

**PENERAPAN NEURAL NETWORK TIME SERIES UNTUK PREDIKSI
SIRKULASI CONTAINER MASUK DAN KELUAR DI
PT. PRIMA MANGGALINDO**



Nama : DAFIQUS SYARIF

NIM : 00.41010.0194

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA**

2007

**PENERAPAN NEURAL NETWORK TIME SERIES UNTUK
PREDIKSI SIRKULASI CONTAINER MASUK DAN KELUAR DI
PT. PRIMA MANGGALINDO**

SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer



UNIVERSITAS

Dinamika

Oleh :

Nama : Dafiqus Syarif

NIM : 00.41010.0194

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

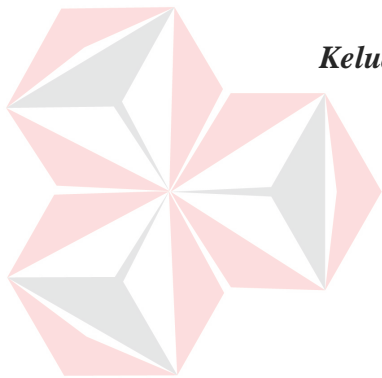
**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA**

2007



*Tuhan tidak akan merubah nasib seseorang,
jika orang tersebut tidak mau berusaha
dan berdoa*

UNIVERSITAS
Dinamika



Kupersembahkan kepada:

Keluarga Besarku yang Berjasa Besar dalam Pendidikanku,

Ayahanda & Ibunda tercinta

Kakak dan Adiku tercinta

Capoeira de Soerabaya

Almamaterku tercinta, STIKOM Surabaya

UNIVERSITAS
Dinamika

**PENERAPAN NEURAL NETWORK TIME SERIES UNTUK PREDIKSI
SIRKULASI CONTAINER MASUK DAN KELUAR DI
PT. PRIMA MANGGALINDO**

Disusun Oleh:

Nama : Dafiqus Syarif

NIM : 00.41010.0194

Surabaya, Juli 2007

Telah diperiksa, diuji dan disetujui :

Pembimbing I

Pembimbing II

Maria Irmina P.,S.Kom.MT

NIDN : 0725057201

Romeo, ST

NIDN : 0705087301

Mengetahui :

Wakil Ketua Bidang Akademik

Drs. Antok Supriyanto, M.MT

NIDN. 0726106201

ABSTRAKSI

Kecenderungan memilih cara yang instan dan otomatis dalam pemenuhan kebutuhan membuat manusia berlomba-lomba menciptakan inovasi baru. Kemajuan teknologi informasi yang telah dicapai manusia memungkinkan untuk bisa diimplementasikan untuk berbagai kebutuhan. Salah satunya dengan Pembuatan Aplikasi Penerapan Neural Network Time Series untuk Prediksi Sirkulasi Container Masuk dan Keluar.

Aplikasi penerapan prediksi sirkulasi container masuk dan keluar yang disajikan secara grafis dan interaktif. Sirkulasi container di tiap waktu adalah tidak tetap. Dibutuhkan suatu metode yang dapat menggali data-data sirkulasi container ini dan menemukan pola-pola yang ada, sehingga dapat diprediksikan.

Backpropagation Neural Network yang memiliki kemampuan untuk belajar sendiri dari data-data yang diumpangkan kepadanya dan membentuk pola-pola terhadap data-data tersebut, merupakan pilihan solusi yang akan digunakan.

Karena sirkulasi *container* yang masuk dan keluar depo juga sangat bergantung pada waktu, sehingga dibutuhkan prediksi yang mencakup deret waktu (*time series*), maka dibutuhkan pengembangan lebih lanjut dari *Backpropagation Neural Network*, yaitu *Neural Network Time Series*.

Dalam Tugas Akhir ini, implementasi *Neural Network* metode *Backpropagation* digunakan untuk menghasilkan konstanta inputan. Dengan pemanfaatan metode ini dapat dilihat dari hasil perhitungan prediksi selisih *neural network* = 0.

KATA PENGANTAR

Puji syukur kepada Allah SWT, hanya berkat rahmat dan hidayah-Nya penulis akhirnya dapat menyelesaikan laporan Tugas Akhir ini.

Dalam penyusunan laporan Tugas Akhir ini penulis mengambil judul PENERAPAN NEURAL NETWORK TIME SERIES UNTUK PREDIKSI SIRKULASI CONTAINER MASUK DAN KELUAR DI PT. PRIMA MANGGALINDO

Dalam kesempatan ini penulis menyampaikan terima kasih kepada:

1. Bapak Romeo, ST selaku dosen pembimbing II atas segala bimbingan dan arahan yang diberikan dalam menyelesaikan Tugas Akhir ini.
2. Ibu Maria Irmina P.,S.Kom,MT selaku dosen pembimbing I atas segala bimbingan dan arahan yang diberikan dalam menyelesaikan Tugas Akhir ini.
3. Bapak Drs.H. Syarful Mudawam dan Ibu Hj. Nurus Su'dah serta kakak dan adik untuk semangat, doa, dan perhatiannya.
4. Mahasiswa STIKOM Angkatan 2000, untuk dukungan dan bantuan.
5. Teman-teman Capoeira de Soerabaya untuk dukungan dan bantuan.
6. Prima Sita W. yang telah banyak memberikan dukungan, motivasi dan pinjaman buku kepada penulis.
7. Semua pihak yang telah membantu penulis yang tidak dapat penulis sebutkan satu persatu.

Ucapan terima kasih juga penulis sampaikan untuk semua pihak yang tidak dapat penulis sebutkan satu per satu dalam tulisan ini.

Penulis sadar bahwa masih banyak kekurangan dalam penulisan laporan Tugas Akhir ini. Untuk itu saran dan kritik yang sifatnya membangun sangat diharapkan demi perbaikan laporan Tugas Akhir ini sehingga dapat bermanfaat bagi pembaca semua.

Surabaya, Juli 2007

Penulis



UNIVERSITAS
Dinamika

DAFTAR TABEL

	Halaman
Tabel 3.1 Neural Network.....	36
Tabel 3.2 Bobot NN	37
Tabel 3.3 Grup Output	37
Tabel 3.4 Obyek.....	38
Tabel 3.5 Grup Input	38
Tabel 3.6 Nilai Obyek.....	39
Tabel 3.7 Contoh Data	45
Tabel 3.8 Uji Coba Mengolah Data	45
Tabel 3.9 Detil Nilai Data	46
Tabel 3.10 Uji Coba Mengolah Detil Nilai Data	48
Tabel 3.11 Data Konfigurasi Neural Network	48
Tabel 3.12 Uji Coba Mengolah Data Neural Network	50
Tabel 3.13 Data Konfigurasi Neural Network yang Dipakai.....	51
Tabel 3.14 Uji Coba Melatih Data	52
Tabel 3.15 Data Konfigurasi Neural Network yang Dipakai.....	52
Tabel 3.16 Uji Coba Prediksi.....	53
Tabel 4.1 Contoh Data	63
Tabel 4.2 Uji Coba Mengolah Data	63
Tabel 4.3 Detil Nilai Data	64
Tabel 4.4 Uji Coba Mengolah Detil Nilai Data	66
Tabel 4.5 Data Konfigurasi Neural Network	67
Tabel 4.6 Uji coba mengolah data <i>neural network</i>	68

Tabel 4.7 Data Konfigurasi Neural Network yang Dipakai.....	69
Tabel 4.8 Uji Coba Melatih Data	70
Tabel 4.9 Data Konfigurasi Neural Network yang Dipakai.....	71
Tabel 4.10 Uji coba Prediksi.....	72

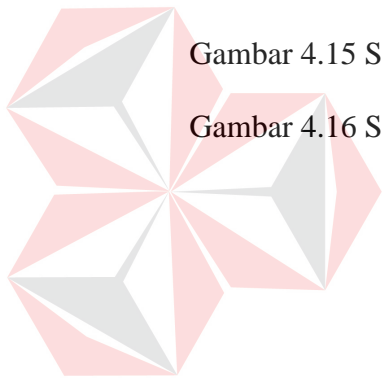


UNIVERSITAS
Dinamika

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Struktur sel syaraf biologis.....	6
Gambar 2.2 Model tiruan neuron.....	7
Gambar 2.3 Grafik fungsi identity	8
Gambar 2.4 Jaringan syaraf umpan maju dengan dua lapisan sel hidden.....	11
Gambar 2.5 Backpropagation neural network dengan satu hidden layer.....	12
Gambar 2.6 Jaringan arus balik menurut Elman.....	19
Gambar 3.1 Sistem flow training backpropagation neural network	26
Gambar 3.2 Sistem flow testing backpropagation neural network	28
Gambar 3.3 Diagram alur sistem neural network time series	31
Gambar 3.4 Context diagram	32
Gambar 3.5 DFD level 0.....	33
Gambar 3.6 Entity relational diagram conceptual data model.....	34
Gambar 3.7 Entity relational diagram physical data model.....	35
Gambar 3.8 Rancangan form utama.....	40
Gambar 3.9 Rancangan form master data.....	41
Gambar 3.10 Rancangan form konfigurasi neural network.....	42
Gambar 3.11 Rancangan form pelatihan.....	43
Gambar 3.12 Rancangan form prediksi	44
Gambar 4.1 Tampilan menu utama.....	56
Gambar 4.2 Tampilan menu	56
Gambar 4.3 Tampilan menu about.....	57
Gambar 4.4 Tampilan menu setup database awal.....	57

Gambar 4.5 Tampilan menu setup database baru	58
Gambar 4.6 Tampilan <i>neural network</i>	58
Gambar 4.7 Tampilan data.....	59
Gambar 4.8 Tampilan konfigurasi <i>neural network</i> (NN).....	59
Gambar 4.9 Tampilan pelatihan <i>neural network</i>	60
Gambar 4.10 Tampilan grafik prediksi <i>neural network</i>	61
Gambar 4.11 Tampilan data detil prediksi <i>neural network</i>	61
Gambar 4.12 Tampilan hasil prediksi <i>neural network</i>	62
Gambar 4.13 Setup Data Berhasil.....	64
Gambar 4.14 Setup Data Berhasil.....	66
Gambar 4.15 Setup Data Berhasil.....	69
Gambar 4.16 Setup Data Berhasil.....	71



DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Listing Program	76
Lampiran 2 Biodata Penulis	131



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

PT Prima Manggalindo merupakan perusahaan yang bergerak di bidang jasa yang menyediakan tempat penyimpanan *container* sementara, sampai digunakan kembali, atau lebih dikenal dengan Depo *Container*. Pengaturan *container* saat penyimpanan pada dasarnya menggunakan model *First In First Out* (FIFO), namun pada praktiknya hal ini sangat susah untuk dilakukan, karena sirkulasi masuk dan keluar *container* yang cukup cepat, dimana *container* dalam penyimpanannya harus dibedakan berdasarkan pemiliknya (*principal*). Oleh sebab itu prediksi sirkulasi *container* per *principal* yang masuk dan keluar depo merupakan salah satu informasi yang sangat berguna dalam pengaturan *container* ini.

Sirkulasi *container* di tiap waktu adalah tidak tetap. Dibutuhkan suatu metode yang dapat menggali data-data sirkulasi *container* ini dan menemukan pola-pola yang ada, sehingga dapat diprediksikan. *Backpropagation Neural Network* yang memiliki kemampuan untuk belajar sendiri dari data-data yang diumpankan kepadanya dan membentuk pola-pola terhadap data-data tersebut, merupakan pilihan solusi yang akan digunakan. Karena sirkulasi *container* yang masuk dan keluar depo juga sangat bergantung pada waktu, sehingga dibutuhkan prediksi yang mencakup deret waktu (*time series*), maka dibutuhkan pengembangan lebih lanjut dari *Backpropagation Neural Network*, yaitu *Neural Network Time Series* (Weigend, dkk, 1990).

Diharapkan model prediksi dengan menggunakan *Neural Network Time Series* ini dapat menjadi model solusi alternatif untuk memecahkan permasalahan dalam memprediksikan sirkulasi container yang masuk dan keluar depo. Dan dengan kemampuan belajarnya, memungkinkan *Neural Network Time Series* dapat lebih adaptif dalam mengakomodasi perubahan-perubahan yang terjadi di keadaan sebenarnya, yang semakin lama semakin cepat dan berfluktuasi.

1.2 Perumusan Masalah

Berdasarkan pada latar belakang diatas maka permasalahan yang akan dibahas dalam Tugas Akhir ini adalah : "Bagaimana menerapkan metode Neural Network Time Series untuk memprediksikan jumlah container yang keluar".

1.3 Pembatasan Masalah

Mengingat luasnya permasalahan dalam sistem ini, maka perlu diberikan batasan untuk lebih memperjelas ruang lingkup permasalahan yang akan dibahas dalam penelitian Tugas Akhir ini. Batasan-batasan masalah tersebut meliputi :

1. Sistem tidak menangani pengolahan data transaksi depo, seperti menambah data, memperbaharui atau menghapus data.
2. Sistem *database* yang dapat digunakan untuk pengambilan data adalah Microsoft SQL Server 2000.
3. Bahasa pemrograman yang digunakan adalah *Microsoft Visual Basic 6.0*.
4. Algoritma *neural network time series* yang digunakan adalah *multilayered perceptron* atau *Back-Propagation*.
5. Data telah siap untuk digali atau telah melalui tahapan proses sebelumnya.

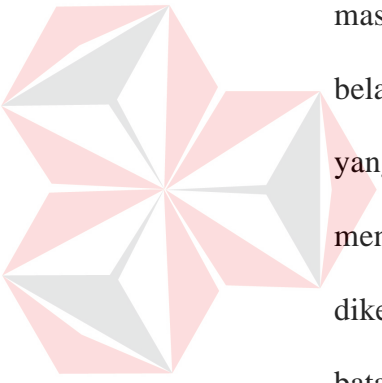
1.4 Tujuan

Adapun tujuan dari penelitian Tugas Akhir ini adalah mengembangkan sistem aplikasi untuk memprediksikan jumlah container per principal yang keluar dari depo.

1.5 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut :

BAB I : PENDAHULUAN



Dalam bab ini diuraikan mengenai latar belakang masalah, perumusan masalah, pembatasan masalah, tujuan dan sistematika penulisan. Latar belakang menjelaskan bagaimana masalah dapat terjadi dan solusi yang diberikan atas masalah tersebut. Perumusan masalah menjelaskan mengenai permasalahan apa yang akan dibahas dan dikerjakan dalam Tugas Akhir. Pembatasan masalah menjelaskan batasan-batasan dari sistem yang dibuat sehingga tidak keluar dari ketentuan yang ditetapkan. Untuk tujuan penelitian berupa harapan dari hasil yang akan dicapai. Untuk sistematika penulisan menjelaskan pembagian dari bab-bab yang ada dalam Tugas Akhir.

BAB II : LANDASAN TEORI

Dalam bab ini diuraikan tentang teori-teori yang berkaitan dalam penyelesaian tugas akhir yang meliputi: Jaringan Saraf Tiruan (*Neural Network*) yang menggunakan algoritma *Backpropagation* dengan metode *Neural Network Time Series*.

BAB III : METODE PENELITIAN/PERANCANGAN SISTEM

Dalam bab ini dibahas tentang langkah-langkah dalam perancangan sistemnya yang berisi analisis perancangan sistem dengan menggunakan diagram alur sistem penyelesaian masalah, diagram penelitian, HIPO (*Hirarki Input Process Output*), *Data Flow Diagram* (DFD) dan *Entity Relationship Diagram* (ERD).

BAB IV : IMPLEMENTASI DAN EVALUASI

Dalam bab ini dibahas tentang Implementasi dan evaluasi dari Tugas Akhir tersebut meliputi instalasi program, implementasi program, dan evaluasi dari implementasi program yang telah dibuat.

BAB V : PENUTUP

Dalam bab ini berisi tentang kesimpulan yang berdasarkan dari hasil akhir aplikasi dan saran yang dirumuskan selama proses penyelesaian Tugas Akhir termasuk saran yang diharapkan untuk pengembangan hasil penelitian ini.



UNIVERSITAS
Dinamika

BAB II

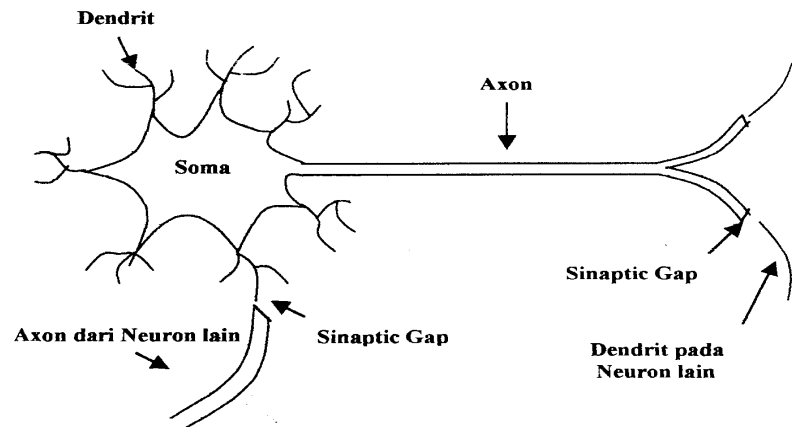
LANDASAN TEORI

2.1 Jaringan Syaraf Tiruan

Menurut Pandjaitan (2007:11) salah satu definisi Jaringan Syaraf Tiruan (JST) adalah suatu teknologi komputasi yang berbasis hanya pada model syaraf biologis dan mencoba mensimulasikan tingkah laku dan kerja model syaraf terhadap berbagai macam model masukan. Sebagai suatu teknologi komputasi, JST merupakan satu teknik pemrosesan informasi yang menggunakan model kuantitatif. Syaraf biologis mengilhami terciptanya suatu proses komputasi yang identik dengan kerja neuron dalam sistem syaraf manusia. Seperti halnya jaringan biologis, model matematik JST menghubungkan sejumlah masukan dan keluaran suatu sistem secara adaptif yang diorganisasikan dalam lapisan elemen pemroses seperti layaknya hubungan antar neuron syaraf biologis.

2.1.1 Struktur sel syaraf biologis

Jaringan saraf manusia terdiri atas sel-sel yang disebut *neuron*. Menurut Setiawan (2003:2) ada tiga komponen utama neuron yang fungsinya dapat dianalogikan dengan yang terjadi pada JST, yaitu *dendrit*, *soma* dan *axon*. *Dendrit* akan menerima sinyal-sinyal dari *neuron* lain. Sinyal tersebut merupakan *impuls* listrik yang ditransmisikan melalui *sinaptic gap* melalui proses kimia. Sedangkan *soma* atau badan sel akan menjumlah sinyal-sinyal input yang masuk. Jika ada input yang masuk, sel akan aktif dan akan mentransmisikan sinyal ke sel lain melalui *axon* dan *sinaptic gap*. Untuk lebih jelasnya struktur sel syaraf biologis dapat dilihat pada gambar 2.1.

