

**ESTIMASI PARAMETER MODEL RESPON IMUN DAN  
PATOGEN MENGGUNAKAN ALGORITMA *HARMONY*  
*SEARCH***

**SKRIPSI**

oleh:

**R. P. HANDONO HAMISENO  
105090400111033**



**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS BRAWIJAYA  
MALANG  
2014**

UNIVERSITAS BRAWIJAYA



**ESTIMASI PARAMETER MODEL RESPON IMUN DAN  
PATOGEN MENGGUNAKAN ALGORITMA *HARMONY*  
*SEARCH***

**SKRIPSI**

Sebagai salah satu syarat untuk memperoleh gelar  
Sarjana Sains dalam bidang Matematika

oleh:

**R. P. HANDONO HAMISENO**  
**105090400111033**



**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS BRAWIJAYA  
MALANG  
2014**

UNIVERSITAS BRAWIJAYA



**LEMBAR PENGESAHAN SKRIPSI**

**ESTIMASI PARAMETER MODEL RESPON IMUN DAN  
PATOGEN MENGGUNAKAN ALGORITMA *HARMONY*  
*SEARCH***

oleh :

**R.P. HANDONO HAMISENO  
105090400111033**

Setelah dipertahankan di depan Majelis Penguji  
pada tanggal 19 Agustus 2014  
dan dinyatakan memenuhi syarat untuk memperoleh gelar  
Sarjana Sains dalam bidang Matematika

**Dosen Pembimbing**

**Dr. Trisilowati, M. Sc.  
NIP. 196309261989032001**

**Mengetahui,  
Ketua Jurusan Matematika  
Fakultas MIPA Universitas Brawijaya**

**Dr. Abdul Rouf Alghofari, M. Sc.  
NIP.196709071992031001**

UNIVERSITAS BRAWIJAYA



## LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : R.P. Handono Hamiseno  
NIM : 105090400111033  
Jurusan : Matematika  
Penulis skripsi berjudul : Estimasi Parameter Model Respon Imun dan Patogen Menggunakan Algoritma *Harmony Search*

dengan ini menyatakan bahwa:

1. skripsi ini adalah benar-benar karya saya sendiri dan bukan hasil plagiat dari karya orang lain. Karya-karya yang tercantum dalam Daftar Pustaka skripsi ini semata-mata digunakan sebagai acuan atau referensi.
2. Apabila di kemudian hari diketahui bahwa isi skripsi saya merupakan hasil plagiat, maka saya bersedia menanggung akibat hukum dari keadaan tersebut.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 19 Agustus 2014  
yang menyatakan,

R.P. Handono Hamiseno  
NIM. 105090400111033

UNIVERSITAS BRAWIJAYA



# ESTIMASI PARAMETER MODEL RESPON IMUN DAN PATOGEN MENGGUNAKAN ALGORITMA *HARMONY* *SEARCH*

## ABSTRAK

Skripsi ini membahas konstruksi model matematika respon imun dan patogen serta estimasi parameter yang terdapat pada model. Model ini menjelaskan interaksi antara populasi respon imun ( $M$ ) dan populasi patogen ( $P$ ). Interaksi antar populasi melibatkan parameter-parameter, yaitu sumber penghasil respon imun non spesifik ( $s_m$ ), laju kerusakan respon imun non spesifik ( $\mu_m$ ), laju berkurangnya respon imun non spesifik oleh patogen ( $k_{mp}$ ), laju pertumbuhan patogen ( $k_{pg}$ ), dan laju respon imun non spesifik dalam membunuh patogen ( $k_{pm}$ ). Pada umumnya parameter-parameter tersebut tidak diketahui nilainya dan dapat diestimasi dengan memanfaatkan data percobaan. Dalam skripsi ini, estimasi parameter dilakukan dengan menggunakan algoritma *harmony search* dengan cara meminimumkan fungsi objektif yang didefinisikan sebagai jumlah kuadrat *error*. Data yang digunakan untuk mengestimasi parameter adalah data bangkitan dari model respon imun dan patogen. Berdasarkan hasil simulasi, algoritma *harmony search* cukup efektif untuk mengestimasi parameter.

