

**LAPORAN AKHIR TAHUN  
PENELITIAN DASAR UNGGULAN PERGURUAN TINGGI  
(PDUPT)**



**OPTIMASI NEURO-FUZZY METHODS UNTUK IDENTIFIKASI  
SISTEM DINAMIK TAK LINIER MODEL PENYAKIT TIPE SEIR**

**TAHUN KE – 1 DARI RENCANA 2 TAHUN**

<b>AULI DAMAYANTI, S.Si., M.Si</b>	<b>0007117502</b>
<b>ASRI BEKTI PRATIWI, S.Si., M.Si</b>	<b>0022128303</b>
<b>Dr. MOCH. IMAM UTOYO, M.Si.</b>	<b>0001036403</b>

**DIBIAYAI OLEH:  
DIREKTORAT RISET DAN PENGABDIAN MASYARAKAT  
DIREKTORAT JENDERAL PENGUATAN RISET DAN PENGEMBANGAN  
KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI  
SESUAI DENGAN PERJANJIAN PENDANAAN PENELITIAN DAN PENGABDIAN  
KEPADA MASYARAKAT  
NOMOR: 122/SP2H/PTNBH/DRPM/2018**

**UNIVERSITAS AIRLANGGA  
NOVEMBER 2018**





LAPORAN AKHIR TAHUN  
PENELITIAN DASAR UNGGULAN PERGURUAN TINGGI  
(PDUPT)

KKC  
KK  
LP 70/19  
Dam  
0



OPTIMASI NEURO-FUZZY METHODS UNTUK IDENTIFIKASI  
SISTEM DINAMIK TAK LINIER MODEL PENYAKIT TIPE SEIR

TAHUN KE – 1 DARI RENCANA 2 TAHUN

AULI DAMAYANTI, S.Si., M.Si	0007117502
ASRI BEKTI PRATIWI, S.Si., M.Si	0022128303
Dr. MOCH. IMAM UTOYO, M.Si.	0001036403

DIBIYAI OLEH:  
DIREKTORAT RISET DAN PENGABDIAN MASYARAKAT  
DIREKTORAT JENDERAL PENGUATAN RISET DAN PENGEMBANGAN  
KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI  
SESUAI DENGAN PERJANJIAN PENDANAAN PENELITIAN DAN PENGABDIAN  
KEPADA MASYARAKAT  
NOMOR: 122/SP2H/PTNBH/DRPM/2018

UNIVERSITAS AIRLANGGA  
NOVEMBER 2018



**HALAMAN PENGESAHAN**

Judul : OPTIMASI NEURO - FUZZY METHODS UNTUK IDENTIFIKASI SISTEM DINAMIK TAK LINIER MODEL PENYAKIT TIPE SEIR

**Peneliti/Pelaksana**

Nama Lengkap : AULI DAMAYANTI, S.Si, M.Si  
 Perguruan Tinggi : Universitas Airlangga  
 NIDN : 0007117502  
 Jabatan Fungsional : Lektor  
 Program Studi : Matematika  
 Nomor HP : 081331333119  
 Alamat surel (e-mail) : aulid@fst.unair.ac.id

**Anggota (1)**

Nama Lengkap : ASRI BEKTI PRATIWI S.Si, M.Si  
 NIDN : 0022128303  
 Perguruan Tinggi : Universitas Airlangga

**Anggota (2)**

Nama Lengkap : Dr. Drs MOHAMMAD IMAM UTOYO M.Si  
 NIDN : 0001036403  
 Perguruan Tinggi : Universitas Airlangga

**Institusi Mitra (jika ada)**

Nama Institusi Mitra : -  
 Alamat : -  
 Penanggung Jawab : -  
 Tahun Pelaksanaan : Tahun ke 1 dari rencana 2 tahun  
 Biaya Tahun Berjalan : Rp 75,000,000  
 Biaya Keseluruhan : Rp 165,000,000

Mengetahui,  
 Dekan FST UNAIR



(Prof. Win Darmanto, M.Si., Ph.D)  
 NIP/NIK 196106161987011001

Kota Surabaya, 12 - 11 - 2018  
 Ketua,



( AULI DAMAYANTI, S.Si, M.Si)  
 NIP/NIK 197511072003122004

Menyetujui,  
 Ketua LPI UNAIR



(Prof. H. Hery Purnobasuki, M.Si., Ph.D)  
 NIP/NIK 196705071991021001

MILIK  
 PERPUSTAKAAN  
 UNIVERSITAS AIRLANGGA  
 SURABAYA



## RINGKASAN

Tujuan utama dari penelitian ini adalah mengembangkan dan mengaplikasikan metode *neuro – fuzzy* untuk identifikasi sistem dinamik tak linier pada model penyakit tipe SEIR. Sistem identifikasi merupakan salah satu cara untuk melakukan validasi hasil pemodelan sistem dengan riil sistem yang ada pada hardware. Identifikasi system dinamik tak linier menjadi alat penting yang dapat digunakan untuk memperbaiki kinerja pengendalian dan mencapai toleransi kesalahan yang kecil. Tujuan adanya metode identifikasi sistem adalah untuk menemukan parameter dari model matematika yang mewakili sistem fisik, dengan memanfaatkan input dan output data yang menggambarkan eksitasi dan respon dari sistem. Pemodelan *neuro-fuzzy* merupakan salah satu metode identifikasi sistem dinamik tak linier, yang memiliki akurasi proses komputasi yang tinggi, terutama dalam mencapai konvergensi minimal. Model *neuro-fuzzy* menggambarkan sistem yang menggunakan aturan fuzzy *if-then* dengan struktur *neural network*.

Pada tahun pertama penelitian, identifikasi sistem dinamik tak linier dilakukan pada model penyakit tipe SIR yaitu model epidemi penyakit yang membagi populasi menjadi empat kelas individu, yaitu individu *susceptible*, *infectious*, dan *recovered*. Model penyakit tipe SIR yang digunakan untuk identifikasi sistem tak linier adalah model penyakit dengue (DBD) dan tuberculosis (TB). Pada tahun pertama penelitian ini, terdapat 3 tahapan pengerjaan identifikasi sistem tak linier pada model penyakit tipe SIR yaitu penentuan model penyakit dengan *neural networks*, estimasi parameter model, dan validasi model. Penentuan model dari system dinamik tak linier yaitu dari model penyakit TB dan DBD dan model *recurrent neural networks*. Estimasi parameter dari model tak linier menggunakan metode heuristik, yaitu *firefly algorithm*, dan *genetic algorithm*. Dan validasi model menggunakan *recurrent neural networks*. Tujuan indentifikasi sistem dinamik tak linier model penyakit tipe SIR adalah untuk mengetahui pola dari model penyakit baik menurut data riil ataupun system model. Dari hasil uji coba dan evaluasi, *firefly algorithm*, *genetic algorithm* dan multilayer perceptron backpropagation dapat mengidentifikasi sistem dinamik tak linier, dengan estimasi parameter optimal pada model penyakit tipe SIR yaitu DBD dan TB.



## PRAKATA

Alhamdulillah, segala puji bagi Allah SWT, atas terselesainya laporan kemajuan Penelitian Data Unggulan Perguruan Tinggi Tahun Anggaran 2018 dengan berjudul *“Optimasi Neuro - Fuzzy Methods Untuk Identifikasi Sistem Dinamik Tak Linier Model Penyakit Tipe Seir”*. Pada kesempatan ini, penulis mengucapkan terima kasih dan penghargaan yang sebesar-besarnya kepada beberapa pihak yang telah membantu terselesainya penelitian ini, yaitu :

1. DP2M ditjen Dikti, melalui Dana DIPA Ditlitabmas Tahun Anggaran 2018 yang telah mendanai penelitian ini melalui program Penelitian Unggulan Perguruan Tinggi tahun anggaran 2016.
2. Rektor UNAIR dan ketua LPI UNAIR yang telah memberikan rekomendasi sehingga penelitian ini dapat didanai Kemenristekdikti.
3. Dekan dan Ketua Departemen Matematika FST Unair yang telah memberikan persetujuan penelitian ini.
4. Semua pihak yang telah memberikan kontribusi dalam penelitian ini.

Semoga hasil penelitian ini bermanfaat bagi semua pihak, terutama bagi perkembangan ilmu matematika dalam penerapannya di dunia kesehatan. Kritik dan saran yang membangun selalu diharapkan untuk kesempurnaan penelitian di masa yang akan datang.

## DAFTAR ISI

HALAMAN SAMPUL	
HALAMAN PENGESAHAN	
RINGKASAN	
PRAKATA	
DAFTAR ISI	
BAB 1. PENDAHULUAN	1
BAB 2. TINJAUAN PUSTAKA	7
2.1 <i>Neuro – Fuzzy Systems</i>	7
2.2 Model Matematika Tipe Sier	8
2.3 Model Penyebaran Penyakit Tipe SEIR dengan Imigran Infektif	9
BAB 3. TUJUAN DAN MANFAAT PENELITIAN	
BAB 4. METODE PENELITIAN	12
4.1 Tempat dan Waktu Penelitian	12
4.2 Bahan dan Alat Penelitian	12
4.3 Tahapan Umum Penelitian	12
4.4 Tahapan Khusus Penelitian	13
BAB 5. HASIL DAN LUARAN YANG DICAPAI	
BAB 6. RENCANA TAHAP BERIKUTNYA	
BAB 7. KESIMPULAN DAN SARAN	15
DAFTAR PUSTAKA	
LAMPIRAN (bukti luaran yang didapat berupa artikel/draft/publikasi)	





## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Identifikasi sistem adalah proses membangun model matematika dari sistem dinamis berdasarkan data input-output yang diamati. Identifikasi model linier sudah berkembang lama baik untuk sistem *linear time invariant* (LTI) maupun sistem *linear time varying* (LTV) (Ljung, 1999; 2006; 2010). Pada kasus model non linier, proses identifikasi menjadi sangat rumit, karena komputer hanya mampu menyelesaikan perhitungan yang linier. Metode yang tepat akan dapat memodelkan sistem nonlinier secara akurat mengingat solusi eksak dari permasalahan nonlinier sangat sulit (Babuška, 2000). Model sistem dinamik tak linier banyak terdapat pada permasalahan mekanika, dinamika, industri, model penyakit dan lain lain. Salah satu model penyakit tak linier adalah model penyakit tipe SEIR.

Epidemiologi adalah ilmu yang ditujukan pada upaya pencegahan penyebaran penyakit yang terjadi pada individu, kelompok atau masyarakat umum (Budiarto, 2002). Penyakit menular adalah penyakit yang disebabkan oleh adanya agen penyebab yang mengakibatkan perpindahan atau penularan penyakit dari orang yang infeksi kepada orang yang rentan, baik secara langsung maupun tidak langsung melalui perantara atau lingkungan hidup (Hanafiah dan Amir, 2007). Penyebab penyakit menular adalah patogen. Patogen merupakan organisme atau substansi seperti bakteri, virus, atau parasit yang mampu menimbulkan penyakit (Timmreck, 2009).

Model SEIR adalah model epidemi penyakit yang membagi populasi menjadi empat kelas individu, yaitu individu *susceptible*, individu *exposed*, individu *infectious*, dan individu *recovered* (Juan dkk, 2006). Pada beberapa jenis penyakit menular memiliki masa inkubasi (laten), yaitu rentang waktu antara masuknya agen penyebab penyakit hingga timbulnya gejala penyakit tersebut. Li dkk. (1994) mengkonstruksi model epidemik SEIR yang terdiri dari empat kompartemen dengan mempertimbangkan periode laten dari infeksi penyakit. Berbagai macam penyakit menular seperti campak (*Measles*), *tuberculous* (TBC), malaria, SARS (*Severe Acute Respiratory Syndrome*), HIV/AIDS (*Human Immunodeficiency Virus/Acquired Immune System Damage*) mempunyai periode laten (Timmreck, 2009).



Pada proses penyebaran penyakit menular, imigran infeksi memiliki peranan yang besar pada percepatan meluasnya penyebaran penyakit menular ke seluruh penjuru dunia. Dengan infeksi yang dibawanya, seorang imigran infeksi dapat menularkan penyakitnya ke manusia yang rentang di daerah ia melakukan imigrasi. Dengan mendapatkan parameter model matematika penyakit tipe SEIR yang optimal, maka dapat diketahui tingkat penyebaran penyakit tipe SEIR dengan pengaruh imigran infeksi. Hal ini merupakan salah satu upaya untuk mengurangi jumlah individu laten dan individu yang telah terinfeksi penyakit dengan biaya yang minimal (Hia dkk, 2013).

Proses identifikasi sistem dinamik tak linier, dapat dilakukan dengan banyak metode atau teknik. Tujuan utama dari setiap metode identifikasi sistem adalah untuk menemukan parameter dari model matematika yang mewakili sistem fisik, dengan memanfaatkan input dan output data yang menggambarkan eksitasi dan respon dari sistem. Di antara teknik-teknik identifikasi nonlinier, metode yang didasarkan pada model *neuro-fuzzy* merupakan model atau metode yang tangguh/kokoh/kuat untuk aplikasi di industri, akademik, maupun kesehatan. Pemodelan *neuro-fuzzy* dapat dianggap sebagai teknik graybox merupakan batas antara model jaringan syaraf tiruan dan fuzzy kualitatif (Babuška, 2002). Jaringan syaraf tiruan dan logika fuzzy menggambarkan kerja fungsi otak manusia yang sangat cocok/tepat dan berguna untuk pemodelan sistem real dengan karakteristik nonlinear yang kompleks dan informasi yang tidak pasti (Lee dkk, 1992)

Jaringan syaraf tiruan memiliki keunggulan yaitu mampu beradaptasi dan mempunyai kemampuan belajar terhadap segala permasalahan (Rumelhart dkk, 1986). Demikian pula dengan logika fuzzy, yang memiliki fitur unggul yaitu kemampuan untuk memperhitungkan ketidakpastian dan ketidaktepatan yang berlaku dari sistem nyata dengan bantuan aturan fuzzy *if-then*. Untuk memanfaatkan keunggulan kemampuan adaptasi dan kemampuan belajar dari jaringan syaraf tiruan dan kemampuan sistem fuzzy untuk memperhitungkan ketidakpastian dan ketidaktepatan sistem real yang ada dengan bantuan aturan fuzzy *if-then*, maka dilakukan proses hybrid dua metode tersebut. Sistem hybrid ini disebut *Adaptive Neuro-Fuzzy Inference System* (ANFIS) (Cherkassky, 1998; Dote dan Ovaska, 2001).

