

## ANALISA PREDIKSI TINGKAT PENGANGGURAN DENGAN JARINGAN SYARAF TIRUAN

Prisa Marga Kusumantara, I Gede Susrama  
Jurusan Teknik Informatika, FTI, UPN "Veteran" Jatim  
e-mail: gde\_susrama@yahoo.com

### ABSTRACT

*Up to now, unemployment is one of unresolved problems despite all efforts carried out to lower the number and always an interesting topic to discuss. This research tried to develop a prediction system to depict unemployment growth year by year using artificial neural network with backpropagation training. Backpropagation was used because of its capability to recognize trained data pattern and was suitable to be used in prediction system with a lot of data input. This application was developed using Visual Basic 6 and aimed to give prediction on unemployment growth to determine the number of workfields should be provided.*

**Keywords:** Artificial Neural Network, Unemployment, Backpropagation, Prediction

### 1. PENDAHULUAN

**"Pengangguran** berarti individu yang tidak mempunyai pekerjaan atau sumber pendapatan tetap. Penganggur terbagi kepada dua, yaitu mereka yang coba mencari pekerjaan secara aktif tetapi gagal, dan mereka yang tidak coba mencari pekerjaan. Secara tepat, hanya mereka yang coba mencari pekerjaan secara aktif tetapi gagal dianggap sebagai penganggur. Sementara mereka yang tidak coba mencari pekerjaan tidak dianggap penganggur karena mereka berkemungkinan hendak berehat, atau sedang menyambung pembelajaran mereka. "Kadar pengangguran" pula ialah jumlah penganggur dibahagi dengan jumlah tenaga kerja, sama ada menganggur ataupun tidak" []

Kolapsnya perekonomian Indonesia sejak krisis pada pertengahan 1997 membuat kondisi ketenaga kerjaan Indonesia ikut memburuk. Sejak itu, pertumbuhan ekonomi Indonesia juga tidak pernah mencapai 7-8 persen. Padahal, masalah pengangguran erat kaitannya dengan pertumbuhan ekonomi. Jika pertumbuhan ekonomi ada, otomatis penyerapan tenaga kerja juga ada. Setiap pertumbuhan ekonomi satu persen, tenaga kerja yang terserap bisa mencapai 400 ribu orang. Jika pertumbuhan ekonomi Indonesia hanya 3-4 persen, tentunya hanya akan menyerap 1,6 juta tenaga kerja, sementara pencari kerja mencapai rata-rata 2,5 juta pertahun. Sehingga, setiap tahun pasti ada sisa.

Pada 1997, jumlah penganggur terbuka mencapai 4,18 juta. Selanjutnya, pada 1999 (6,03 juta), 2000 (5,81 juta), 2001 (8,005 juta), 2002 (9,13 juta) dan 2003 (11,35 juta). Sementara itu, data pekerja dan pengangguran menunjukkan, pada 2001: usia kerja (144,033 juta), angkatan kerja (98,812 juta), penduduk yang kerja (90,807 juta), penganggur terbuka (8,005 juta), setengah penganggur terpaksa (6,010 juta), setengah penganggur sukarela (24,422 juta); pada 2002: usia kerja (148,730 juta), angkatan kerja (100,779 juta), penduduk yang kerja (91,647 juta), penganggur

terbuka (9,132 juta), setengah penganggur terpaksa (28,869 juta), setengah penganggur sukarela (-).

Untuk dapat menentukan jumlah lapangan kerja yang harus disediakan dalam setiap tahunnya maka di perlukan sebuah sistem yang dapat memprediksi jumlah pengangguran yang ada. Jaringan Syaraf Tiruan adalah sistem komputasi yang didasarkan pemodelan sistem saraf biologis (*neurons*) melalui pendekatan dari sifat-sifat komputasi biologis (*biological computation*). Dalam memecahkan permasalahan sebuah sistem yang menggunakan Jaringan Syaraf Tiruan dilatih terlebih dahulu untuk mengenali pola-pola data yang ada pada permasalahan kemudian sistem menghasilkan bobot-bobot dan dapat menentukan hasil prediksi pola data periode selanjutnya.