**Kata kunci** : model respon imun dan patogen, estimasi parameter, algoritma *harmony search*

UNIVERSITAS BRAWIJAYA



# PARAMETER ESTIMATION OF IMMUNE RESPONSE AND PATHOGEN MODEL BY HARMONY SEARCH ALGORITHM

## ABSTRACT

This final project discusses a construction of a response immune and pathogen mathematical model and parameter estimation of this model. This model explains interactions between immune response population ( $M$ ) and pathogen population ( $P$ ). The interactions between population involve some parameters, namely the source of non-specific immune response ( $s_m$ ), decay rate for the non-specific response ( $\mu_m$ ), rate at which the non-specific response is exhausted by pathogen ( $k_{mp}$ ), the growth rate of pathogen ( $k_{pg}$ ), and the rate at which the non-specific response eliminates pathogen ( $k_{pm}$ ). Generally, those parameters are did not known about its value and could be estimated by using experimental data. In this final project, parameter estimation is done by using harmony search algorithm that used to minimize the objective function which is defined as sum of squares error. Generated data from response immune and pathogen model's is used to estimate the parameter. Based on the simulation results, harmony search algorithm is quite effective for estimate the parameters.

**Key words** : immune response and pathogen model, parameter estimation, harmony search algorithm

UNIVERSITAS BRAWIJAYA



## KATA PENGANTAR

Segala puji dan syukur ke hadirat Allah SWT yang telah memberikan bimbingan dan petunjuk-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Estimasi Parameter Model Respon Imun dan Patogen Menggunakan Algoritma *Harmony Search*” dengan baik. Shalawat serta salam selalu tercurahkan kepada Nabi Muhammad SAW sebagai suri tauladan bagi penulis.

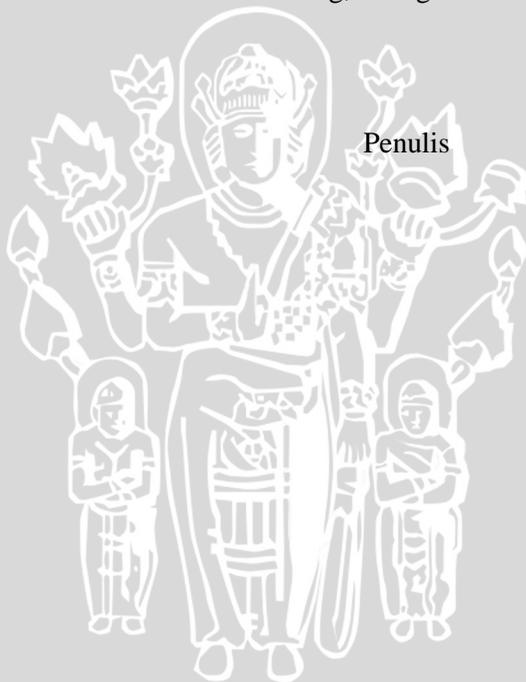
Skripsi ini tidak dapat diselesaikan dengan baik tanpa bantuan, bimbingan serta motivasi dari berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Dr. Trisilowati. M. Sc. selaku dosen pembimbing skripsi atas segala bimbingan, motivasi, kritik, saran, waktu, serta kesabaran yang telah diberikan selama pembimbingan skripsi ini dan selalu mendorong penulis untuk menyelesaikan skripsi ini.
2. Prof. Dr. Agus Suryanto, M.Sc. dan Dr. Wuryansari Muharini K., M.Si. selaku dosen penguji atas segala saran yang diberikan untuk perbaikan skripsi ini.
3. Dr. Abdul Rouf Alghofari, M. Sc. selaku Ketua Jurusan Matematika dan Dr. Sobri Abusini M.T. selaku Ketua Program Studi Matematika.
4. Prof. Dr. Marjono, M. Phil. selaku dosen pembimbing akademik yang telah membimbing penulis sejak awal kuliah.
5. Semua bapak/ ibu dosen Jurusan Matematika yang telah memberikan ilmunya kepada penulis, serta segenap staf dan karyawan TU Jurusan Matematika atas segala bantuannya.
6. Ibu, Ayah, dan kakak tercinta atas segala doa dan dukungan yang tak pernah henti diberikan kepada penulis.
7. Sahabat-sahabatku M. Nurullah, Soni H. W., Faizal A. R. A., Ruvita I. P., Wety L., Nur Hamid yang tidak pernah bosan menghibur, memberi dukungan dan bantuan yang diberikan selama ini.
8. Teman-teman Matematika A 2010 atas doa, dukungan, dan kebersamaan selama ini.

