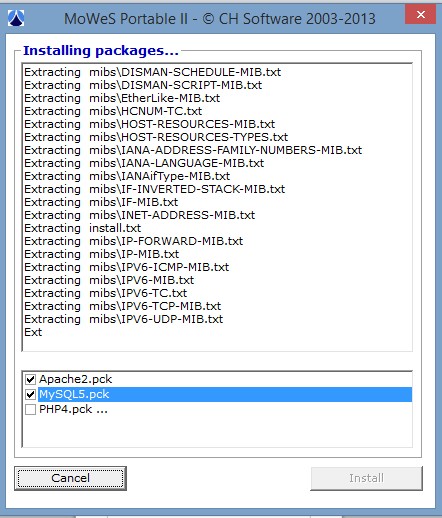


**Gambar 5.4 Halaman *List* *Packed* *Mowes***

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.

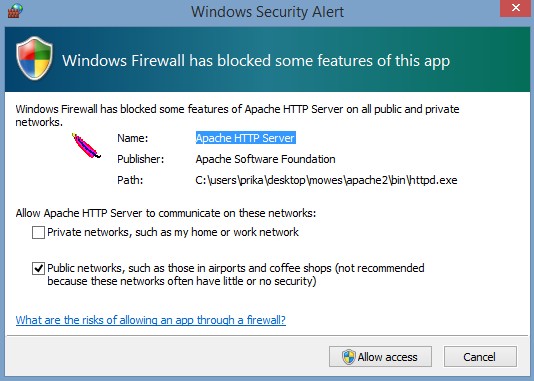
**Gambar 5.6 Proses *Instal* *Mowes***

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*”.