Pada metode konvensional jaringan saraf, terdapat permasalahan pada pemilihan atau pengaturan nilai bobot (*weight*) dan *bias* yaitu mengalami konvergensi lambat untuk minimal lokal dan global, hal ini berakibat jaringan saraf kesulitan dalam memperoleh

output yang diharapkan. Hal tersebut dapat diatasi dengan metode heuristik antara lain, *Artificial bee colony*, *Firefly Algorithm (FA)*, *genetic algorithm*, *simulated annealing* dan *Cuckoo Search (CS)* untuk mencapai nilai minimal global (C. Ozturk and D. Karaboga, 2011; X.-S. Yang, S. Deb, 2009; Awad, M., 2010; Fang, L., Chen, P. dan Liu, S., 2007).

Pada tahun pertama penelitian, akan dikembangkan suatu teknik optimasi menggunakan *firefly algorithm*, dan *genetic algorithm*, pada proses estimasi parameter dan proses identifikasi model menggunakan *recurrent neural network*. Sehingga diharapkan proses identifikasi model tak linier menjadi optimal.

## 1.2 Permasalahan

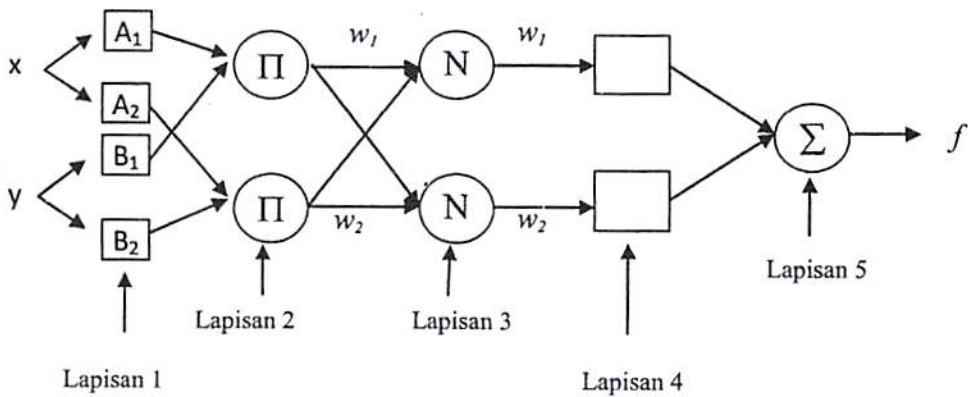
Berdasarkan uraian diatas, permasalahan dalam penelitian ini sebagai berikut :

1. Bagaimana menerapkan teknik optimasi pada metode *neuro-fuzzy* untuk mendapatkan parameter model penyakit tipe SIR (*tuberculoses (TBC)*, malaria, SARS (*Severe Acute Respiratory Syndrome*), dengue, HIV/AIDS) yang optimal.
2. Bagaimana mengidentifikasi sistem dinamik tak linier model penyakit tipe SIR menggunakan metode *neuro-fuzzy*

**BAB 2. TINJAUAN PUSTAKA**

**2.1 Neuro – Fuzzy System**

*Adaptive Neuro-Fuzzy Inference System* (ANFIS) adalah penggabungan mekanisme fuzzy inference system yang digambarkan dalam arsitektur jaringan syaraf. Sistem inferensi fuzzy yang digunakan adalah sistem inferensi fuzzy model Tagaki-Sugeno-Kang (TSK) orde satu dengan pertimbangan kesederhanaan dan kemudahan komputasi.



Gambar 2.1 Struktur *Adaptive Neuro-Fuzzy Inference System*

Pada Gambar 2.1 terlihat sistem neuro-fuzzy terdiri atas lima lapisan dengan fungsi yang berbeda untuk tiap lapisannya. Tiap lapisan terdiri atas beberapa simpul yang dilambangkan dengan kotak atau lingkaran. Lambang kotak menyatakan simpul adaptif artinya nilai parameternya bisa berubah dengan pembelajaran dan lambang lingkaran menyatakan simpul nonadaptif yang nilainya tetap.

Lapisan-lapisan pada struktur ANFIS dapat dijelaskan sebagai berikut :

*Lapisan 1.* Semua simpul pada lapisan ini adalah simpul adaptif (parameter dapat berubah) dengan fungsi simpul :

$$\begin{aligned}
 O_{1,i} &= \mu_{A_i}(x), \text{ untuk } i = 1, 2, \text{ atau} \\
 O_{1,i} &= \mu_{B_{i-2}}(y), \text{ untuk } i = 3, 4
 \end{aligned}
 \tag{2.1}$$

dengan x dan y adalah masukan pada simpul i,  $A_i$  (atau  $B_{i-2}$ ) adalah fungsi keanggotaan masing-masing simpul. Simpul  $O_{1,i}$  berfungsi untuk menyatakan derajat keanggotaan tiap masukan terhadap himpunan fuzzy A dan B. Fungsi keanggotaan yang dipakai adalah jenis generalized bell (*gbell*). Parameter a, b, c, pada fungsi keanggotaan *gbell* dinamakan parameter premis yang adaptif.





*Lapisan 2.* Semua simpul pada lapisan ini adalah nonadaptif (parameter tetap). Fungsi simpul ini adalah mengalikan setiap sinyal masukan yang datang. Fungsi simpul :

$$O_{2,i} = w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y), \quad i = 1, 2 \quad (2.2)$$

Tiap keluaran simpul menyatakan derajat pengaktifan (*firing strength*) tiap aturan fuzzy. Fungsi ini dapat diperluas apabila bagian premis memiliki lebih dari dua himpunan fuzzy. Banyaknya simpul pada lapisan ini menunjukkan banyaknya aturan yang dibentuk. Fungsi perkalian yang digunakan adalah interpretasi kata hubung and dengan menggunakan operator t-norm.

*Lapisan 3.* Setiap simpul pada lapisan ini adalah simpul nonadaptif yang menampilkan fungsi derajat pengaktifan ternormalisasi (*normalized firing strength*) yaitu rasio keluaran simpul ke-i pada lapisan sebelumnya terhadap seluruh keluaran lapisan sebelumnya, dengan bentuk fungsi simpul:

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \quad (2.3)$$

Apabila dibentuk lebih dari dua aturan, fungsi dapat diperluas dengan membagi  $w_i$  dengan jumlah total  $w$  untuk semua aturan.

*Lapisan 4.* Setiap simpul pada lapisan ini adalah simpul adaptif dengan fungsi simpul:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (2.4)$$

dengan  $\bar{w}_i$  adalah derajat pengaktifan ternormalisasi dari lapisan 3 dan parameter  $p, q, r$  menyatakan parameter konsekuen yang adaptif.

*Lapisan 5.* Pada lapisan ini hanya ada satu simpul tetap yang fungsinya untuk menjumlahkan semua masukan. Fungsi simpul :

$$O_{5,1} = \sum_i \bar{w}_i f_i = \frac{\sum w_i f_i}{\sum w_i} \quad (2.5)$$

Jaringan adaptif dengan lima lapisan tersebut ekuivalen dengan sistem inferensi fuzzy TSK.

(Jang, 1997)

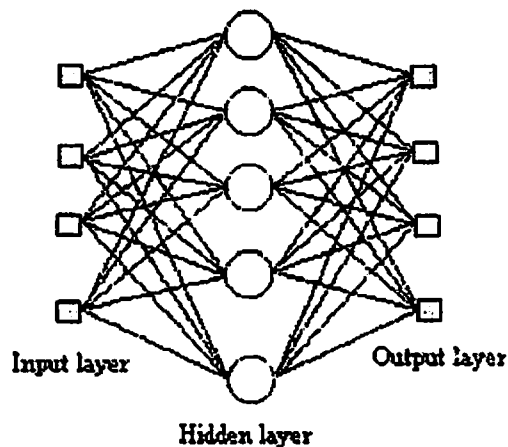
## 2.2 Multilayer Perceptron Neural Network (MLPNN)

Pelatihan (*training*) dengan MLPNN terdiri atas tiga langkah, yaitu umpan maju (*feedforward*), propagasi error (*backpropagation of error*), dan pembaharuan (*update*) bobot dan bias (Fausett, 2003). Pada tiap langkah yang dijalani, neuron-neuron diaktifkan

dengan menggunakan fungsi aktivasi *sigmoid biner*. Fungsi aktivasi tersebut digunakan pada proses *input* ke *hidden* maupun dari *hidden* ke *output*.

Saat umpan maju (*feedforward*), setiap unit *input* ( $X_i$ ) akan menerima sinyal *input* dan akan menyebarkan sinyal tersebut pada tiap unit *hidden* ( $Z_j$ ). Setiap unit *hidden* kemudian akan menghitung aktivasinya dan mengirim sinyal ( $z_j$ ) ke tiap unit *output*. Kemudian, setiap unit *output* ( $Y_k$ ) juga akan menghitung aktivasinya ( $y_k$ ) untuk menghasilkan respon terhadap *input* yang diberikan jaringan (Fausett, 2003). Gambar 2.10 menunjukkan struktur jaringan *MLPNN*.

Saat proses pelatihan (*training*), setiap unit *output* membandingkan aktivasinya ( $y_k$ ) dengan nilai target untuk menentukan besarnya *error*. Berdasarkan *error* tersebut, dihitung faktor  $\delta_k$ . Faktor  $\delta_k$  digunakan untuk mendistribusikan *error* dan *output* kembali ke *layer* sebelumnya. Dengan cara yang sama, faktor  $\delta_j$  juga dihitung pada unit *hidden*  $Z_j$ . Faktor  $\delta_k$  digunakan untuk memperbaharui bobot antara *hidden layer* dan *input layer*. Setelah semua faktor  $\delta$  ditentukan, bobot untuk semua *layer* diperbaharui secara bersamaan. Pembaharuan bobot  $w_{jk}$  (dari unit *hidden*  $Z_j$  ke unit *output*  $Y_k$ ) dilakukan berdasarkan faktor  $\delta_k$  dan aktivasi  $z_j$  dari unit *hidden*  $Z_j$ . Sedangkan, pembaharuan bobot  $v_i$  (dari unit *input*  $X_i$  ke unit *hidden*  $Z_j$ ) dilakukan berdasarkan faktor  $\delta_j$ , dan aktivasi  $x_i$ , dari *input* (Fausett, 2003).



Gambar 2.2 Arsitektur MLPNN

Selengkapnya, notasi-notasi yang akan digunakan pada algoritma pelatihan dan pengujian yaitu sebagai berikut :

1.  $x_i$  adalah vektor data *input*  $x = (x_1, \dots, x_i, \dots, x_n)$
2.  $t$  adalah vektor target  $t = (t_1, \dots, t_k, \dots, t_m)$

3.  $\alpha$  adalah *learning rate*. *Learning rate* yaitu parameter yang mengontrol perubahan bobot selama pelatihan. Jika *learning rate* besar, jaringan semakin cepat belajar, tetapi hasilnya kurang akurat. *Learning rate*, biasanya dipilih antara 0 dan 1.
4.  $X_i$  adalah unit *input* ke- $i$ . Untuk unit *input*, sinyal yang masuk dan keluar pada suatu unit dilambangkan dengan variabel yang sama, yaitu  $x_i$ .
5.  $Z_j$  adalah unit *hidden* ke- $j$ . Sinyal *input* pada  $Z_j$  dilambangkan dengan  $z\_in_j$ . Sinyal *output* (aktivasi) untuk  $Z_j$  dilambangkan dengan  $z_j$ .
6.  $v_{0j}$  adalah bias untuk unit *hidden* ke- $j$ .
7.  $v_{ij}$  adalah bobot antara unit *input* ke- $i$  dan unit *hidden* ke- $j$ .
8.  $Y_k$  adalah unit *output* ke- $k$ . Sinyal *input* ke  $Y_k$  dilambangkan  $y\_in_k$ . Sinyal *output* (aktivasi) untuk  $Y_k$  dilambangkan dengan  $y_k$ .
9.  $w_{0k}$  adalah bias untuk unit *output* ke- $k$ .
10.  $w_{jk}$  adalah bobot antara unit *hidden* ke- $j$  dan unit *output* ke- $k$ .
11.  $\delta_k$  adalah faktor koreksi *error* untuk bobot  $w_{jk}$ .
12.  $\delta_j$  adalah faktor koreksi *error* untuk bobot  $v_{ij}$ .

### Prosedur Pelatihan (*Training*)

Langkah-langkah pelatihan *backpropagation* (Fausett, 2003) sebagai berikut :

1. Inisialisasi bobot (sebaiknya diatur pada bilangan acak yang kecil).
2. Jika *stopping criteria* belum terpenuhi, lakukan langkah 3-10.
3. Untuk setiap data *training*, lakukan langkah 4-9.
4. Setiap unit *input* ( $X_i, i=1, \dots, n$ ) menerima sinyal *input*  $x_i$  dan menyebarkan sinyal tersebut pada seluruh unit lapisan di atasnya (unit *hidden*).
5. Setiap unit *hidden* ( $Z_j, j=1, \dots, p$ ) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot, termasuk biasanya

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.7)$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output*, dan unit *hidden* yang bersangkutan



$$z_j = f(z\_in_j) = \frac{1}{1 + e^{-z\_in_j}} \quad (2.8)$$

lalu mengirim sinyal *output* tersebut ke seluruh unit pada lapisan *output*.

6. Setiap unit *output* ( $Y_k, k=1, \dots, m$ ) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot, termasuk biasanya

$$y\_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (2.9)$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dan unit *output* yang bersangkutan :

$$y_k = f(y\_in_k) = \frac{1}{1 + e^{-y\_in_k}} \quad (2.10)$$

lalu mengirim sinyal *output* ini ke seluruh unit pada unit *output*.

7. Setiap unit *output* ( $Y_k, k=1, \dots, m$ ) menerima suatu target yang sesuai dengan *input training* untuk menghitung kesalahan (*error*) antara target dengan *output* yang dihasilkan jaringan :

$$\delta_k = (t_k - y_k) f'(y\_in_k) = (t_k - y_k) y_k (1 - y_k) \quad (2.11)$$

Faktor  $\delta_k$  digunakan untuk menghitung koreksi *error* ( $\Delta w_{jk}$ ) yang nantinya akan dipakai untuk memperbaharui  $w_{jk}$ , dengan :

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

Selain itu, juga dihitung koreksi bias  $\Delta w_{0k}$  yang nantinya akan dipakai untuk memperbaharui  $w_{0k}$ , dengan :

$$\Delta w_{0k} = \alpha \delta_k$$

Faktor  $\delta_k$  kemudian dikirimkan ke *lapisan* yang berada di bawahnya.