9. Semua pihak yang tidak dapat penulis sebutkan satu persatu.

Semoga Allah SWT memberikan anugerah dan barokahnya kepada semua pihak yang turut membantu menyelesaikan skripsi ini. Penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat kekurangan. Untuk itu penulis mengharapkan kritik dan saran melalui email penulis [Handono.hamiseno@gmail.com](mailto:Handono.hamiseno@gmail.com). Akhir kata, semoga skripsi ini dapat bermanfaat bagi semua pihak.

Malang, 19 Agustus 2014



Penulis

## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>LEMBAR PENGESAHAN</b> .....	<b>iii</b>
<b>LEMBAR PERNYATAAN</b> .....	<b>v</b>
<b>ABSTRAK</b> .....	<b>vii</b>
<b>ABSTRACT</b> .....	<b>ix</b>
<b>KATA PENGANTAR</b> .....	<b>xi</b>
<b>DAFTAR ISI</b> .....	<b>xiii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xv</b>
<b>DAFTAR TABEL</b> .....	<b>xvii</b>
<b>DAFTAR LAMPIRAN</b> .....	<b>xix</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	3
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>5</b>
2.1 Persamaan Diferensial .....	5
2.2 Sistem Otonomus .....	6
2.2.1 Sistem Otonomus Linear.....	7
2.2.2 Sistem Otonomus Nonlinear.....	8
2.3 Model Logistik.....	10
2.4 Metode Runge-Kutta.....	11
2.5 Metode Kuadrat Terkecil .....	12
2.6 Optimasi.....	13
2.7 Algoritma <i>Harmony Search</i> .....	15
2.8 Imun non Spesifik.....	19
2.9 Patogen .....	20
<b>BAB III PEMBAHASAN</b> .....	<b>23</b>
3.1 Struktur Model.....	23

3.2	Konstruksi Model Respon Imun dan Patogen.....	24
3.2.1	Laju Perubahan Populasi Imun .....	24
3.2.2	Laju Perubahan Populasi Patogen .....	24
3.2.3	Model Respon Imun dan Patogen .....	25
3.3	Titik Keseimbangan.....	25
3.3.1	Titik Keseimbangan $E_1$ .....	26
3.3.2	Titik Keseimbangan $E_2$ .....	26
3.3.3	Titik Keseimbangan $E_3$ .....	26
3.4	Analisis Kestabilan Titik Keseimbangan.....	27
3.4.1	Analisis Kestabilan Titik Keseimbangan $E_1$ .....	27
3.4.2	Analisis Kestabilan Titik Keseimbangan $E_2$ .....	27
3.4.3	Analisis Kestabilan Titik Keseimbangan $E_3$ .....	28
3.5	Estimasi Parameter Model Respon Imun dan Patogen (kasus satu).....	29
3.5.1	Estimasi Parameter Menggunakan Data yang Dibangkitkan .....	29
3.5.2	Solusi Numerik dengan Metode Runge-Kutta orde empat .....	31
3.5.3	Implementasi Algoritma <i>Harmony Search</i> ...	33
3.5.4	Uji Perubahan Parameter Algoritma .....	43
3.6	Estimasi Parameter Model Respon Imun dan Patogen (kasus dua) .....	50
<b>BAB IV PENUTUP.....</b>		<b>55</b>
<b>DAFTAR PUSTAKA .....</b>		<b>57</b>
<b>LAMPIRAN .....</b>		<b>59</b>