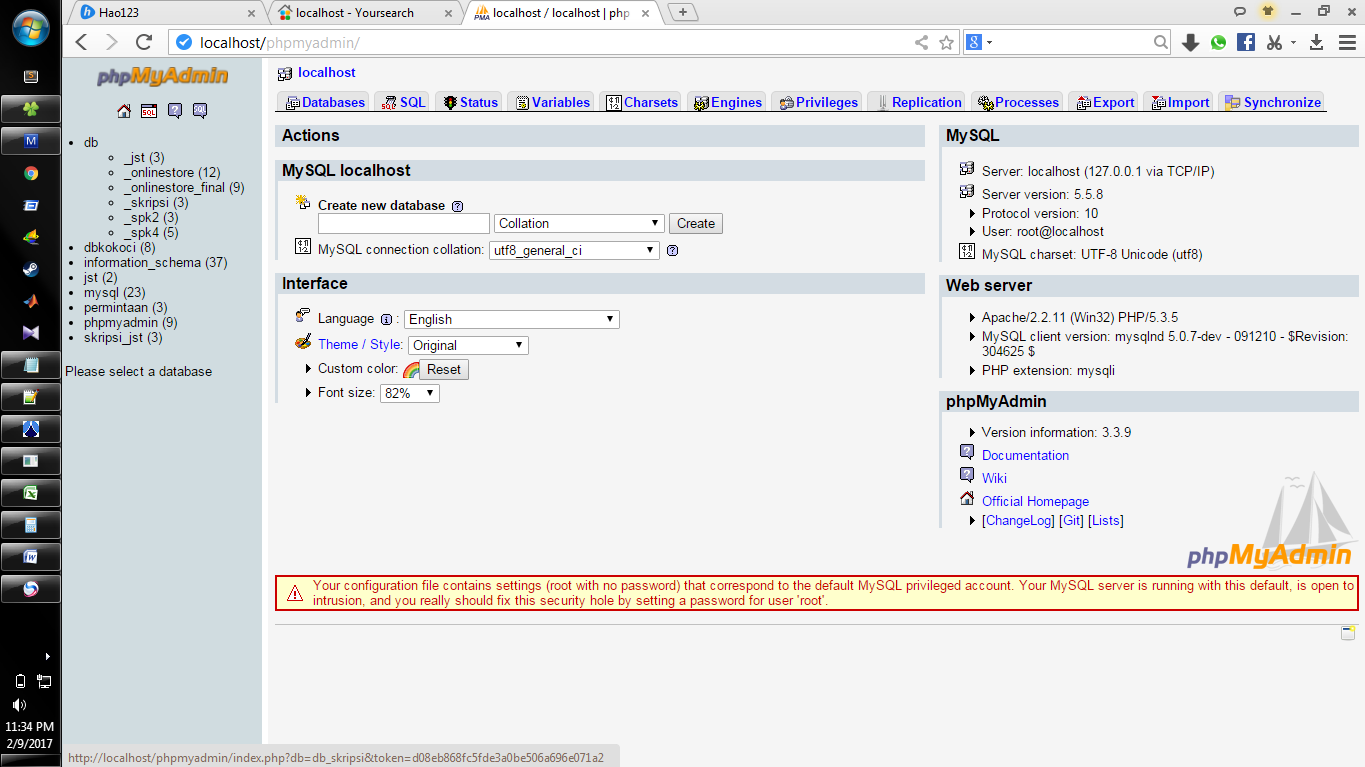


**Gambar 5.8 *Windows* *security* *apache* *HTTP* *Server***

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**

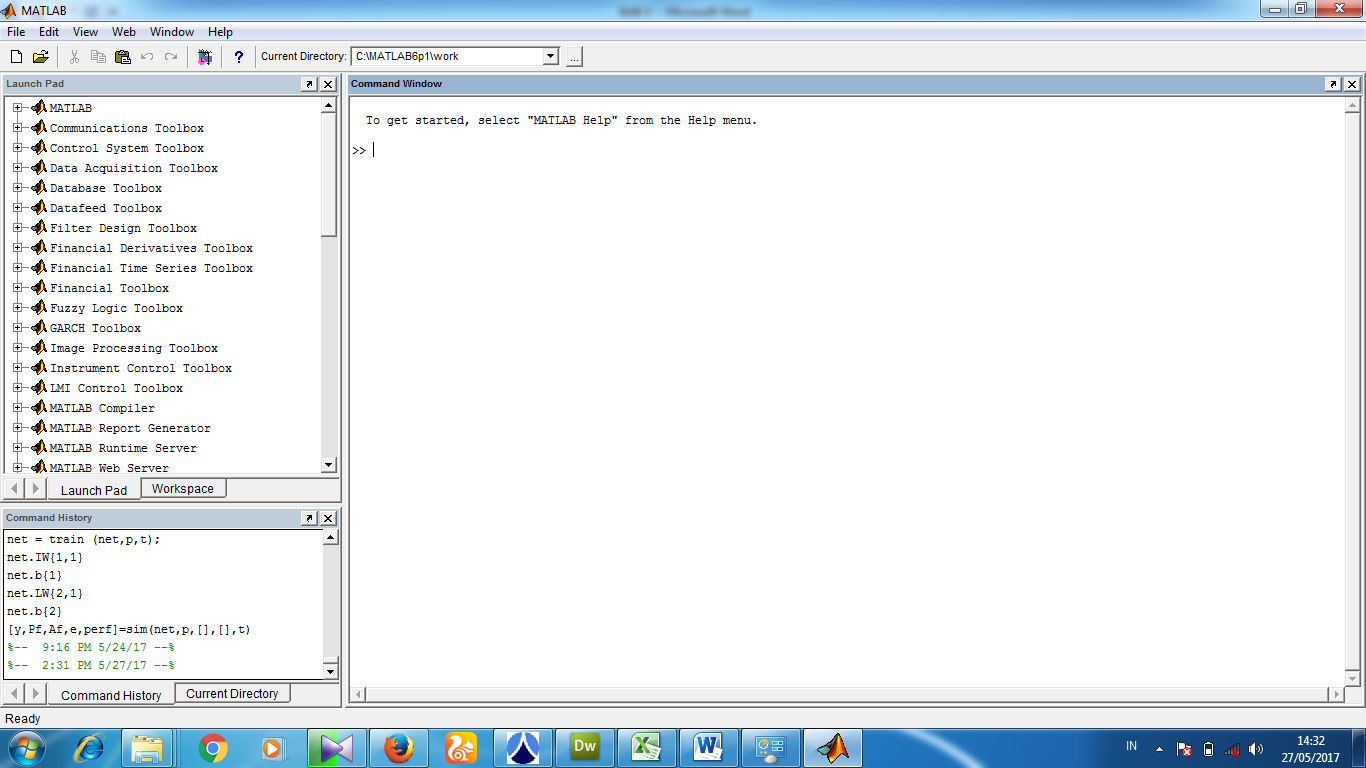
Pengujian dan implementasi sistem bertujuan untuk melihat apakah sistem yang dirancang sudah sesuai dengan apa yang diinginkan atau belum, setelah dilakukannya pengujian dan implementasi, kualitas sebuah *system* akan terlihat. Tampilan program yaitu merupakan sub bab yang menjelaskan tentang proses 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 :

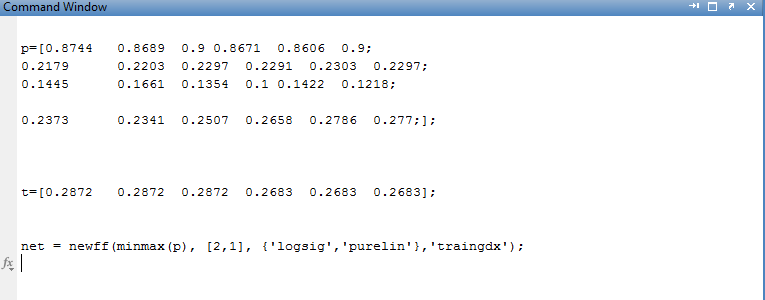
1. Tampilan Awal Matlab.

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



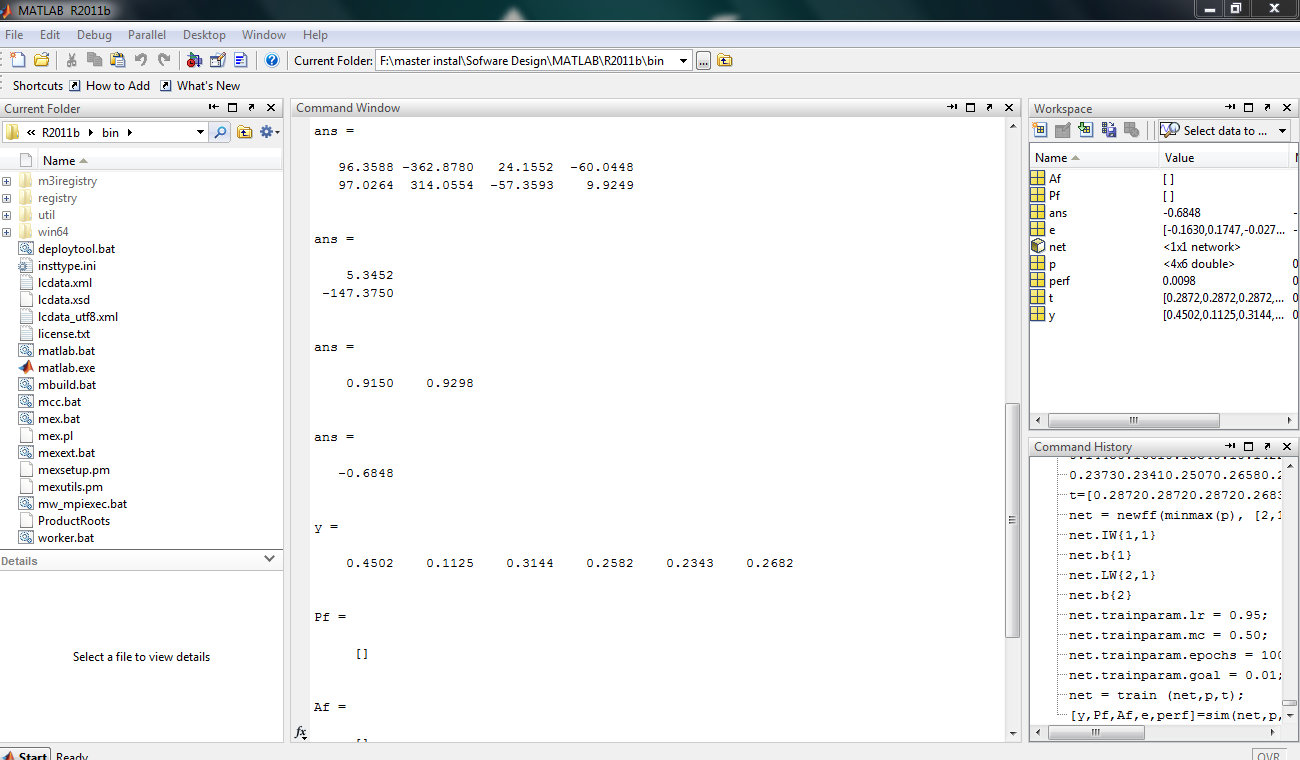
**Gambar 5.11 Tampilan Awal Matlab**

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 :