8. Setiap unit *hidden* ( $Z_j, j=1, \dots, p$ ) menjumlah *input* delta (yang dikirim dari langkah 6) yang sudah berbobot :

$$\delta\_in_j = \sum_{k=1}^m \delta_k w_{jk} \quad (2.12)$$

Kemudian, hasilnya dikalikan dengan turunan dari fungsi aktivasi yang digunakan jaringan untuk menghasilkan faktor koreksi *error* ( $\delta_j$ ), dengan :

$$\delta_j = \delta\_in_j f'(z\_in_j) = \delta\_in_j z_j (1 - z_j) \quad (2.13)$$

Faktor  $\delta_j$  digunakan untuk menghitung koreksi *error* ( $\Delta v_{ij}$ ) yang nantinya akan dipakai untuk memperbaharui  $v_{ij}$ , dengan :

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

Selain itu, juga dihitung koreksi bias  $\Delta v_{0j}$  yang nantinya akan dipakai untuk memperbaharui  $v_{0j}$ , dengan :

$$\Delta v_{0j} = \alpha \delta_j \quad (2.15)$$

9. Setiap unit *output* ( $Y_k, k = 1, \dots, m$ ) akan memperbaharui bias dan bobotnya dari setiap unit *hidden* ( $Z_j, j = 1, \dots, p$ ),

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.16)$$

Demikian pula, setiap unit *hidden* ( $Z_j, j = 1, \dots, p$ ) akan memperbaharui bias dan bobotnya dari setiap unit *input* ( $X_i, i = 1, \dots, n$ ),

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.17)$$

10. Memeriksa *stop criteria*.

Untuk menentukan *stopping criteria* terdapat dua cara yaitu, membatasi iterasi (*epoch*) yang ingin dilakukan dan membatasi *error*.

Cara menghitung *error* dengan *Mean Square Error (MSE)* adalah berikut :

- a. Dengan bobot yang ada saat itu, lakukan langkah umpan maju (langkah ke-3 sampai dengan langkah ke-5), di mana *inputnya* diambil dari *input training set* jika yang ingin dihitung adalah *training set error* dan atau *input test set* jika yang ingin dihitung adalah *test set error*. Langkah tersebut dilakukan untuk semua data *training/test* yang ada.
- b. Kemudian, dicari selisih antara target *output* ( $t_k$ ) dengan *output* jaringan ( $y_k$ ) dan diimplementasikan pada persamaan *Mean Square Error*, maka :

$$MSE = \frac{1}{n} \times \{(t_{k1} - y_{k1})^2 + (t_{k2} - y_{k2})^2 + \dots + (t_{km} - y_{km})^2\}. \quad (2.18)$$

### Prosedur Pengujian (*Testing*)

Setelah pelatihan, dilakukan pengujian. Pengujian MLPNN diaplikasikan dengan hanya menggunakan umpan maju dari algoritma pelatihan.

Prosedur pengujian (Fausett, 2003) sebagai berikut :

1. Inisialisasi bobot (dari algoritma pelatihan).
2. Untuk setiap unit *input*, lakukan langkah 3-5.
3. Untuk  $i = 1, \dots, n$ , atur aktivasi unit *input*  $x_i$ .

4. Untuk  $j = 1, \dots, p$

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.19)$$

$$z_j \doteq f(z_{in_j}) = \frac{1}{1 + e^{-z_{in_j}}} \quad (2.20)$$

5. Untuk  $k = 1, \dots, m$  :

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (2.21)$$

$$y_k = f(y_{in_k}). \quad (2.22)$$

### 2.3 Model Matematika Tipe SEIR

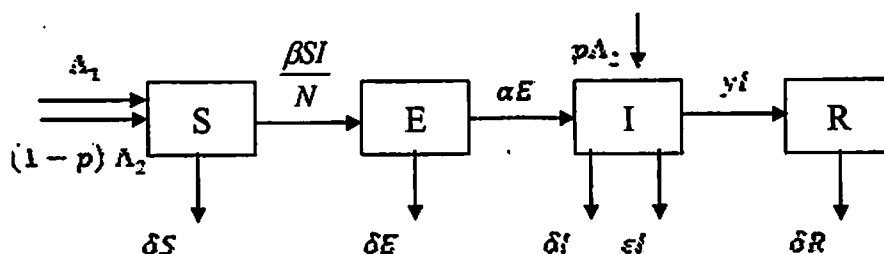
Di dalam epidemiologi dikenal istilah individu *susceptible*, individu *exposed*, individu *infectious*, dan individu *recovered*. Individu *susceptible* adalah individu yang sehat dan rentan terinfeksi penyakit. Individu *exposed* adalah individu yang terinfeksi penyakit tetapi belum dapat menularkannya ke individu yang lain. Individu *infectious* adalah individu yang terinfeksi penyakit dan dapat menularkannya ke individu yang lain. Individu *recovered* adalah individu yang telah sembuh dari penyakit. Model epidemi penyakit yang membagi populasi menjadi empat kelas individu, yaitu individu *susceptible*, individu *exposed*, individu *infectious*, dan individu *recovered* disebut dengan model SEIR (Juan dkk, 2006).

Model epidemik SEIR dikonstruksi dengan empat kompartemen dengan mempertimbangkan periode laten dari infeksi penyakit. Berbagai macam penyakit menular seperti campak, tuberkulosis, malaria, SARS (Severe Acute Respiratory Syndrome), HIV/AIDS mempunyai periode laten. Adanya periode laten menjadi alasan pembentukan model SEIR, yakni munculnya kelas ekspos (*exposed*). Dan dengan adanya imigran yang masuk secara konstan ke dalam kelas rentan dan kelas laten, dimana imigran tersebut terdiri dari individu-individu rentan dan laten (Li dkk, 1994)

Li dkk, (1999) mengasumsikan bahwa besarnya laju rekrutmen dan kematian alami tidak sama. Pada perkembangan berikutnya Li dkk, (2004) menambahkan asumsi pada kompartemen E dan I, bahwa kematian individu selain karena sebab alami juga dapat disebabkan oleh infeksi penyakit (fatal). Juan dkk (2006) mengembangkan model SEIR dengan faktor imigrasi di kompartemen yang berbeda.



Berikut diagram transmisi dari model matematika penyebaran penyakit tipe SEIR dengan pengaruh imigran infeksi sebagai berikut :



Gambar 2.3 Diagram Transmisi Model Matematika Penyebaran Penyakit Tipe SEIR dengan pengaruh Imigran Infektif

Berdasarkan Gambar 2.2, maka dapat dibentuk suatu model penyebaran penyakit tipe SEIR dengan pengaruh imigran infeksi sebagai berikut :

$$\frac{dS}{dt} = A_1 + (1-p)A_2 - \frac{\beta SI}{N} - \delta S \tag{2.6}$$

$$\frac{dE}{dt} = \frac{\beta SI}{N} - (\alpha + \delta)E \tag{2.7}$$

$$\frac{dI}{dt} = pA_2 + \alpha E - (\epsilon + y + \delta)I \tag{2.8}$$

$$\frac{dR}{dt} = yI - \delta R \tag{2.9}$$

Persamaan (2.6) merepresentasikan laju perubahan populasi *susceptible* per satuan waktu. Populasi *susceptible* akan bertambah karena adanya kelahiran dengan laju konstan sebesar  $A_1$  dan penambahan akibat masuknya imigran dengan laju sebesar  $(1-p)A_2$ . Kemudian, populasi *susceptible* berkurang karena adanya interaksi (kontak efektif) antara individu *susceptible* dan *infectious* dengan laju sebesar  $\beta$  dan karena adanya kematian alami dengan laju sebesar  $\delta$ .

Persamaan (2.7) merepresentasikan laju perubahan populasi *exposed* per satuan waktu. Populasi *exposed* akan meningkat dengan adanya kontak efektif antara individu *susceptible* dan *infectious* dengan laju sebesar  $\beta$ . Kemudian, berkurang karena adanya kematian alami dengan laju sebesar  $\delta$  dan akibat adanya perpindahan individu dari kelas *exposed* ke kelas *infectious* dengan laju perpindahannya sebesar  $\alpha$ .

Persamaan (2.8) merepresentasikan laju perubahan populasi *infectious* per satuan waktu. Populasi *infectious* bertambah karena masuknya individu *infectious* yang berstatus sebagai imigran dengan laju konstan sebesar  $pA_2$  dan karena adanya individu yang

masuk akibat perpindahan individu dari kelas *exposed* ke kelas *infectious*. Kemudian, penurunannya dipengaruhi karena adanya individu yang sembuh sehingga berpindah ke kelas *recovered* dengan laju  $\gamma$  dan akibat adanya kematian alami dan kematian akibat penyakit yang menginfeksi dengan masing-masing lajunya sebesar  $\delta$  dan  $\epsilon$ .

Persamaan (2.9) merepresentasikan laju perubahan populasi *recovered* per satuan waktu. Populasi *recovered* akan meningkat dengan adanya individu yang sembuh dari penyakit dengan laju sebesar  $\gamma$  dan menurun akibat adanya kematian alami pada individu *recovered* dengan lajunya sebesar  $\delta$ .

(Hia dkk, 2013)

#### 2.4 Model Penyebaran Penyakit Dengue dan Penyakit Tuberculosis tipe SIR

(Pandey, A., Mubayi, A., Medlock, J., 2013. comparing vector host and SIR models for dengue transmission, *Mathematical Biosciences*, vol 246 (2), page 252-259).

(Side, S., 2015. A susceptible-infected-recovered model and simulation for transmission of tuberculosis. *Adv. Sci. Lett.*,21: 137-139.)

Transmisi Dengue telah banyak dimodelkan menggunakan model tipe SIR, yang hanya menggambarkan secara eksplisit infeksi manusia. Model SIR ini lebih sederhana daripada model host-vektor, karena hanya mengkaji penyebaran penyakit DBD pada populasi manusia, sehingga membuat analisis dan estimasi parameter menjadi lebih mudah. Model penyebaran penyakit Dengue tipe SIR sebagai berikut :

$$\begin{aligned}\frac{dS}{dt} &= B - \beta \frac{I}{N} S - \mu S \\ \frac{dI}{dt} &= \beta \frac{I}{N} S - \gamma I - \mu I - \alpha I \\ \frac{dR}{dt} &= \gamma I - \mu R\end{aligned}$$

Dengan S adalah populasi manusia yang susceptible terhadap penyakit dengue, I adalah populasi manusia terinfeksi (infection) dengue, dan R adalah populasi manusia yang sembuh (recovered) dengue. B menyatakan laju kelahiran manusia.  $\beta$  menyatakan laju transmisi penyakit dari manusia ke manusia,  $\mu$  menyatakan laju kematian alami,  $\gamma$  menyatakan laju recovery dan  $\alpha$  menyatakan laju mortality akibat penyakit. Total populasi dinyatakan dengan  $N = S + I + R$ .

Pada model penyebaran penyakit tuberculosis tipe SIR sebagai berikut :

$$\begin{aligned}\frac{dS}{dt} &= -b \frac{I}{N} S - \mu S + \pi \\ \frac{dI}{dt} &= b \frac{I}{N} S - (\mu + \mu_i + c) I \\ \frac{dR}{dt} &= c I - \mu R\end{aligned}$$

Dengan  $N = S + I + R$ .  $S$  adalah populasi manusia yang susceptible terhadap penyakit tuberkulosis,  $I$  adalah populasi manusia terinfeksi (infection) tuberkulosis, dan  $R$  adalah populasi manusia yang sembuh (recovered) tuberculosi. Jumlah populasi  $S$  akan bertambah karena kelahiran sebesar  $\pi$ , dengan  $\pi$  adalah konstan. Populasi  $S$  akan berkurang karena kematian dengan laju  $\mu$ . Kontak langsung dengan individu yang terinfeksi menyebabkan individu pada populasi rentan akan ikut terinfeksi dan akan masuk menjadi populasi  $I$ . Hal ini menyebabkan berkurangnya populasi  $S$ . Laju penularan penyakit TB adalah  $b$ . Berkurangnya populasi  $I$  disebabkan oleh kematian karena faktor lain dengan laju  $\mu$  dan kematian karena penyakit TB dengan laju  $\mu_i$ . Individu yang terinfeksi TB dapat sembuh secara spontan dengan laju  $c$  dan masuk dalam populasi  $R$ . Hal ini juga menyebabkan berkurangnya populasi  $I$ . Populasi  $R$  dapat berkurang karena disebabkan oleh kematian dengan laju  $\mu$ .

## 2.5 Firefly Algorithm

Algoritma kunang-kunang atau Firefly Algorithm (FA) pertama kali dikembangkan oleh Xin-She Yang pada tahun 2007 di Cambridge University. Algoritma ini merupakan sebuah algoritma yang terinspirasi dari pola berkedip dan perilaku kunang-kunang. *Firefly* atau Kunang-kunang, termasuk dalam keluarga *Lampyridae*, kumbang bersayap kecil yang mampu memproduksi *cold light* (cahaya dingin atau sebuah cahaya yang dihasilkan kecil) berkedip untuk menarik pasangan. (Encyclopedia Britannica, 2013)

Ciri-ciri khas dari kunang-kunang yaitu kalaupun cahaya yang dihasilkan dari proses kimia dalam tubuhnya yaitu proses Bioluminisensi (*Bioluminisense*). Bioluminisense berasal dari kata “bio” yang berarti hidup dan “luminisensi” yang berarti emisi cahaya. Luminisensi adalah fenomena dimana materi memancarkan cahaya dalam kisaran cahaya tampak. (Holsa, 2009)

Menurut (Yang, 2010) perumusan *firefly algorithm* adalah sebagai berikut :

1. Semua kunang-kunang itu unisex, jadi satu kunang-kunang akan tertarik pada kunang-kunang lain.

2. Daya tarik sebanding dengan tingkat kecerahan kunang-kunang, kunang-kunang dengan tingkat kecerahan lebih rendah akan tertarik dan bergerak ke kunang-kunang dengan tingkat kecerahan lebih tinggi, kecerahan dapat berkurang seiring dengan bertambahnya jarak dan adanya penyerapan cahaya akibat faktor udara.
3. Kecerahan atau intensitas cahaya kunang-kunang ditentukan oleh nilai fungsi tujuan dari masalah yang diberikan. Untuk masalah maksimisasi, intensitas cahaya sebanding dengan nilai fungsi tujuan.