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Analisis kurva data pengamatan .....	13
Gambar 2.2 (a). Fungsi yang memuat titik minimum lokal sekaligus minimum global (b). Fungsi yang memuat titik minimum lokal dan minimum global .....	15
Gambar 2.3 Analogi algoritma HS .....	16
Gambar 2.4 Matriks <i>harmony memory</i> .....	18
Gambar 3.1 Diagram kompartemen .....	23
Gambar 3.2 Solusi numerik $M$ dan $P$ terhadap waktu $t$ .....	30
Gambar 3.3 Solusi numerik $M$ dan $P$ terhadap waktu $t$ dengan berbagai nilai awal.....	32
Gambar 3.4 Perbandingan antara data dan solusi numerik pada iterasi ke-1.....	40
Gambar 3.5 Perbandingan antara data dan solusi numerik pada iterasi ke-50.....	41
Gambar 3.6 Perbandingan antara data dan solusi numerik pada iterasi ke-500.....	42
Gambar 3.7 Grafik konvergensi algoritma HS.....	43
Gambar 3.8 Perbandingan antara data dan solusi numerik pada iterasi ke-5000.....	48
Gambar 3.9 Perbandingan kurva solusi antara parameter data bangkitan dan parameter hasil estimasi .....	49
Gambar 3.10 Solusi numerik $M$ dan $P$ terhadap waktu $t$ .....	51
Gambar 3.11 Perbandingan antara data dan solusi numerik pada iterasi ke-5000.....	52
Gambar 3.12 Perbandingan kurva solusi antara parameter data bangkitan dan parameter hasil estimasi .....	53

UNIVERSITAS BRAWIJAYA



## DAFTAR TABEL

	Halaman
Tabel 2.1 Kestabilan titik kesetimbangan.....	8
Tabel 3.1 Parameter hasil estimasi iterasi ke-1 .....	39
Tabel 3.2 Parameter hasil estimasi iterasi ke-50 .....	39
Tabel 3.3 Parameter hasil estimasi iterasi ke-500 .....	39
Tabel 3.4 <i>Output</i> uji perubahan parameter algoritma bagian I.....	44
Tabel 3.5 Parameter hasil estimasi iterasi ke-5000 .....	45
Tabel 3.6 <i>Output</i> uji perubahan parameter algoritma bagian II .....	46
Tabel 3.7 <i>Output</i> uji perubahan parameter algoritma bagian III .....	47
Tabel 3.8 Parameter hasil estimasi iterasi ke-5000 .....	50



UNIVERSITAS BRAWIJAYA



## DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Program RK4 .....	59
Lampiran 2. Program Algoritma <i>Harmony Search</i> untuk Estimasi Parameter.....	61

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Mikroorganisme penyebab penyakit yang disebut patogen merupakan salah satu agen eksternal berbahaya bagi manusia. Manusia dilengkapi dengan sistem pertahanan internal yang kompleks dan sistem imun yang memberi perlindungan terus menerus terhadap masuknya patogen atau benda-benda asing ke dalam tubuh. Jika patogen masuk ke dalam tubuh, maka sistem imun bereaksi terhadap patogen.

Respon imun merupakan respon yang ditimbulkan oleh sel-sel dan molekul yang menyusun sistem imunitas saat berhadapan dengan substansi asing (antigen). Respon imun bertanggung jawab mempertahankan kesehatan tubuh, yaitu mempertahankan tubuh terhadap serangan sel patogen. Respon imun terbagi menjadi dua jenis berdasarkan mekanisme pertahanan tubuh yaitu respon imun spesifik dan respon imun non spesifik. Respon imun spesifik menghancurkan senyawa asing yang sudah dikenalnya sedangkan respon imun non spesifik merupakan lini pertahanan pertama terhadap sel-sel atipikal, seperti sel asing atau mutan yang cedera.

Salah satu model matematika yang menggambarkan sistem imun adalah model respon imun non spesifik dan patogen. Menurut Reynolds dkk. (2006) model dibentuk melalui dua interaksi yaitu populasi respon imun non spesifik ( $M$ ) dan populasi patogen ( $P$ ). Model matematika yang terbentuk berupa sistem dinamik yang memuat parameter-parameter yang belum diketahui nilainya. Suatu model matematika tidak dapat diinterpretasikan dengan baik bila nilai parameter yang ada pada model tidak diketahui. Oleh karena itu, proses estimasi parameter dilakukan untuk mengetahui nilai parameter yang ada pada model. Pada persamaan diferensial, estimasi parameter merupakan masalah yang sulit. Proses estimasi memerlukan data dan solusi numerik persamaan diferensial. Sistem persamaan diferensial dapat diselesaikan secara numerik dengan menggunakan nilai awal parameter yang dibangkitkan. Selanjutnya, solusi numerik dicocokkan dengan data menggunakan fungsi objektif yang berupa *error* dengan pendekatan metode kuadrat terkecil (*least squares method*).