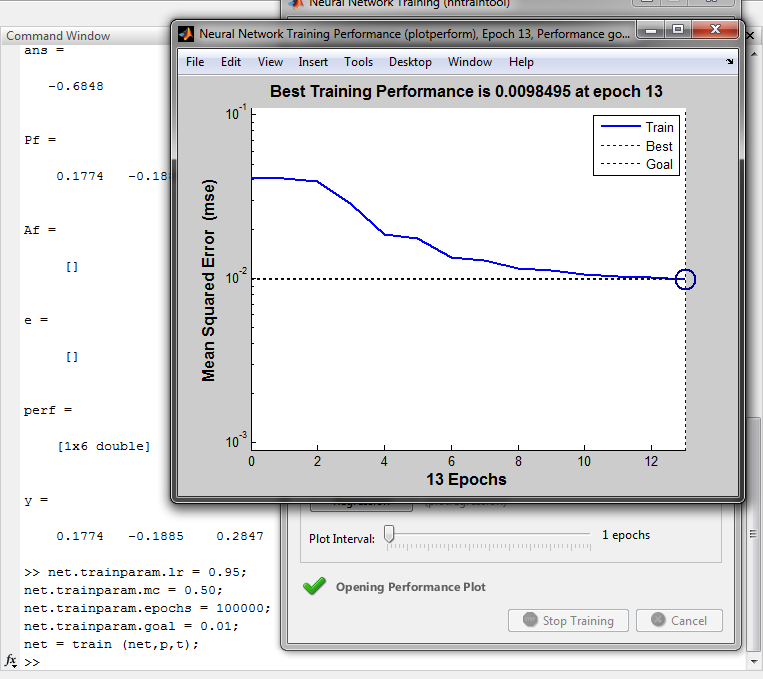


**Gambar 5.12 Tampilan Masukan Input Dan Target**

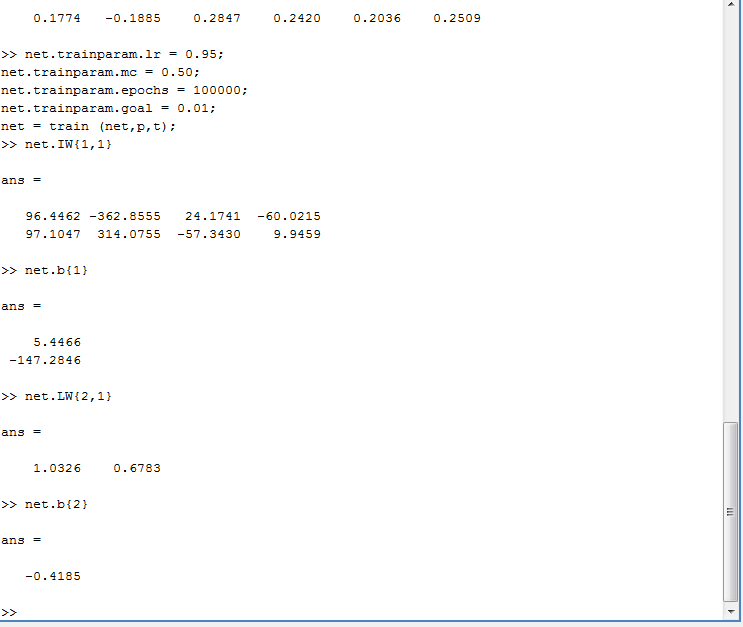
1. Setelah masukan nilai input dan target, selanjutnya masukan beberapa *codding* untuk mendapatkan nilai dari *bias input* ke *hidden* dan *nilai* *bias* dari *hidden* ke *output*, seperti gambar dibawah ini :



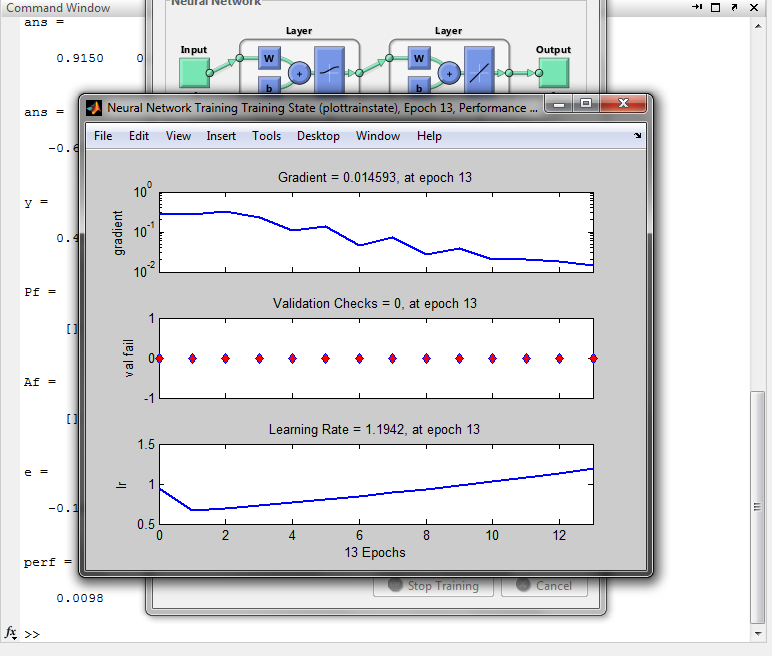
**Gambar 5.13 Tampilan Mencari Nilai Bias**

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***

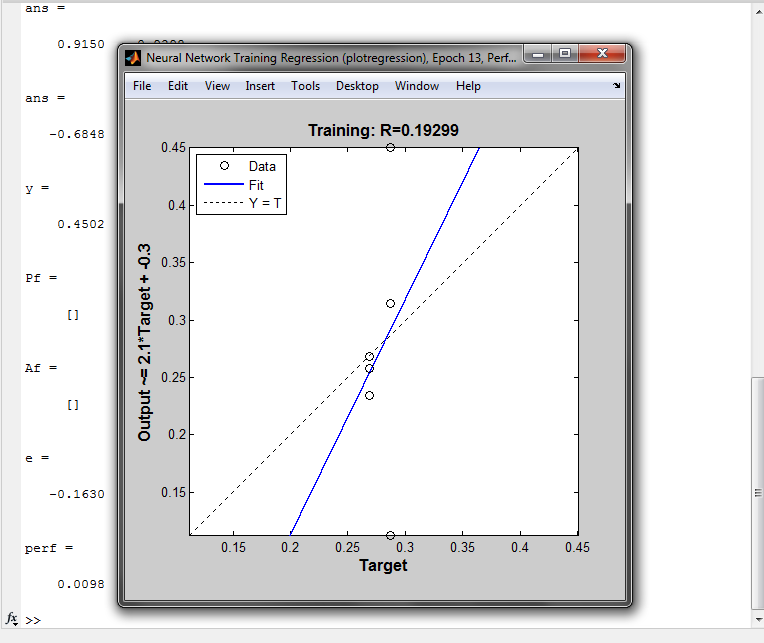
1. Setelah melakukan pengujian untuk mendaptkan *epoch*, perlu melakukan lagi pencarian nilai *bias* agar mendapatkan nilai *bias* akurat , seperti gambar dibawah ini :