Untuk masalah maksimisasi, kecerahan hanya sebanding dengan nilai fungsi tujuan. Berikut beberapa istilah yang digunakan dalam *Firefly Algorithm* (Yang, 2010) :

1. Populasi adalah sebuah kumpulan populasi yang direpresentasikan dengan *fireflies*.
2. *Firefly* adalah individu dalam populasi yang terdiri dari kumpulan simbol atau kode yang merepresentasikan solusi dari permasalahan.
3. Intensitas cahaya adalah nilai untuk mengevaluasi *firefly*.
4. *Attractiveness* (Daya Tarik) adalah daya tarik *firefly* yang dinilai oleh *firefly* lainnya berdasarkan intensitas cahayanya.
5. *Distance* adalah jarak antara 2 *firefly*.
6. *Movement* adalah pergerakan yang dilakukan masing-masing *firefly* menuju *firefly* lain yang intensitas cahayanya lebih terang.

### 2.5.1. Keatraktifan *Firefly*

Didalam *firefly algorithm* terdapat 2 hal penting yaitu variasi intensitas cahaya dan daya tarik kunang-kunang (*attractiveness*). Untuk mempermudah, dapat diasumsikan bahwa daya tarik kunang-kunang ditentukan oleh kecerahan intensitas cahaya yang sebanding dengan nilai fungsi tujuan. Dalam kasus yang paling sederhana contohnya masalah optimasi maksimum, tingkat intensitas cahaya pada sebuah kunang-kunang  $x$  dapat dilihat sebagai,

$$I(x) \approx f(x)$$

Dengan nilai  $I$  merupakan tingkat intensitas cahaya pada  $x$  kunang-kunang yang sebanding terhadap solusi fungsi tujuan permasalahan yang akan dicari  $f(x)$ . Daya  $\beta$  tarik adalah relatif, karena intensitas cahaya yang harus dilihat dan dinilai oleh kunang-kunang lain. Dengan demikian, hasil penelitian akan berbeda tergantung dari jarak antara kunang-kunang yang satu dengan yang lainnya ( $r_{ij}$ ). Selain itu, intensitas cahaya akan menurun seiring dengan jarak dari sumbernya, dan cahaya juga diserap oleh udara

dengan jarak yang berlebih, hal ini akan memungkinkan adanya daya tarik ( $\beta$ ) yang bervariasi dengan berbeda-beda tingkat penyerapannya.

Fungsi keaktraifan ialah sebagai berikut :

$$E(r) = E_0 \cdot e^{-\gamma r^2}$$

dengan :

$\beta_0$  : nilai keatraktifan kunang-kunang awal

$\gamma$  : koefisien penyerapan cahaya

$r$  : jarak antar kunang-kunang

(Yang, 2010)

### 2.5.2. Distance dan Movement

Jarak antara dua kunang-kunang  $i$  dan  $j$  pada  $x_i$  dan  $x_j$ , masing-masing adalah sebagai berikut :

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^n (x_{i,k} - x_{j,k})^2}$$

Dengan  $x_{i,k}$  adalah komponen atau lokus ke- $k$  dari untai kunang-kunang  $i$ , dalam kasus dua dimensi maka,

$$r_{ij} = \sqrt{(x_{i,1} - x_{j,1})^2 + (x_{i,2} - x_{j,2})^2}$$

Pergerakan kunang-kunang  $i$  yang tertarik ke kunang-kunang  $j$  yang lebih menarik (terang) ditentukan oleh persamaan berikut :

$$x_i = x_i - E_0 \cdot \exp(-\gamma r_{ij}^2) \cdot (x_j - x_i) - \alpha \cdot (\text{rand} - \frac{1}{2})$$

Dengan variabel awal  $x_i$  menunjukkan posisi awal kunang-kunang yang berada pada lokasi  $x$ , kemudian persamaan kedua menyatakan daya tarik *firefly* kepada *firefly* yang memiliki intensitas cahaya yang lebih besar, yang terdiri  $\beta_0 = 1,0$  parameter ini merupakan nilai keatraktifan awal pada *firefly*, ( $\exp$ ) bilangan eksponensial, parameter  $\gamma \in [0, \infty)$  biasa ditentukan antara 0,1 sampai 10 merupakan nilai untuk tingkat penyerapan pada lingkungan sekitar *firefly*  $i$  dan  $j$ . selanjutnya persamaan ketiga terdiri dari selisih nilai solusi pada *firefly*  $i$  terhadap *firefly*  $j$ . kemudian fungsi persamaan pergerakan *firefly* secara random menunjukkan adanya bilangan *random* yang kisarannya antara  $[0,1]$ , variabel  $\alpha$  yang memiliki kisaran antara  $[0,1]$  biasa ditentukan dengan nilai 0,2. (Broersma, 2010)



## 2.6 Genetic Algorithm

Algoritma adalah suatu himpunan langkah-langkah atau instruksi yang telah dirumuskan dengan baik (*well-defined*) untuk memperoleh suatu keluaran khusus (*specific output*) dari suatu masukan khusus (*specific input*) dalam langkah yang jumlahnya berhingga (Chartrand dan Ollermann, 1993).

Algoritma genetika adalah algoritma pencarian yang berdasarkan pada mekanisme sistem natural yakni genetik dan seleksi alam. Berbeda dengan teknik pencarian konvensional, algoritma genetika berangkat dari himpunan solusi yang dihasilkan secara acak. Himpunan ini disebut populasi. Sedangkan setiap individu dalam populasi disebut kromosom yang merupakan representasi solusi dari populasi. Kromosom-kromosom berevolusi dalam suatu proses iterasi yang berkelanjutan yang disebut dengan generasi (Gen dan Cheng, 1994).

Algoritma genetika dikembangkan oleh Jhon Holland di era 1960-an adalah teknik optimasi yang stokastik. Mirip dengan algoritma heuristik yang lain seperti *Simulated Anneling*, algoritma genetika bisa mendapatkan solusi lokal dengan menggunakan operasi yang disebut mutasi (Osman dan Kelly, 1996).

Algoritma genetika diawali dengan inisialisasi populasi yang merupakan inisialisasi sekumpulan alternatif pemecahan yang didapat secara acak. Dalam populasi tersebut, terdapat individu-individu yang dinamakan kromosom. Kromosom-kromosom berevolusi dalam suatu proses iterasi yang berkelanjutan yang disebut generasi. Pada setiap generasi, kromosom dievaluasi berdasarkan suatu fungsi evaluasi (Gen dan Cheng, 1997). Generasi baru diperoleh dari proses kawin silang (*crossover*) dan mutasi (*mutation*). Sebelum membuat generasi baru, diperlukan proses seleksi kawin silang dan mutasi untuk menentukan kromosom induk. Kemudian melalui proses kawin silang dan mutasi, diperoleh kromosom anak. Selanjutnya dilakukan evaluasi terhadap kromosom induk dan anak untuk membentuk generasi baru (Goldberg, 1989).

### 2.6.1. Inisialisasi Parameter

Informasi pada bagian ini diambil berdasarkan (Gen dan Cheng, 1997). Algoritma genetika mempunyai parameter-parameter yang digunakan untuk proses optimalisasi, parameter-parameter yang digunakan adalah probabilitas *crossover* ( $P_c$ ), probabilitas mutasi ( $P_m$ ), dan jumlah kromosom ( $Pop\_size$ ).

#### a. Probabilitas *crossover*

Probabilitas *crossover* ( $P_c$ ) adalah rasio dari jumlah anak yang diproduksi pada setiap generasi untuk ukuran populasi.

b. Probabilitas mutasi

Probabilitas mutasi ( $P_m$ ) adalah prosentase dari jumlah total gen-gen dalam populasi yang mengalami mutasi.

### 2.6.2. Fungsi *Fitness*

Informasi pada bagian ini diambil berdasarkan (Zomaya, 1996). Di dalam algoritma genetika, pemilihan kromosom sangat penting, guna melihat apakah suatu solusi itu baik atau buruk. Oleh karena itu, algoritma genetika memerlukan nilai objektif yang dapat membedakan hal tersebut. Nilai objektif itu disebut juga dengan fungsi evaluasi (*fitness*).

Fungsi evaluasi (*fitness*) adalah fungsi yang menunjukkan keandalan suatu individu untuk bertahan dalam populasi. Di dalam seleksi alam, individu yang bernilai *fitness* tinggi yang akan bertahan hidup, sedangkan individu yang bernilai *fitness* rendah akan mati. Untuk menghitung nilai *fitness* digunakan persamaan sebagai berikut:

$$f(x) = \frac{1}{\text{max}}$$

### 2.6.4. Seleksi

Seleksi adalah proses penentuan induk yang akan dipilih sebagai induk untuk *crossover* atau mutasi untuk mendapatkan individu baru. Setiap kromosom yang terdapat dalam populasi akan dipilih menjadi induk melalui proses seleksi (Zomaya, 1996). Langkah-langkah metode seleksi *roulette wheel* adalah sebagai berikut:

- Menghitung nilai *fitness* setiap kromosom, dinotasikan  $f_i, i = 1, 2, \dots, \text{pop\_size}$ .
- Menghitung jumlah total nilai *fitness* semua kromosom pada populasi, dinotasikan  $F$ .
- Menghitung peluang terpilih dari setiap kromosom ( $p_i$ ),

$$p_i = \frac{f_i}{F}$$

- Menghitung peluang kumulatif setiap kromosom ( $q_i$ ),

$$q_i = \sum_{j=1}^i p_j$$

- Bangkitkan bilangan real secara acak dari interval (0,1), dinotasikan  $r$ .
- Jika  $r \leq q_1$ , maka kromosom pertama terpilih; yang lain, pilih kromosom ke- $i$  apabila  $q_{i-1} < r \leq q_i$  dengan  $2 \leq i \leq n$ .

### 2.6.5. Crossover

Dalam Gen dan Cheng (1997) dijelaskan bahwa *crossover* adalah persilangan dua individu yang menghasilkan anak yang masih mewarisi sebagian sifat induknya. Menurut Obitko (1998), tipe dan bentuk *crossover* tergantung pada pengkodean dan jenis permasalahannya.

Ada beberapa jenis *crossover* diantaranya adalah:

a. *Crossover* satu titik (*Single Point Crossover*)

Satu titik *crossover* diseleksi, barisan kode dari awal individu sampai titik *crossover* dikopi dari induk pertama, sisanya dikopi dari induk kedua.

b. *Crossover* dua titik (*Two Point Crossover*)

Dua titik diseleksi, barisan kode dari individu awal sampai titik *crossover* pertama dikopi dari induk pertama, bagian dari yang pertama sampai titik *crossover* kedua dikopi dari induk kedua, dan sisanya dikopi dari induk pertama.

c. *Convex crossover*

Secara umum, rata-rata dari dua vektor  $X_1$  dan  $X_2$  dihitung sebagai berikut:

$$X_1' = \beta_1 X_1 + \beta_2 X_2$$

$$X_2' = \beta_1 X_2 + \beta_2 X_1$$

dengan

$$\beta_1 + \beta_2 = 1, \beta_1 > 0, \beta_2 > 0$$

$X_1$  adalah kromosom induk 1 yang mengalami *convex crossover*

$X_2$  adalah kromosom induk 2 yang mengalami *convex crossover*

$X_1'$  adalah kromosom anak 1 hasil *convex crossover*

$X_2'$  adalah kromosom anak 2 hasil *convex crossover*

*Convex Crossover* mungkin salah satu yang paling umum digunakan. Banyak persoalan menggunakan *convex crossover* dengan  $\beta_i = 0.5$  ( $i = 1, 2$ ) Gen dan Cheng (1997).

### 2.6.6. Mutasi

Informasi pada bagian ini diambil berdasarkan Obitko (1998). Mutasi merupakan proses perubahan sebagian sifat individu sehingga menghasilkan struktur genetik baru. Untuk mengetahui berapa banyak individu yang akan dihasilkan dari proses mutasi diberikan laju mutasi ( $P_m$ ) yaitu rasio jumlah gen yang dimutasi dari total gen pada suatu populasi.

### BAB 3. TUJUAN DAN MANFAAT PENELITIAN

#### 3.1 Tujuan Penelitian

Berdasarkan uraian diatas, tujuan dalam penelitian ini dapat diperinci sebagai berikut :

Tujuan tahun I penelitian ini adalah

1. Melakukan estimasi parameter model dinamik tak linier penyakit tipe SIR tuberculosis (TB) dan dengue (DBD) menggunakan algoritma heuristic (optimasi model) *firefly algorithm* dan *genetic algorithm*.
2. Menerapkan teknik optimasi pada proses jaringan saraf menggunakan *firefly algorithm* dan *genetic algorithm* untuk mengoptimalkan bobot jaringan.

Tujuan tahun II penelitian ini adalah

1. Mengidentifikasi sistem dinamik tak linier model penyakit tipe SIR menggunakan metode *recurrent neural networks*.
2. Melakukan validasi model penyakit tipe SIR berdasarkan data lapangan yaitu *tubercoluses* (TBC) dan dengue (DBD) untuk mengetahui pola penyebaran penyakitnya.

#### 3.2 Manfaat Penelitian

Manfaat yang dapat diambil dari hasil penelitian ini adalah :

1. Hasil dari proses identifikasi system dinamik tak linier model penyakit tipe SIR adalah dapat mengetahui penyebaran dari suatu penyakit tipe SIR berdasarkan model tak linier dan estimasi parameter model yang didapatkan.
2. Pengembangan ilmu pengetahuan, yaitu pengembangan penerapan ilmu matematika pada bidang kesehatan terutama pada bidang epidemiologi.