Permasalahannya adalah bagaimana meminimumkan fungsi objektif tersebut menggunakan metode optimasi.

Optimasi merupakan suatu cara yang dilakukan untuk mendapatkan nilai minimum atau maksimum secara sistematis dari suatu fungsi maupun nilai lainnya dalam berbagai kasus. Menurut Geem (2009), penyelesaian masalah optimasi sering menggunakan algoritma berbasis kalkulus yang membutuhkan gradien untuk memperoleh solusi optimal. Kelemahan algoritma berbasis kalkulus tersebut tidak dapat memberikan nilai optimum dari suatu fungsi yang tidak kontinu. Dikarenakan fungsi yang tidak kontinu pada suatu titik tidak memiliki gradient pada titik tersebut. Oleh karena itu, diperlukan algoritma untuk mendapatkan nilai optimum tanpa membutuhkan gradien. Salah satu algoritma yang tidak membutuhkan gradien adalah algoritma *Harmony Search*.

Algoritma *Harmony Search* (HS) adalah algoritma baru yang dikenalkan oleh Zong Woo Geem pada tahun 2001. Algoritma HS terinspirasi oleh proses pertunjukan musik jazz. Dalam proses tersebut, dianalogikan seorang musisi mengimprovisasi *pitch instrument* di mana proses tersebut bertujuan untuk memperoleh harmoni yang lebih baik. Proses improvisasi musik hampir mirip dengan proses untuk mencari solusi optimum. *Pitch* dari masing-masing instrumen musik menentukan kualitas harmoni tertentu seperti halnya fungsi objektif dalam menentukan bobot variabel, sedangkan harmoni dalam musik tersebut merupakan representasi dari vektor solusi.

Sebelumnya Kartiko (2009) melakukan estimasi parameter menggunakan algoritma genetika. Ketika melakukan optimasi menggunakan algoritma genetika, dalam satu iterasi memerlukan proses perhitungan yang panjang. Selain itu, fungsi *fitness* dari fungsi objektif harus ditentukan terlebih dahulu. Berbeda dari kajian yang telah dilakukan sebelumnya dan berdasarkan uraian di atas pada skripsi ini dilakukan estimasi parameter model respon imun dan patogen menggunakan metode optimasi *Harmony Search*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, pokok permasalahan yang dibahas dalam skripsi ini adalah sebagai berikut.

1. Bagaimana mengkonstruksi model respon imun non spesifik dan patogen?
2. Bagaimana mengestimasi parameter-parameter pada model respon imun non spesifik dan patogen menggunakan algoritma *Harmony Search*?

### 1.3 Tujuan

Berdasarkan rumusan masalah yang telah dipaparkan, tujuan penulisan skripsi ini adalah sebagai berikut.

1. Mengkonstruksi model respon imun non spesifik dan patogen.
2. Mengestimasi parameter-parameter pada model respon imun non spesifik dan patogen menggunakan algoritma *Harmony Search*.



UNIVERSITAS BRAWIJAYA



## BAB II TINJAUAN PUSTAKA

Pada bab ini diberikan beberapa teori untuk membantu memahami permasalahan yang dibahas dan digunakan sebagai acuan dalam pembahasan.

### 2.1 Persamaan Diferensial

#### Definisi 2.1.1 (Persamaan Diferensial Biasa)

Persamaan diferensial biasa adalah persamaan diferensial yang menyangkut satu atau lebih fungsi (peubah tak bebas) beserta turunannya terhadap satu peubah bebas (Ross, 2004).

#### Definisi 2.1.2 (Persamaan Diferensial Biasa Orde $n$ )

Suatu persamaan yang mengandung satu atau beberapa turunan dari suatu fungsi yang tidak diketahui disebut persamaan diferensial. Khususnya, suatu persamaan berbentuk

$$F(x, y, y^{(1)}, y^{(2)}, \dots, y^{(n)}) = 0$$

di mana  $y^{(n)}$  menyatakan turunan  $y$  terhadap  $x$  yang ke  $n$ , disebut persamaan diferensial biasa orde  $n$  (Boyce dan DiPrima, 2000).