Karena itu maka Jaringan Syaraf Tiruan dapat digunakan untuk memprediksi jumlah pengangguran pada tahun-tahun mendatang berdasarkan data pengangguran pada tahun-tahun sebelumnya.

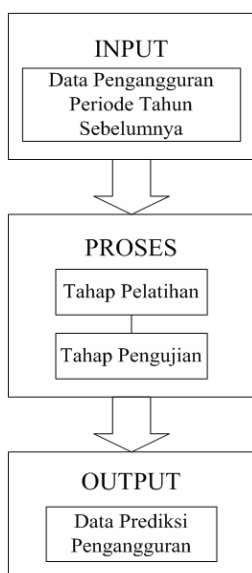
Berdasarkan hal-hal di atas maka pada penelitian ini diterapkan Jaringan Syaraf Tiruan menggunakan metode pelatihan *Backpropagation* untuk prediksi tingkat pengangguran di propinsi Jawa Timur, dengan tujuan membuat sistem prediksi untuk menentukan jumlah pengangguran di masa mendatang (Satu tahun kedepan dari data masukan terakhir).

### 2. ANALISA PERMASALAHAN

Semakin meningkatnya jumlah pengangguran yang ada di sebuah daerah khususnya di Jawa Timur membuat daerah tersebut diharapkan dapat menyediakan lapangan pekerjaan bagi pengangguran tersebut. Pihak yang terkait dengan ketenaga kerjaan diharapkan dapat memprediksikan jumlah pengangguran yang ada pada masa mendatang sehingga pihak tersebut dapat mempersiapkan jumlah lapangan kerja yang harus disiapkan. Mengingat pentingnya prediksi tentang jumlah pengangguran di masa yang akan datang maka diperlukan sebuah sistem yang dapat

memprediksikan jumlah pengangguran di waktu yang akan datang dengan menggunakan data jumlah pengangguran pada periode-periode sebelumnya. Proses mula-mula dilatihkan dengan data pengangguran periode masa lalu, misalnya data yang ada adalah data pengangguran tahun 2001, 2002, dan 2003. Maka data tahun 2001 dan 2002 dijadikan sebagai masukan dan untuk data 2003 digunakan sebagai target keluaran yang diinginkan. Dari penyesuaian antara data masukan dan data keluaran Jaringan Syaraf Tiruan akan menghasilkan bobot-bobot yang dapat memetakan data masukan ke data target keluaran sistem yang diinginkan. Apabila *error* (kesalahan) yang dihasilkan oleh Jaringan Syaraf Tiruan sudah lebih kecil di bandingkan toleransi error yang ditetapkan maka pelatihan Jaringan Syaraf Tiruan tersebut sudah dapat di sebut optimal.

Gambaran umum dari sistem yang digunakan dalam sistem prediksi tingkat pengangguran adalah sebagai berikut:



Gambar 1. Gambaran Umum Sistem

### 3. NORMALISASI DATA

Normalisasi data adalah proses perubahan data pengangguran asli menjadi data yang *range* nya antara 0,1 dan 0,9 karena fungsi aktivasi yang digunakan adalah fungsi sigmoid biner yang nilai fungsi tersebut tidak pernah mencapai 0 ataupun 1 (Siang, JJ, 2005).