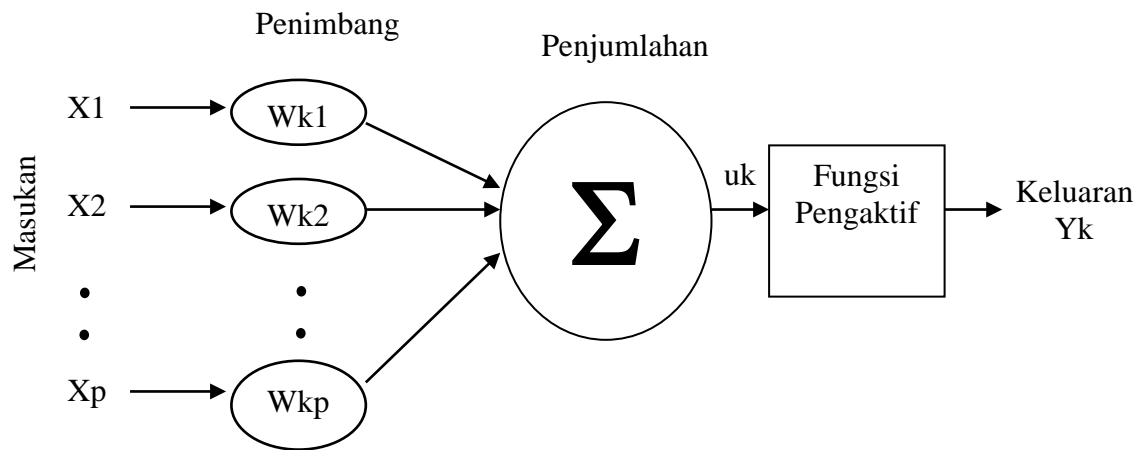


Gambar 2.1 Struktur sel syaraf biologis

2.1.2 Struktur jaringan syaraf tiruan

Menurut Fu (1994:18) Jaringan Saraf Tiruan (*Neural Network*) mempunyai arsitektur distribusi paralel yang terdiri dari banyak *node* dan penghubung. Antara titik penghubung dari satu *node* ke lainnya dihubungkan dengan sebuah bobot.

Tiruan neuron dalam struktur JST adalah elemen pemroses seperti pada gambar 2.2 yang dapat berfungsi seperti halnya sebuah neuron (Purnomo & Kurniawan, 2006:14). Dengan serangkaian inputan di luar sistem yang diberikan kepadanya jaringan ini dapat memodifikasi bobot yang akan dihasilkannya, sehingga akan menghasilkan output yang konsisten sesuai dengan input yang diberikan kepadanya. Setiap elemen pemroses melaksanakan operasi matematika yang sudah ditentukan dan menghasilkan (hanya) sebuah harga keluaran dari satu ataupun banyak masukan.



Gambar 2.2 Model tiruan *neuron*

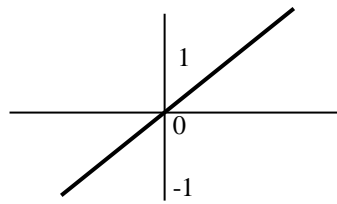
Sebuah pemodelan neuron memiliki masukan X_p sebanyak p , yang berasal dari sel lain atau dari masukan luar (bukan dari neuron). Selanjutnya setiap masukan diberi pembobot W_{kp} . Masing – masing masukan X_p akan dikalikan dengan pembobot W_k yang berkesesuaian. Untuk semua hasil perkalian akan dijumlahkan sebagaimana pada persamaan dibawah ini:

$$u_k = \sum_{j=1}^p W_{kj} X_j \quad (2.1)$$

dan hasil persamaan tersebut akan menjadi masukan bagi fungsi aktivasi untuk mendapatkan tingkat derajat sinyal keluaran pada neuron. Untuk jenis fungsi aktivasi *identity* dapat dideskripsikan dengan persamaan:

$$Y_k = f(u) \quad (2.2)$$

dengan grafik yang ditunjukkan sebagai berikut :



Gambar 2.3 Grafik fungsi *identity*

Pada umumnya sinyal fungsi aktivasi yang dikeluarkan tiap *neuron* berbeda, hal ini dikarenakan berbedanya nilai bobot yang diterima tiap *neuron* berbeda.

Jaringan syaraf dapat diklasifikasikan menjadi dua jenis, yaitu *single layer* dan *multi layer* (Setiawan, 2003:5). Secara umum, tiap unit pada lapisan (*Layer*) yang sama atau dapat kita sebut *neuron* mempunyai tingkah laku yang sama untuk pemrosesan sinyal data. Hanya hal terpenting yang perlu diperhatikan adalah penentuan penggunaan jenis fungsi *aktivasi* pada masing-masing unit pada lapisan tersebut dan pola koneksi pembobot antar lapisan. Namun biasanya unit pada lapisan yang sama mempunyai jenis fungsi *aktivasi* yang sama dan pola koneksi pembobot yang sama pula.

Untuk pemilihan jumlah *layer* bukan berarti pemilihan *layer* untuk *neuron*, namun pemilihan *layer* untuk penghubung jalur pembobot antar *neuron*. Jadi variabel terpenting untuk pengenalan pola adalah pembobotnya.

Fungsi aktivasi yang digunakan untuk melakukan proses pelatihan atau prediksi antara lain :

a. Fungsi aktivasi *identity*

Merupakan sebuah jaringan *single layer* yang sering digunakan dalam sebuah fungsi guna penyesuaian jaringan *input*, dimana berisi variabel nilai yang

terdapat pada sebuah unit *output* yang berbentuk sebuah *binary* (1 atau 0) atau *bipolar* (1 atau -1). *Identity* dapat dirumuskan sebagai berikut :

$$f(x) = x$$

dimana : x berisi nilai *input*

b. Fungsi aktivasi *binary sigmoid*

Merupakan salah satu fungsi *sigmoid* yang dapat diskalakan untuk beberapa *range* dari nilai-nilai yang sesuai dari sebuah permasalahan (inputan), *range* yang paling umum digunakan adalah *range* dari -1 sampai 1, dapat dirumuskan sebagai berikut :

$$f(x) = \frac{1}{1 + \exp(-\sigma x)}$$

$$f'(x) = \sigma f(x)[1-f(x)]$$

c. Fungsi aktivasi *bipolar sigmoid*

Merupakan sebuah relasi yang tertutup untuk sebuah fungsi *hiperbolic tangent*, dimana seringkali digunakan sebagai fungsi aktivasi ketika menginginkan sebuah nilai-nilai dengan *range output* antara -1 sampai 1, dirumuskan sebagai berikut :

$$g(x) = 2f(x) - 1 = \frac{2}{1 + \exp(-\sigma x)}$$

$$= \frac{1 - \exp(-\sigma x)}{1 + \exp(-\sigma x)}$$

d. Fungsi aktivasi *hyperbolic tangent*

Secara umum mengacu pada fungsi *signum*. Untuk sebuah *sigmoid* kita mungkin menggunakan fungsi *hyperbolic tangent*, yang didefinisikan sebagai berikut :

$$\varphi(v) = \tanh\left(\frac{v}{2}\right) = \frac{1 - \exp(-v)}{1 + \exp(-v)}$$

Memperbolehkan fungsi aktivasi dari tipe *sigmoid* untuk mengasumsikan nilai negatif sebagaimana yang digambarkan persamaan diatas mempunyai keuntungan analitik. Lebih jauh, psikologi *neuro* membuktikan bahwa sifat eksperimental (Eekman dan Freema, 1986), secara kasar dengan antisimetris yang sempurna tentang keaslian karakteristik dari fungsi *tangent hiperbolic*.

e. Fungsi aktivasi *gaussian*

Merupakan salah satu fungsi aktivasi dengan sebuah lokal *field* dari suatu respon, juga sering digunakan dalam jaringan *neural bacpropagation*. Respon sama seperti sebuah fungsi nilai positif untuk seluruh nilai-nilai dari x , suatu respon berkurang untuk suatu nilai 0 seperti $|x-c\} \rightarrow \infty$. Fungsi *gaussian* dapat di rumuskan sebagai berikut:

$$f(x) = \exp(-x^2)$$

sehingga diperoleh persamaan :

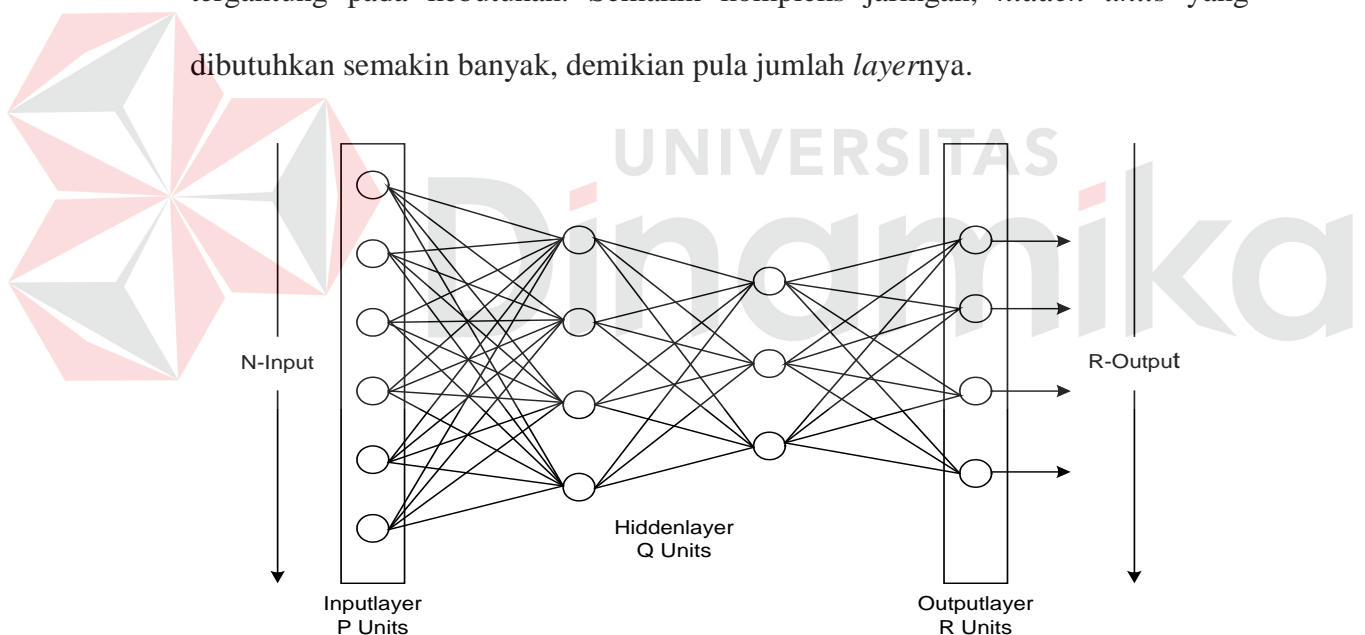
$$f'(x) = -2x \exp(-x^2) = -2xf(x)$$

2.1.3 Single layer

Dalam jaringan syaraf *single layer*, neuron-neuron dapat dikelompokkan menjadi dua bagian, yaitu *input units* dan *output units*. *Input units* menerima masukan dari luar, sedangkan *output units* akan mengeluarkan respon dari jaringan sesuai dengan masukannya.

2.1.4 Multi layer

Dalam jaringan *multi layer*, selain ada *input units* dan *output units* juga terdapat unit-unit yang tersembunyi (*hidden units*). Jumlah *hidden units* tersebut tergantung pada kebutuhan. Semakin kompleks jaringan, *hidden units* yang dibutuhkan semakin banyak, demikian pula jumlah *layernya*.

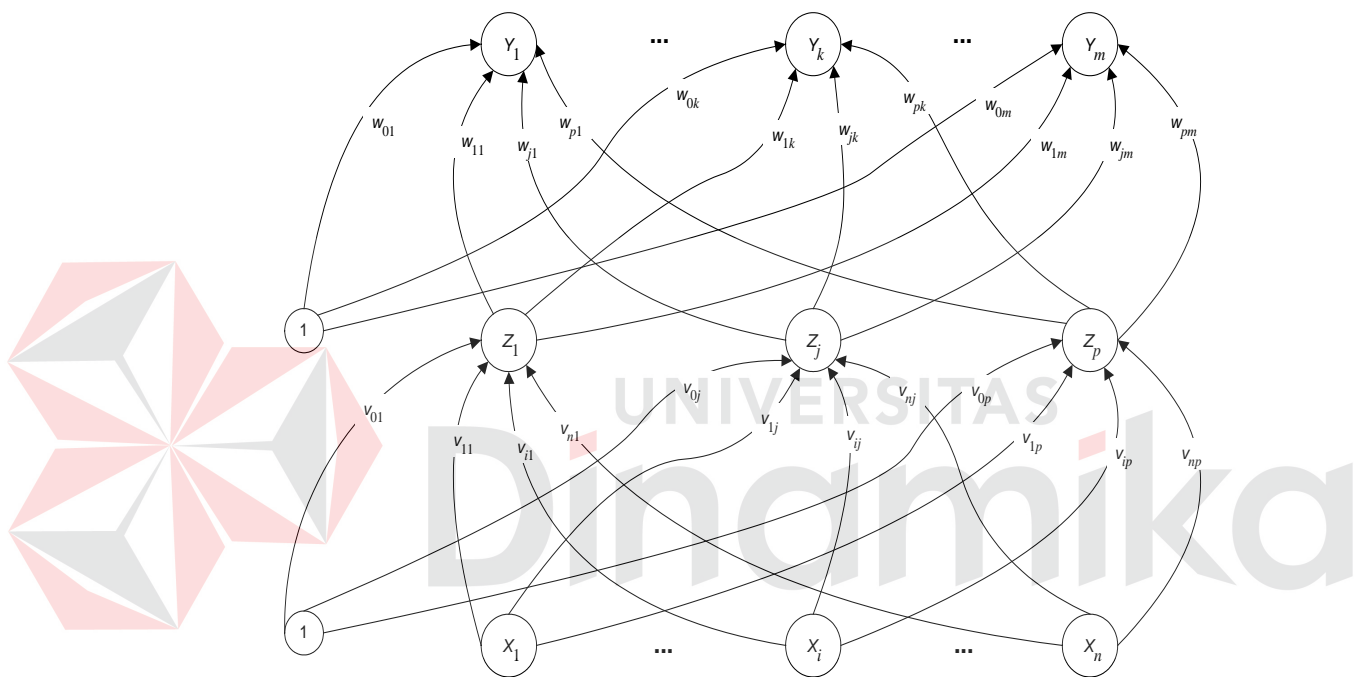


Gambar 2.4 Jaringan syaraf umpan maju dengan dua lapisan sel hidden

2.1.5 Backpropagation Neural Network (BPN)

Jaringan *neural multi layer* dengan satu *hidden layer* (unit Z) ditunjukkan pada gambar 2.5 Unit *output* (unit Y) dan unit *hidden* juga mungkin mempunyai

pembiasan (sebagaimana yang ditunjukkan pada gambar). Bias pada unit *ouput* Y_k dinotasikan dengan w_{0k} ; bias pada unit *hidden* Z_j dinotasikan dengan v_{0j} . Bias ini berindak seperti bobot ada koneksi dari unit yang outputnya selalu satu. (Unit ini ditunjukkan pada gambar 2.5 tapi biasanya tidak ditunjukkan secara eksplisit). Hanya arah dari alur informasi untuk fase operasi *feedforward* yang ditunjukkan. Selama fase pembelajaran, sinyal dikirim dalam arah berlawanan.



Gambar 2.5 *Backpropagation neural network* dengan satu *hidden layer*

Pelatihan *Neural Network* dengan *backpropagation* melibatkan tiga tahap: *feedforward* dari pola pelatihan *input*, *backpropagation* dari *error* yang terhubung, dan penyesuaian bobot (Fausett, 1994:290). Selama *feedforward* tiap *input unit* (X_i) menerima sebuah sinyal *input* dan menyebarkan sinyal ini pada tiap *unit hidden* Z_1, \dots, Z_p . Tiap *unit hidden* kemudian mengkomputasi aktivasi (Y_k) dan mengirimkan sinyal tersebut untuk *output unit*. Tiap *output unit*

mengkomputasi aktivasi dirinya pada bentuk respon dari jaringan yang memberikan pola *input*. Selama pelatihan, tiap *output unit* membandingkan aktivasi komputasi Y_k dengan nilai *target* t_k untuk menentukan *error* yang berhubungan pada pola dengan *unit* tersebut. Berdasarkan *error* ini, faktor δ_k ($k= 1, \dots, m$) dilakukan komputasi. δ_k digunakan untuk mendistribusikan kesalahan pada *output unit* Y_k kembali pada semua *unit* dalam *layer* sebelumnya (*hidden unit* yang berhubungan ke Y_k). Hal ini juga digunakan untuk melakukan *update* bobot diantara *output* dan *hidden layer*. Dalam perilaku yang serupa faktor δ_j ($j = 1, \dots, p$) dikomputasi untuk tiap *hidden unit* Z_j . Adalah tidak diperlukan untuk melakukan pembelajaran terhadap *error* pada *input layer*, tapi δ_j digunakan untuk melakukan *update* bobot diantara *hidden layer* dan *input layer*. Setelah semua faktor δ telah ditentukan bobot semua *layer* disesuaikan secara *simultan*. Penyesuaian terhadap bobot w_{jk} (dari *hidden unit* Z_j pada *output unit* Y_k) didasarkan pada faktor δ_k dan aktivasi z_j dari *hidden unit* Z_j . Penyesuaian pada bobot v_{ij} (dari *input unit* X_i pada *hidden unit* Z_j) didasarkan pada faktor δ_j dan aktivasi x_i dari *input unit*. Tata aturan penulisan yang digunakan dalam algoritma pelatihan untuk jaringan *backpropagation* mengikuti aturan:

- x *Input* vektor pelatihan: $x = (x_1, \dots, x_i, \dots, x_n)$
- t *Output* vektor target: $t = (t_1, \dots, t_i, \dots, t_n)$
- δ_k Bagian dari penyesuaian bobot koreksi *error* untuk w_{jk} yang dikarenakan suatu *error* dari *output unit* Y_k ; jadi informasi tentang *error* di *unit* Y_k yang dikembalikan ke *hidden unit* itu dimasukkan ke dalam *unit* Y_k

δ_j Bagian dari penyesuaian bobot koreksi *error* untuk v_{ij} yang dikarenakan informasi yang salah *backpropagation* dari *output layer* ke *hidden unit* Z_j

α Rata-rata pembelajaran

X_i *Input unit* i : untuk sebuah *input unit*, sinyal *input* dan sinyal *output* adalah sama, penamaan, x_i

v_{oj} Bias pada *hidden unit* j

Z_j *Hidden unit* j : *Input net* ke Z_j dinotasikan dengan z_{in_j} :

$$z_{in_j} = v_{oj} + \sum_{i=1}^r x_i v_{ij} \quad (2.3)$$

Sinyal *output* (aktivasi) dari Z_j dinotasikan dengan z_j :

$$z_j = f(z_{in_j}) \quad (2.4)$$

w_{ok} Bias pada *output unit* k

Y_k *Output unit* k : *Input net* ke Y_k dinotasikan dengan y_{in_k} :

$$y_{in_k} = w_{ok} + \sum_{j=1}^m z_j w_{jk} \quad (2.5)$$

Sinyal *output* (aktivasi) dari Y_k dinotasikan dengan y_k :

$$y_k = f(y_{in_k}) \quad (2.6)$$

Secara lebih rinci algoritma *backpropagation* adalah sebagai berikut:

Step 0. Inisialisasi bobot dan bias

(Set ke nilai *random* kecil)

Step 1. Ketika kondisi berhenti adalah salah, kerjakan langkah 2-9

Step 2. Untuk tiap pasang pelatihan, kerjakan langkah 3-8

Step 3. Tiap *input unit* (X_i , $i = 1, \dots, n$) menerima sinyal *input* x_i dan meneruskan sinyal ini ke semua *unit* pada *layer* yang bersangkutan (*hidden unit*)

Step 4. Tiap *hidden unit* (Z , $j = 1, \dots, p$) dijumlahkan dengan bobot sinyal *input*

$$z_{in_j} = v_{oj} + \sum_{i=1}^r x_i v_{ij}$$

pergunakan fungsi aktivasi ini untuk menghitung sinyal *output*,

$$z_j = f(z_in_j),$$

dan mengirimkan sinyal ini ke seluruh *unit* pada *layer* yang bersangkutan (*output unit*).