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

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

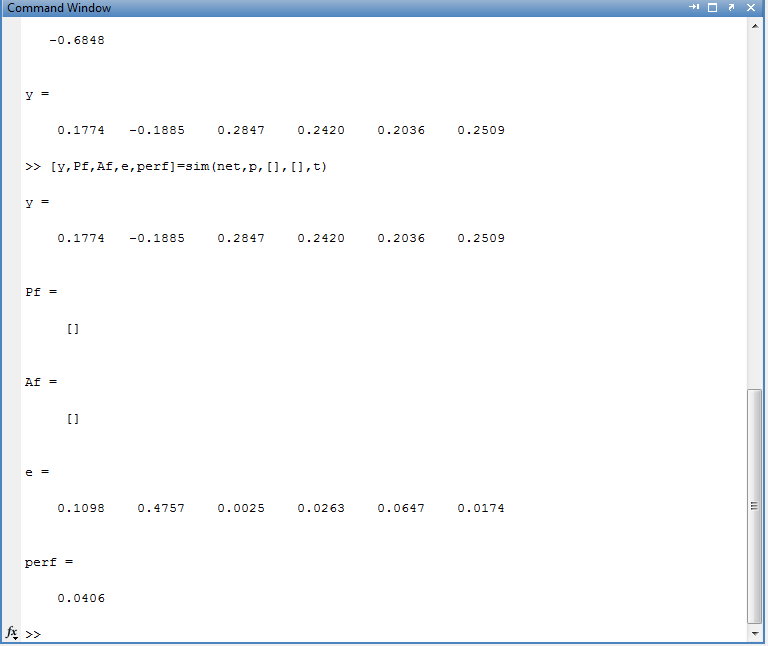
**Gambar 5.16 Tampilan *Tranning State PlottrainState***

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.



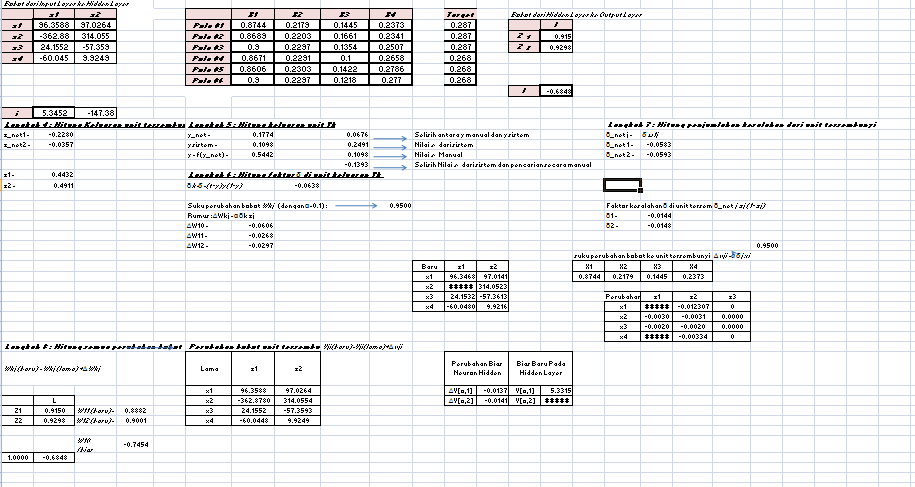
**Gambar 5.18 Tampilan Nilai Y dan E dalam Matlab**

* + 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**

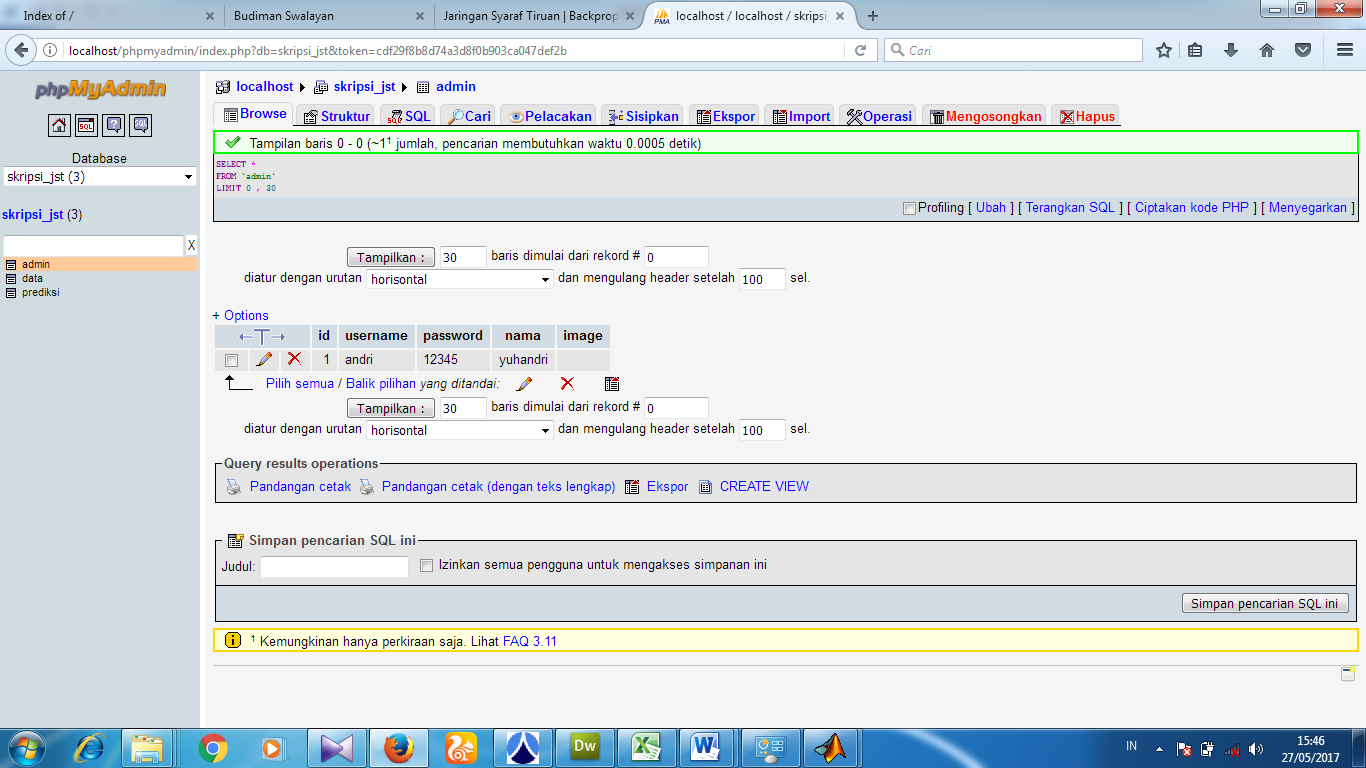
Pada pengujian interface ini, dilihat apakah interface sistem sudah memenuhi syarat sebagai interface yang *userfriendly* bagi pengguna nantinya. Pengujian interface dilakukan agar sistem berbasis website mudah digunakan oleh pengguna nantinya.