Rumus yang digunakan pada proses ini adalah:

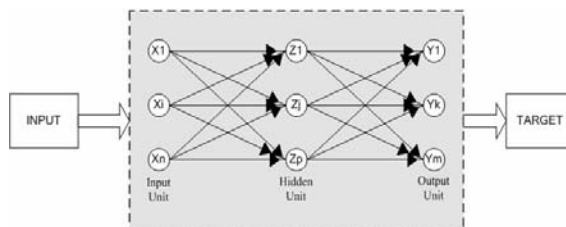
$$X' = \frac{0.8 * (X - a)}{b - a} + 0.1 \quad (1)$$

Keterangan:

- X' = Data yang telah di normalisasi
- X = Data yang belum dinormalisasi
- a = Data terkecil
- b = Data terbesar

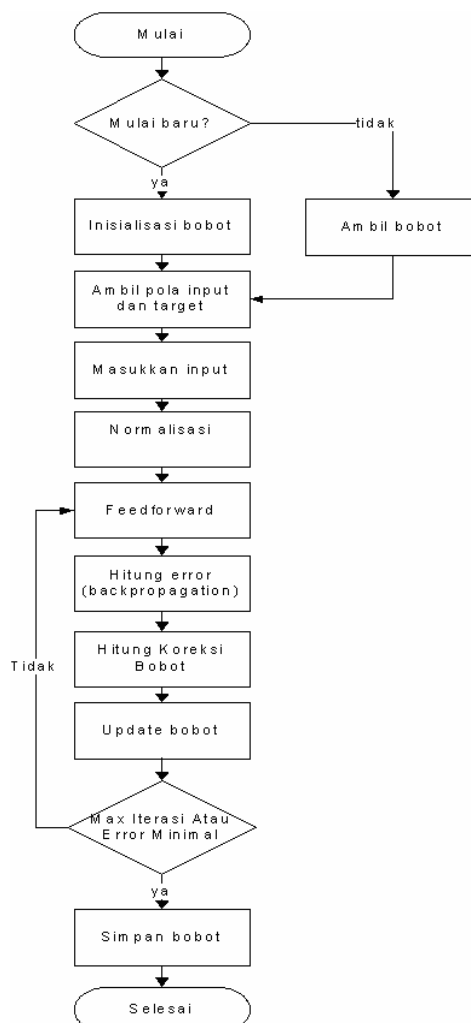
### 3.1 Tahap pelatihan

Tahap ini adalah proses pengenalan pola-pola data yang telah di normalisasi agar sistem dapat menentukan bobot-bobot yang dapat memetakan antara data *input* dengan data target *output* yang diinginkan.



Gambar 2. Blok Diagram Tahap Pelatihan

*Input* yang digunakan adalah berupa data pengangguran perbulan dari periode lalu misalkan tahun 2000 dan tahun 2001. Target yang digunakan adalah berupa nilai *output* yang diinginkan misalkan data pengangguran periode tahun 2002. Masing-masing *input* yang dilatihkan memiliki target *output* tersendiri. Alur proses pelatihannya dapat dilihat pada gambar 3.



Gambar 3. Alur Proses Tahap Pelatihan

Notasi yang digunakan dalam algoritma pelatihan:

- x → Data training *input*  
x = (x<sub>1</sub>, ..., x<sub>i</sub>, ..., x<sub>n</sub>)
- t → Data *training* untuk target *output*  
t = (t<sub>1</sub>, ..., t<sub>k</sub>, ..., t<sub>m</sub>)
- α → *Learning rate* yaitu parameter untuk mengontrol perubahan bobot selama pelatihan. Semakin besar *learning rate*, maka jaringan syaraf tiruan akan semakin cepat belajar tetapi hasilnya kurang akurat. Semakin kecil *learning rate*, maka jaringan syaraf tiruan akan semakin lambat belajar tetapi hasilnya lebih akurat.
- X<sub>i</sub> → Unit *input* ke-i
- Z<sub>j</sub> → Hidden unit ke-j
- Y<sub>k</sub> → Unit *output* ke-k
- v<sub>0j</sub> → Bias untuk *hidden* unit ke-j
- v<sub>ij</sub> → Bobot antara unit *input* ke-i dengan hidden unit ke-j
- w<sub>0k</sub> → Bias untuk unit *output* ke-k
- W<sub>jk</sub> → Bobot antara hidden unit ke-j dengan unit *output* ke-k
- δ<sub>k</sub> → Faktor koreksi *error* untuk bobot w<sub>jk</sub>
- δ<sub>j</sub> → Faktor koreksi *error* untuk bobot v<sub>ij</sub>
- m → Momentum