## BAB 4. METODE PENELITIAN

### 4.1 Tempat dan Waktu Penelitian

- Waktu penelitian : Maret 2018 sampai dengan Oktober 2018 (Tahun I)  
Maret 2019 sampai dengan Oktober 2019 (Tahun II)
- Tempat Penelitian : 1. Laboratorium Komputer Matematika, FST, Universitas Airlangga, Surabaya.  
2. Rumah Sakit/Dinas Kesehatan/BPS di Wilayah Jawa Timur  
3. Ruang Diskusi R.205

### 4.2 Bahan dan Alat Penelitian

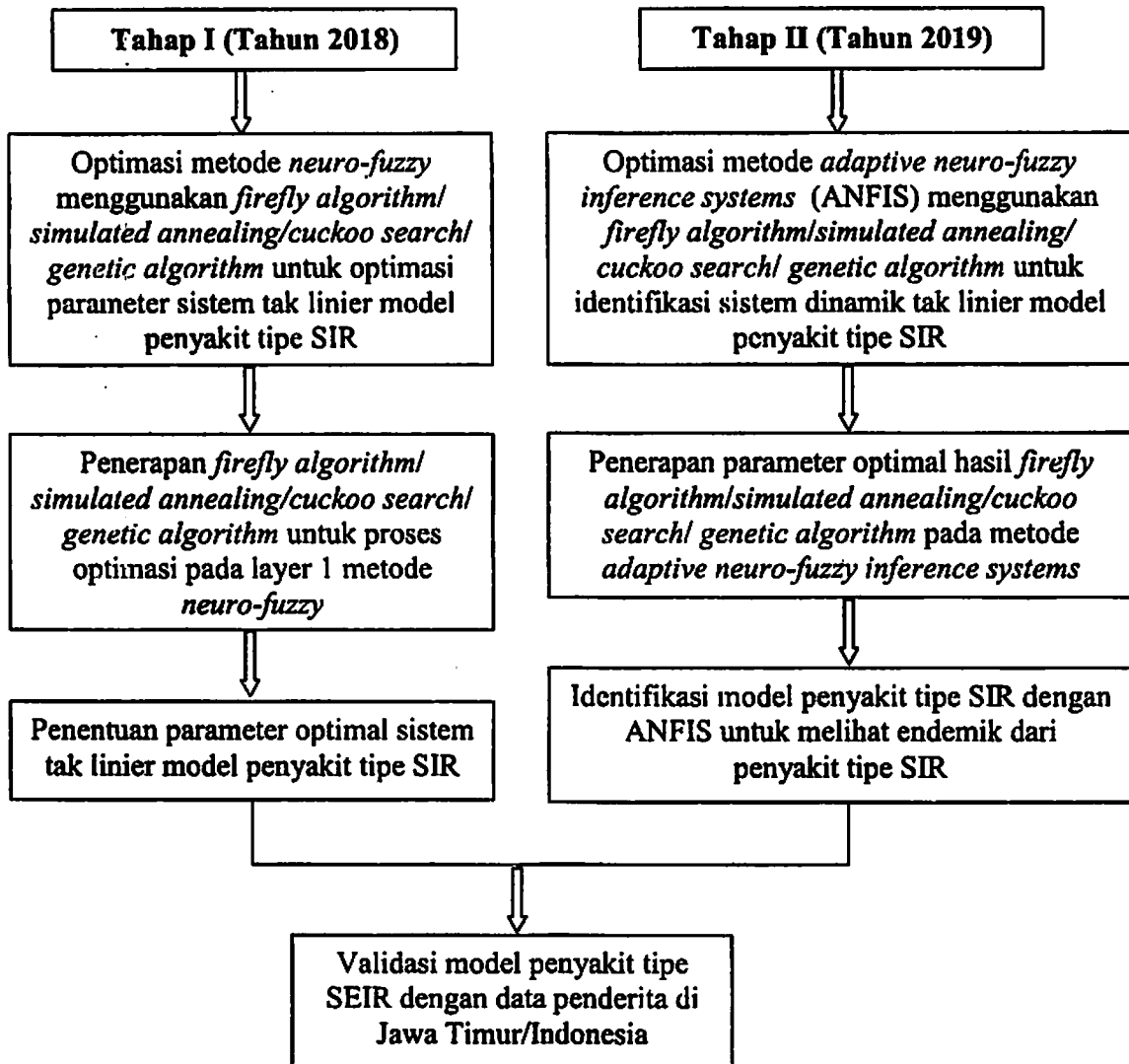
Bahan dan alat yang digunakan pada penelitian ini adalah

1. Data penderita penyakit tipe SEIR
2. Software *Matlab*

### 4.3 Tahapan Umum Penelitian

Secara garis besar, tahapan penelitian Optimasi *Neuro-Fuzzy Methods* Untuk Identifikasi Sistem Dinamik Tak Linier Model Penyakit Tipe SEIR dibagi menjadi dua tahap yakni pada tahap I (2018), akan dilakukan proses optimasi menggunakan metode heuristik antara lain *firefly algorithm*, *simulated annealing*, *cuckoo search*, *genetic algorithm* dan *cat swarm optimization* pada metode *neuro-fuzzy*, diharapkan proses penentuan nilai parameter model penyakit dapat optimal. Sedangkan pada tahap II (2019), akan dilakukan proses identifikasi sistem dinamik tak linier model penyakit tipe SIR dengan menggunakan metode *adaptive neuro fuzzy inference systems*. Pada tahapan penelitian tersebut diatas, diharapkan diperoleh parameter model penyakit tipe SIR yang optimal berdasarkan data penyakit tipe SIR dan dapat diketahui tingkat penyebaran dari penyakit tipe SEIR. Tahapan umum penelitian diperlihatkan melalui diagram alir pada Gambar 4.1.





Gambar 4.1 Diagram Alir Tahapan Umum Penelitian

#### 4.4 Tahapan Khusus Penelitian

##### 4.4.1. Prosedur Penelitian Tahap I (Tahun 2018)

Tahapan pada penelitian tahap I yaitu proses optimasi menggunakan metode heuristik antara lain *firefly algorithm*, *genetic algorithm* dan *cat swarm optimization* pada metode *neuro-fuzzy* dengan langkah-langkah sebagai berikut :

1. Inisialisasi parameter metode heuristik, maksimal iterasi, toleransi error.
2. Membangkitkan bobot dan bias jaringan saraf secara random.
3. Menentukan asumsi-asumsi nilai awal parameter pada model sistem dinamik tak linier model penyakit tipe SIR.

4. Melakukan proses pelatihan untuk mendapatkan bobot dan bias optimal dengan metode heuristik sampai maksimal iterasi.
5. Menentukan nilai parameter dari model sesuai dengan batas toleransi error.

#### 4.4.2. Prosedur Penelitian Tahap II (Tahun 2019)

Tahapan pada penelitian tahap II yaitu proses identifikasi sistem dinamik tak linier model penyakit tipe SIR dengan menggunakan metode *adaptive neuro fuzzy inference systems*, dengan langkah-langkah sebagai berikut :

1. Inisialisasi nilai parameter model.
2. Melakukan proses fuzifikasi dari nilai parameter yang sudah diperoleh.
3. Melakukan proses perhitungan dari derajat keaggotaan masing-masing nilai parameter.
4. Menentukan nilai premis dan konsekuen dari aturan-aturan fuzzy yang berhubungan dengan nilai parameter model.
5. Melakukan proses defuzifikasi dari keluaran yang didapatkan dari perhitungan langkah 4 untuk mengidentifikasi parameter sistem dinamik tak linier dari model penyakit tipe SIR.
6. Melakukan proses validasi model dari parameter yang telah didapatkan dengan data real dari penyakit tipe SIR untuk mengetahui penyebaran dari penyakit tipe SIR.

## BAB 5. HASIL DAN LUARAN YANG DICAPAI

Pada bab ini akan dibahas tentang hasil proses identifikasi sistem dinamik tak linier model penyakit tipe seir untuk tahapan penentuan jaringan yang optimal dan penentuan nilai parameter untuk model tersebut

### 5.1 Hasil yang dicapai

Data yang digunakan pada penelitian ini diambil dari Dinas Kesehatan Provinsi Jawa Timur Jl. Ahmad Yani 118 Surabaya 60231. Data berupa jumlah penderita per bulan selama 3 tahun, yang terkena tuberculosis dan dengue, sebanyak . Data riil ini, digunakan pada proses identifikasi model penyebaran penyakit TB dan DBD tipe SIR yang bertujuan untuk mengetahui behavior dari penyebaran penyakit TB dan DBD. Proses identifikasi system tak linier model penyakit TB dan DBD mempunyai tiga tahapan, yaitu membangun model, estimasi parameter dan validasi model.

Pada model penyebaran penyakit Dengue (DBD) tipe SIR, mempunyai tiga kompartemen, yaitu populasi manusia yang *susceptible* terhadap DBD (S), populasi manusia terinfeksi DBD (I) dan populasi manusia *recovered* dari DBD (R). Model penyebarar. penyakit DBD, sebagai berikut :

$$\begin{aligned}\frac{dS}{dt} &= B - \beta \frac{I}{N} S - \mu S \\ \frac{dI}{dt} &= \beta \frac{I}{N} S - \gamma I - \mu I - \alpha I \\ \frac{dR}{dt} &= \gamma I - \mu R\end{aligned}$$

Dengan :

B menyatakan laju kelahiran manusia.

$\beta$  menyatakan laju transmisis penyakit dari manusia ke manusia,

$\mu$  menyatakan laju kematian alami,

$\gamma$  menyatakan laju *recovery*

$\alpha$  menyatakan laju mortality akibat penyakit.

Total populasi dinyatakan dengan  $N = S + I + R$  .

Sehingga laju perubahan total populasi :

MILIK  
PERPUSTAKAAN  
UNIVERSITAS AIRLANGGA  
SURABAYA

$$\begin{aligned}
\frac{dN}{dt} &= \frac{dS}{dt} + \frac{dI}{dt} + \frac{dR}{dt} = B - \beta \frac{I}{N} S - \mu S + \beta \frac{I}{N} S - \gamma I - \mu I - \alpha I + \gamma I - \mu R \\
&= B - \mu S - \mu I - \mu R - \alpha I \\
&= B - \mu(S + I + R) - \alpha I \\
&= B - \mu N - \alpha I
\end{aligned}$$

Total populasi manusia di asumsikan konstan, artinya tidak mengalami perubahan yang signifikan dalam kurun 3 tahun. Sehingga model penyebaran penyakit DBD sebagai berikut :

$$\frac{dN}{dt} = 0$$

$$\frac{dI}{dt} = \beta \frac{I}{N} (N - I - R) - \gamma I - \mu I - \alpha I$$

$$\frac{dR}{dt} = \gamma I - \mu R$$

Tahapan pertama pada penelitian ini adalah melakukan estimasi parameter terhadap parameter  $\beta$ ,  $\gamma$ ,  $\mu$ , dan  $\alpha$  menggunakan firefly algorithm dengan ODE45. Langkah-langkah estimasi parameter menggunakan *Firefly Algorithm* adalah sebagai berikut :

1. Input parameter *Firefly Algorithm*.
2. Membangkitkan secara random populasi awal sebanyak  $m$  *firefly*.
3. Menentukan solusi numerik dari model DBD dengan ODE45.
4. Menghitung intensitas cahaya tiap *firefly*  $I(x)$  berdasarkan nilai fungsi tujuan  $f(x)$ .
5. Membandingkan intensitas cahaya tiap *firefly* dengan *firefly* lainnya. Apabila terdapat *firefly* yang intensitas cahayanya lebih besar, lakukan update pergerakan *firefly* menggunakan proses *movement*.
6. Menentukan *G-best*. Untuk iterasi pertama, *firefly* terbaik (*firefly* dengan intensitas cahaya terbesar) adalah *G-best*.
7. Membandingkan *firefly* terbaik tiap iterasi dengan *G-best* yang diperoleh. Apabila intensitas cahaya *firefly* terbaik saat itu lebih besar dari pada *G-best* maka *firefly* tersebut menjadi *G-best*.
8. Melakukan proses *movement* seperti pada persamaan 2.28 kepada *firefly* terbaik dengan memasukkan nilai  $\beta = 0$  dan menggabungkannya dengan *firefly* yang lain untuk menjadi populasi awal pada iterasi selanjutnya.
9. Melakukan proses diatas sampai batas iterasi dipenuhi



Pada model penyebaran penyakit tuberculosis (TB) tipe SIR, mempunyai jumlah dan jenis kompartemen yang sama dengan model DBD, dan model penyebaran penyakit tuberculosis (TB) sebagai berikut :

$$\begin{aligned}\frac{dS}{dt} &= -b \frac{I}{N} S - \mu S + \pi \\ \frac{dI}{dt} &= b \frac{I}{N} S - (\mu + \mu_1 + c)I \\ \frac{dR}{dt} &= cI - \mu R\end{aligned}$$

Dengan :

$\pi$  menyatakan laju kelahiran dan bernilai konstan

$\mu$  menyatakan laju kematian

Laju penularan penyakit TB adalah  $b$ .

$\mu_1$  menyatakan laju kematian karena penyakit TB.

$c$  menyatakan laju individu yang terinfeksi TB dapat sembuh secara spontan.

Langkah-langkah estimasi parameter penyebaran TB menggunakan *genetic algorithm* sebagai berikut :

- a. Menentukan parameter-parameter program, antara lain ukuran populasi (*npop*), jumlah generasi yang menyatakan jumlah iterasi (*ngen*), probabilitas pindah silang (*Pc*), probabilitas mutasi (*mutrate*), *selection rate* (*xrate*), jumlah gen yang menyatakan jumlah parameter yang diestimasi (*nvar*), serta menginputkan data riil.
- b. Membangkitkan kromosom secara random [0,1] berukuran *npop* \**nvar*. Kromosom-kromosom yang terbentuk pada tahap ini dijadikan populasi awal.
- c. Menghitung *ninduk* = *npop*\**xrate*, lalu hasil perhitungan dibulatkan.
- d. Membuat urutan peringkat dari *ninduk* sampai dengan satu.
- e. Menghitung *peringkat kumulatif*
- f. Membuat matriks probabilitas kumulatif berukuran *ninduk* x1 dengan elemen ke-k adalah
- g. Menghitung solusi numerik model dengan *software* MATLAB.
- h. Melakukan evaluasi nilai fungsi objektif untuk setiap kromosom pada populasi. Nilai fungsi objektif yang digunakan adalah meminimumkan rata-rata dari *mean magnitude of relative error* yang disingkat MMRE
- i. Mengurutkan populasi berdasarkan nilai fungsi objektif sehingga kromosom dengan nilai MMRE meminimum berada di urutan satu dan seterusnya.