* + - 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*

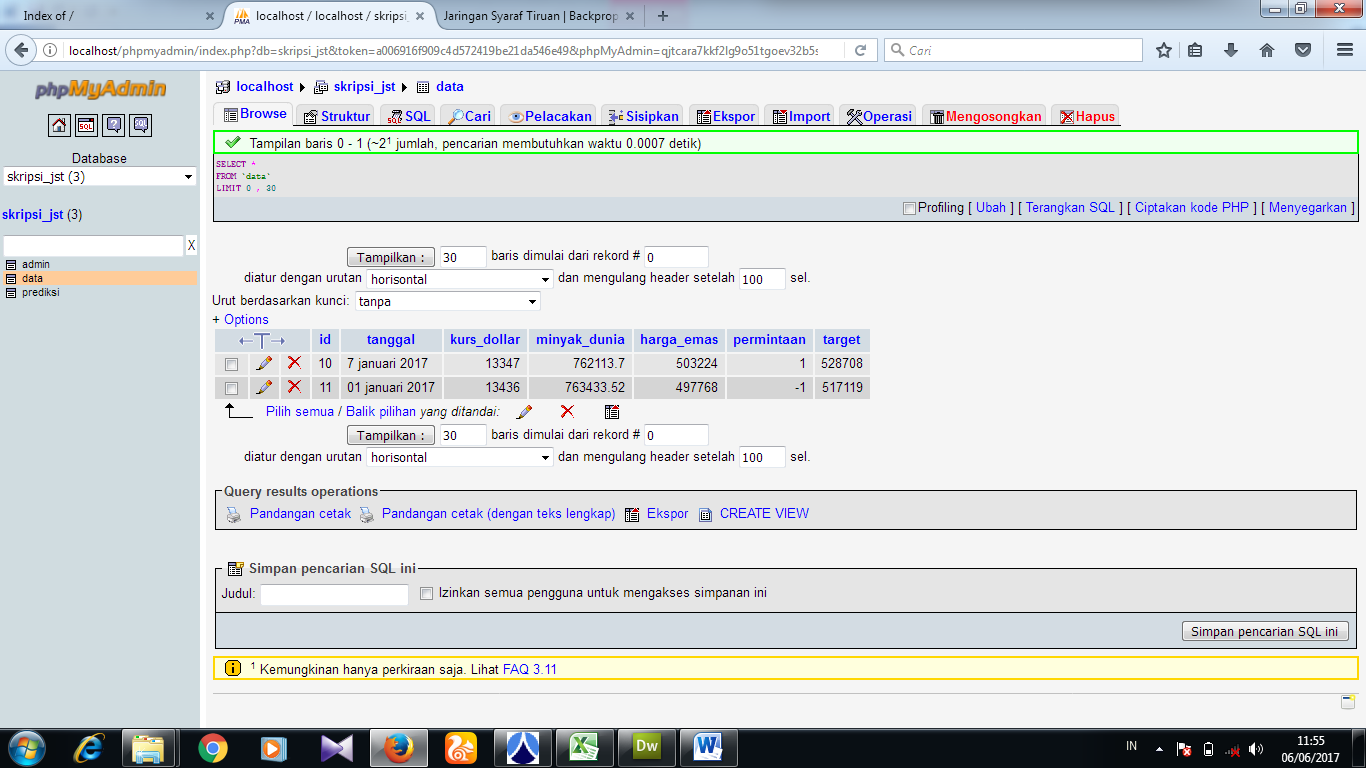
Tampilan *table manager* berfungsi sebagai table yang menyimpan secara keseluruhan biodata yang sudah di inputkan oleh *admin*.



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

1. *Tampilan Tabel Data*

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



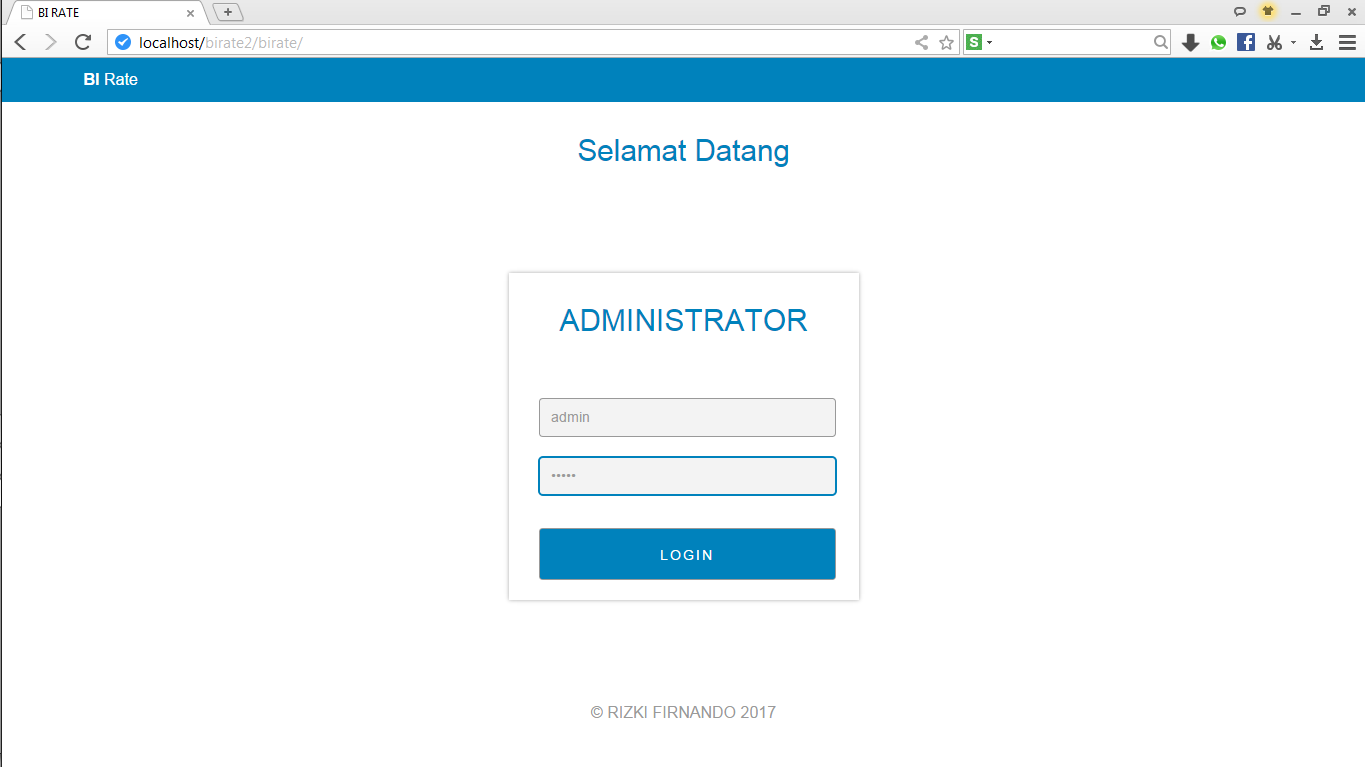
**Gambar 5.21 Halaman Tabel *Data Database***