**Step 1:** Inisialisasi, Inisialisasi nilai bobot dan bias dapat *diset* dengan sembarang angka (acak) antara -0.5 dan 0.5 Inisialisasi *learning rate*, maksimal iterasi dan toleransi *error*

**Step 2:** Lakukan selama *stopping condition* masih belum terpenuhi. Untuk menentukan *stopping condition* dengan menggunakan maksimal iterasi atau toleransi *error*. Jika iterasi sudah melebihi maksimal iterasi maka pelatihan dihentikan. Jika menggunakan toleransi *error* dengan metode *Mean Absolute Percentage Error* (MAPE), bila nilai MAPE kurang dari atau sama dengan toleransi *error* maka pelatihan dihentikan.

**Step 3:** Setiap unit *input* (X<sub>i</sub>, i = 1, ..., n) menerima sinyal *input* dan menyebarkannya pada seluruh *hidden* unit

**Step 4:** Setiap hidden unit (Z<sub>j</sub>, j = 1, ..., p) akan menghitung sinyal-sinyal *input* dengan bobot dan biasnya.

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2)$$

Kemudian dengan menggunakan fungsi aktivasi yang telah ditentukan diperoleh sinyal *output* dari *hidden* unit tersebut.

$$z_j = f(z\_in_j) \quad (3)$$

**Step 5:** Setiap unit *output* (Y<sub>k</sub>, k = 1, ..., m) akan menghitung sinyal-sinyal dari *hidden* unit dengan bobot dan biasnya.

$$y\_in_k = v_{0k} + \sum_{j=1}^p Z_j W_{jk} \quad (4)$$

Kemudian dengan menggunakan fungsi aktivasi yang telah ditentukan diperoleh sinyal *output* dari unit *output* tersebut.

$$y_k = f(y\_in_k) \quad (5)$$

**Step 6:** Dihitung kesalahan antara target *output* dengan *output* yang dihasilkan menggunakan metode *Mean Absolute Percentage Error* dengan persamaan 6.

Jika masih belum memenuhi syarat, dilakukan penghitungan faktor koreksi *error* (δ<sub>k</sub>).

$$\delta_k = (t_k - y_k) f'(y\_in_k) \quad (6)$$

Faktor koreksi *error* digunakan untuk menghitung koreksi *error* (ΔW<sub>jk</sub>) untuk memperbaharui W<sub>jk</sub>.

$$\Delta W_{jk} = \alpha \delta_k z_j \quad (7)$$

**Step 7:** Setiap hidden unit (Z<sub>j</sub>, j = 1, ..., p) akan menghitung bobot yang dikirimkan *output* unit. Jika kondisi iterasi pertama yang rumus yang digunakan

$$\delta\_in_j = \sum_{k=1}^m \delta_k w_{jk} \quad (8)$$

Kemudian hasilnya dikalikan dengan turunan dari fungsi aktivasi untuk mendapatkan faktor koreksi *error*

$$\delta_j = \delta\_in_j f'(z\_in_j) \quad (9)$$

$$\Delta V_{ij} = \alpha \delta_j x_i$$

Pada iterasi kedua dan selanjutnya

$$\delta\_in_j = \sum_{k=1}^m \delta_k w_{jk} \quad (10)$$

Kemudian hasilnya dikalikan dengan turunan dari fungsi aktivasi untuk mendapatkan faktor koreksi *error*

$$\delta_j = \delta\_in_j f'(z\_in_j)$$

$$\Delta V_{ij} = (m \delta_{j-1}) + (\alpha \delta_j x_i) * (1-m) \quad (11)$$

**Step 8:** Setiap unit *output* (Y<sub>k</sub>, k = 1, ..., m) akan memperbaharui bobotnya dari setiap hidden unit.

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (12)$$

Demikian pula setiap hidden unit (Z<sub>j</sub>, j = 1, ..., p) akan memperbaharui bobotnya dari setiap unit *input*.