- j. Menyeleksi populasi menjadi sejumlah *ninduk*.
- k. Memilih induk yang akan melakukan pindah silang.
- l. Melakukan pindah silang antara dua induk dengan metode kombinasi linier dari kedua induk sehingga diperoleh dua anak dan tempatkan anak hasil persilangan ke populasi kromosom.
- m. Melakukan elitisme sehingga kromosom dengan nilai fungsi objektif terbaik tidak mengalami mutasi.
- n. Lakukan mutasi pada beberapa gen dari kromosom-kromosom secara acak dengan mengganti nilai gen terpilih dengan nilai acak.
- o. Bentuk populasi baru dengan menggabungkan kromosom hasil langkah n.
- p. Pilih kromosom dengan nilai fungsi objektif minimal. Mengecek kriteria penghentian prosedur yaitu iterasi ke-*n*gen. Jika belum terpenuhi, kembali ke langkah g. Jika terpenuhi, tampilkan hasil pada langkah p.

Source code program m-file matlab untuk proses estimasi parameter pada model penyakit TB dan DBD, adalah sebagai berikut :

```
%%Estimasi Paramater model tak linier penyakit tipe SIR DBD%%
global beta miu gamma alfa;
** parameter didapat dari estimasi parameter sebelumnya **
beta = 0.98964;
miu = 0.25707;
gamma = 0.11941;
alfa = 0.19959;
data = xlsread('DataDBD.xlsx',1,'B3:E14');
tdata = data(:,1);
adata = data(:,2);
bdata = data(:,3);
cdata = data(:,4);
z0 = [adata(1) bdata(1) cdata(1)];
[t,x] = ode45('Modelpenyakit',tdata,z0);

x(:,1);x(:,2);x(:,3);

galat_a = x(:,1) - adata;
galat_b = x(:,2) - bdata;
galat_c = x(:,3) - cdata;
galat_relatif_a = abs(galat_a ./ adata);
galat_relatif_b = abs(galat_b ./ bdata);
galat_relatif_c = abs(galat_c ./ cdata);
biaya_pop = (1/12)* (1/(3*3))* (sum(galat_relatif_a) +
sum(galat_relatif_b) + sum(galat_relatif_c))

figure(1);
plot(t,x(:,1),'k',t,adata(:,1),'ko');
title('Populasi S');
```

```

xlabel('waktu(tahun)');
grid on;

figure(2);
plot(t,x(:,2),'k',t,bdata(:,1),'ko');
title('Populasi 2');
xlabel('waktu(tahun)');
grid on;

figure(3);
plot(t,x(:,3),'k',t,cdata(:,1),'ko');
title('Populasi 3');
xlabel('waktu(tahun)');
grid on;

%%Estimasi Paramater model tak linier penyakit tipe SIR TB %%
global npop pop nftujuan Z0 data ti tspan Z nilai pm pc ngen;
npop = input('Masukan banyaknya populasi      : ');
pm = input('Masukan probabilitas mutasi      : ');
pc = input('Masukan probabilitas crossover : ');
ngen = input('Masukan banyaknya generasi      : ');
Initialization()
Z0;
data;
RandomPop(npop)
FTujuan(1)
nilai; Z;
Elitism()
nftujuan(1,1)
for i=1:ngen
    Cost(npop,1)
    CRelatif(npop,1)
    IndukC(pc)
    Crossover()
    IndukM(pm)
    Mutation()
    AllGen()
    NewPop()
    Elitism()
    %clc
    b = pop(1,1,1);
    c = pop(1,1,2);
    miu = pop(1,1,3);
    miuT = pop(1,1,4);
    Pi = pop(1,1,5);
    sprintf('*) Pada iterasi ke-%d, nilai fungsi tujuannya adalah
    %0.6e',i,nftujuan(1,1))
    sprintf(' Dengan nilai parameter :')
    [b c miu miuT Pi]
end
sprintf('Proses telah berhenti.')
sprintf('Diperoleh nilai parameter: ')
b = pop(1,1,1);

```

```

c      = pop(1,1,2);
miu    = pop(1,1,3);
miuT   = pop(1,1,4);
Pi     = pop(1,1,5);
hasil=[b c miu miuT Pi]
%Tu
Nilai_fungsi_tujuan= nftujuan(1,1)

function Initialization()
global nilai data Z0 tspan ti sA;
A=xlsread('datareal.xlsx','input');
sA=size(A);
ti=A(:,1)
data=A(:,2:sA(2))
sdata=[sA(1) sA(2)-1];
Z0=data(1,:);
nilai(1,:)=Z0;
tspan=[ti(1) ti(sA(1))]
end

function RandomPop(npop)
global pop lb;
lb=5; %banyak parameter yang akan diestimasi
pop(1, :, :)=rand(npop, lb);
sprintf('Nilai awal parameter :')
pop
end

function FTujuan(c)
global npop pop Z0 nilai tspan ti nindukm ncross parameter nilai
data;
global nftujuan data sA Z;
if c==1
    a=1;
    b=npop;
else
    a=npop;
    b=npop+ncross+nindukm-1;
end
for i=a:b
    parameter=pop(c,i,:);
    [t,Z]=RungaKutta4(@model,tspan(1),Z0,tspan(2),0.1,ti);
    nilai;
    selisih=abs(data-nilai);
    pselisih=selisih./data;
    k=0;
    for j=2:sA(1)
        l=0;
        for m=1:sA(2)-1
            l=l+(pselisih(j,m));
        end
        k=k+l/(sA(2)-1);
    end
    nftujuan(c,i)=k/(sA(1)-1);
end

```

```

end
Z;
end

function [T,Y] = RungaKutta4 (Eq,t,y,tStop,h,ti)
global nilai
T(1) = t;
Y(1,:) = y;
i = 1;
j=2;
while t < tStop
    i = i + 1;
    h = min(h,tStop - t);
    K1 =feval(Eq,t,y);
    K2 =feval(Eq,t + h/2,y + h*K1/2);
    K3 =feval(Eq,t + h/2,y + h*K2/2);
    K4 =feval(Eq,t + h,y + h*K3);
    y = y + h*(K1 + 2*K2 + 2*K3 + K4)/6;
    t = t + h;
    if j<=length(ti)
        if t-ti(j)<h & t-ti(j)>-h
            nilai(j,:)=y; j=j+1;
        end
    end
    T(i,1) = t;
    Y(i,:) = y;
end

function dy=model(t,y)
global parameter;
b = parameter(1);
c = parameter(2);
miu = parameter(3);
miuT = parameter(4);
Pi = parameter(5);
dy=zeros(1,3);
dy(1) = Pi-b*(y(2)/(y(1)+y(2)+y(3)))-miu*y(1);
dy(2) = b*(y(2)/(y(1)+y(2)+y(3)))-(miu+miuT+c)*y(2);
dy(3) = c*y(2)-miu*y(3);
end

function Elitism()
global npop pop nftujuan;
elit=nftujuan(1);
e=1;
for i=2:npop
    if nftujuan(i)<elit
        elit=nftujuan(i);
        e=i;
    end
end
sweep=pop(1,1,:);
pop(1,1,:)=pop(1,e,:);
pop(1,e,:)=sweep;

```

```

nftujuan(e)=nftujuan(1);
nftujuan(1)=elit;
end

function Cost(b,c)
global npop pop allpop cinduk indukc minduk frelatif;
global indukm nindukm ncross lb parameter nilai data nftujuan
ncost;
if c==1
    a=2;
else
    a=1;
end
Sum=sum(nftujuan(c,a:b));
ncost(c,a:b)=Sum./nftujuan(c,a:b);
end

function CRelatif(b,c)
global crelatif ckomulatif ncost;
if c==1
    a=2;
else
    a=1;
end
Sum=sum(ncost(c,a:b));
crelatif=ncost/Sum;
for i=a:b
    ckomulatif(c,i)=sum(crelatif(c,a:i));
end
end

function IndukC(pc)
global npop indukc nindukc ckomulatif cinduk;
random=rand(1,npop-1);
for i=1:npop-1
    for k=2:npop
        k;
        ckomulatif(1,k);
        random(i);
        if ckomulatif(1,k)>random(i)
            cinduk(i)=k;
            break;
        end
    end
    j=1;
end
end
cinduk;
random=rand(1,npop-1);
for i=1:npop-1
    if random(i)<pc;
        indukc(j)=cinduk(i);
        j=j+1;
    end
end
end

```



```

if mod(j-1,2)~=0
    indukc(j-1)=0;
    nindukc=j-2;
else
    nindukc=j-1;
end
end

function Crossover()
global pop indukc nindukc ncross cross lb randomc;
k=1;
lb;
for i=1:2:nindukc
    randomc=ceil(lb*rand());
    indukc(i);
    indukc(i+1);
    if indukc(i)~=indukc(i+1)
        cross(k,:)=pop(1,indukc(i),:);
        cross(k+1,:)=pop(1,indukc(i+1),:);
        if randomc(1)==1
            randomc=randomc+1;
        end
    end
end
end
cross(k,randomc:lb)=pop(1,indukc(i+1),randomc:lb);
cross(k+1,randomc:lb)=pop(1,indukc(i),randomc:lb);
k=k+2;
ncross=k-1;
end

function IndukM(pm)
global npop nindukm indukm ckomulatif;
random=rand(1,npop-1);
for i=1:npop-1
    for k=2:npop
        if ckomulatif(1,k)>random(i)
            indukm(i)=k;
            break;
        end
    end
end
end
j=1;
random=rand(1,npop-1);
for i=1:npop-1
    if random(i)<pm
        indukm(j)=indukm(i);
        j=j+1;
    end
end
end
nindukm=j-1;
end

function Mutation()

```

```

global npop pop indukc nindukc nindukm nindukm crelatif
ckumulatif randomm;
global indukm mutation cross lb parameter nilai data nftujuan
ncost;
for i=1:nindukm
    randomm=ceil(lb*rand(1,2));
    mutation(i,:)=pop(1,indukm(i),:);
    mutation(i,randomm(1))=pop(1,indukm(i),randomm(2));
    mutation(i,randomm(2))=pop(1,indukm(i),randomm(1));
end
end

function AllGen()
global npop pop ncross nindukm nindukm mutation cross;
pop(2,1:npop-1,:)=pop(1,2:npop,:);
if ncross~=0
    pop(2,npop:npop+ncross-1,:)=cross(1:ncross,:);
end
if nindukm~=0
    pop(2,npop+ncross:npop+ncross+nindukm-
1,:)=mutation(1:nindukm,:);
end
end

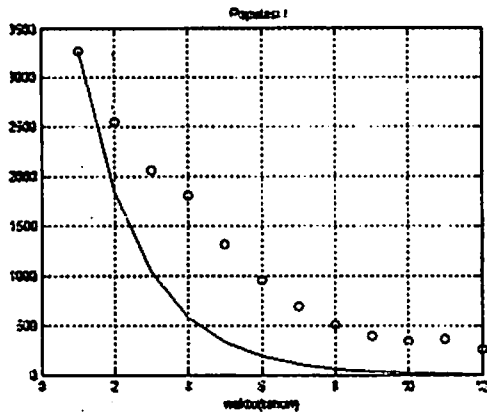
function NewPop()
global npop pop ncross nindukm nindukm ckumulatif nftujuan;
for i=1:npop-1
    nftujuan(2,i)=nftujuan(1,i+1);
end
FTujuan(2);
nftujuan(2,:);
all=npop+ncross+nindukm-1;
random=rand(1,npop-1);
for i=2:npop
    Cost(all,2)
    CRelatif(all,2)
    for j=1:all
        if ckumulatif(2,j)>random(i-1)
            pop(1,i,:)=pop(2,j,:);
            nftujuan(1,i)=nftujuan(2,j);
            pop(2,j,:)=pop(2,all,:);
            nftujuan(2,j)=nftujuan(2,j);
            all=all-1;
            break;
        end
    end
end
end
end

```

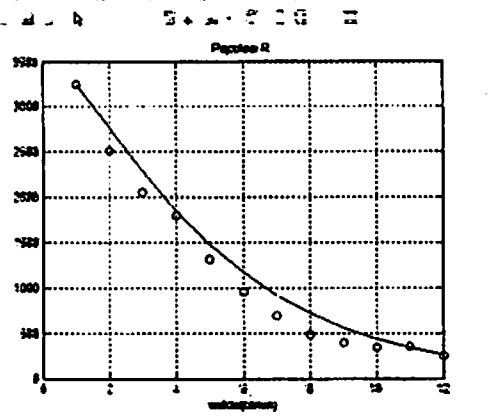
Hasil dari estimasi parameter model penyakit tipe SIR TB dan DBD adalah sebagai berikut :



File Edit View Insert Tools Database Window Help



File Edit View Insert Tools Database Window Help



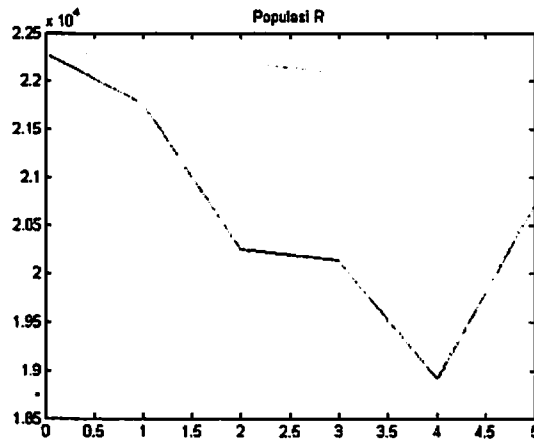
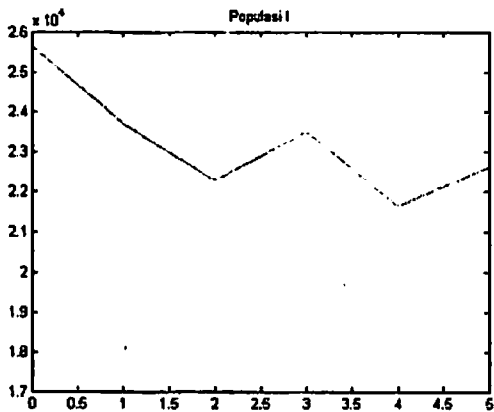
Diperoleh nilai parameter:

hasil =

0.2289    0.1204    0.0941    0.0721    0.2343

Nilai\_fungsi\_tujuan =

0.3188



## 5.2. Luaran yang dicapai

Pada penelitian ini, luaran yang ingin dicapai adalah publikasi pada seminar Internasional, serta publikasi pada jurnal internasional. Saat ini luaran yang telah dicapai adalah :

1. Satu artikel sudah dipresentasikan pada seminar internasional :  
The 2<sup>nd</sup> International Conference on Mathematics: Education, Theory, and Application (ICMETA) pada tanggal October 30<sup>th</sup>-31<sup>th</sup> 2018, at Best Western Premier Solo Baru, Solo, Indonesia yang diselenggarakan oleh Universitas Sebelas Maret Solo dengan judul paper "*Nonlinear system identification model of the spread of TB disease using genetic algorithm and multilayer perceptron*".
2. Artikel yang berjudul "*Nonlinear system identification model of the spread of TB disease using the genetic algorithm and multilayer perceptron*" telah submit di Journal of Physics : Conference Series.