* + - 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*

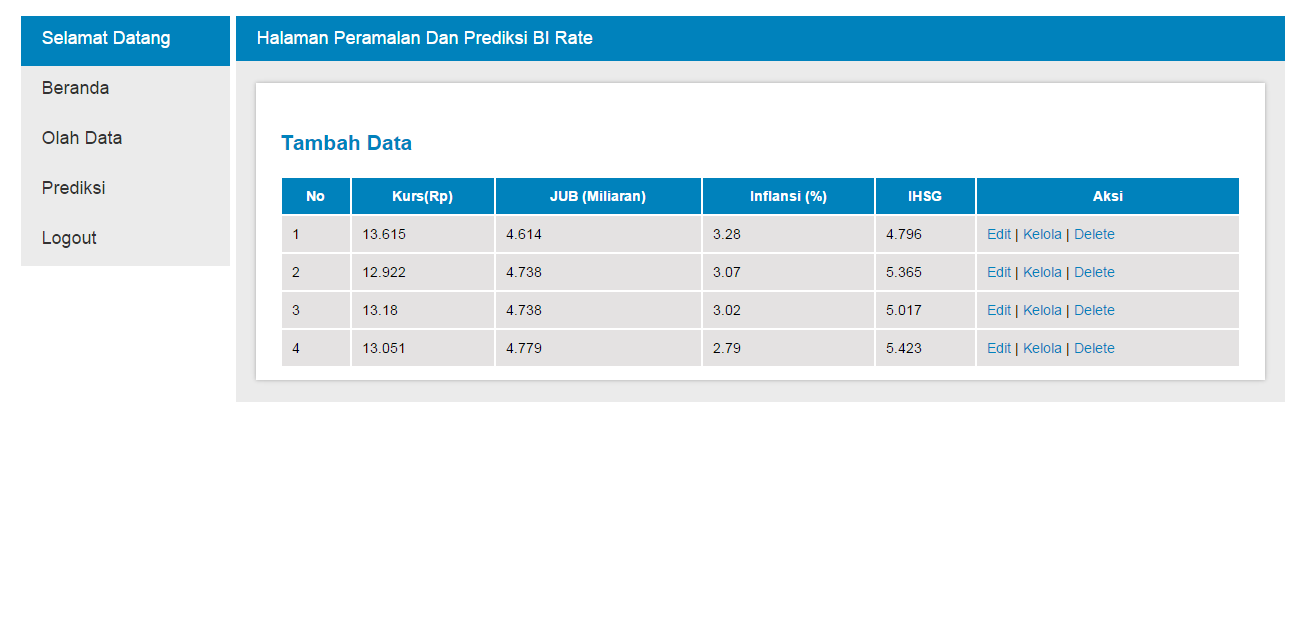
Tampilan *Home* merupakan tampilan awal pada *form menu* utama. Pada tampilan menu utama atau *home* ini terdapat beberapa menu yang dapat diakses oleh Admin Tampilan *form menu* utama dapat dilihat pada Gambar dibawah ini :



**Gambar 5.23 Halaman *Home Website***

* + - * 1. *Tampilan Olah Data*

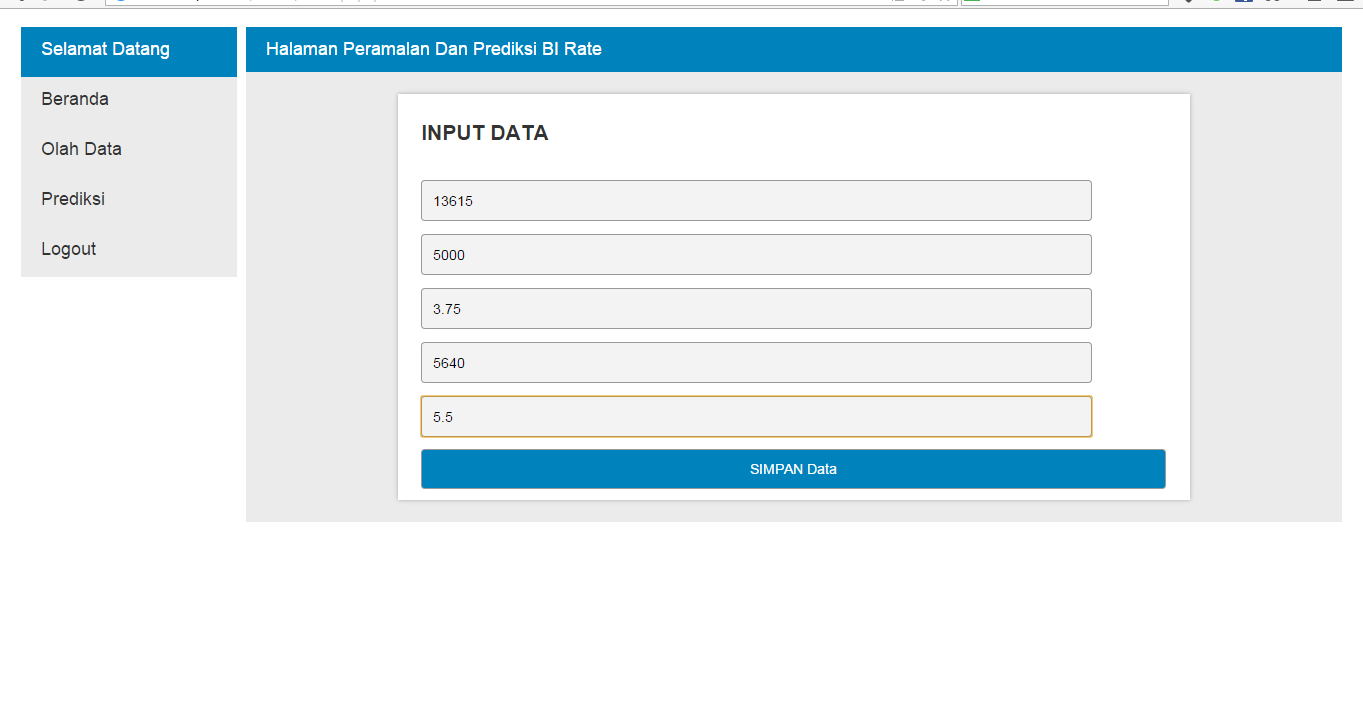
Pada tampilan ini adalah semua data yang ada di system dan admin dapat menambah data pada sistem. Seperti gambar berikut :