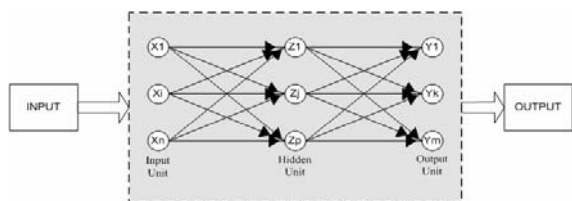
$$V_{ij}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij} \quad (13)$$

**Step 9:** Memeriksa *stopping condition*.

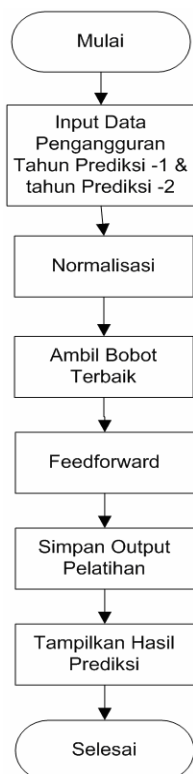
### 3.2 Tahap Pengujian

Setelah bobot yang terbaik pada tahap pelatihan didapat, maka nilai pembobot tersebut digunakan untuk mengolah data masukan untuk menghasilkan keluaran yang sesuai. Hal ini

digunakan untuk menguji apakah JST dapat bekerja dengan baik yaitu dapat memprediksi pola data yang telah dilatihkan dengan tingkat kesalahan yang kecil.



Gambar 4. Blok Diagram Tahap Pengujian



Gambar 5. Alur Proses Pengujian Jaringan Syaraf Tiruan

Notasi yang digunakan dalam algoritma pengujian:

- $X_i$  → Unit *input* ke- $i$
- $Z_j$  → Hidden unit ke- $j$
- $Y_k$  → Unit *output* ke- $k$
- $v_{0j}$  → Bias untuk hidden unit ke- $j$
- $v_{ij}$  → Bobot antara unit *input* ke- $i$  dengan hidden unit ke- $j$
- $w_{0k}$  → Bias untuk unit *output* ke- $k$
- $w_{jk}$  → Bobot antara hidden unit ke- $j$  dengan unit *output* ke- $k$

**Step 1:** Inisialisasi, Inisialisasi nilai bobot dan bias sesuai dengan bobot yang dihasilkan pada proses pelatihan.

**Step 2:** Setiap unit *input* ( $X_i$ ,  $i = 1, \dots, n$ ) menyebarkan sinyal *input* pada seluruh hidden unit.

**Step 3** Setiap hidden unit ( $Z_j$ ,  $j = 1, \dots, p$ ) akan menghitung sinyal-sinyal *input* dengan bobot dan biasnya.

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (14)$$

Kemudian dengan menggunakan fungsi aktivasi yang telah ditentukan diperoleh sinyal *output* dari hidden unit tersebut.

$$z_j = f(z\_in_j)$$

**Step 4:** Setiap unit *output* ( $Y_k$ ,  $k = 1, \dots, m$ ) akan menghitung sinyal-sinyal dari hidden unit dengan bobot dan biasnya.

$$y\_in_k = v_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (15)$$

Kemudian dengan menggunakan fungsi aktivasi yang telah ditentukan diperoleh sinyal *output* dari unit *output* tersebut.