Dear Sir Madam,

Thank you for submitting a full paper in The 2<sup>nd</sup> International Conference on Mathematics: "Education, Theory, and Application".

Your full paper is well recieved and being reviewed. We will notice you when the reviewing process is finished.

If you have any questions, comments, or remarks, please don't hesitate to contact us at [ICMETA@umsida.ac.id](mailto:ICMETA@umsida.ac.id).

Best regards,

The 2<sup>nd</sup> ICMETA 2018 organizer

3. Draft artikel yang akan di submit pada jurnal internasional terindeks scopus "*Indonesian Journal of Electrical Engineering and Computer Science*"

## BAB 6. RENCANA TAHAPAN BERIKUTNYA

Pada penelitian ini, rencana tahapan pengerjaan selanjutnya sebagai berikut :

1. Melakukan proses identifikasi sistem dinamik tak linier model penyakit tipe SEIR menggunakan metode *neuro-fuzzy*.
2. Melakukan validasi model penyakit tipe SIR berdasarkan data lapangan yaitu *tubercoluses* (TBC) dan dengue (DBD).



## BAB 7. KESIMPULAN DAN SARAN

### 7.1. Kesimpulan

Pada proses pengerjaan penelitian ini dapat diambil kesimpulan sebagai berikut :

1. Tahapan identifikasi sistem dinamik tak linier model penyakit tipe SIR tuberculosis dan dengue berupa estimasi parameter, mendapatkan nilai parameter yang mendekati pola model sebenarnya.
2. Algoritma genetika dan firefly mampu mendapatkan nilai parameter dari model penyakit yang optimal yaitu mampu sesuai dengan model sebenarnya.

### 7.2. Saran

Pada penelitian selanjutnya akan dicoba mengidentifikasi model system dinamik tak linier yang lain.





## DAFTAR PUSTAKA

- Awad, M., 2010, *Optimization RBFNNs Parameters Using Genetic Algorithms: Applied on Function Approximation*. International Journal of Computer Science and Security (IJCSS), 4 (3), 295-307.
- Babuška, R., 2000, *Neuro-fuzzy methods for modeling and identification*, In A. Abraham, L. C. Jain, & J. Kacprzyk (Eds.), Recent advances in intelligent paradigms and applications (pp. 161–186). Heidelberg: Springer-Verlag. Reprinted with permission.
- Babuška, R., 2002, *Neuro-fuzzy methods for modeling and identification*. In A. Abraham, L.C. Jain, and J. Kacprzyk, editors, Recent Advances in Intelligent Paradigms and Applications, pages 161–186. Copyright Springer-Verlag, Heidelberg. Reprinted with permission.
- Britz, Denny, 17 September 2015 <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/> Tanggal akses : 1 Agustus 2018.
- Budiarto, E., 2002, Pengantar Epidemiologi. Penerbit Buku Kedokteran EGC, Jakarta.
- Cherkassky, V., 1998, Fuzzy Inference Systems : A Critical Review, Computational Intelligence:SoftComputing and Fuzzy-Neuro Integration with Applications, O. Kayak and L. A. Zadeh, Eds. Springer, Berlin.
- Dote, Y., dan S. J. Ovaska, S.J., 2001, Industrial applications of soft computing : a review, in *Proc.of the IEEE*, 2001, pp. 1243–1265.
- Fang, L., Chen, P. dan Liu, S, 2007, Particle Swarm Optimization with Simulated Annealing for TSP, *Proceeding of the 6th WSEAS Int. Conf. On Artificial Intelligence, Knowledge Engineering and data Bases*, pp.206-210.
- Hanafiah, M.J., dan Amir, A., 2007, Etika Kedokteran dan Hukum Kesehatan, Penerbit Buku Kedokteran EGC, Jakarta.
- Hia, M.E., Balatif, O., Rachik, M., dan Bouyaghroumni, J., 2013, *Application of Optimal Control Theory to an SEIR model with Immigration of Infectives*, International Journal of Computer Science Issues, Vol 10, Issue 2 No. 2.
- Jang, J., 1997, ANFIS : Adaptive Neuro-Fuzzy based Inference Systems, IEEI Trans. On System, Man and Cybernetics.

- Juan, Z., Jianquan, L., dan Zhien, M., 2006, Global Dynamics of An SEIR Epidemic Model with Immigration of different Compartments, *Acta Mathematica Scientia*, 26B (3): 551-567.
- Karaboga, D., and Ozturk, C., 2011, *A novel clustering approach: Artificial bee colony (abc) algorithm*, *AppliedSoft Computing*, vol. 11, no. 1, pp. 652–657.
- Lee, M., Lee, S. Y., dan Park, C. H., 1992, Neural controller of nonlinear dynamic systems using higher order neural networks, *Electronics Letters* 28(3), 276-277.
- Li, M.Y., Muldowney, J.S., 1994, Global Stability for The SEIR Model in Epidemiology, *Math Biosci*, 125:155-64.
- Li, M.Y., Graef, J.,R., Wang, L.C., dan Karsai, J., 1999, Global Dynamics of SEIR Model with A Varying Total Population Size, *Math Biosci*, 160: 191-213.
- Ljung, L., 1999, *System Identification - Theory for the User*. Prentice-Hall, Upper Saddle River, N.J., 2nd edition.
- Ljung, L., 2006, Some aspects on nonlinear system identification. In Proc 14th IFAC Symposium on System Identification, Newcastle, Australia.
- Ljung, L., 2010, *Perspectives on system identification*. Annual Reviews in Control, 34(1).
- Rumelhart, D., Hinton, G., and Williams, R., 1986, Learning representations by backpropagation errors, *Nature*, vol. 323, pp. 533–536.
- Timmreck, T.C., 2009, *Epidemiologi Suatu Pengantar Edisi 2*, Penerbit Buku Kedokteran EGC, Jakarta.
- Yang, X.S., and Deb, S., 2009, Cuckoo search via Levy flights, in: Proc. of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), December 2009, India, IEEE Publications, USA, pp. 210-214.
- Chartrand, G. dan Oellermann, O.R. 1993. *Applied and Algorithmic Graph Theory*. McGraw-Hill, New York.
- Gen, M. dan Cheng, R.1997.*Genetic Algorithms and Engineering Design*.John Wiley & Sons, Inc, New York.
- Goldberg, D.E. 1989. *Genetic Algorithms in Search , Optimization, and Machine Learning*. Addition-Wesley Publishing Company, Canada.
- Obitko, M. 1998. *Introduction To Genetic Algorithm*, University of Applied Sciences.
- Osman, I.H., dan Kelly, J.P. 1996. *Meta-heuristic: Theory and Applications*, Kluwer Academic Publisher, Boston.

Yang, X.S., 2010, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, Frome, United Kingdom

Zomaya, A. Y. 1996. *Parallel and Distributed Computing Handbook*, MacGraw-Hill: New York.



# C E R T I F I C A T E

This is to certify that


*Priboowo Subiyanto*

has presented a paper with entitled

Nonlinear System Identification Model of the Spread of TB Disease  
using the Genetic Algorithm and Multilayer Perceptron

The 2<sup>nd</sup> International Conference on Mathematics : Education, Theory & Application  
(ICMETA) 30<sup>th</sup> - 31<sup>st</sup> October 2018

Surakarta, 31<sup>st</sup> October 2018



*[Signature]*  
Prof. Ir. Ari ~~Hando~~ Ramelan, M.Sc. (Hons), Ph.D.  
Dean



*[Signature]*  
Dr. Siswanto, M.Si.  
Chairman of 2<sup>nd</sup> ICMETA 2018



## **Nonlinear system identification model of the spread of TB disease using genetic algorithm and multilayer perceptron**

**A Damayanti<sup>1</sup>, P Subiyanto<sup>2</sup>, and A B Pratiwi<sup>3</sup>**

**<sup>1,2,3</sup>Department of Mathematics, Faculty of Science and Technology, Universitas Airlangga, Surabaya, Indonesia**

**Email : <sup>1</sup>aulid@fst.unair.ac.id**

**Abstract.** System identification is the process of building a mathematical model of a dynamic system based on the observed input-output data. Artificial neural networks have been shown to complete the identification of nonlinear systems with various learning algorithms. Identification of nonlinear dynamic systems in the model of TB disease spread is needed to predict the spread of TB disease. The model for the spread of TB disease has three compartments and six parameters. The process of identifying models for the spread of TB disease by estimating parameters using the genetic algorithm and multilayer perceptron. The choice of multilayer perceptron architecture determines the minimum error in the nonlinear system identification process for the model of TB disease spread. From the results of the trial, it was shown that the multilayer perceptron and genetic algorithm were able to identify the spread model of TB disease with MMRE of 0.00347.



# Nonlinear system identification model of SIR type based on ga-fa and multilayer perceptron

A Damayanti<sup>1</sup>, P Subiyanto<sup>2</sup>, and A B Pratiwi<sup>3</sup>

<sup>1,2,3</sup>Department of Mathematics, Faculty of Science and Technology, Universitas Airlangga, Surabaya, Indonesia

Email : [aulid@fst.unair.ac.id](mailto:aulid@fst.unair.ac.id)

**Abstract.** System identification is the process of building a mathematical model of a dynamic system based on the observed input-output data. Artificial neural networks have been shown to complete the identification of nonlinear systems with various learning algorithms. Identification of nonlinear dynamic systems in the model of TB disease spread is needed to predict the spread of TB disease. The model for the spread of TB disease has three compartments and six parameters. The process of identifying models for the spread of TB disease by estimating parameters using the genetic algorithm and multilayer perceptron. The choice of multilayer perceptron architecture determines the minimum error in the nonlinear system identification process for the model of TB disease spread. From the results of the trial, it was shown that the multilayer perceptron and genetic algorithm were able to identify the spread model of TB disease with MMRE of 0.00347

## 1. Introduction

In this modern era, computers have become an indispensable need for various groups. Today, systems on computers have been able to do many things. One of them is the ability to identify nonlinear systems on various problems with certain algorithms. System identification is a way to regulate a system so that output on the system is in accordance with the target. The identification of non-linear systems is very complicated because computers are only able to solve linear calculations. Some problems can occur when predictions are made on a system that has never been trained before. These problems include prediction of the system for a less accurate time. However, in the presence of various methods, it will be able to model nonlinear systems accurately even though the exact solution of nonlinear problems is very difficult. One method that can be used to identify non-linear systems is the Artificial Neural Network method [1].

Artificial neural networks are attempts to model information processing based on the capabilities of biological nervous systems that exist in humans. So, it can be concluded that artificial neural networks are biological neural networks viewed from the point of view of information processing. So that we are able to design models that can then be simulated and analyzed [2]. Neural networks can be used to identify complicated nonlinear systems. The excellent performance of a neural network can be attributed to the nonlinear nature of the network. It has been proven that a continuous function can be arbitrarily well approximated by a feedforward network with only one hidden layer [3].

The conventional backpropagation method experiences slow local and global minimal convergence and randomly assigns initial weight values, which results in difficulties in obtaining the expected output. So that many researchers carry out hybrid processes with other methods, one of



which is Particle Swarm Optimization. According to [4], Particle Swarm Optimization is one of the best Artificial Intelligence methods for optimizing and estimating parameters. According to [5], genetic algorithm (GA) is a metaheuristic algorithm inspired by the process of natural selection and evolution. This algorithm can determine solutions by applying genetic operators to form solutions. So that by using GA, it can minimize operating costs and improve solutions [6].

In this research, the model discussed is a mathematical model of the spread of tuberculosis. Tuberculosis (TB) is an infectious disease caused by the bacterium *Mycobacterium Tuberculosis*, which can attack various organs, especially the lungs. If the disease is not treated or the treatment is not complete, it can cause dangerous complications to death. TB was reported to have existed in the world since 5000 BC, but progress in the discovery and control of TB has only occurred in the last two centuries [7]. In the field of mathematics, mathematical models play an important role in studying the dynamics of the spread of TB. The mathematical model is modeled in the form of a system of differential equations, in this case, most mathematical models are nonlinear in nature which are very difficult to find solutions in exact.

Therefore, in this study, the identification of the nonlinear system TB model parameter estimation process, using a genetic algorithm and multilayer perceptron. Genetic algorithms are used in the parameter estimation process, especially in the completion process of the TB model in the form of the ordinary differential equation, and optimized using multilayer perceptron. It is expected that the process of identifying nonlinear TB model systems can approach the actual model.

## 2. Methodology

In general, the stages to identify the nonlinear system are model constructing, model data determination, model parameter estimation, and model validation. In this research, the processes of nonlinear system identification are TB model constructing, parameter estimation, and model simulation.

### 2.1. System Identification

System identification is an experimental approach to modeling a process or plant from unknown parameters. In this context, modeling means building a mathematical representation of the dynamic behavior of a system or process in either the time or frequency of the domain. System identification is also known as the evaluation process of building a model at any time or domain frequency. Evaluation can be done by testing into the model approach used by entering the existing data input. From the process, the results of the model output are obtained which are then compared with the system output data so that it can be evaluated by improving certain parameters in the model used. System identification includes several stages of the process, including experimental planning, selection of model structures to be used, parameter estimation, and model validation. The steps for identifying the system will continue until the results of a satisfactory model development are obtained [8].