**Gambar 5.24 Tampilan Olah Data**

* + - * 1. *Tampilan Tambah Data*

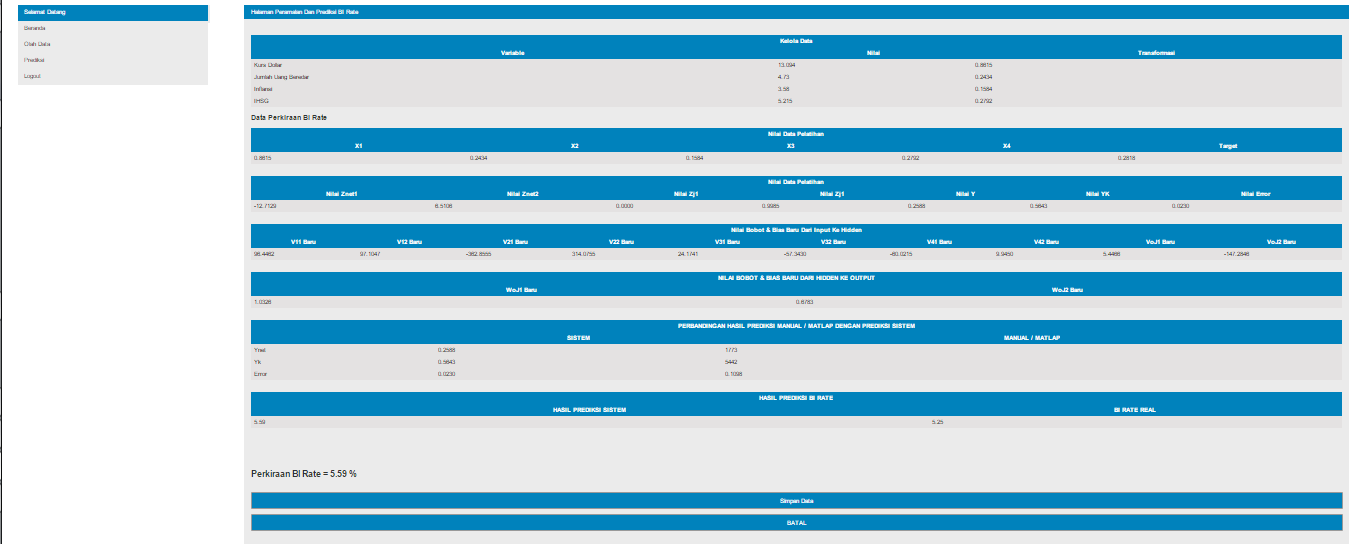
Tampilan input new data merupakan halaman untuk admin mnambahkan Data untuk kebutuhan memprediksi.



**Gambar 5.25 Tampilan Form Input New Data**

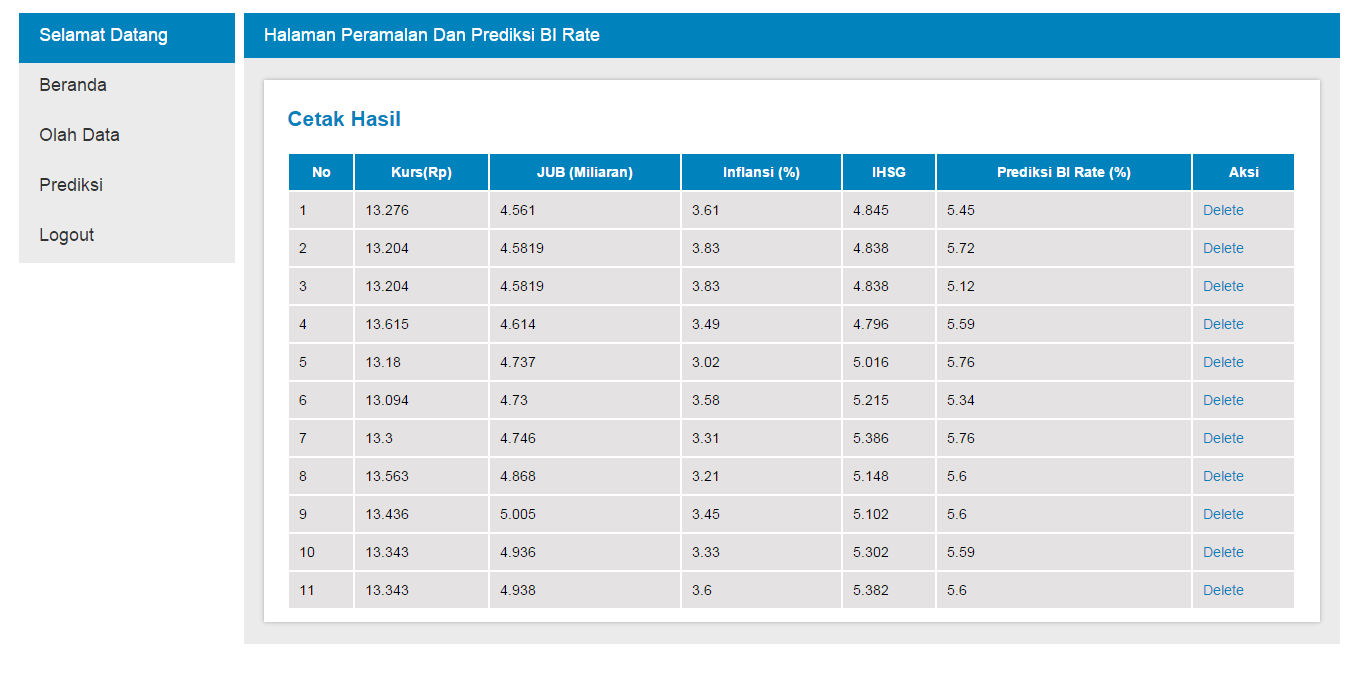
* + - * 1. *Tampilan Kelola Data dan Proses Prediksi*