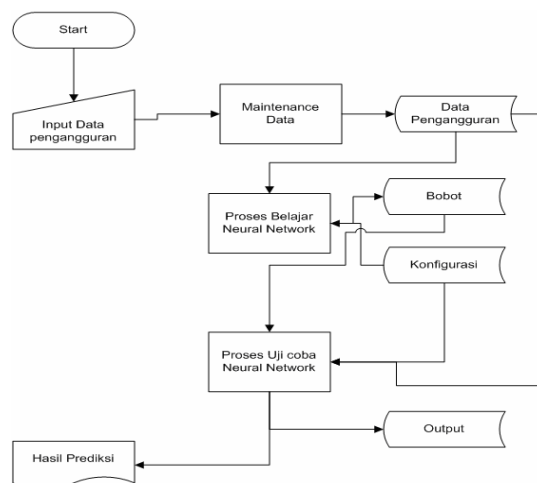
$$y_k = f(y\_in_k) \quad (16)$$

#### 4. PERANCANGAN SISTEM

Sebelum membuat program aplikasi, terlebih dahulu dilakukan proses perancangan sistem. Hal ini dilakukan supaya aplikasi yang dibuat dapat berfungsi sesuai dengan yang diharapkan sehingga mampu menghasilkan prediksi tingkat pengangguran yang akurat. Dalam perancangan sistem ini ada beberapa tahapan-tahapan yang harus dilakukan. Adapun tahapan-tahapan dalam perancangan sistem yang dilakukan adalah pembuatan Diagram Alir, Data Flow Diagram (DFD), Entity Relationship Diagram (ERD), dan Struktur Database.

##### 4.1 Diagram alir

Diagram alir menunjukkan jalannya program aplikasi secara garis besar. Diagram alir sangat membantu dalam pembuatan suatu program aplikasi. Karena selain menunjukkan jalannya program aplikasi dan pengguna, diagram alir juga memperlihatkan database yang dibutuhkan oleh aplikasi. Diagram alir yang dibuat dalam aplikasi ini sebagai berikut:



Gambar 6. Diagram Alir Sistem Prediksi

### 4.2 Data Flow Diagram

Data flow diagram digunakan untuk analisa desain sistem yang menggambarkan sistem secara garis besar dan memecahnya menjadi sub bagian-sub bagian yang lebih terperinci. Di dalam sistem prediksi ini terdapat beberapa level dari Data Flow Diagram seperti dijelaskan berikut ini.



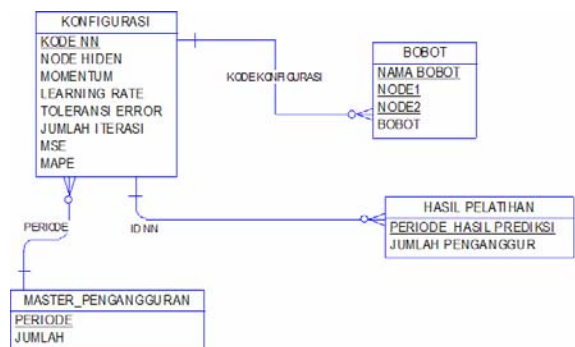
Gambar 7. Context Diagram Sistem Prediksi

Gambar 7 menunjukkan context diagram sistem prediksi. Context diagram merupakan level paling awal dari suatu DFD. Dalam context diagram terlihat entity yang berperan dalam program aplikasi ini, yaitu user. User memberikan data ke sistem berupa data pengangguran yang kemudian digunakan oleh sistem untuk proses pembelajaran dan uji coba. Sistem akan memberikan data hasil prediksi kepada user.