### 2.2. Nonlinear Model of the Spread of Tuberculosis (TB)

Tuberculosis is an infectious disease caused by *Mycobacterium tuberculosis* and can cause death. In general, bacteria attack the lungs causing pulmonary TB disease. But in certain cases, these bacteria can attack other organs that cause extrapulmonary TB disease. An individual can be infected with TB if the bacteria is inhaled and forms colonies in the body. The spread of this disease can occur if individuals infected with TB sneeze, cough, even talk. In general, the main symptoms of TB include shortness of breath, coughing up phlegm, more than one month of fever, feeling unwell (malaise), weight loss, and cold sweat [9].

The following formulas are for SIR type of tuberculosis spreading model:

$$\begin{aligned}\frac{dS}{dt} &= -\beta \frac{I}{N} S - \mu S + \pi \\ \frac{dI}{dt} &= \beta \frac{I}{N} S - (\mu + \mu_i + \lambda) I \\ \frac{dR}{dt} &= \lambda I - \mu R\end{aligned}$$

where  $N = S + I + R$ .  $S$  is a human population that is susceptible for TB,  $I$  is a human population infected with TB, and  $R$  is a recovered TB population. the number of populations of  $S$  will increase due to birth by  $\pi$ , where  $\pi$  is constant. The population of  $S$  will decrease due to death at the rate of  $\mu$ . Direct contact with infected individuals causes individuals in susceptible populations to become infected and will never into population  $I$ . this causes a decrease in population  $S$ . The rate of TB spreading is  $\beta$ . Decreased population  $I$  was caused by death due to other factors with the rate of  $\mu$  and death due to TB at a rate of  $\mu_i$ . Individuals who are infected with TB can recover spontaneously at the rate of  $\lambda$  and enter the population  $R$ . this also results in reduced population  $I$ . Population  $R$  can be reduced due to death by the rate of  $\mu$  [10].

### 2.3. Genetic Algorithm (GA)

Genetic algorithm is a metaheuristic algorithm inspired from evolution theory process. The concept begins with determining solution set or so-called population. Each individual which is the solution of any problem is stated as chromosome. So, each individual in the population is called as chromosome. Every chromosome is formed from independent structure which is called gene. Gene declares the special characteristic of an individual. Every gene has its position in chromosome which is called as locus. This chromosome will develop through iteration, respectively. The iteration is called generation [11].

According to [12], fitness function is a function which provides a value for each individual in a population. This fitness function shows the quality of each solution which represented by the individual. In this study, the stages on genetic algorithm are as followed:

1. [Start] Generating random populations of  $n$  individuals. Each individual consists of a number of parameters from the model.
2. Calculate numerical solutions for models with four order Runge-Kutta for each individual.
3. [Fitness] Calculating fitness for each individual in population.

$$f(x) = MMRE = \frac{\sum_{i=1}^n \frac{1}{3} \left( \sum_{j=1}^3 \frac{|y_{i,j} - y_{i,j}^*|}{y_{i,j}} \right)}{n}$$

4. [New population] Form new individuals after passing the following genetic processes:
  - a. [Selection] Choose the best individual with the fitness to be used as a parent.

The stages for roulette wheel selection are as followed:

- i. Calculating fitness value for each chromosome, notated with  $f_i$ ,  $i=1, 2, \dots, \text{pop\_size}$ .
- ii. Calculating total amount of fitness value of all chromosomes in population, notated by  $F$ .
- iii. Calculating selected probability from each chromosome ( $p_i$ ),

$$p_i = \frac{f_i}{F}$$

- iv. Calculating cumulative probability of each chromosome ( $q_i$ ),

$$q_i = \sum_{j=1}^i p_j$$

- v. Generating real number randomly with the interval of (0,1), notated as  $r$ . If  $r \leq q_i$ , then the first chromosome is selected; for the others, select the  $i$ -th chromosome if  $q_{i-1} < r \leq q_i$  with  $2 \leq i \leq n$ .
- b. [Crossover] Parental crossover will bear new individual. This process considerate the value of crossover probability.
- c. [Mutasi] Parental mutation will bear new individual using parental characteristic. This process considerate mutation probability.
5. [Replacing] Combining dan comparing the fitness of the population generated on process 1 with the population generated on process 3.  $n$  individual with the best fitness will be maintained, the rest will be discarded.
6. [Testing] If  $N$  iteration is achieved or optimum, the process stops and the best solution is obtained. Else, back to process 2 until the best solution is achieved.

#### 2.4. Multilayer Perceptron (MLP)

Multilayer perceptron is an artificial neural network that has one or more hidden layers between the input layer and the output layer. The architecture MLP can be seen in Figure 1. In MLP, an activation function is used for mapping the sum of input processing elements (weighted inputs) to its output. The activation function on each neuron uses a differentiable nonlinear function. The method used in MLP training is the backpropagation method. During the training there are two main phases, namely the forward phase and the backward phase. The architecture of multilayer perceptron can be seen in Figure 1.

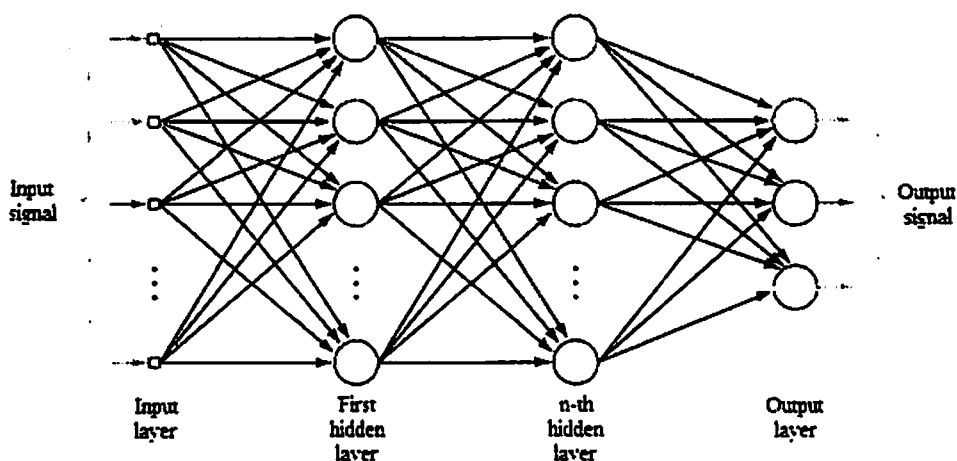


Figure 1. Multilayer Perceptron Architecture with n Layers

In the feedforward, every input unit will receive input signal and spread it to every hidden unit. After that, the hidden unit will calculate its activation and deliver the signal to every output unit. Every output will then calculate the activation too to produce response towards the given input.

In the training process, each of the output compares their activation with target value to determine the error. Based on this error, the factor used to distribute the error is calculated and the output will return to the previous layer. Using the following way, the factor of hidden unit will be used to update the weights connecting hidden layer to output layer [13].

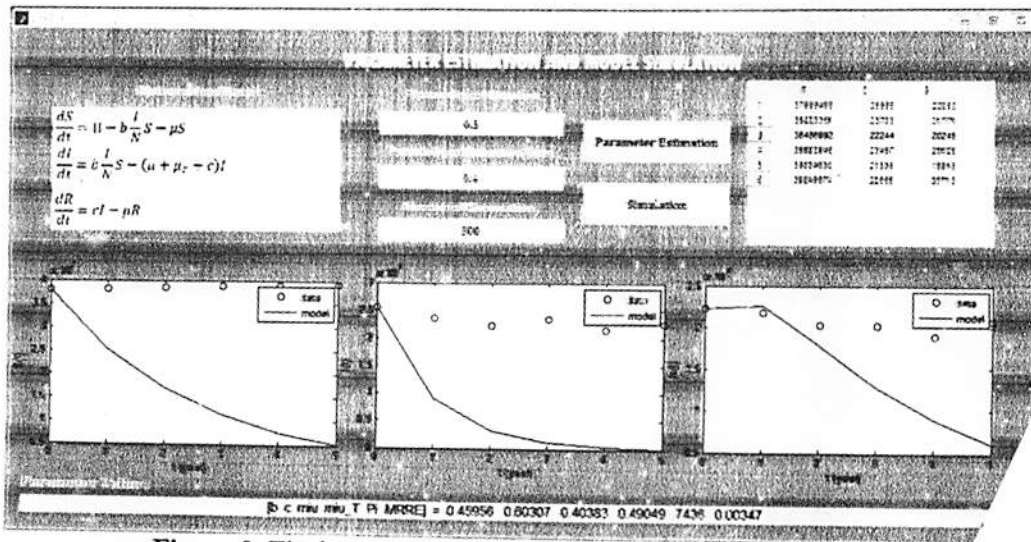
### 3. The Experimental Result

Data on TB disease used in this study was obtained from Dinas Kesehatan Provinsi Jawa Timur Jl. Ahmad Yani Surabaya for 6 years starting from 2010 to 2016. Data obtained in the form of data on patients who suspect TB, TB infection, and TB recovery. The data is used in the process of estimating the parameters of TB spreading, namely the parameters of the  $\mu$  mortality rate,  $\beta$  birth rate,  $\gamma$  TB transmission rate,  $\lambda$  mortality rate due to TB virus, and  $\delta$  cure rate from three of compartment of model TB. Parameter estimation of TB spreading model using the genetic algorithm and multilayer

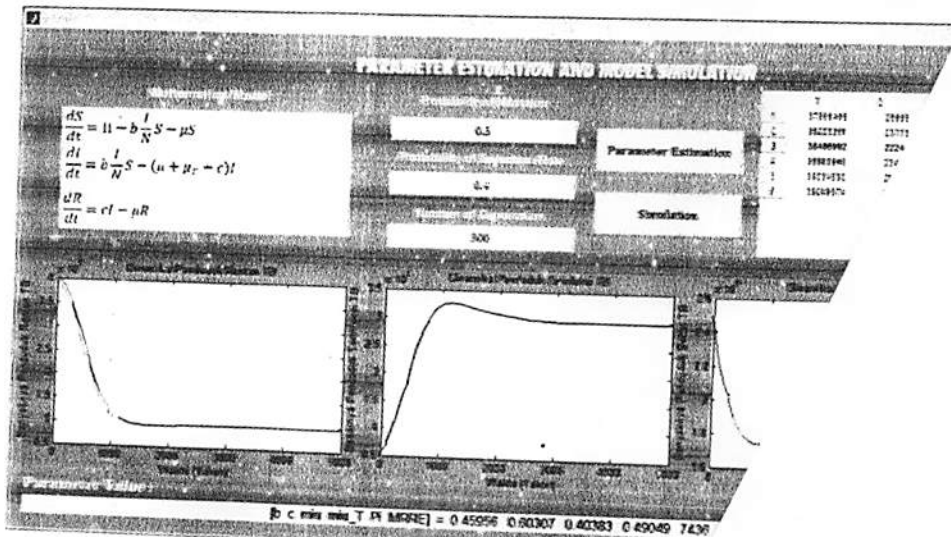
perceptron. The best parameters are obtained at 500 generation with a various pm, pc and learning rate. The result of parameter estimation as shown in Table 1. The identification of nonlinear systems with estimated parameter obtained at 0.5 pm, 0.4 pc, 0.1 learning rate, and 500 generations, with an error of 0.00347 as shown in Figure 2. The result of the model simulation of the identification process can be seen in Figure 3.

**Table 1.** The result of parameter estimation of model spread TB

Experiment to	pm	pc	Learning rate	Error	Parameter of model				
					$\pi$	$\beta$	$\mu$	$\mu_c$	$\lambda$
1	0.4	0.4	0.1	0.06755	345	0.85653	0.38965	0.40562	0.22187
2	0.4	0.4	0.9	0.18999	6543	0.7876	0.1765	0.38981	0.5643
3	0.5	0.4	0.1	0.00347	7436	0.45956	0.40383	0.49049	0.60307
4	0.5	0.5	0.1	0.05444	212	0.83182	0.29299	0.39286	0.20495
5	0.5	0.5	0.9	0.17643	5436	0.6540	0.1652	0.3207	0.5210



**Figure 2.** The best result of Parameters Estimation of the Model



**Figure 3.** Result of identification of th



#### 4. Conclusions

The process of identifying models for the spread of TB disease by estimating parameters using the genetic algorithm and multilayer perceptron. The choice of multilayer perceptron architecture determines the minimum error in the nonlinear system identification process for the model of TB disease spread. From the results of the trial, it was shown that the multilayer perceptron and genetic algorithm were able to identify the spread model of TB disease with MMRE of 0.00347.

#### References

- [1] Soelaiman R and Rifa'i R 2009 Identifikasi Sistem Nonlinier dengan Menggunakan Recurrent Neural Network dan Algoritma Dead-Zone Kalman Filter. Industrial Electronic Seminar.
- [2] Rojas R 1997 Neural Network: A Systematic Introduction, Springer-Verlag, Berlin
- [3] Cybenko G 1989 Approximations by superpositions of a sigmoidal function, Math Control Signal System 2. 303-314
- [4] Neto M, Petry G G, Aranildo R L, and Ferreira T A, 2009 Combining Artificial Neural network and Particle Swarm System for Time Series Forecasting, *International Joint Conference on Neural Networks*, pp. 2230-2237
- [5] Otobiko M 1998 Introduction to Genetic Algorithms, Czech Technical University Prague
- [6] Ren C 2012 Applying Genetic Algorithm to Vehicle Routing Problem. Heilongjiang University, China
- [7] Kemenkes 2011 Pedoman Pelayanan Gizi pada Pasien Tuberkulosis, Kementerian Kesehatan RI Ditjen Bina Gizi dan KIA, Jakarta
- [8] Oliver Nelles 2001 Nonlinear System Identification, Springer-Verlag, Berlin
- [9] World Health Organization 2017 Fact Sheets on Tuberculosis. <http://www.who.int/en/news-room/fact-sheets/detail/tuberculosis>
- [10] Diekmann, O. and J.A.P. Heesterbeek. 2000. *Mathematical Epidemiology of Infectious Diseases: Model Building, Analysis and Interpretation*. Simon Levis, Princeton University, USA
- [11] Gen M and Cheng R 1997 Genetic Algorithms and Engineering Design, John Wiley and Sons Inc., New York
- [12] Beasley D, Bull D R and Martin R R 1993 An Overview of Genetic Algorithm: Part I Fundamentals, University Computing, Cardiff
- [13] Simon S Haykin (2008) Neural Networks and Learning Machines Third Edition, Pearson Education, New Jersey