Step 5. Tiap *output unit* (Y_k , $k = 1, \dots, m$) dijumlahkan dengan bobot sinyal *input*

$$y_in_k = w_{ok} + \sum_{j=1}^m z_j w_{jk}$$

dan pergunakan fungsi aktivasi ini untuk menghitung sinyal *output*,

$$y_k = f(y_in_k).$$

Error dari *backpropagation*:

$$(t_k - y_k) \tag{2.7}$$

Step 6. Tiap *output unit* (Y_k , $k = 1, \dots, m$) menerima satu pola target yang cocok untuk pola pelatihan *input*, syarat perhitungan informasi *error* ini

$$\delta_k = (t_k - y_k) f'(y_in_k) \tag{2.8}$$

syarat penghitungan bobot koreksi ini (digunakan untuk meng-*update* w_{jk} nantinya),

$$\Delta w_{jk} = \alpha \delta_k z_j \tag{2.9}$$

syarat penghitungan bias koreksi ini (digunakan untuk meng-*update* w_{ok} nantinya),

$$\Delta w_{ok} = \alpha \delta_k \tag{2.10}$$

dan mengirimkan δ_k ke *unit* pada *layer* sebelumnya.

Step 7. Tiap *hidden unit* (Y_k , $k = 1, \dots, p$) menjumlahkan ke *input* delta (dari *unit* pada *layer* sesudahnya)

$$\delta_in_j = \sum_{k=1}^p \delta_k w_{jk} \tag{2.11}$$

kalikan dengan nilai dari fungsi aktivasi untuk menghitungnya syarat informasi *error*,

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (2.12)$$

hitung koreksi bobot syaratnya (nanti digunakan untuk meng-*update* v_{ij})

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.13)$$

dan hitung koreksi bias syaratnya (nanti digunakan untuk meng-*update* v_{oj})

$$\Delta v_{oj} = \alpha \delta_j \quad (2.14)$$

Meng-*update* bobot dan bias:

Step 8. Tiap *output unit* ($Y_k, k = 1, \dots, m$) *update* bias dan bobot ($j = 0, \dots, p$):

$$w_{jk} \text{ (baru)} = w_{jk} \text{ (lama)} + \Delta w_{jk} \quad (2.15)$$

Tiap *hidden unit* ($Z_j, j = 1, \dots, p$) *update* bias dan bobot ($i = 0, \dots, n$):

$$v_{ij} \text{ (baru)} = v_{ij} \text{ (lama)} + \Delta v_{ij} \quad (2.16)$$

Step 9. Kondisi pemberhentian *test* (memeriksa kondisi berhenti)

Setelah pelatihan, suatu jaringan *Neural Network* digunakan hanya pada fase *feedforward* algoritma pelatihan. Prosedur aplikasinya meliputi antara lain:

Step 0. Inisialisasi bobot (dari algoritma pelatihan)

Step 1. Untuk tiap vektor *input*, kerjakan step 2-4

Step 2. Untuk $i = 1, \dots, n$; set aktivasi dari *input unit* x_i ;

Step 3. Untuk $j = 1, \dots, p$:

$$z_{in_j} = v_{oj} + \sum_{i=1}^n x_i v_{ij}$$

$$z_j = f(z_{in_j}),$$

Step 4. Untuk $k = 1, \dots, m$:

$$y_{in_k} = w_{ok} + \sum_{j=1}^m z_j w_{jk}$$

$$y_k = f(y_{in_k}).$$

2.2 Analisa Time Series

Menurut Makridakis (1999), penggunaan metode-metode peramalan umum meliputi dua tugas dasar, yaitu analisis deret data dan seleksi model peramalan yang paling cocok dengan deret data tersebut. Dalam pemakaian metode pemulusan, analisis deret data untuk memeriksa adanya faktor musiman akan membantu penetapan metode pemulusan tertentu yang dapat menangani ada atau tidaknya pengaruh musiman tersebut. Pengaruh musiman adalah kecenderungan pengulangan suatu pola data yang dipengaruhi oleh adanya musim.

Analisa *time series*, memperkirakan apa yang akan terjadi berdasarkan data masa lalu. *Time series* adalah kumpulan dari pengamatan yang teratur pada sebuah variabel selama periode waktu tertentu. Dengan mempelajari bagaimana sebuah variabel (berdasarkan sejarah) berubah setiap waktu, sebuah relasi diantara kebutuhan dan waktu dapat diformulasikan dan digunakan untuk memprediksi tingkat kebutuhan yang akan datang. Dalam analisa *time series*, data historis dikomposisikan untuk mengenali komponen yang relevan yang mempengaruhi variabel yang akan diperkirakan.

Sebuah *time series* (deret waktu) merupakan urutan dari serangkaian vektor-vektor : $x(t), t = 0, 1, \dots$,

dimana t = menunjukkan kesamaan waktu

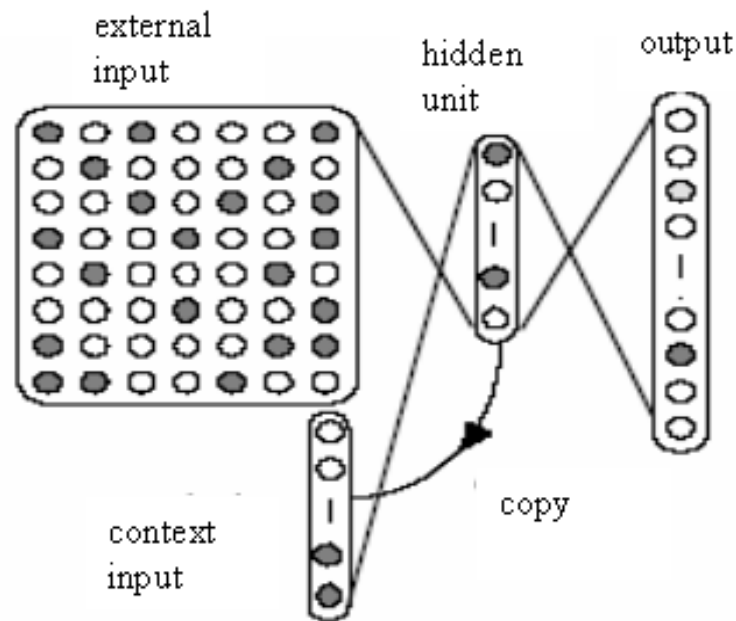
Secara teori, x mungkin merupakan sebuah nilai, dimana variasinya terhubung secara terus menerus dengan t . Dalam prakteknya, untuk beberapa masukan ke sistem, x merupakan sampling suatu deret titik-titik data tertentu (diskrit). Kecepatan pada tiap-tiap sampling diambil dari resolusi maksimum

model. Pada tahapan ini sebuah model dengan resolusi tertinggi akan memiliki prediksi terbaik. Jadi hasil yang lebih besar mungkin dihasilkan dengan mengakses setiap titik dalam suatu deret.

Pemilihan rentang waktu dalam melakukan sampling, adalah proses *delay* (penundaan) dan diasumsikan bahwa rentang waktu merupakan pergerakan setiap titik data. Dalam *neural network* harus memfokuskan pada peramalan dengan deret waktu dari nilai-nilai x sampai waktu saat ini.

2.2.1 Jaringan menurut Elman dan model ruang

Perulangan terhadap *layer-layer* merupakan bagian dari *hidden layer* atau *output layer*, dimana status (*node*) pada saat waktu t adalah “hasil penggandaan (*copy*)” dalam sebuah penambahan inputan dari perhitungan pada waktu $t + 1$. Dimana perhitungan sebuah status (*node*) baru, didapat dari *context input*. Penambahan *input* membawa sebuah informasi mengenai nilai status sebelumnya. Jaringan Elman ini menggambarkan proses pengambilan kembali urutan-urutan nilai (informasi) yang sudah tersimpan untuk melanjutkan proses selanjutnya guna mendapatkan *output* yang lebih optimal.



Gambar 2.6 Jaringan arus balik menurut Elman

Arsitektur ini menggambarkan arah pada pengelompokan dan pengambilan status, ini adalah salah satu bentuk dari sebuah *neural network*. Fungsi lain yang juga disediakan oleh model *neural network* lainnya adalah *Classification* (Pengelompokan) dan *Optimization* (mencari fungsi yang optimal).

2.2.2 Penerapan time series pada neural network

Pada MLP (*Multi Layer Perceptron*) terdiri dari tiga *input layer*, *hidden layer*, dan *output layer*, atau lebih dikenal dengan *backpropagation*, dan umumnya digunakan oleh sistem-sistem yang statis terhadap perubahan waktu. Untuk *time series* yang sifatnya dinamis (berubah terhadap waktu), maka MLP harus bisa dimodifikasi dengan cara bagaimana bisa mengingat dan memanggil kembali status yang lalu. Dalam hal ini MLP (*Multi Layer Perceptron*) bisa dijabarkan sebagai berikut: baik *hidden layer* (model Elman) ataupun *output*

layer (model Jordan), status pada waktu t di-copy-kan sebagai suatu suplemen *vektor input* untuk komputasi pada waktu $t + 1$. Pada waktu perhitungan status baru informasi bergerak turun secara drastis, suplemen *vektor input* memberikan konteks informasi yang lalu, sehingga NN (*Neural Network*) dapat mengingat *time delay* NN (*Neural Network*), yang merupakan dimensi *temporal* (sebagai koneksi antar *delay* waktu pada variabel koneksi), dan memanggil kembali status yang lalu.

Pada model Elman, nilai pada *layer hidden* dikembalikan sebagai input. Pada *layer input*, *time series* (deret waktu) dalam *neural network* merupakan pengukuran secara sekuensial dari satu atau lebih variabel yang ada pada suatu sistem yang dinamis, dimana status akan berubah dalam waktu. Asumsi dasar yang digunakan pada model Elman ini, disebut asumsi Markov, yang artinya bahwa elemen sekuensial berikutnya dapat diprediksikan oleh suatu sistem yang menghasilkan deret waktu. Dengan kata lain semua sejarah deret waktu yang dibutuhkan untuk menghasilkan suatu elemen sekuensial dapat diekspresikan oleh satu vektor status yang diwakili oleh *hidden layer*.

Beberapa orang mempunyai pendapat yang berbeda dalam menyebutkan tipe-tipe dari *neural network* untuk digunakan dalam proses *time series* (deret waktu). Perbedaan *neural network* dititikberatkan pada tipe mekanismenya, guna menjelaskan informasi yang sifatnya sementara. Perbedaan proses tersebut, antara lain (1) *Layer delay* tanpa pengembalian (*feed back*), (2) *Layer delay* dengan pengembalian, (3) *Unit delay* tanpa pengembalian (*feed back*), dan (4) *Unit delay* dengan pengembalian. Pada *neural network*, *time delay* digunakan untuk mengingat dan mengambil kembali sekuen (status) yang lalu.

2.3 Perhitungan Error dengan MSE dan MAPE

MSE (*Mean Squared Error*) merupakan suatu ukuran ketepatan perhitungan, dengan mengkuadratkan masing-masing galat untuk masing-masing item dalam sebuah susunan data dan kemudian memperoleh rata-rata atau nilai tengah jumlah kuadrat tersebut. Nilai tengah galat kuadrat memberikan bobot yang lebih besar terhadap galat yang besar daripada kesalahan kecil sebab galat dikuadratkan sebelum dijumlahkan.

Sedangkan MAPE (*Mean Absolute Percentage Error*) merupakan nilai tengah atau rata-rata jumlah seluruh persentasi galat untuk sebuah susunan data yang diberikan. Ia merupakan salah satu ukuran ketepatan model prediksi yang digunakan dalam metode kuantitatif atau peramalan.

Rumusan untuk mencari nilai MSE dan MAPE (Mertens 1994) adalah sebagai berikut :

$$MAE = (\Sigma |\text{Error}|) / \text{jumlah data} \quad (2.17)$$

$$MAPE = (\Sigma |\text{PE}|) / \text{jumlah data} \quad (2.18)$$

$$\text{Error} = \text{Target} - \text{Hasil klasifikasi } \textit{Neural Network} \quad (2.19)$$

$$\text{PE} = (\text{Error} / \text{Target}) \times 100 \quad (2.20)$$

BAB III

PERANCANGAN SISTEM

Dalam pembuatan Tugas Akhir ini diperlukan perancangan sistem guna mempermudah dan memperjelas dalam pembuatan suatu program, sehingga program yang dibuat nanti dapat terstruktur, jelas dan sesuai dengan yang diharapkan. Perancangan sistem di sini meliputi metode penelitian, identifikasi permasalahan, analisa permasalahan, *database management system* (DBMS), desain *input output*, desain uji coba dan analisa serta perancangan sistem yang terdiri dari bagan sistem, bagan berjenjang, *data flow diagram* dan *entity relationship diagram*.

3.1 Metode Penelitian

3.1.1 Studi pustaka

Studi pustaka dilakukan untuk menentukan teori apa saja yang akan digunakan dalam pembuatan sistem pengambilan keputusan dalam prediksi sirkulasi container masuk dan keluar, yang menjadi bahasan utama tugas akhir ini. Adapun teori yang digunakan dapat dilihat pada bab II landasan teori.

3.1.2 Survei

Survei dan wawancara dilakukan untuk mengumpulkan data-data di PT Prima Manggalindo, yang menunjang dan dibutuhkan untuk pembuatan tugas akhir ini. Data-data yang dibutuhkan diantaranya adalah data-data transaksi masuk dan keluar *container* beberapa periode untuk tiap principal.

3.1.3 Wawancara

Wawancara dilakukan dengan tujuan untuk mengumpulkan informasi yang berkaitan dengan masalah-masalah tentang sirkulasi masuk dan keluarnya container.

3.2 Identifikasi Permasalahan

Pada saat melakukan proses identifikasi masalah ini, perumusan masalah dan tujuan penelitian serta konsep atau metode yang mendukung dalam penelitian ini ditentukan dan dipelajari. PT. Prima Manggalindo adalah suatu perusahaan yang bergerak di bidang jasa yang menyediakan tempat penyimpanan container sementara.

Pada saat ini perusahaan tersebut mengalami kesulitan dalam memprediksi sirkulasi container masuk dan keluar yang akan datang. Berbagai faktor yang harus dipertimbangkan dalam memprediksi sirkulasi container masuk dan keluar yang akan datang dimana container dalam penyimpanannya harus dibedakan berdasarkan pemiliknya (principal) dan juga sangat bergantung pada waktu.

Oleh karena itu, diperlukan sistem penerapan *Neural Network Time Series* untuk memprediksi sirkulasi container masuk dan keluar yang akan datang pada perusahaan tersebut. Untuk dapat memberikan solusi dalam memprediksi sirkulasi container masuk dan keluar yang optimal.

Tugas akhir ini diharapkan dapat membantu untuk mengembangkan sebuah sistem untuk memprediksikan sirkulasi *container* yang masuk dan keluar depo di PT Prima Manggalindo.

3.3 Analisa Permasalahan

Setelah melakukan survey, wawancara dan identifikasi permasalahan, langkah selanjutnya adalah melakukan analisa permasalahan dan pemecahannya. Penelitian ini dilakukan untuk mengembangkan sebuah sistem untuk memprediksikan sirkulasi *container* yang masuk dan keluar depo di PT Prima Manggalindo. Untuk pengembangan sistem ini diperlukan data-data transaksi depo container beberapa periode untuk tiap principal yang didapat dari PT Prima Manggalindo. *Backpropagation Neural Network Time Series* merupakan metode yang digunakan sebagai solusi dari permasalahan.

3.3.1 Sirkulasi Container Masuk dan Keluar dengan menggunakan Artificial Neural Network

Sejauh ini masih belum ada sistem yang dapat membantu sirkulasi container masuk dan keluar tersebut dalam menentukan dengan tepat dan cepat.