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*

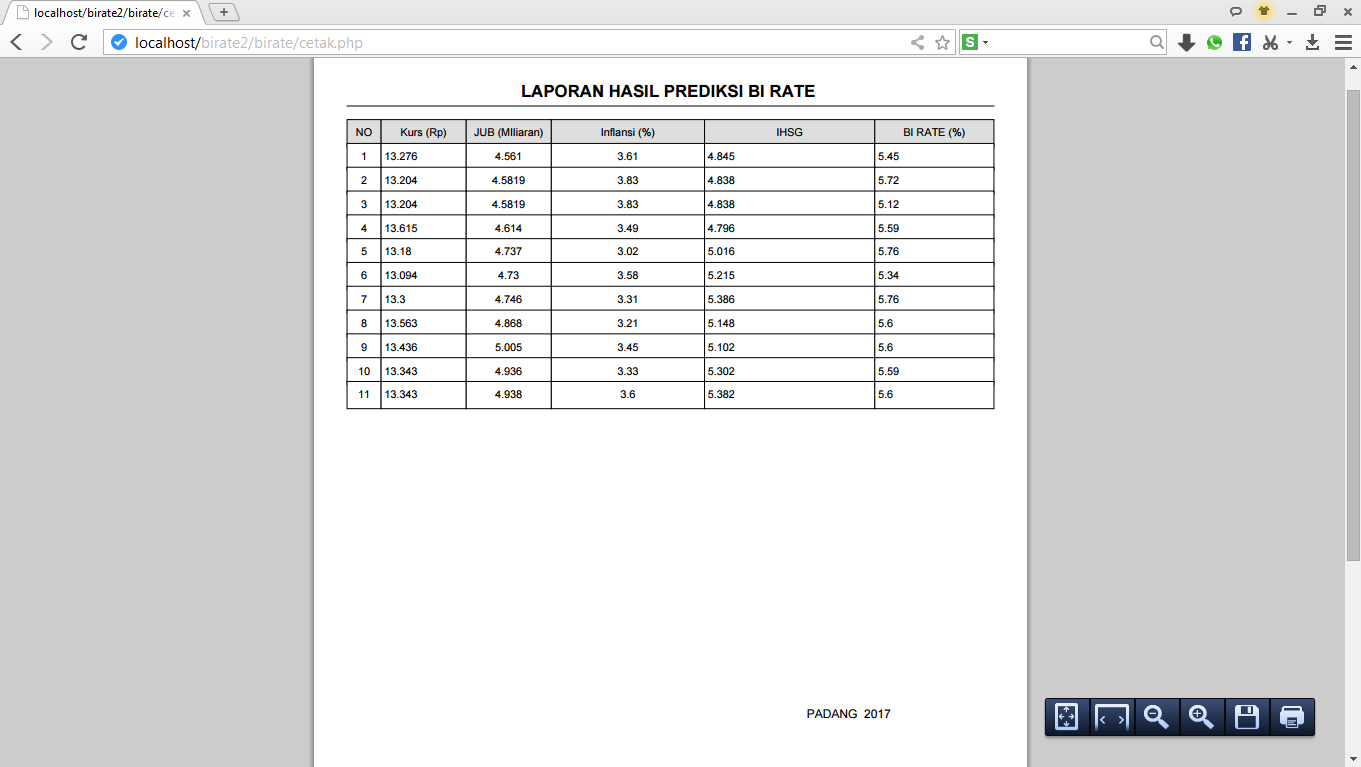
Tampilan data prediksi merupakan halaman data yang sudah ada hasil prediksi *BI Rate*.



**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**.

**6.1 Kesimpulan**

Dari serangkaian proses penelitian yang dilakukan penulis dalam topik penelitian ini, maka penulis dapat menarik kesimpulan bahwa:

1. Dengan merancang Jaringan Syaraf Tiruan dapat digunakan untuk melakukan proses prediksi harga *bitcoin*.
2. Dengan membangun Jaringan Syaraf Tiruan dengan menggunakan metode *backpropagation* dapat digunakan untuk memprediksi harga *bitcoin*.
3. Dengan pengimplementasian sistem aplikasi Jaringan Syaraf Tiruan berbasis web dapat membantu para *trader bitcoin* untuk mendapatkan informasi mengenai harga *bitcoin* untuk satu jam kedepan.

**6.2 Keterbatasan Sistem**

Sistem aplikasi yang dirancang untuk memprediksi harga *bitcoin*, 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 diimplementasikan untuk memprediksi *bitcoin* dengan menggunakan data 12 jam terakir.
2. Aplikasi Jaringan Syaraf Tiruan dengan metode *Backpropagation* ini menggunakan pola 5-2-1 pada pencarian *matlab*.
3. Aplikasi Jaringan Syaraf Tiruan dengan metode *Backpropagation* ini hanya bisa digunakan pada keadaan pasar yang stabil.

**6.3 Saran**

Dari penelitian yang telah dilakukan tentu masih banyak hal-hal yang masih kurang, maka dari itu terdapat beberapa saran :

1. Penulis mengharapkan adanya pengembangan terhadap aplikasi karena fitur-fitur pada aplikasi masih sangat kurang.
2. Lebih baik perhitungan prediksi dilakukan secara otomatis dan data diambil secara *realtime*, sehingga nantinya *user* tidak perlu lagi melakukan *input* data.
3. Sebaiknya perhitungan Jaringan Syaraf Tiruan juga menggunakan metode selain *backpropogation* untuk perbandingan hasil yang lebih akurat.

DAFTAR PUSTAKA

[1] Deyi Li and Yi Du, 2017, Artificial Intelligence With Uncertainty Second Edition, Tsinghua University Beijing, China

[2] Dahriani Hakim Tanjung. 2015. Jaringan Saraf Tiruan Dengan Backpropagation Untuk Memprediksi Penyakit Asma.Volume 2 No.1 Januari 2015 : 2354-5771.

[3] Ferry Mulyanto. 2015. Pemanfaatan Cryptocurrency Sebagai Penerapan Mata Uang Rupiah Kedalam Bentuk Digital Menggunakan Teknologi Bitcoin. Volume 4 No.4 2015 : 2302-5700.

[4] Dimaz A.W, 2017. Blockchain Dari Bitcoin Untuk Dunia”. Yogyakarta : Jasakom.

[5] Janner Simarmata, 2009.Rekayasa Perangkat Lunak.Yogyakarta : Andi.

[6] S, Rossa A. dan M. Shalahudin 2013. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung: Informatika Bandung.

[7] Jong Jek Siang, 2009.Jaringan Syaraf Tiruan. Yogyakarta : Andi.

[8]Nawi, N.M., Rehman, M.Z., Aziz, M.A., Herawan, T. and Abawajy, J.H., 2014, November. An Accelerated Particle Swarm Optimization Based Levenberg Marquardt Back Propagation Algorithm.

[9] Riedmiller, M. and Braun, H., 1993. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In Neural Networks, 1993., IEEE International Conference On (pp. 586-591). IEEE.

[10] Ganatra, A. (2011). Initial Classification Through Back Propagation In a Neural Network Following Optimization Through GA to Evaluate the Fitness of an Algorithm. International Journal of Computer Science & Information Technology.

[11] Pakaja, Fachrudin, Naba, Agus dan Purwanto, 2012. Peramalan Penjualan Mobil Menggunakan Jaringan Syaraf Tiruan dan Certainty Factor.

[12] https://www.bitcoin.ac.id/ Bitcoin Data 26 april 2018