#### Definisi 2.1.3 (Persamaan Diferensial Parsial)

Persamaan diferensial parsial adalah persamaan diferensial yang menyangkut satu atau lebih fungsi (peubah tak bebas) beserta turunannya terhadap lebih dari satu peubah bebas (Ross, 2004).

#### Definisi 2.1.4 (Persamaan Diferensial Biasa Linear)

Persamaan diferensial biasa linear orde  $n$ , dalam variabel tak bebas  $y$  dan variabel bebas  $x$  adalah persamaan yang dapat dinyatakan dalam bentuk

$$a_0(x) \frac{d^n y}{dx^n} + a_1(x) \frac{d^{n-1} y}{dx^{n-1}} + \dots + a_{n-1}(x) \frac{dy}{dx} + a_n(x) y = f(x) \quad (2.1)$$

dengan  $a_0(x) \neq 0$ . Pada persamaan (2.1) fungsi  $a_0(x), a_1(x), \dots, a_n(x)$  disebut koefisien persamaan diferensial dan  $f(x)$  disebut bagian non homogen. Persamaan diferensial (2.1) disebut homogen jika  $f(x)$  sama dengan nol (Ross, 2004).

### Definisi 2.1.5 (Sistem Persamaan Diferensial)

Sistem persamaan diferensial berdimensi  $n$  adalah suatu sistem yang memuat  $n$  buah persamaan diferensial dengan  $n$  buah fungsi yang tidak diketahui, di mana  $n$  merupakan bilangan bulat positif dan  $n \geq 2$  (Boyce dan DiPrima, 2000).

### Definisi 2.1.6 (Persamaan Diferensial Nonlinear)

Persamaan diferensial nonlinear adalah persamaan diferensial biasa yang variabel tak bebas dan turunannya berderajat lebih dari satu, dan atau memuat perkalian antara variabel tak bebas dan turunannya (Ross, 2004).

## 2.2 Sistem Otonomus

Suatu sistem persamaan diferensial yang berbentuk

$$\begin{cases} \frac{dx}{dt} = F(x, y) \\ \frac{dy}{dt} = G(x, y) \end{cases} \quad (2.2)$$

dengan fungsi-fungsi  $F$  dan  $G$  secara eksplisit tidak dipengaruhi oleh variabel waktu  $t$  disebut sebagai sistem otonomus atau dalam hal ini disebut bidang otonomus.

Diasumsikan bahwa fungsi  $F$  dan  $G$  adalah kontinu dalam suatu domain  $D$  di bidang  $xy$  yang disebut bidang fase untuk sistem (2.2). Jika diberikan kondisi awal pada saat  $t = t_0$  yaitu  $x(t_0) = x_0$ ,  $y(t_0) = y_0$  dengan  $(x_0, y_0) \in D$ , maka sistem (2.2) memiliki solusi tunggal  $x = x(t)$ ,  $y = y(t)$  yang terdefinisi pada suatu interval  $I$  yang memuat titik  $t_0$  (Boyce dan DiPrima, 2000).

### Definisi 2.2.1 (Titik Kesetimbangan)

Pandang sistem otonomus (2.2). Titik  $(x^*, y^*)$  yang memenuhi  $F(x^*, y^*) = G(x^*, y^*) = 0$  disebut titik kritis sistem otonomus (2.2). Titik kritis  $(x^*, y^*)$  merupakan solusi sistem (2.2) yang bernilai konstan, sebab nilai  $\frac{dx}{dt} = 0$  dan  $\frac{dy}{dt} = 0$ . Keadaan yang menyebabkan  $\frac{dx}{dt} = 0$  dan  $\frac{dy}{dt} = 0$  disebut keadaan setimbang, sehingga titik kritis disebut juga titik kesetimbangan (Boyce dan DiPrima, 2000).

### Definisi 2.2.2 (Kestabilan Titik Kesetimbangan)

Titik kesetimbangan  $\vec{x}^* = (x^*, y^*)$  dikatakan

- a. Stabil, jika untuk setiap  $\varepsilon > 0$ , terdapat  $\delta > 0$  sedemikian sehingga untuk setiap solusi  $\vec{x} = \vec{x}(t)$  yang memenuhi  $\|\vec{x}(0) - \vec{x}^*\| < \delta$ ,  
maka berlaku

$$\|\vec{x}(t) - \vec{x}^*\| < \varepsilon,$$

untuk setiap  $t > 0$ .