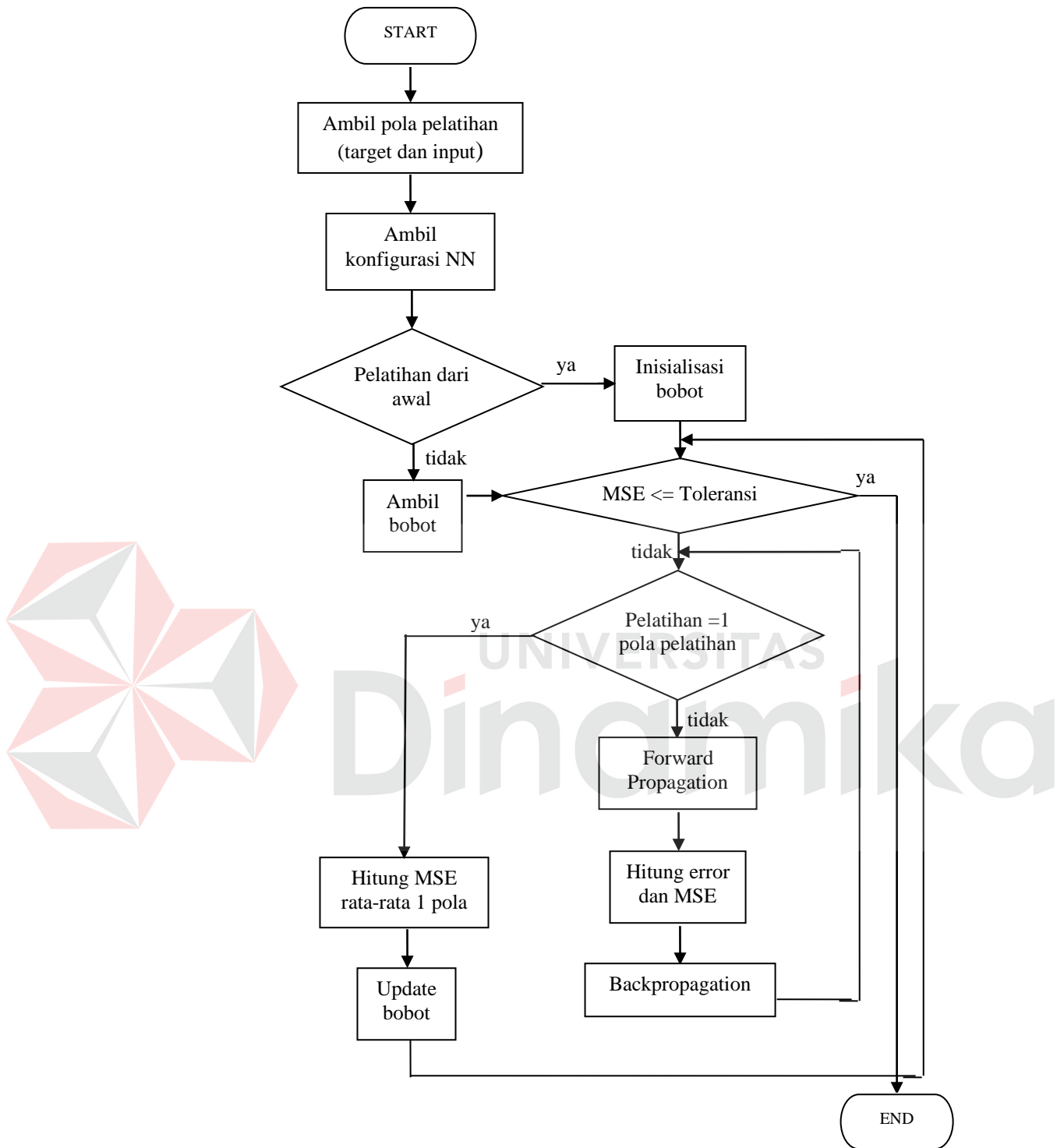
Apabila dilakukan secara manual tentunya hal tersebut kurang efisien karena membutuhkan waktu dan tenaga yang tidak sedikit mengingat banyaknya container masuk dan keluar yang harus diamati perkembangannya serta banyaknya variabel pengukuran yang harus ditekankan. Oleh sebab itu dibutuhkan suatu sistem yang dapat membantu dalam membuat keputusan dengan tepat dan cepat mengenai sirkulasi container masuk dan keluar tentunya dengan metode yang sesuai dan dengan tetap memperhatikan variabel ukuran dan aturan-aturan yang ada.

Pengaturan *container* saat penyimpanan pada dasarnya menggunakan model *First In First Out* (FIFO), namun pada praktiknya hal ini sangat susah

untuk dilakukan, karena sirkulasi masuk dan keluar container yang cukup cepat, dimana container dalam penyimpanannya harus dibedakan berdasarkan pemiliknya (principal). Oleh sebab itu prediksi sirkulasi *container* per principal yang masuk dan keluar depo merupakan salah satu informasi yang sangat berguna dalam pengaturan *container* ini.

Artificial Neural Network adalah suatu model yang merupakan analogi dari jaringan saraf biologis untuk dapat diterapkan ke dalam sistem komputasi, sehingga sistem diharapkan dapat belajar untuk mengenali pola-pola dari data yang dilatihkan kepadanya. Oleh sebab itu *Artificial Neural Network* telah banyak digunakan untuk prediksi, pengenalan pola, dan klasifikasi. Pada Tugas Akhir ini.

Sirkulasi container di tiap waktu adalah tidak tetap. Dibutuhkan suatu metode yang dapat menggali data-data sirkulasi container ini dan menemukan pola-pola yang ada, sehingga dapat diprediksikan. *Backpropagation Neural Network* yang memiliki kemampuan untuk belajar sendiri dari data-data yang diumpunkan kepadanya dan membentuk pola-pola terhadap data-data tersebut, merupakan pilihan solusi yang akan digunakan. Karena sirkulasi *container* yang masuk dan keluar depo juga sangat bergantung pada waktu, sehingga dibutuhkan prediksi yang mencakup deret waktu (*time series*), maka dibutuhkan pengembangan lebih lanjut dari *Backpropagation Neural Network*, yaitu *Neural Network Time Series*.

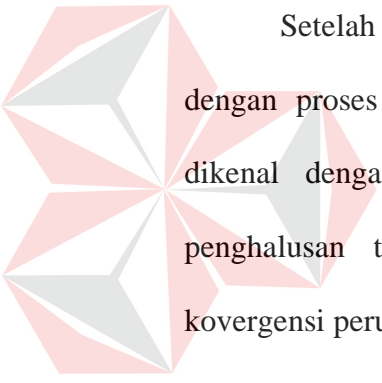


Gambar 3.1 Sistem flow training *Backpropagation Neural Network*

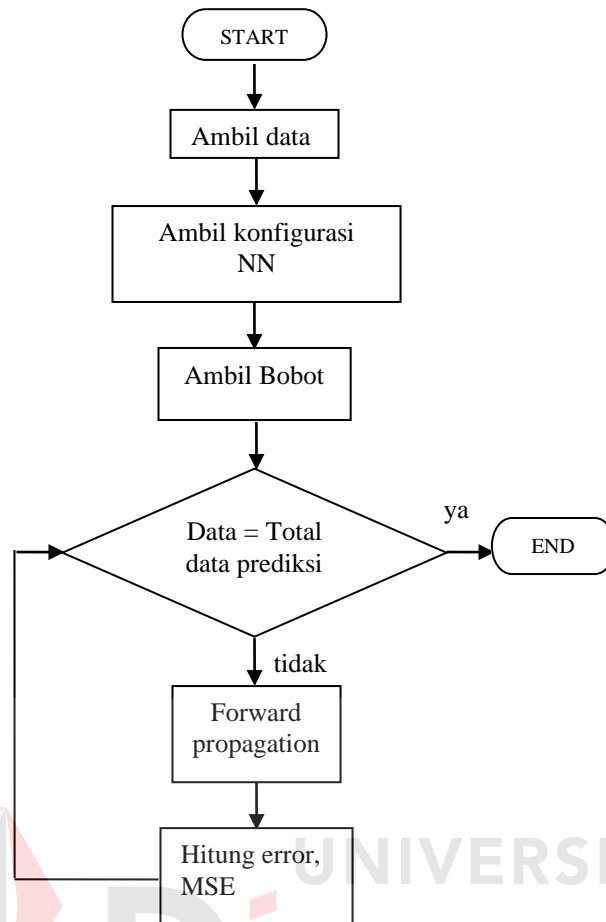
Keterangan gambar:

Proses dimulai dengan mengambil pola pelatihan (*input* dan *target*) dan konfigurasi *neural network*, kemudian terdapat pilihan apakah kita memulai *training* dari awal atau tidak. Jika ya, langsung ke proses inisialisasi bobot, dan jika tidak, maka ambil bobot.

Selanjutnya iterasi proses pelatihan akan dimulai dan berhenti pada saat $MSE \leq$ toleransi. Satu kali iterasi pelatihan akan meliputi proses *forward propagation*, perhitungan *error* dan MSE, dan *back propagation* dari tiap pola masukan, hingga keseluruhan pola masukan telah dieksekusi.



Setelah semua pola masukan telah dieksekusi, maka akan dilanjutkan dengan proses peng-*update*-an bobot, dari *neural network*. Teknik ini lebih dikenal dengan *batch updating*. Prosedur *batch updating* memiliki efek penghalusan terhadap koreksi, yang memungkinkan untuk meningkatkan konvergensi perubahan ke titik lokal minimum.



Gambar 3.2 Sistem flow testing *Backpropagation Neural Network*

Keterangan gambar :

Proses testing dimulai dengan mengambil data prediksi, mengambil konfigurasi NN dan ambil bobot, kemudian dilanjutkan hingga data sudah sama dengan total data prediksi. Pada testing NN hanya terdapat proses *forward*, dan proses perhitungan *error*, MSE.

3.3.2 Inisialisasi bobot dan bias

Langkah pertama dari proses *neural network* ini adalah menentukan *nodes*, baik *input nodes*, *hidden nodes*, maupun *output nodes*. Setelah *nodes* tergambar,

langkah berikutnya adalah menentukan bobot dan bias. Bias biasanya selalu bernilai 1. Sedangkan untuk bobot pertama kali didapat dari proses random dari -0.5 sampai 0.5.

3.3.3 Forwardpropagation

Setiap *input unit* menyebarkan sinyal pada *hidden unit* dan setiap *hidden unit* kemudian menjumlahkan sinyal-sinyal *input* yang sudah berbobot, termasuk biasnya dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari *hidden unit* yang bersangkutan. Kemudian *hidden unit* tersebut akan mengirim sinyal *output* yang telah dihitung tersebut ke seluruh unit pada *output unit*. Setelah sinyal *output* dikirim ke *output unit* kemudian setiap *output unit* akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot, termasuk biasnya dan kemudian memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari *output unit* yang bersangkutan lalu mengirimkannya ke seluruh unit pada *output unit*. Untuk klasifikasi, proses berhenti sampai *Forwardpropagation* ini.

3.3.4 Propagasi kesalahan

Setiap *output unit* menerima suatu pola target yang sesuai dengan pola pelatihan *input* untuk menghitung faktor koreksi kesalahan (*error*) antara target dengan *output* yang dihasilkan jaringan. Sebagaimana pelatihan *input*, pelatihan *output* juga diskalakan menurut fungsi aktivasi yang dipakai. Faktor koreksi *error* (dalam hal ini merupakan faktor koreksi *error output-hidden*) digunakan untuk menghitung delta bobot *output-hidden* yang nantinya akan dipakai untuk memperbarui bobot dan bias. Setelah delta bobot diketahui, setiap *hidden unit*

menjumlah *input delta* yang sudah berbobot. Kemudian hasilnya dikalikan dengan turunan dari fungsi aktivasi yang digunakan jaringan untuk menghasilkan faktor koreksi *error* (dalam hal ini merupakan faktor koreksi *error hidden-input*). Faktor koreksi *error* ini digunakan untuk menghitung delta bobot *hidden-input* yang nantinya akan dipakai untuk memperbarui bobot dan bias.

3.3.5 Pembaharuan bobot dan bias

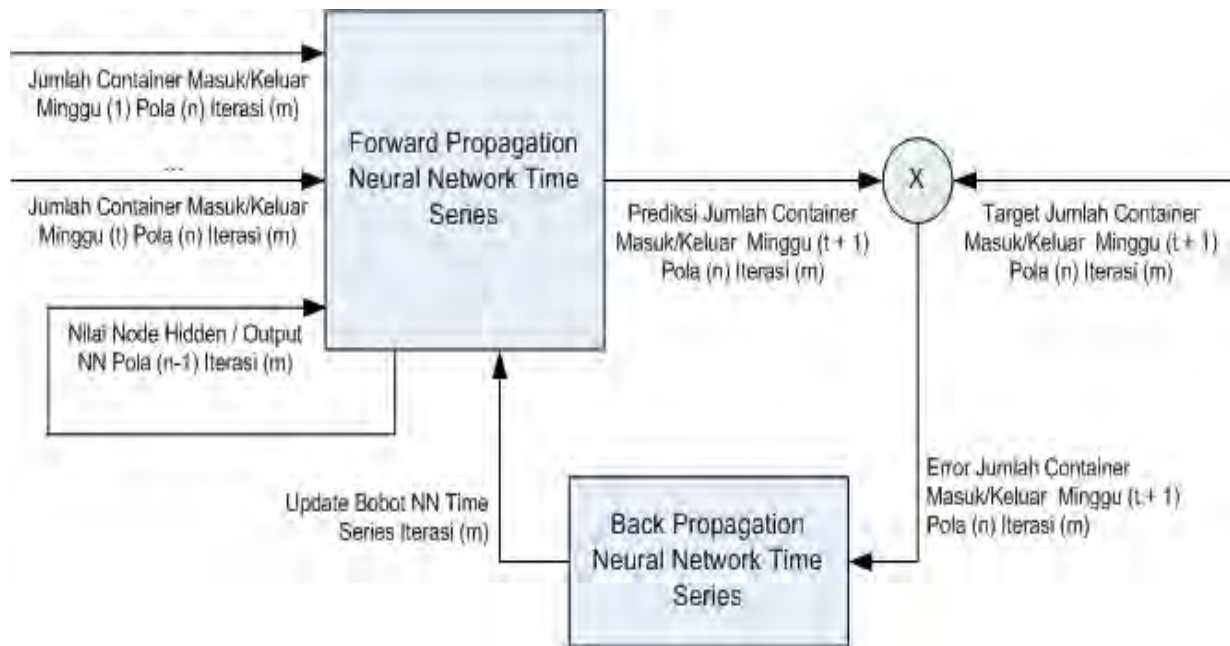
Setiap *output unit* kemudian akan memperbarui bias dan bobotnya dari setiap *hidden unit*. Demikian pula setiap *hidden unit* akan memperbarui bias dan bobotnya dari setiap *input unit*.

3.4 Perancangan Sistem

Perancangan sistem di sini menggunakan *data architect* dan *process analyst* dari *power designer 6 32 bit*, dan Microsoft Visio 2000 . Perancangan sistem digunakan untuk menggambarkan aliran data yang ada antara proses dalam bentuk diagram.

3.4.1 Bagan sistem

Berikut gambaran umum dari sistem pengambilan keputusan proses sirkulasi container masuk dan keluar yang dikembangkan.



Gambar 3.3 Diagram alir sistem *NN Time Series*

Secara garis besar, alur sistem *NN Time Series* akan diberi umpan maju, yaitu jumlah container masuk / keluar per minggu selama t minggu. Data ini akan bergeser untuk tiap pola data yang akan dilatihkan sampai semua minggu yang akan dilatihkan selesai. 1 pola data terdiri dari inputan jumlah container masuk / keluar dari minggu ke 1 hingga minggu ke t dengan target adalah jumlah container masuk / keluar di minggu ke $t+1$. *NN Time Series* juga mendapatkan umpan nilai node hidden / node output dari pola sebelumnya (pola ke $n-1$). Hasil keluaran dari *NN Time Series* adalah prediksi jumlah container masuk/keluar untuk minggu berikutnya (minggu ke $t+1$), dimana hasil keluaran ini akan dibandingkan dengan target jumlah container masuk/keluar di minggu ke $t+1$.

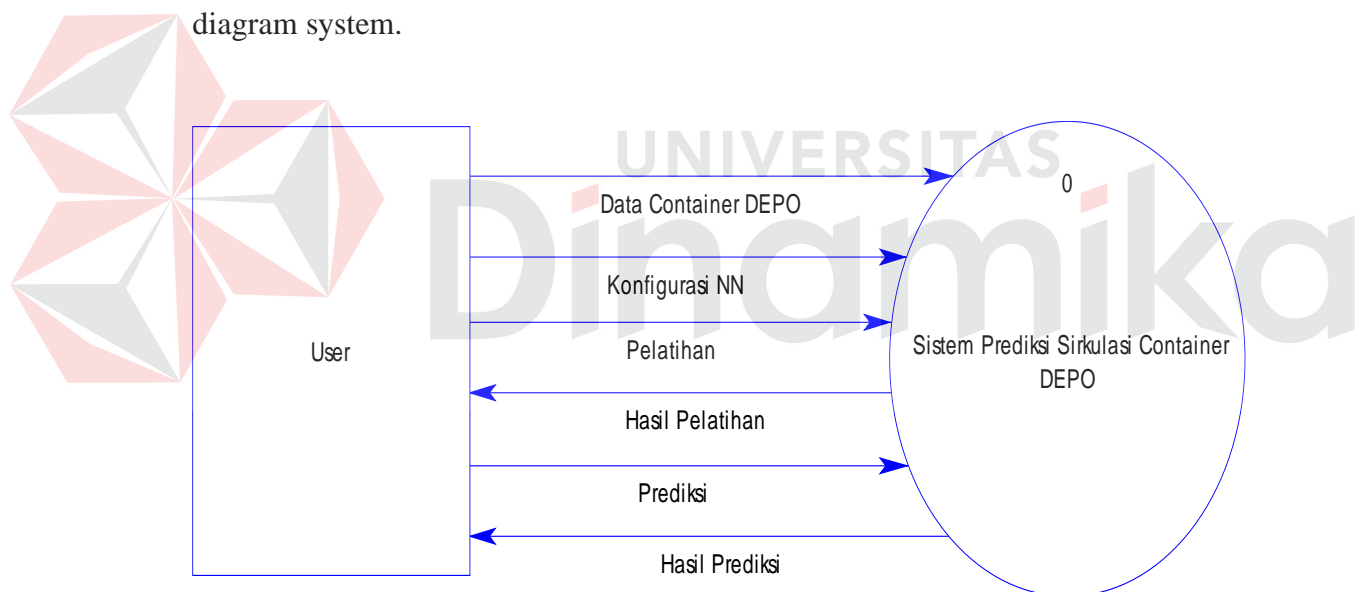
Selisih antar target dan hasil prediksi adalah error yang akan menjadi masukan bagi proses umpan balik *NN Time Series* dan menghasilkan update bobot. Nilai

update bobot akan diakumulasikan selama 1 iterasi untuk tiap pola data yang dilatihkan, dan bobot-bobot NN Time Series akan diubah terhadap akumulasi nilai update bobot setelah pelatihan menyelesaikan 1 iterasi untuk digunakan pada iterasi berikutnya.