### 4.3 Entity Relational Diagram

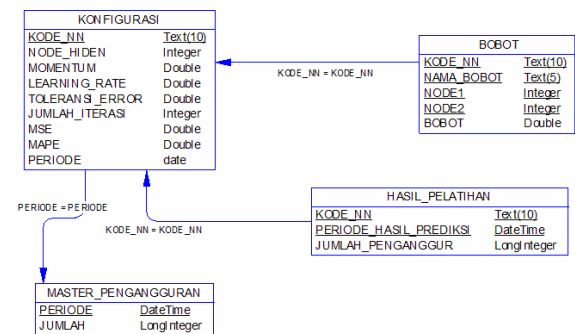
Entity Relational Diagram adalah diagram yang menggambarkan hubungan antara tabel-tabel pada proses prediksi pengangguran.

### 4.4 Conceptual Data Model (CDM)



Gambar 8. Conceptual Data Model

### 4.5 Physical Data Model (PDM)



Gambar 9. Physical Data Model

### 4.6 Disain Input dan Output

Untuk menjalankan aplikasi prediksi pengangguran ini dibutuhkan beberapa form yang digunakan sebagai sarana untuk melakukan proses prediksi.

#### Form Maintenance Data

Form ini digunakan untuk meng-inputkan data pengangguran sekaligus maintenance data pengangguran.

#### Form Pelatihan

Form ini digunakan untuk melatih data input sistem sehingga sistem dapat mengenali pola data dan menghasilkan bobot-bobot yang nantinya digunakan dalam tahap validasi,ujicoba dan peramalan.

#### Form Validasi

Form ini digunakan untuk melakukan validasi pola data output yang telah dilatihkan dalam sistem.

#### Form Uji Coba

Form ini digunakan untuk menguji keakuratan sistem dalam melakukan peramalan. Data yang dimasukkan adalah data yang belum latihan pada sistem.

## 5. IMPLEMENTASI PROGRAM

Berikut ini adalah penjelasan langkah-langkah penggunaan program beserta fasilitas-fasilitas yang disediakan dalam aplikasi prediksi dengan menggunakan Jaringan Syaraf Tiruan.

### 5.1 Form Utama

Saat program dijalankan, form yang pertama kali muncul adalah sebagai berikut:



Gambar 10. Tampilan Form Utama Sistem

Tombol pelatihan digunakan untuk menampilkan form pelatihan yang digunakan untuk melakukan pelatihan terhadap pola-pola data pengangguran masa lalu yang ditentukan oleh user. Tombol Validasi digunakan untuk membuka form validasi yang digunakan untuk mengetahui ketepatan prediksi sistem dari pola data yang sudah diinputkan. Tombol Uji coba digunakan untuk membuka form Uji Coba yang digunakan untuk

menguji ketepatan prediksi sistem dari pola data yang belum dilatihkan.

### 5.2 Form Maintenance Data

Form *maintenance* data digunakan untuk menambah atau mengubah data pengangguran yang ada pada *database* sistem.

### 5.3 Form Pelatihan

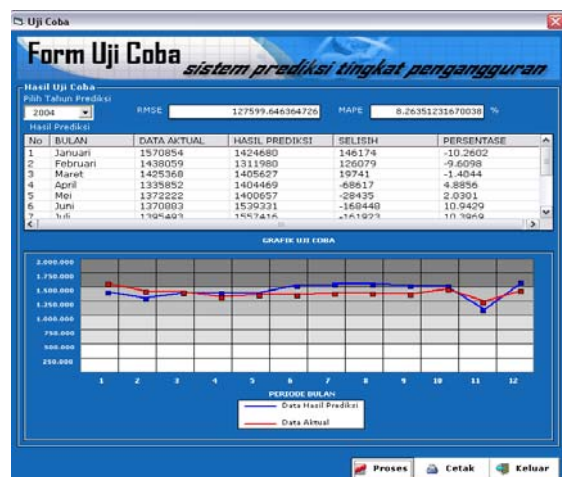
Form pelatihan digunakan untuk melatih pola-pola data periode tahun-tahun sebelumnya yang ditentukan oleh user.

### 5.4 Form Validasi

Form validasi digunakan untuk melihat ketepatan sistem dalam mengenali pola-pola data yang telah diinputkan.