- b. Stabil asimtotik, jika stabil dan terdapat  $\delta_0, 0 < \delta_0 < \delta$  sedemikian sehingga sebuah solusi  $\vec{x} = \vec{x}(t)$  yang memenuhi  $\|\vec{x}(t) - \vec{x}^*\| < \delta_0$   
maka

$$\lim_{t \rightarrow \infty} \vec{x}(t) = \vec{x}^*.$$

- c. Tak stabil, jika titik tersebut tidak memenuhi kriteria (a) (Boyce dan DiPrima, 2000).

### 2.2.1 Sistem Otonomus Linear

Perhatikan sistem otonomus linear berikut.

$$\begin{aligned} \frac{dx}{dt} &= ax + by, \\ \frac{dy}{dt} &= cx + dy. \end{aligned} \tag{2.3}$$

Misalkan  $\vec{x} = (x, y)$ , maka sistem persamaan (2.3) dapat ditulis sebagai

$$\frac{d\vec{x}}{dt} = A\vec{x},$$

dengan

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \text{ dan } \vec{x} = \begin{bmatrix} x \\ y \end{bmatrix}, \text{ dengan } \det(A) \neq 0$$

Dalam hal ini  $A$  disebut matriks koefisien sistem (2.3). Untuk memperoleh akar persamaan karakteristik  $\lambda_1$  dan  $\lambda_2$  digunakan  $\det(A - \lambda I) = 0$ . Akar persamaan karakteristik yang diperoleh digunakan untuk menentukan jenis kestabilan titik kesetimbangan sistem (2.3). Kriteria jenis kestabilan titik kesetimbangan sistem (2.3) berdasarkan akar persamaan karakteristik adalah sebagai berikut.

Tabel 2.1 Kestabilan titik kesetimbangan sistem (2.3)

Nilai Eigen	Kestabilan	Jenis	
$\lambda_1, \lambda_2 \in \mathbb{R}$	$\lambda_1, \lambda_2 > 0$	Tidak stabil	Simpul
	$\lambda_1, \lambda_2 < 0$	Stabil asimtotik	Simpul
	$\lambda_1 > 0$ $\lambda_2 < 0$	Tidak stabil	Pelana
$\lambda_{1,2} = p \pm iq \in \mathbb{C}$	$p > 0$	Tidak stabil	Spiral
	$p < 0$	Stabil asimtotik	Spiral
	$p = 0$	Stabil	Center

(Boyce dan DiPrima, 2000).

### 2.2.2 Sistem Otonomus Nonlinear

Perhatikan sistem otonomus nonlinear berikut.

$$\begin{aligned} \frac{dx}{dt} &= f(x, y), \\ \frac{dy}{dt} &= g(x, y), \end{aligned} \tag{2.4}$$

dengan  $f$  dan  $g$  adalah fungsi nonlinear yang mempunyai turunan parsial yang kontinu di titik kesetimbangan  $(x^*, y^*)$ . Deret Taylor fungsi  $f$  dan  $g$  di sekitar  $(x^*, y^*)$  dapat dinyatakan sebagai berikut

$$\begin{aligned} f(x, y) &= f(x^*, y^*) + \frac{\partial f(x^*, y^*)}{\partial x}(x - x^*) + \frac{\partial f(x^*, y^*)}{\partial y}(y - y^*) \\ &+ \eta_1(x, y), \end{aligned} \tag{2.5}$$

$$\begin{aligned} g(x, y) &= g(x^*, y^*) + \frac{\partial g(x^*, y^*)}{\partial x}(x - x^*) + \frac{\partial g(x^*, y^*)}{\partial y}(y - y^*) \\ &+ \eta_2(x, y), \end{aligned} \tag{2.6}$$

dengan  $\eta_1(x, y)$  dan  $\eta_2(x, y)$  adalah suku sisa. Untuk hampiran orde satu di atas, suku sisa memenuhi sifat