3.4.2 Data flow diagram

A. Context diagram

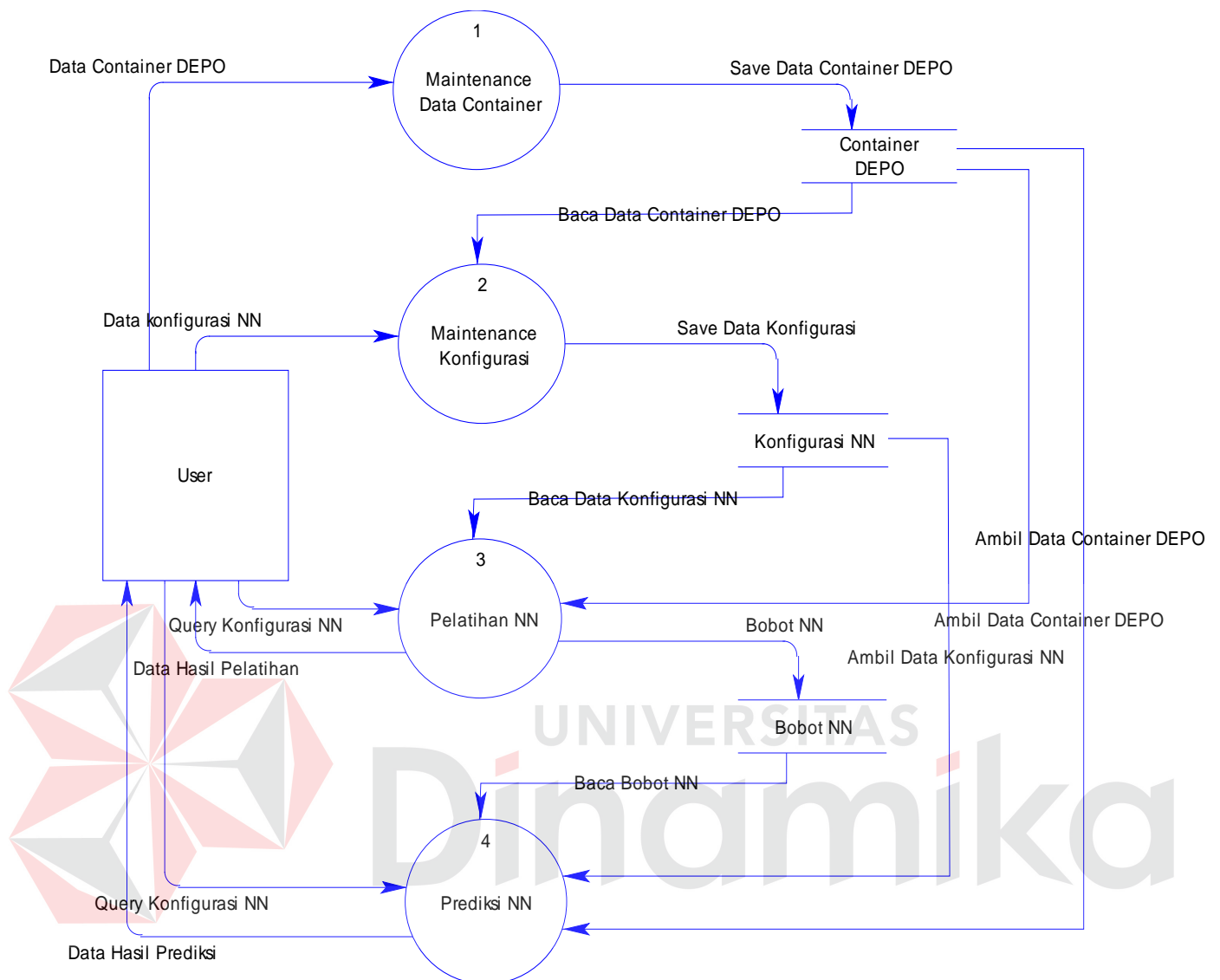
Merupakan pengembangan proses yang tertinggi dalam tingkatan/level dan berhubungan dengan beberapa *entity* yang terlibat langsung dengan pengolahan data dalam sistem yang dibuat. Gambar 3.7 berikut adalah context diagram system.



Gambar 3.4 Context Diagram Sistem

B. DFD level 0

DFD level 0 merupakan hasil dari *decompose* dari proses yang ada di context diagram sebelumnya. Gambar 3.8 berikut merupakan DFD level 0.



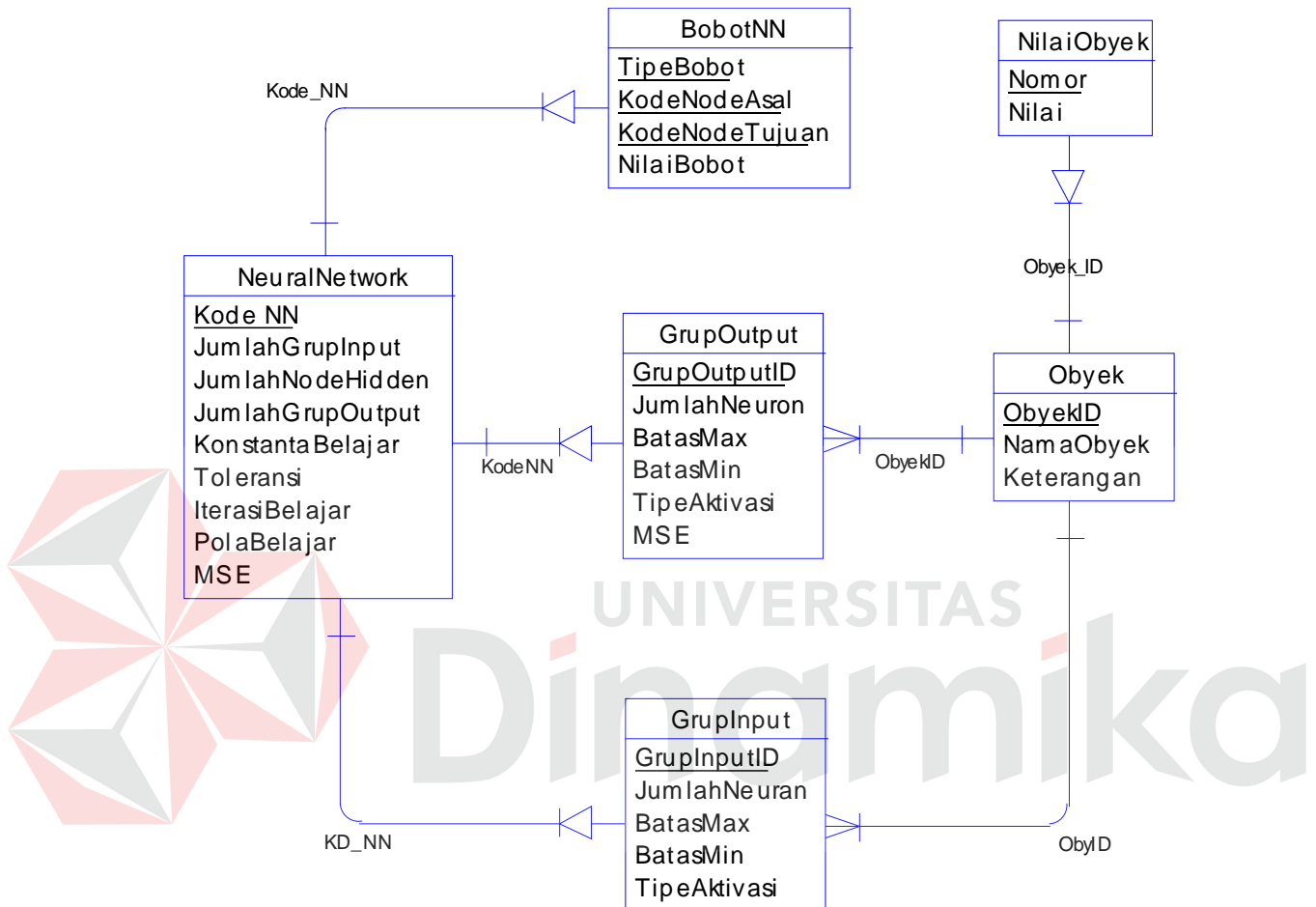
Gambar 3.5 DFD level 0

3.4.3 Entity relational diagram (ERD)

ERD (Entity Rational Diagram) digunakan untuk mengintegrasikan, menentukan dan mendokumentasikan kebutuhan sistem akan pemrosesan database. Model ini dapat menggambarkan struktur basis data secara detail sampai dengan field-field yang terdapat dalam suatu tabel serta tipe-tipe data yang menyertainya.

A. Conceptual Data Model (CDM)

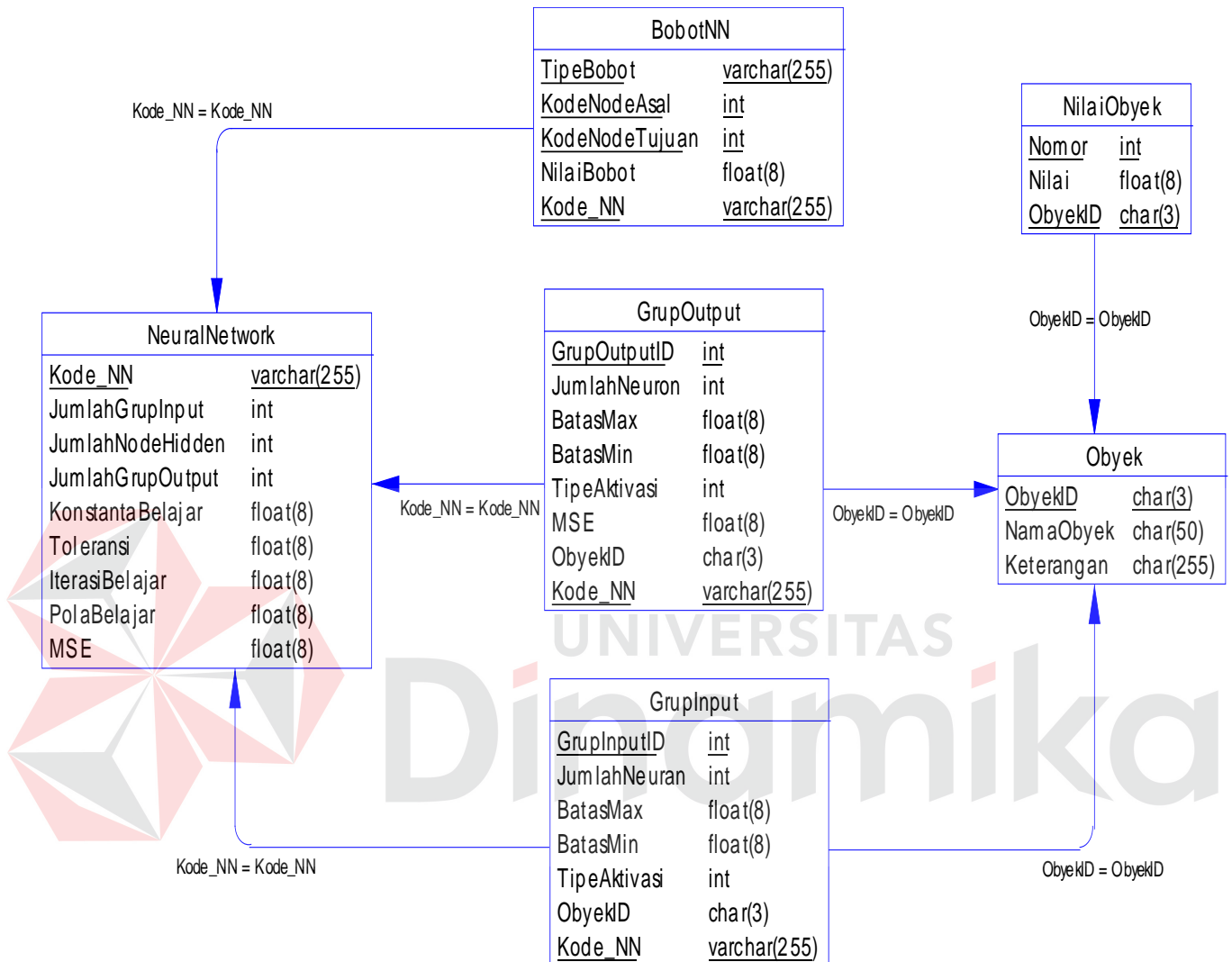
Model konseptual dari database yang telah dirancang adalah sebagai berikut:



Gambar 3.6 Entity Relational Diagram Conceptual Data Model

B. Physical Data Model (PDM)

Model fisik dari database yang telah dirancang adalah sebagai berikut:



Gambar 3.7 Entity Relational Diagram Physical Data Model

3.5 DBMS (Database Management System)

Dalam penyusunan sistem ini digunakan Microsoft SQL (Structure Query Language) server 2000 untuk penyimpanan datanya, rincian dari ER-Diagram digambarkan dengan struktur database yang terdiri atas kolom-kolom yang memiliki atribut berupa nama kolom, tipe data, batasan atau aturan yang

mengarah pada tabel tertentu dan keterangan. Struktur database menunjukkan daftar kebutuhan tabel yang digunakan untuk menyimpan data yang diperlukan dalam sistem ini. adapun struktur dari database tersebut adalah sebagai berikut :

Tabel *Neural Network*

Primary Key : KodeNN

Jumlah field : 8

Foreign key : -

Relasi tabel : Tabel BobotNN, GrupInput, GrupOutput

Tabel 3.1 NN

Kolom	Tipe Data	Panjang	Keterangan
KodeNN	Varchar	8	Primary Key
JumlahGrupInput	Integer	4	-
JumlahNodeHidden	Integer	4	-
JumlahGrupOutput	Integer	4	-
Konstanta Belajar	Float	8	-
Toleransi	Float	8	-
Iterasi Belajar	Float	8	-
Pola Belajar	Float	8	-

Tabel Bobot NN

Primary key : Kode NN, Tipe Bobot, Kode *Node* Asal, Kode *Node* Tujuan

Jumlah field : 5

Foreign key : KodeNN

Relasi tabel : *Neural Network*

Tabel 3.2 Bobot NN

Kolom	Tipe Data	Panjang	Keterangan
KodeNN	Varchar	255	Primary Key
Tipe Bobot	Varchar	255	Primary Key
KodeNode Asal	Integer	4	Primary Key
Kode Node Tujuan	Integer	4	Foreign Key
Nilai Bobot	Float	8	-

Tabel GrupOutput

Primary key : GrupOutputID, Kode NN

Jumlah field : 8

Foreign key : ObyekID, KodeNN

Relasi tabel : *Neural Network*, Obyek

Tabel 3.3 Grup Output

Kolom	Tipe Data	Panjang	Keterangan
GroupOutputID	Integer	4	Primary Key
KodeNN	Varchar	255	Primary Key
JumlahNeuron	Integer	4	-
BatasMax	Float	8	-
BatasMin	Float	8	-
TipeAktivasi	Integer	4	-
ObyekID	Char	3	-
MSE	Float	8	-

Tabel Obyek

Primary key : ObyekID

Jumlah field : 3

Foreign key : -

Relasi tabel : GrupInput, Grupoutput, Nilai Obyek

Tabel 3.4 Obyek

Kolom	Tipe Data	Panjang	Keterangan
ObyekID	Char	3	Primary Key
NamaObyek	Char	50	-
Keterangan	Char	255	-

Tabel GrupInput

Primary key : GrupInputID, KodeNN

Jumlah field : 7

Foreign key : ObyekID, KodeNN

Relasi tabel : Obyek, Neural Network

Tabel 3.5 Grup Input

Kolom	Tipe Data	Panjang	Keterangan
GrupInputID	Integer	4	Primary Key
KodeNN	Varchar	255	Primary Key
JumlahNeuron	Integer	4	-
BatasMax	Float	8	-
BatasMin	Float	8	-
TipeAktivasi	Integer	4	-
ObyekID	Char	3	-

Tabel Nilai Obyek

Primary key : Nomor, ObyekID

Jumlah field : 3

Foreign key : ObyekID

Relasi tabel : Obyek

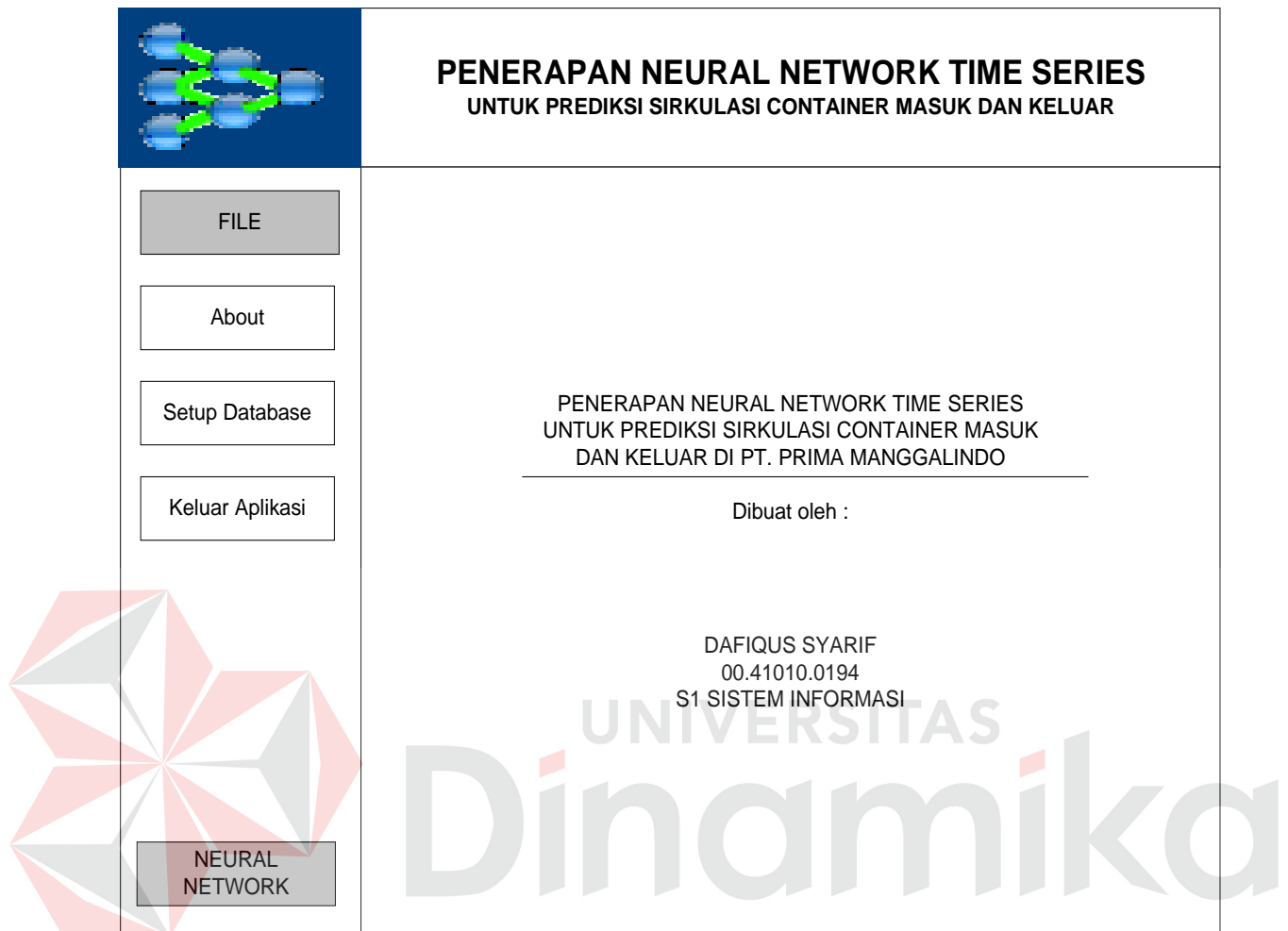
Tabel 3.6 Nilai Obyek

Kolom	Type Data	Panjang	Keterangan
Nomor	Integer	4	Primary Key
ObyekID	Char	3	Primary Key
Nilai	Float	8	

3.6 Desain Input Output

Desain *input output* adalah suatu perancangan dari form-form yang mengimplementasikan masukan dan rancangan *output* yang berupa laporan-laporan yang akan digunakan sebagai dokumentasi. Perancangan *input* dan *output* yang dipakai untuk mengimplementasikan sistem yang dibuat dapat dilihat pada gambar berikut :

3.6.1 Form Utama



Gambar 3.8 Rancangan form Utama

Form ini didesain sebagai media untuk menampilkan pertama kali aplikasi dijalankan. Untuk menu File terdiri dari sub-sub menu About, Setup Database dan Keluar Aplikasi. Untuk menu Neural Network terdiri dari sub-sub menu Master Data, Konfigurasi NN, Pelatihan NN, dan Prediksi.