### 5.5 Form Uji Coba

Form ini digunakan untuk melakukan pengujian keakuratan sistem dalam memprediksi pola data yang tidak dilatihkan.



Gambar 18. Tampilan Form Uji Coba

## 6. KESIMPULAN

Hasil dari implementasi dan evaluasi sistem prediksi menggunakan Jaringan Syaraf Tiruan dan membandingkan hasil prediksinya dengan metode *xponential Smoothing* maka dapat diambil kesimpulan sebagai berikut:

- Aplikasi prediksi dengan menggunakan Jaringan Syaraf Tiruan dapat dijadikan salah satu solusi untuk memprediksi tingkat pengangguran di Jawa Timur.
- Dalam implementasinya output yang dihasilkan merupakan angka prediksi pengangguran tahun depan perbulan yang didapat dari inputan angka pengangguran 2 tahun terakhir perbulan.
- Konfigurasi parameter untuk pelatihan sistem prediksi menggunakan Jaringan Syaraf Tiruan terbaik yang didapat dari hasil percobaan menghasilkan konfigurasi jumlah hidden node 50, nilai learning rate 0,15, batas tahun pelatihan adalah tahun 2002, nilai momentum 0,6 dan toleransi MAPE 0,001%.

- Perbandingan keakuratan prediksi dengan membandingkan nilai MAPE antara metode *Exponential Smoothing* dan Jaringan Syaraf Tiruan untuk tahun prediksi 2002 adalah 10.77% untuk metode *Exponential Smoothing* dan 7.46% untuk metode Jaringan Syaraf Tiruan.
- Perbandingan keakuratan prediksi dengan membandingkan nilai MAPE antara metode *Exponential Smoothing* dan Jaringan Syaraf Tiruan untuk tahun prediksi 2003 adalah 22.54% untuk metode *Exponential Smoothing* dan 7.67% untuk metode Jaringan Syaraf Tiruan.
- Perbandingan keakuratan prediksi dengan membandingkan nilai MAPE antara metode *Exponential Smoothing* dan Jaringan Syaraf Tiruan untuk tahun prediksi 2004 adalah 9.07% untuk metode *Exponential Smoothing* dan 8.26% untuk metode Jaringan Syaraf Tiruan.

## PUSTAKA

- Agung, Rachmad, 2005, *Tugas Akhir Aplikasi Pengenalan Pola Tanda Tangan Menggunakan Metode Backpropagation Neural Network Untuk Keamanan Transaksi Perbankan*, STIKOM, Surabaya.
- Aritonang R. Lerbin R, 2002, *Peramalan Bisnis*, Ghalia Indonesia. Jakarta.
- Jawa Timur, BPS, 2000, *Hasil Survei Sosial Ekonomi Nasional Tahun 2000 Propinsi Jawa Timur*, BPS Jawa Timur, Surabaya
- Fausett, L, 1994, *Fundamentals of Neural Network*, Prentice-Hall Inc, New Jersey.
- Kurniadi, A, 2000, *Pemrograman Microsoft Visual Basic 6*, PT Elex Media Komputindo, Jakarta.
- Kusumadewi, S, 2004, *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB dan EXCEL LINK*, Graha Ilmu, Yogyakarta.
- Kosko, Bart., 1992. *Neural Networks and Fuzzy System*, Prentice-Hall Inc, Engelwood Cliffs, New Jersey.
- Makridakis, S. and Wheelwright, S.C., 1996, *Metode – metode peramalan untuk manajemen*, Penerbit Binarupa Aksara, Jakarta
- Setiawan, K, 2003, *Paradigma Sistem Cerdas*, Bayumedia Publishing, Malang.
- Siang, JJ, 2005 *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*, Andi, Yogyakarta.
- <http://ms.wikipedia.org/wiki/Pengangguran>
- <http://www.nakertrans.go.id>
- <http://www.tempointeraktif.com>