3.6.2 Form Master Data

PENERAPAN NEURAL NETWORK TIME SERIES

UNTUK PREDIKSI SIRKULASI CONTAINER MASUK DAN KELUAR

FILE

NEURAL NETWORK

Data

Konfigurasi Neural Network

Pelatihan

Prediksi

Data ID

Nama Data

Keterangan

Nomor

Nilai

Detil Nilai Data

	Nomor	Nilai	

|<

<

.../...

>

>|

Data Baru

Edit Data

Hapus Data

Simpan Data

Batal

Refres Data

Gambar 3.9 Rancangan form Master Data

Form master data digunakan untuk proses menampilkan melalui tombol-tombol navigasi, menambah, mengubah, menghapus, membatalkan dan merefresh data container dari tabel obyek dan nilai obyek.

3.6.3 Form Konfigurasi Neural Network

PENERAPAN NEURAL NETWORK TIME SERIES
UNTUK PREDIKSI SIRKULASI CONTAINER MASUK DAN KELUAR

FILE
NEURAL NETWORK
Data
Konfigurasi Neural Network
Pelatihan
Prediksi

Kode NN

Jml Grup Input Jml Node Hidden Toleransi

Jml Node Input Konst. Belajar Pola Belajar

Jml Grup Output Iterasi Belajar

Grup Input	Grup Output
Grup ID <input type="text"/>	Obyek ID <input type="text"/> ▾
Jumlah Neuron <input type="text"/>	Batas Maximum <input type="text"/>
Tipe Aktivasi <input type="text"/> ▾	Batas Minimum <input type="text"/> <input type="button" value="Edit Data"/>

Detil Grup Input

	Grup ID	Jml Neuron	Obyek ID	Batas Max	Batas Min	Tipe Aktivasi
▾	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

UNIVERSITAS DINDAMIKA

◀ ▶

Gambar 3.10 Rancangan form konfigurasi *neural network*

Form konfigurasi *neural network* ini digunakan untuk proses menampilkan melalui tombol-tombol navigasi, menambah, mengubah, menghapus, membatalkan dan merefres data-data konfigurasi *neural network* ke dalam tabel Neural Network.

3.6.4 Form Pelatihan

PENERAPAN NEURAL NETWORK TIME SERIES

UNTUK PREDIKSI SIRKULASI CONTAINER MASUK DAN KELUAR

FILE

NEURAL NETWORK

Data

Konfigurasi Neural Network

Pelatihan

Prediksi

Kode Neural Network

Iterasi

Latihan Dari Awal

Data Awal

Data Awal

Grafik Pelatihan

Data Detil Neural Network


Grafik Pelatihan				
[Line Graph showing fluctuating data points]				

Pengukuran	
Output	MSE

Gambar 3.11 Rancangan form pelatihan

Form pelatihan ini digunakan untuk melakukan proses pelatihan data *neural network*. Dalam form ini simpan bobot digunakan untuk menyimpan bobot hasil dari pelatihan. Bobot disimpan ke dalam tabel bobot_NN.

3.6.5 Form Pelatihan



PENERAPAN NEURAL NETWORK TIME SERIES

UNTUK PREDIKSI SIRKULASI CONTAINER MASUK DAN KELUAR

FILE

NEURAL NETWORK

Data

Konfigurasi Neural Network

Pelatihan

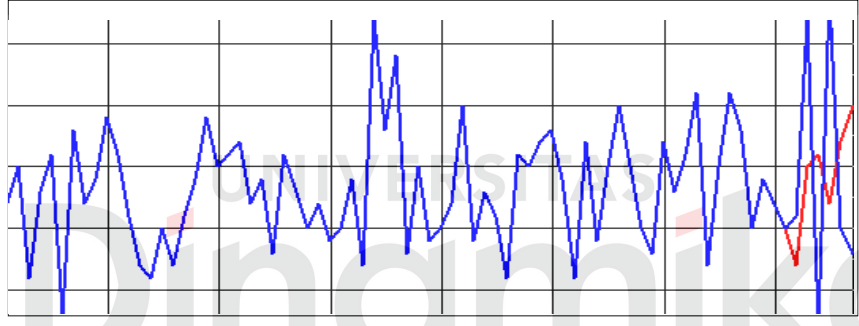
Prediksi

Kode Neural Network

Data Awal

Data Awal

Grafik Prediksi
Data Detil Neural Network
Hasil Prediksi



Gambar 3.12 Rancangan form prediksi

Form prediksi ini digunakan untuk melakukan proses tes dan mengetahui hasil output. Dalam form ini simpan bobot hasil dari pelatihan digunakan untuk menampilkan hasil prediksi .

3.7 Desain Uji Coba dan Analisa

Desain uji coba dan analisa adalah untuk melihat bagaimana suatu proses pada sistem bekerja, tujuan dari proses, dan *output* yang diharapkan. Uji coba dan analisa yang dilakukan meliputi uji coba dan analisa fitur dasar sistem, uji coba dan analisa perhitungan sistem serta uji coba dan analisa validasi sistem.

3.7.1 Desain Uji Coba

A Uji coba fitur dasar sistem

Uji coba fitur dasar sistem ini dilakukan dengan metode *black box testing*, yang bertujuan untuk memastikan fitur-fitur dasar dalam sistem telah berfungsi dengan benar. Uji coba fitur dasar sistem dalam tugas akhir ini meliputi :

A.1 Uji coba mengolah data container

Tujuan dari uji coba ini adalah untuk memastikan keberhasilan dari obyek-obyek yang berada pada form data container.

Tabel 3.7 Contoh data container

Nama Field	Data-1	Data-2	Data-3	Data-4	Data-5	Data-6
Data ID	A40	A20	B40	B20	C40	C20
Nama Data	A40GP	A20GP	B40GP	B20GP	C40GP	C20GP
Keterangan	Bulan	Bulan	Minggu	Minggu	Hari	Hari

Tabel 3.8 Uji coba mengolah data container

No	Tujuan	Input	Output yang diharapkan
1	Memastikan data container	Tabel master_data	Seluruh data dari tabel master_data
2	Memastikan penambahan data baru dapat dilakukan	Memasukkan data container ke dalam tabel master_data.	Data container yang baru dimasukkan dapat disimpan pada tabel master_data

Tabel 3.8 Uji coba mengolah data container (lanjutan)

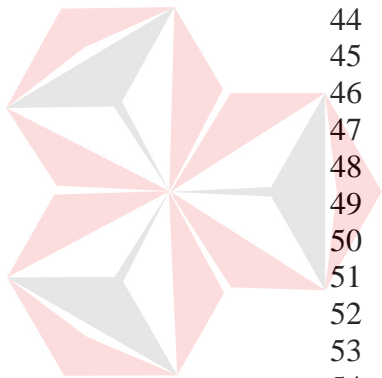
No	Tujuan	Input	Output yang diharapkan
3	Memastikan perubahan data container yang sudah ada dapat dilakukan	Tabel master_data kolom data ID : A40 dengan nama data A40GP diubah menjadi A40GA	Pada field kolom yang diubah akan tampil perubahan dari data yang baru. Data terbaru disimpan dalam tabel master_data
4	Memastikan penghapusan data container dapat dilakukan	Pada tampilan data dengan data ID : A40 tekan tombol Hapus	Data pada tabel master_data dengan data ID : A40 dihapus. Pada tampilan tidak nampak lagi

Tabel 3.9 Detil nilai data

Nomor	Nilai A40	Nilai A20	Nilai B40	Nilai B20	Nilai C40	Nilai C20
1	379	271	94	68	15	12
2	333	267	105	75	11	7
3	340	263	79	61	18	11
4	367	250	101	67	13	8
5	362	259	83	72	12	5
6	388	245	97	60	14	10
7	395	251	82	67	10	15
8	348	256	71	68	12	14
9	397	248	85	66	15	12
10	402	235	99	70	6	8
11	400	252	73	60	13	10
12	390	248	83	67	16	13
13			92	71	3	9
14			81	62	18	6
15			114	60	12	13
16			80	57	14	7
17			90	65	19	18
18			113	72	16	9
19			71	63	11	6
20			88	59	7	8
21			97	61	6	5
22			128	64	10	12
23			80	57	7	14
24			83	63	11	17
25			98	72	14	8
26			111	60	19	9
27			85	63	15	5
28			101	56	16	6

Tabel 3.9 Detil nilai data (lanjutan)

Nomor	Nilai A40	Nilai A20	Nilai B40	Nilai B20	Nilai C40	Nilai C20
29			87	64	17	9
30			92	61	12	6
31			83	77	14	8
32			86	54	8	7
33			99	62	16	11
34			102	58	13	14
35			91	60	10	10
36			105	68	12	9
37			100	59	9	7
38			92	65	10	10
39			108	60	14	13
40			102	51	7	6
41			97	63	27	11
42			94	56	18	5
43			100	67	24	9
44			109	66	8	12
45			97	62	15	5
46			102	70	9	7
47			92	57	10	10
48			99	59	12	14
49					20	15
50					9	10
51					13	13
52					11	15
53					6	6
54					16	8
55					15	7
56					17	5
57					18	11
58					14	8
59					6	13
60					17	7
61					9	10
62					15	9
63					20	4
64					15	7
65					10	8
66					8	9
67					17	6
68					13	5
69					16	10
70					21	14
71					7	13
72					15	10



Tabel 3.9 Detil nilai data (lanjutan)

Nomor	Nilai A40	Nilai A20	Nilai B40	Nilai B20	Nilai C40	Nilai C20
73					21	11
74					18	9
75					10	8
76					14	5
77					12	7
78					10	12
79					7	6
80					15	10
81					16	14
82					12	8
83					17	7
84					20	5

Tabel 3.10 Uji coba mengolah detil nilai data

No	Tujuan	Input	Output yang diharapkan
5	Memastikan pengubahan detil nilai data dapat dilakukan	Tabel detil_nilai_data kolom detil nilai data: 1 dengan nilai 379 diubah menjadi 300	Pada field kolom yang akan tampil perubahan dari data yang baru. Data terbaru disimpan dalam tabel detil_nilai_data

A.2 Uji coba mengolah data konfigurasi neural network

Tujuan dari uji coba ini adalah untuk memastikan keberhasilan dari obyek-obyek yang berada pada form data konfigurasi *neural network*.

Tabel 3.11 Data konfigurasi *neural network*

Kode NN	Jml Grup Input	Jmh Node Input	Jml Grup Output	Jml Node Hidden	Konstanta Belajar	Iterasi Belajar	Toleransi	Pola belajar
C40a1	1	2	1	4	0.01	<NULL>	<NULL>	20
C40a2	1	2	1	4	0.01	<NULL>	<NULL>	40
C40a3	1	2	1	4	0.01	<NULL>	<NULL>	60
C40b1	1	3	1	6	0.01	<NULL>	<NULL>	20
C40b2	1	3	1	6	0.01	<NULL>	<NULL>	40
C40b3	1	3	1	6	0.01	<NULL>	<NULL>	60

Tabel 3.11 Data konfigurasi *neural network* (lanjutan)

Kode NN	Jml Grup Input	Jmh Node Input	Jml Grup Output	Jml Node Hidden	Konstanta Belajar	Iterasi Belajar	Toleransi	Pola belajar
C40c1	1	4	1	8	0.01	<NULL>	<NULL>	20
C40c2	1	4	1	8	0.01	<NULL>	<NULL>	40
C40c3	1	4	1	8	0.01	<NULL>	<NULL>	60
C40d1	1	5	1	10	0.01	<NULL>	<NULL>	20
C40d2	1	5	1	10	0.01	<NULL>	<NULL>	40
C40d3	1	5	1	10	0.01	<NULL>	<NULL>	60
C40e1	1	6	1	12	0.01	<NULL>	<NULL>	20
C40e2	1	6	1	12	0.01	<NULL>	<NULL>	40
C40e3	1	6	1	12	0.01	<NULL>	<NULL>	60
C40f1	1	7	1	14	0.01	<NULL>	<NULL>	20
C40f2	1	7	1	14	0.01	<NULL>	<NULL>	40
C40f3	1	7	1	14	0.01	<NULL>	<NULL>	60
C20a1	1	2	1	4	0.01	<NULL>	<NULL>	20
C20a2	1	2	1	4	0.01	<NULL>	<NULL>	40
C20a3	1	2	1	4	0.01	<NULL>	<NULL>	60
C20b1	1	3	1	6	0.01	<NULL>	<NULL>	20
C20b2	1	3	1	6	0.01	<NULL>	<NULL>	40
C20b3	1	3	1	6	0.01	<NULL>	<NULL>	60
C20c1	1	4	1	8	0.01	<NULL>	<NULL>	20
C20c2	1	4	1	8	0.01	<NULL>	<NULL>	40
C20c3	1	4	1	8	0.01	<NULL>	<NULL>	60
C20d1	1	5	1	10	0.01	<NULL>	<NULL>	20
C20d2	1	5	1	10	0.01	<NULL>	<NULL>	40
C20d3	1	5	1	10	0.01	<NULL>	<NULL>	60
C20e1	1	6	1	12	0.01	<NULL>	<NULL>	20
C20e2	1	6	1	12	0.01	<NULL>	<NULL>	40
C20e3	1	6	1	12	0.01	<NULL>	<NULL>	60
C20f1	1	7	1	14	0.01	<NULL>	<NULL>	20
C20f2	1	7	1	14	0.01	<NULL>	<NULL>	40
C20f3	1	7	1	14	0.01	<NULL>	<NULL>	60

Tabel 3.12 Uji coba mengolah data *neural network*

No	Tujuan	Input	Output yang diharapkan
6	Memastikan penambahan data baru dapat dilakukan	Memasukkan jumlah grup input, jumlah node input, jumlah grup output, jumlah node hidden, konstanta belajar, toleransi, pola belajar data konfigurasi <i>neural network</i> ke dalam tabel NN . Kolom kodeNN akan di- <i>looping</i> dengan otomatis ketika tombol simpan data ditekan. Kolom lainnya harus dimasukkan oleh Admin	Data disimpan pada tabel NN
7	Memastikan perubahan data yang sudah ada dapat dilakukan	Tabel NN kolom : jml grup input = 1, jml node input = 2, jmlgrup output = 1, jml node hidden = 20, konst.belajar = 0.01, toleransi = 0, pola belajar = 20 diganti dengan jml grup input =1, jml node input = 2, jmlgrup output = 1, jml node hidden = 15, konst.belajar = 0.001, toleransi = 0, pola belajar = 20	Pada field kolom yang diubah akan tampil perubahan dari data yang baru. Data terbaru disimpan dalam tabel NN
8	Memastikan penghapusan data dapat dilakukan	Pada tampilan data dengan kodeNN = C40a1 tekan tombol hapus	Data pada tabel NN dengan kodeNN = C40a1 dihapus. Pada tampilan tidak nampak lagi

B Uji coba perhitungan sistem

Uji coba perhitungan sistem di sini menguji ketepatan perhitungan untuk proses-proses utama dalam sistem. Uji coba ini dilakukan pada proses pelatihan. Karena banyaknya data baik untuk data yang dilatihkan maka pada uji coba ini diberikan data sampel atau data yang diskenariokan.

1. Pelatihan

Tujuan dari uji coba perhitungan sistem pelatihan ini adalah untuk memastikan keberhasilan dari obyek-obyek yang berada pada form pelatihan.

Tabel 3.13 Data konfigurasi *neural network* yang dipakai

KodeNN	Jml Grup Input	Jml Grup Output	Jml Node Hidden	IterasiBelajar
C40a1	1	1	4	<NULL>
C40a2	1	1	4	<NULL>
C40a3	1	1	4	<NULL>
C40b1	1	1	6	<NULL>
C40b2	1	1	6	<NULL>
C40b3	1	1	6	<NULL>
C40c1	1	1	8	<NULL>
C40c2	1	1	8	<NULL>
C40c3	1	1	8	<NULL>
C40d1	1	1	10	<NULL>
C40d2	1	1	10	<NULL>
C40d3	1	1	10	<NULL>
C40e1	1	1	12	<NULL>
C40e2	1	1	12	<NULL>
C40e3	1	1	12	<NULL>
C40f1	1	1	14	<NULL>
C40f2	1	1	14	<NULL>
C40f3	1	1	14	<NULL>
C20a1	1	1	4	<NULL>
C20a2	1	1	4	<NULL>
C20a3	1	1	4	<NULL>
C20b1	1	1	6	<NULL>
C20b2	1	1	6	<NULL>
C20b3	1	1	6	<NULL>
C20c1	1	1	8	<NULL>
C20c2	1	1	8	<NULL>
C20c3	1	1	8	<NULL>
C20d1	1	1	10	<NULL>
C20d2	1	1	10	<NULL>
C20d3	1	1	10	<NULL>
C20e1	1	1	12	<NULL>
C20e2	1	1	12	<NULL>
C20e3	1	1	12	<NULL>
C20f1	1	1	14	<NULL>
C20f2	1	1	14	<NULL>
C20f3	1	1	14	<NULL>

Pada pelatihan *Neural Network* ini, langkah pertama adalah memilih data konfigurasi *neural network* seperti pada tabel diatas. Langkah selanjutnya adalah tekan tombol Start Pelatihan pada form pelatihan. Ketika pelatihan berhasil (MAE=0), langkah berikutnya adalah menyimpan bobot.

Tabel 3.14 Uji coba melatih data

No	Tujuan	Input	Output yang diharapkan
9	Memastikan pelatihan berhasil	Memilih kodeNN dan pola dipilih kemudian tombol mulai.	Setelah belajar tekan tombol dinyatakan berhasil). <i>Mean Absolute Error</i> mencapai 0 (pelatihan dinyatakan berhasil).
10	Memastikan penyimpanan bobot dapat dilakukan	Setelah pelatihan berhasil, bobot dapat menekan tombol simpan bobot.	Bobot akan disimpan pada tabel sesuai dengan <i>layernya</i> dan kodeNNnya.

2. Prediksi

Tujuan dari uji coba prediksi ini adalah untuk memastikan keberhasilan dari obyek-obyek yang berada pada form pelatihan.

Tabel 3.15 Data konfigurasi *neural network* yang dipakai

KodeNN	Jml Grup Input	Jml Grup Output	Jml Node Hidden	IterasiBelajar
C40a1	1	1	4	<NULL>
C40a2	1	1	4	<NULL>
C40a3	1	1	4	<NULL>
C40b1	1	1	6	<NULL>
C40b2	1	1	6	<NULL>
C40b3	1	1	6	<NULL>
C40c1	1	1	8	<NULL>
C40c2	1	1	8	<NULL>
C40c3	1	1	8	<NULL>
C40d1	1	1	10	<NULL>
C40d2	1	1	10	<NULL>
C40d3	1	1	10	<NULL>
C40e1	1	1	12	<NULL>

Tabel 3.15 Data konfigurasi *neural network* yang dipakai (lanjutan)

KodeNN	Jml Grup Input	Jml Grup Output	Jml Node Hidden	IterasiBelajar
C40e2	1	1	12	<NULL>
C40e3	1	1	12	<NULL>
C40f1	1	1	14	<NULL>
C40f2	1	1	14	<NULL>
C40f3	1	1	14	<NULL>
C20a1	1	1	4	<NULL>
C20a2	1	1	4	<NULL>
C20a3	1	1	4	<NULL>
C20b1	1	1	6	<NULL>
C20b2	1	1	6	<NULL>
C20b3	1	1	6	<NULL>
C20c1	1	1	8	<NULL>
C20c2	1	1	8	<NULL>
C20c3	1	1	8	<NULL>
C20d1	1	1	10	<NULL>
C20d2	1	1	10	<NULL>
C20d3	1	1	10	<NULL>
C20e1	1	1	12	<NULL>
C20e2	1	1	12	<NULL>
C20e3	1	1	12	<NULL>
C20f1	1	1	14	<NULL>
C20f2	1	1	14	<NULL>
C20f3	1	1	14	<NULL>

Pada prediksi *Neural Network* ini, langkah pertama adalah memilih data konfigurasi *neural network* seperti pada tabel diatas. Langkah selanjutnya adalah tekan tombol Start pelatihan pada form prediksi.

Tabel 3.16 Uji coba prediksi

No	Tujuan	Input	Output yang diharapkan
11	Memastikan proses prediksi berhasil	Memilih kodeNN dan pola dipilih kemudian tombol start pelatihan. Kemudian pilih data master	Setelah dan pola belajar tekan pelatihan. Hasil perhitungan prediksi dan hasil pengukuran ketepatan prediksi

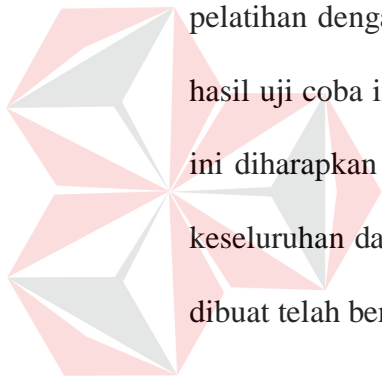
3.7.1 Desain Analisa

A Analisa fitur dasar sistem

Hasil uji coba dari keseluruhan uji yang dilakukan akan menentukan kelayakan fitur dasar sistem berdasarkan desain yang telah ditetapkan. Fitur-fitur dasar sistem dinilai layak bilamana keseluruhan hasil uji coba ini sesuai dengan *output* yang diharapkan.

B Analisa perhitungan sistem

Hasil uji coba dari keseluruhan uji yang dilakukan akan menentukan kelayakan proses pelatihan berdasarkan desain yang telah ditetapkan. Proses pelatihan dengan data yang diskenariokan ini dinilai layak bilamana keseluruhan hasil uji coba ini sesuai dengan *output* yang diharapkan. Data yang diskenariokan ini diharapkan dapat mewakili keseluruhan pola belajar yang ada pada tabel dan keseluruhan data nilai yang ada pada tabel untuk memastikan bahwa sistem yang dibuat telah benar.



BAB IV

IMPLEMENTASI DAN EVALUASI

Pembahasan dan pengujian sistem adalah tahap penjelasan secara nyata penggunaan dari rancang bangun program yang telah dibuat, mulai dari evaluasi sampai dengan implementasi.

Sebelum menjalankan aplikasi ini, ada hal yang harus diperhatikan yaitu kebutuhan sistem. Sesuai dengan kebutuhan untuk merancang sistem diperlukan perangkat keras dan perangkat lunak. Adapun perangkat keras yang dibutuhkan untuk menjalankan aplikasi ini yaitu :

4.1. Kebutuhan Sistem

Adapun Hardware yang digunakan dalam pembuatan sistem ini adalah :

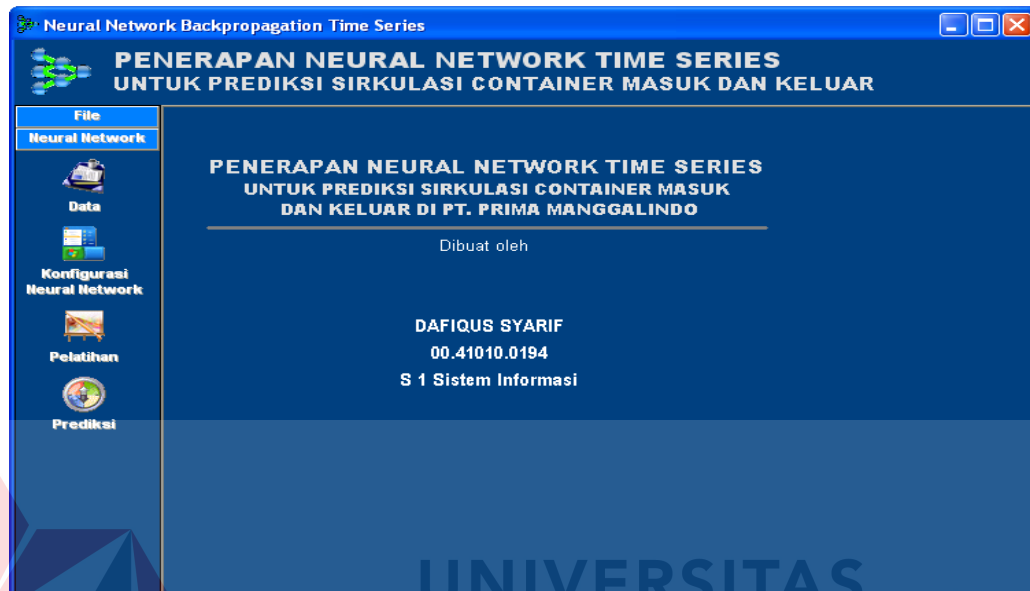
1. AMD Sempron Processor 3100
2. Harddisk 40 Gb
3. Monitor VGA 15.4" WXGA wide TFT LCD
4. DVDROM Philips 16 X
5. Memory 512 DDR
6. Keybord
7. Printer Canon Bjc – 265 sp
8. Mouse

Sedangkan software yang digunakan :

1. Microsoft SQL Server 2000
2. Microsoft Visual Basic 6.0
3. Power Designer 6.1

4.2. Analisa dan Implementasi Sistem

Pada form utama terdapat menu-menu yang dapat mengarahkan pada form-form yang ada, menu-menu tersebut, antara lain (1) menu file, (2) menu *neural network*.



Gambar 4.1 Tampilan menu utama



Gambar 4.2 Tampilan menu

Form ini menunjukkan sub menu file yang ada di menu utama terdiri dari menu : (1)About, (2) Setup Database, (3) Keluar aplikasi yang digunakan untuk keluar dari program / aplikasi.



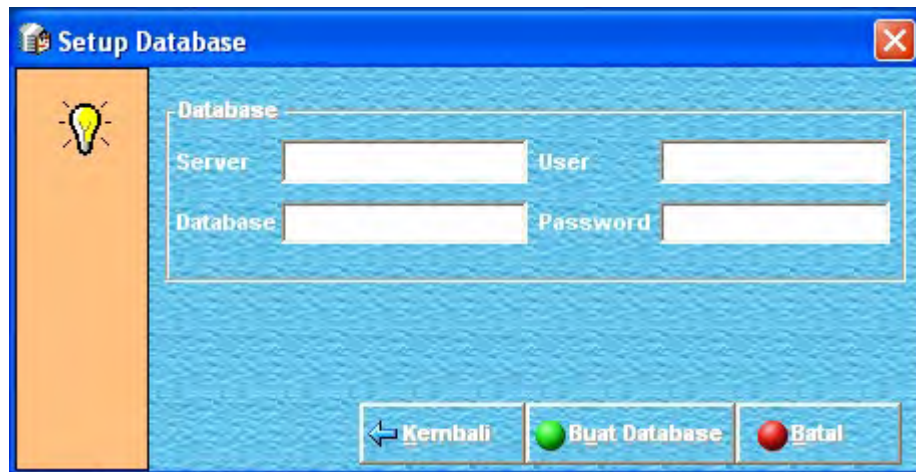
Gambar 4.3 Tampilan menu about

Form diatas merupakan tampilan sub menu about di dalam menu file yang berisi tentang tampilan aplikasi penerapan *neural network* untuk prediksi sirkulasi container masuk dan keluar.



Gambar 4.4 Tampilan menu setup database awal

Tampilan ini digunakan untuk proses database pada aplikasi, terdiri dari (1) pembuatan database baru, (2) memakai database yang telah ada.



Gambar 4.5 Tampilan menu setup database baru

Form ini merupakan sub menu setup *database*, yang digunakan untuk membuat suatu *database* baru guna melakukan koneksi ke aplikasi.



Gambar 4.6 Tampilan *neural network*

Tampilan ini menunjukkan sub-sub menu di dalam *neural network* yaitu (1) data, (2) konfigurasi *neural network*, (3) pelatihan, (4) prediksi.

Detil Nilai Data	
Nomor	Nilai
1	15.00
2	11.00
3	18.00
4	13.00
5	12.00
6	14.00
7	10.00
8	12.00
9	15.00
10	6.00
11	13.00
12	16.00

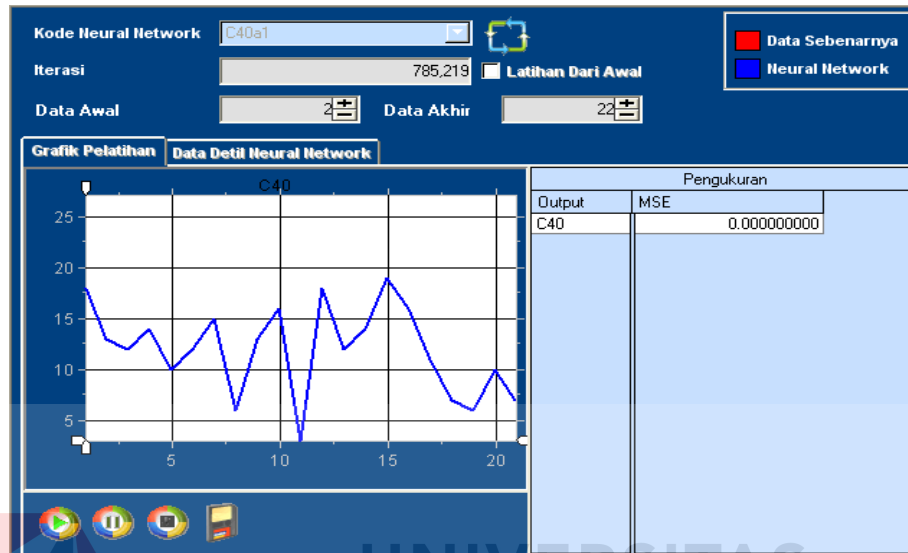
Gambar 4.7 Tampilan data

Tampilan diatas merupakan sub menu dari menu *neural network* yang digunakan untuk menginputkan data baru serta menampilkan data-data yang sudah tersimpan (*database*) guna memilih (*sorting*) data-data tertentu dijadikan acuan untuk melakukan proses prediksi.

Detil Grup Input					
Grup ID	Jml Neuron	Obyek ID	Batas Max	Batas Min	Tipe Aktivasi
1	2.00	C40	27.00	3.00	Bipolar Sigmoid

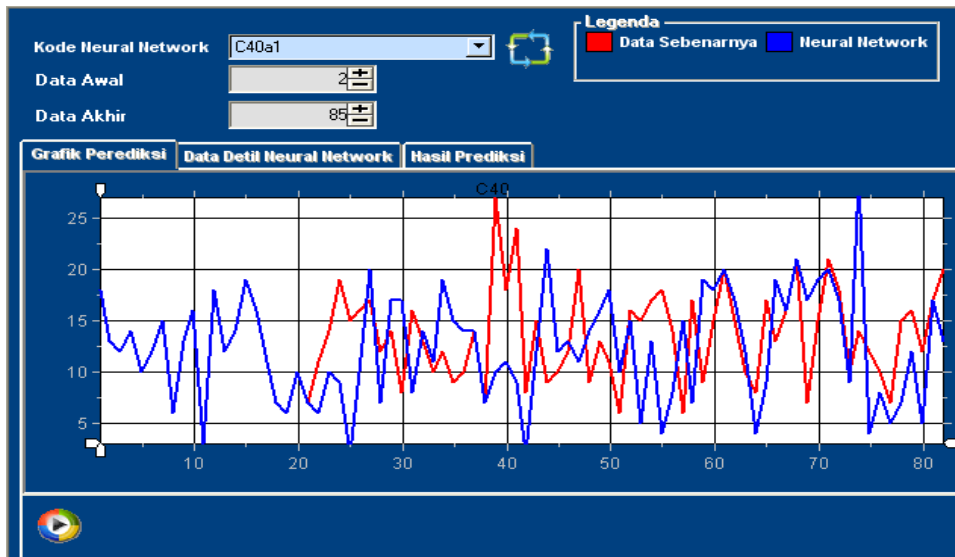
Gambar 4.8 Tampilan konfigurasi *neural network* (NN)

Menu ini merupakan sub menu dari menu neural network yang digunakan untuk menginputkan, menampilkan serta mengupdate semua konfigurasi *neural network* sebagai acuan prediksi pergerakan sirkulasi container masuk dan keluar.



Gambar 4.9 Tampilan pelatihan *neural network*

Form diatas merupakan tab grafik pelatihan pada sub menu pelatihan yang digunakan untuk menampilkan proses pelatihan (prediksi) berupa grafik data sebenarnya dan grafik *neural network*, serta menampilkan proses pengukuran berupa output yang diinginkan , MSE (*Mean Squared Error*).



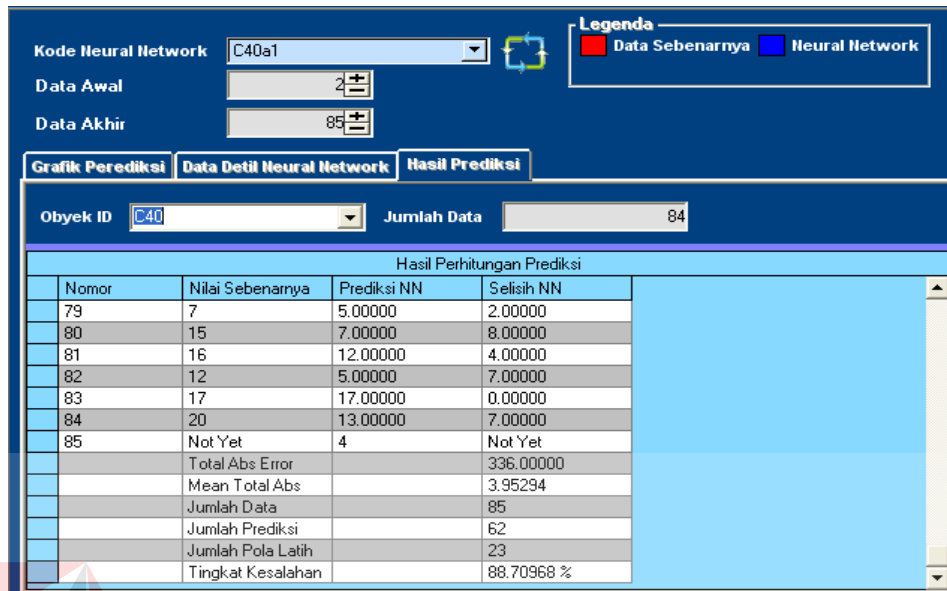
Gambar 4.10 Tampilan grafik prediksi *neural network*

Form ini merupakan tampilan tab grafik prediksi pada sub menu prediksi yang digunakan untuk menampilkan grafik hasil proses pelatihan berupa grafik data sebenarnya dan grafik *neural network*.

Grup ID	Jml Neuron	Obyek ID	Batas Max	Batas Min	Tipe Aktivasi
1	2.00	C40	27.00	3.00	Bipolar Sigmoid

Gambar 4.11 Tampilan data detil prediksi *neural network*

Form ini merupakan tampilan tab data detil *neural network* pada sub menu prediksi untuk menampilkan data detil *neural network* yang dihasilkan selama proses prediksi.



Gambar 4.12 Tampilan hasil prediksi *neural network*

Form ini merupakan tampilan tab hasil prediksi *neural network* pada sub menu prediksi untuk menampilkan hasil pengukuran ketepatan prediksi dan semua hasil perhitungan prediksi yang dihasilkan selama proses pelatihan data.

4.3 Desain Uji Coba dan Analisa

Desain uji coba dan analisa adalah untuk melihat bagaimana suatu proses pada sistem bekerja, tujuan dari proses, dan *output* yang diharapkan. Uji coba dan analisa yang dilakukan meliputi uji coba dan analisa fitur dasar sistem, uji coba dan analisa perhitungan sistem serta uji coba dan analisa validasi sistem.

4.3.1 Desain Uji Coba

A Uji coba fitur dasar sistem

Uji coba fitur dasar sistem ini dilakukan dengan metode *black box testing*, yang bertujuan untuk memastikan fitur-fitur dasar dalam sistem telah berfungsi dengan benar. Uji coba fitur dasar sistem dalam tugas akhir ini meliputi :

A.1 Uji coba mengolah data

Tujuan dari uji coba ini adalah untuk memastikan keberhasilan dari obyek-obyek yang berada pada form data.

Tabel 4.1 Contoh data

Nama Field	Data-1	Data-2	Data-3	Data-4	Data-5	Data-6
Data ID	A40	A20	B40	B20	C40	C20
Nama Data	A40GP	A20GP	B40GP	B20GP	C40GP	C20GP
Keterangan	Bulan	Bulan	Minggu	Minggu	Hari	Hari

Tabel 4.2 Uji coba mengolah data

No	Tujuan	Input	Output yang diharapkan	Status
1	Memastikan data container	Tabel master_data	Seluruh data dari tabel master_data	Sukses
2	Memastikan penambahan data baru dapat dilakukan	Memasukkan data container ke dalam tabel master_data.	Data container yang baru dimasukkan dapat disimpan pada tabel master_data	Sukses (Gambar 4.13)
3	Memastikan perubahan data container yang sudah ada dapat dilakukan	Tabel master_data kolom data ID : A40 dengan nama data A40GP diubah menjadi A20GP	Pada field kolom yang diubah akan tampil perubahan dari data yang baru. Data terbaru disimpan dalam tabel master_data	Sukses (Gambar 4.13)
4	Memastikan penghapusan data container dapat dilakukan	Pada tampilan data dengan data ID : A40 tekan tombol Hapus	Data pada tabel master_data dengan data ID : A40 dihapus. Pada tampilan tidak nampak lagi	Sukses



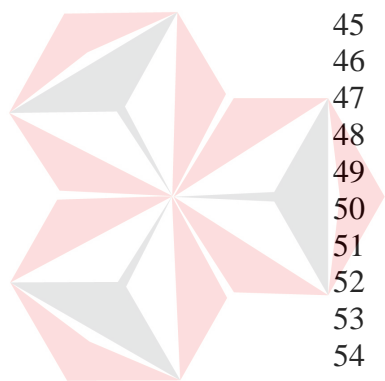
Gambar 4.13 Setup Data Berhasil

Tabel 4.3 Detil nilai data

Nomor	Nilai A40	Nilai A20	Nilai B40	Nilai B20	Nilai C40	Nilai C20
1	379	271	94	68	15	12
2	333	267	105	75	11	7
3	340	263	79	61	18	11
4	367	250	101	67	13	8
5	362	259	83	72	12	5
6	388	245	97	60	14	10
7	395	251	82	67	10	15
8	348	256	71	68	12	14
9	397	248	85	66	15	12
10	402	235	99	70	6	8
11	400	252	73	60	13	10
12	390	248	83	67	16	13
13			92	71	3	9
14			81	62	18	6
15			114	60	12	13
16			80	57	14	7
17			90	65	19	18
18			113	72	16	9
19			71	63	11	6
20			88	59	7	8
21			97	61	6	5
22			128	64	10	12
23			80	57	7	14
24			83	63	11	17
25			98	72	14	8
26			111	60	19	9
27			85	63	15	5
28			101	56	16	6
29			87	64	17	9
30			92	61	12	6

Tabel 4.3 Detil nilai data (lanjutan)

Nomor	Nilai A40	Nilai A20	Nilai B40	Nilai B20	Nilai C40	Nilai C20
31			83	77	14	8
32			86	54	8	7
33			99	62	16	11
34			102	58	13	14
35			91	60	10	10
36			105	68	12	9
37			100	59	9	7
38			92	65	10	10
39			108	60	14	13
40			102	51	7	6
41			97	63	27	11
42			94	56	18	5
43			100	67	24	9
44			109	66	8	12
45			97	62	15	5
46			102	70	9	7
47			92	57	10	10
48			99	59	12	14
49					20	15
50					9	10
51					13	13
52					11	15
53					6	6
54					16	8
55					15	7
56					17	5
57					18	11
58					14	8
59					6	13
60					17	7
61					9	10
62					15	9
63					20	4
64					15	7
65					10	8
66					8	9
67					17	6
68					13	5
69					16	10
70					21	14
71					7	13
72					15	10
73					21	11

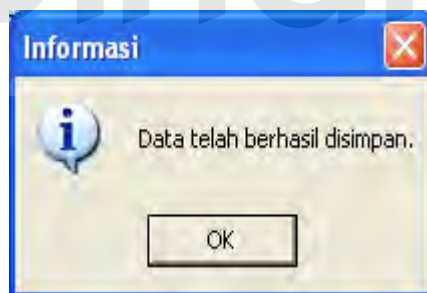


Tabel 4.3 Detil nilai data (lanjutan)

Nomor	Nilai A40	Nilai A20	Nilai B40	Nilai B20	Nilai C40	Nilai C20
74					18	9
75					10	8
76					14	5
77					12	7
78					10	12
79					7	6
80					15	10
81					16	14
82					12	8
83					17	7
84					20	5

Tabel 4.4 Uji coba mengolah detil nilai data

No	Tujuan	Input	Output yang diharapkan	Status
5	Memastikan perubahan detil nilai data dapat dilakukan	Tabel detil_nilai_data kolom detil nilai data: 1 dengan nilai 379 diubah menjadi 300	Pada field kolom yang diubah akan tampil perubahan dari data yang baru. Data terbaru disimpan dalam tabel detil_nilai_data	Sukses (gambar 4.14)



Gambar 4.14 Setup Data Berhasil

A.2 Uji coba mengolah data konfigurasi neural network

Tujuan dari uji coba ini adalah untuk memastikan keberhasilan dari obyek-obyek yang berada pada form data konfigurasi *neural network*.

Tabel 4.5 Data konfigurasi *neural network*

Kode NN	Jml Grup Input	Jmh Node Input	Jml Grup Output	Jml Node Hidden	Konstanta Belajar	Iterasi Belajar	Toleransi	Pola belajar
C40a1	1	2	1	4	0.01	<NULL>	<NULL>	20
C40a2	1	2	1	4	0.01	<NULL>	<NULL>	40
C40a3	1	2	1	4	0.01	<NULL>	<NULL>	60
C40b1	1	3	1	6	0.01	<NULL>	<NULL>	20
C40b2	1	3	1	6	0.01	<NULL>	<NULL>	40
C40b3	1	3	1	6	0.01	<NULL>	<NULL>	60
C40c1	1	4	1	8	0.01	<NULL>	<NULL>	20
C40c2	1	4	1	8	0.01	<NULL>	<NULL>	40
C40c3	1	4	1	8	0.01	<NULL>	<NULL>	60
C40d1	1	5	1	10	0.01	<NULL>	<NULL>	20
C40d2	1	5	1	10	0.01	<NULL>	<NULL>	40
C40d3	1	5	1	10	0.01	<NULL>	<NULL>	60
C40e1	1	6	1	12	0.01	<NULL>	<NULL>	20
C40e2	1	6	1	12	0.01	<NULL>	<NULL>	40
C40e3	1	6	1	12	0.01	<NULL>	<NULL>	60
C40f1	1	7	1	14	0.01	<NULL>	<NULL>	20
C40f2	1	7	1	14	0.01	<NULL>	<NULL>	40
C40f3	1	7	1	14	0.01	<NULL>	<NULL>	60
C20a1	1	2	1	4	0.01	<NULL>	<NULL>	20
C20a2	1	2	1	4	0.01	<NULL>	<NULL>	40
C20a3	1	2	1	4	0.01	<NULL>	<NULL>	60
C20b1	1	3	1	6	0.01	<NULL>	<NULL>	20
C20b2	1	3	1	6	0.01	<NULL>	<NULL>	40
C20b3	1	3	1	6	0.01	<NULL>	<NULL>	60
C20c1	1	4	1	8	0.01	<NULL>	<NULL>	20
C20c2	1	4	1	8	0.01	<NULL>	<NULL>	40
C20c3	1	4	1	8	0.01	<NULL>	<NULL>	60
C20d1	1	5	1	10	0.01	<NULL>	<NULL>	20
C20d2	1	5	1	10	0.01	<NULL>	<NULL>	40
C20d3	1	5	1	10	0.01	<NULL>	<NULL>	60
C20e1	1	6	1	12	0.01	<NULL>	<NULL>	20
C20e2	1	6	1	12	0.01	<NULL>	<NULL>	40
C20e3	1	6	1	12	0.01	<NULL>	<NULL>	60
C20f1	1	7	1	14	0.01	<NULL>	<NULL>	20
C20f2	1	7	1	14	0.01	<NULL>	<NULL>	40
C20f3	1	7	1	14	0.01	<NULL>	<NULL>	60

Tabel 4.6 Uji coba mengolah data *neural network*

No	Tujuan	Input	Output yang diharapkan	Status
6	Memastikan penambahan data baru dapat dilakukan	Memasukkan jumlah grup input, jumlah node input, jumlah grup output, jumlah node hidden, konstanta belajar, toleransi, pola belajar data konfigurasi <i>neural network</i> ke dalam tabel NN . Kolom kodeNN akan di- <i>looping</i> dengan otomatis ketika tombol simpan data ditekan. Kolom lainnya harus dimasukkan oleh Admin	Data disimpan pada tabel NN	Sukses (Gambar 4.15)
7	Memastikan perubahan data yang sudah ada dapat dilakukan	Tabel NN kolom : jml grup input = 1, jml node input = 2, jmlgrup output = 1, jml node hidden = 20, konst.belajar = 0.01, toleransi = 0, pola belajar = 20 diganti dengan jml grup input =1, jml node input = 2, jmlgrup output = 1, jml node hidden = 15, konst.belajar = 0.001, toleransi = 0, pola belajar = 20	Pada field kolom yang diubah akan tampil perubahan dari data yang baru. Data terbaru disimpan dalam tabel NN	Sukses (Gambar 4.15)
8	Memastikan penghapusan data dapat dilakukan	Pada tampilan data dengan kodeNN = C40a1 tekan tombol Hapus	Data pada tabel NN dengan kodeNN = C40a1 dihapus. Pada tampilan tidak nampak lagi	Sukses



Gambar 4.15 Setup Data Berhasil

B Uji coba perhitungan sistem

Uji coba perhitungan sistem di sini menguji ketepatan perhitungan untuk proses-proses utama dalam sistem. Uji coba ini dilakukan pada proses pelatihan.

Karena banyaknya data baik untuk data yang dilatihkan maka pada uji coba ini diberikan data sampel atau data yang diskenariokan.

1. Pelatihan

Tujuan dari uji coba perhitungan sistem pelatihan ini adalah untuk memastikan keberhasilan dari obyek-obyek yang berada pada form pelatihan.

Tabel 4.7 Data konfigurasi *neural network* yang dipakai

KodeNN	Jml Grup Input	Jml Grup Output	Jml Node Hidden	IterasiBelajar
C40a1	1	1	4	<NULL>
C40a2	1	1	4	<NULL>
C40a3	1	1	4	<NULL>
C40b1	1	1	6	<NULL>
C40b2	1	1	6	<NULL>
C40b3	1	1	6	<NULL>
C40c1	1	1	8	<NULL>
C40c2	1	1	8	<NULL>
C40c3	1	1	8	<NULL>
C40d1	1	1	10	<NULL>
C40d2	1	1	10	<NULL>
C40d3	1	1	10	<NULL>
C40e1	1	1	12	<NULL>

Tabel 4.7 Data konfigurasi *neural network* yang dipakai (lanjutan)

KodeNN	Jml Grup Input	Jml Grup Output	Jml Node Hidden	IterasiBelajar
C40e2	1	1	12	<NULL>
C40e3	1	1	12	<NULL>
C40f1	1	1	14	<NULL>
C40f2	1	1	14	<NULL>
C40f3	1	1	14	<NULL>
C20a1	1	1	4	<NULL>
C20a2	1	1	4	<NULL>
C20a3	1	1	4	<NULL>
C20b1	1	1	6	<NULL>
C20b2	1	1	6	<NULL>
C20b3	1	1	6	<NULL>
C20c1	1	1	8	<NULL>
C20c2	1	1	8	<NULL>
C20c3	1	1	8	<NULL>
C20d1	1	1	10	<NULL>
C20d2	1	1	10	<NULL>
C20d3	1	1	10	<NULL>
C20e1	1	1	12	<NULL>
C20e2	1	1	12	<NULL>
C20e3	1	1	12	<NULL>
C20f1	1	1	14	<NULL>
C20f2	1	1	14	<NULL>
C20f3	1	1	14	<NULL>

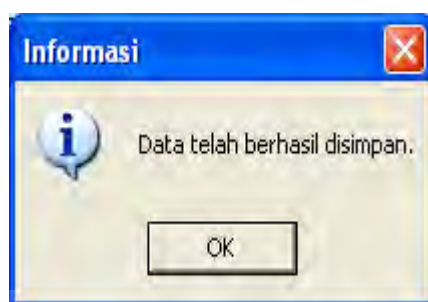
Pada pelatihan *Neural Network* ini, langkah pertama adalah memilih data konfigurasi *neural network* seperti pada tabel diatas. Langkah selanjutnya adalah tekan tombol Start Pelatihan pada form pelatihan. Ketika pelatihan berhasil (MAE=0), langkah berikutnya adalah menyimpan bobot.

Tabel 4.8 Uji coba melatih data

No	Tujuan	Input	Output yang diharapkan	Status
9	Memastikan pelatihan berhasil	Memilih kodeNN dan pola belajar kemudian tekan tombol mulai.	<i>Mean Absolute Error</i> mencapai 0 (pelatihan dinyatakan berhasil).	Sukses

Tabel 4.8 Uji coba melatih data (lanjutan)

No	Tujuan	Input	Output yang diharapkan	Status
10	Memastikan penyimpanan bobot dapat dilakukan	Setelah pelatihan berhasil, bobot disimpan dengan menekan tombol simpan bobot.	Bobot akan disimpan pada tabel sesuai dengan <i>layernya</i> dan kodeNNnya.	Sukses (Gambar 4.16)



Gambar 4.16 Setup Data Berhasil

2. Prediksi

Tujuan dari uji coba prediksi ini adalah untuk memastikan keberhasilan dari obyek-obyek yang berada pada form pelatihan.

Tabel 4.9 Data konfigurasi *neural network* yang dipakai

KodeNN	Jml Grup Input	Jml Grup Output	Jml Node Hidden	IterasiBelajar
C40a1	1	1	4	<NULL>
C40a2	1	1	4	<NULL>
C40a3	1	1	4	<NULL>
C40b1	1	1	6	<NULL>
C40b2	1	1	6	<NULL>
C40b3	1	1	6	<NULL>
C40c1	1	1	8	<NULL>
C40c2	1	1	8	<NULL>
C40c3	1	1	8	<NULL>
C40d1	1	1	10	<NULL>
C40d2	1	1	10	<NULL>
C40d3	1	1	10	<NULL>

Tabel 4.9 Data konfigurasi *neural network* yang dipakai (lanjutan)

KodeNN	Jml Grup Input	Jml Grup Output	Jml Node Hidden	IterasiBelajar
C40e1	1	1	12	<NULL>
C40e2	1	1	12	<NULL>
C40e3	1	1	12	<NULL>
C40f1	1	1	14	<NULL>
C40f2	1	1	14	<NULL>
C40f3	1	1	14	<NULL>
C20a1	1	1	4	<NULL>
C20a2	1	1	4	<NULL>
C20a3	1	1	4	<NULL>
C20b1	1	1	6	<NULL>
C20b2	1	1	6	<NULL>
C20b3	1	1	6	<NULL>
C20c1	1	1	8	<NULL>
C20c2	1	1	8	<NULL>
C20c3	1	1	8	<NULL>
C20d1	1	1	10	<NULL>
C20d2	1	1	10	<NULL>
C20d3	1	1	10	<NULL>
C20e1	1	1	12	<NULL>
C20e2	1	1	12	<NULL>
C20e3	1	1	12	<NULL>
C20f1	1	1	14	<NULL>
C20f2	1	1	14	<NULL>
C20f3	1	1	14	<NULL>

Pada prediksi *Neural Network* ini, langkah pertama adalah memilih data konfigurasi *neural network* seperti pada tabel diatas. Langkah selanjutnya adalah tekan tombol Start pelatihan pada form prediksi.

Tabel 4.10 Uji coba prediksi

No	Tujuan	Input	Output yang diharapkan
11	Memastikan proses prediksi berhasil	Memilih kodeNN dan pola dipilih kemudian tombol start pelatihan. Kemudian pilih data master	Setelah belajar tekan pelatihan. Hasil perhitungan prediksi dan hasil pengukuran ketepatan prediksi

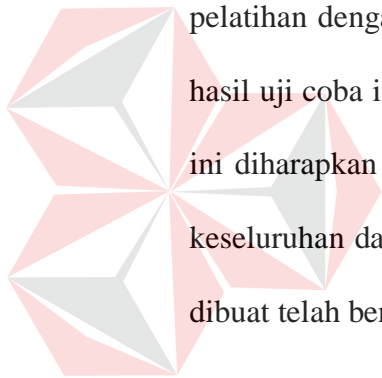
4.3.2 Desain Analisa

A Analisa fitur dasar sistem

Hasil uji coba dari keseluruhan uji yang dilakukan akan menentukan kelayakan fitur dasar sistem berdasarkan desain yang telah ditetapkan. Fitur-fitur dasar sistem dinilai layak bilamana keseluruhan hasil uji coba ini sesuai dengan *output* yang diharapkan.

B Analisa perhitungan sistem

Hasil uji coba dari keseluruhan uji yang dilakukan akan menentukan kelayakan proses pelatihan berdasarkan desain yang telah ditetapkan. Proses pelatihan dengan data yang diskenariokan ini dinilai layak bilamana keseluruhan hasil uji coba ini sesuai dengan *output* yang diharapkan. Data yang diskenariokan ini diharapkan dapat mewakili keseluruhan pola belajar yang ada pada tabel dan keseluruhan data nilai yang ada pada tabel untuk memastikan bahwa sistem yang dibuat telah benar.



BAB V

PENUTUP

5.1 Kesimpulan

Dari pembuatan buku Tugas Akhir ini dapat diambil beberapa kesimpulan sebagai berikut :

Berdasarkan hasil uji coba validasi sistem yang telah dilakukan, membuktikan bahwa aplikasi telah dapat berjalan sesuai dengan yang diharapkan. Implementasi *neural network* time series untuk prediksi jumlah container yang keluar dapat dilakukan dengan menerapkan fungsi aktivasi non linier yaitu binary sigmoid atau bipolar sigmoid, hal ini disebabkan data yang ada bersifat non linier.

5.2 Saran

Untuk pengembangan lebih lanjut dari penulisan Tugas Akhir ini, berikut berisi saran yang berguna untuk pemikiran maupun implementasi :

1. Konfigurasi dan kinerja neural network dapat ditingkatkan dengan beberapa algoritma optimasi dan kemungkinan untuk mempergunakan algoritma selain *backpropagation* dapat dipertimbangkan untuk meningkatkan kinerja sistem.
2. Akan lebih baik apabila data pelatihan diperbanyak, untuk menghasilkan peramalan yang lebih akurat lagi.

DAFTAR PUSTAKA

- Chappelier J.C, Grumbach A, 1994, *Time In Neural Networks*, Sigart Buletin, ACM
- Elman, J.L, 1991, *Finding Structure In Time*, *Cognitive Science*.
- Fauset, Lauren, 1994, *Fundamental of Neural Network: architectures, algorithms, and applications*, *Prentice Hall, Inc*.
- Fu, LiMin., 1994, *Neural Networks in Computer Intelligence*, McGraw-Hill, Inc, USA.
- Haykin, Simon, 1994, *Neural Network A Comprehensive Foundation*, Prentice Hall, Inc.
- Lee C.H, Park K.C, 1992, *Prediction of Monthly Transition of The Composition Stock Price Index Using Recurrent Back-propagation*, North-Holland, Amsterdam.
- Makridakis, S , Wheelwright, S. C. , McGee V.E. , *Metode dan Aplikasi peramalan jilid 1*.
- Pandjaitan, Lanny W., 2007, *Dasar-dasar Komputasi Cerdas*, Andi Offset, Yogyakarta.
- Purnomo, Mauridhi Hery., Kurniawan, Agus., 2006, *Supervised Neural Networks dan Aplikasinya*, Graha Ilmu, Yogyakarta.
- Rao, Valuru, Dr. & Rao, Hayagriva, 1995, *C++ Neural Network and Fuzzy Logic 2nd Edition*, MIS:Press.
- Rohwer R, 1994, *The Time Dimension of Neural Network Model*, Sigart Buletin, ACM
- Setiawan, Kuswara., 2003, *Paradigma Sistem Cerdas*, Bayumedia Publishing, Malang.
- Weigend A.S, Rumelhart D.E, Huberman B.A, 1990, *Backpropagation weight-Elimination and Time Series Prediction*, San Malco, CA