$$\lim_{(x,y) \rightarrow (x^*,y^*)} \frac{\eta_1(x,y)}{\|\vec{w}\|} = 0 \quad (2.7)$$

$$\lim_{(x,y) \rightarrow (x^*,y^*)} \frac{\eta_2(x,y)}{\|\vec{w}\|} = 0$$

dengan  $\vec{w} = (x - x^*, y - y^*)^T$ . Dengan menggunakan persamaan (2.5) dan (2.6) serta mengingat  $\frac{dx}{dt} = \frac{d(x-x^*)}{dt}$ , dan  $\frac{dy}{dt} = \frac{d(y-y^*)}{dt}$ , persamaan tersebut dapat ditulis dalam bentuk matriks, yaitu

$$\frac{d}{dt} \begin{bmatrix} x - x^* \\ y - y^* \end{bmatrix} = \begin{bmatrix} f(x^*, y^*) \\ g(x^*, y^*) \end{bmatrix} + \begin{bmatrix} \frac{\partial f(x^*, y^*)}{\partial x} & \frac{\partial f(x^*, y^*)}{\partial y} \\ \frac{\partial g(x^*, y^*)}{\partial x} & \frac{\partial g(x^*, y^*)}{\partial y} \end{bmatrix} \begin{bmatrix} x - x^* \\ y - y^* \end{bmatrix} + \begin{bmatrix} \eta_1(x, y) \\ \eta_2(x, y) \end{bmatrix}. \quad (2.8)$$

Matriks

$$J(x^*, y^*) = \frac{\partial f(x^*, y^*)}{\partial(x, y)} = \begin{bmatrix} \frac{\partial f(x^*, y^*)}{\partial x} & \frac{\partial f(x^*, y^*)}{\partial y} \\ \frac{\partial g(x^*, y^*)}{\partial x} & \frac{\partial g(x^*, y^*)}{\partial y} \end{bmatrix},$$

disebut matriks Jacobi atau *partial derivative matrix*.

Jika dimisalkan  $u = x - x^*$ , dan  $v = y - y^*$ , sehingga  $\vec{w} = (u, v)^T$  dan dengan mengingat bahwa  $f(x^*, y^*) = g(x^*, y^*) = 0$ , maka persamaan (2.8) dapat ditulis sebagai

$$\begin{bmatrix} \frac{du}{dt} \\ \frac{dv}{dt} \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x^*, y^*)}{\partial x} & \frac{\partial f(x^*, y^*)}{\partial y} \\ \frac{\partial g(x^*, y^*)}{\partial x} & \frac{\partial g(x^*, y^*)}{\partial y} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}$$

atau

$$\frac{d\vec{w}}{dt} = J\vec{w} + \vec{\eta}. \quad (2.9)$$

Berdasarkan persamaan (2.9) bila  $\vec{w} \rightarrow 0$  maka  $\vec{\eta} \rightarrow 0$ , sehingga nilai  $\vec{\eta}$  dapat diabaikan dan di sekitar  $(x^*, y^*)$  sistem nonlinear (2.4) dapat dihampiri oleh sistem linear

$$\frac{d\vec{w}}{dt} = J\vec{w} \quad (2.10)$$

Untuk  $x = x^*$ , dan  $y = y^*$  diperoleh  $(u^*, v^*) = (0, 0)$ , sehingga sistem linear (2.10) memiliki titik kesetimbangan  $(u^*, v^*) = (0, 0)$  (Boyce dan DiPrima, 2000).

### **Teorema 2.1 (Kestabilan Sistem Otonomus Nonlinear)**

Titik kesetimbangan sistem otonomus nonlinear (2.4) bersifat

1. stabil asimtotik jika titik kesetimbangan sistem hasil linearisasi (2.10) stabil asimtotik,
2. tak stabil jika titik kesetimbangan sistem hasil linearisasi (2.10) tak stabil (Boyce dan DiPrima, 2000)

### **2.3 Model Logistik**

Model pertumbuhan populasi mengasumsikan bahwa laju pertumbuhan terhadap waktu berbanding lurus dengan jumlah populasi yang ada. Dengan menggunakan kaidah logistik (*logistik law*) bahwa persediaan logistik ada batasnya, model ini mengasumsikan pada masa tertentu jumlah populasi akan mendekati titik kesetimbangan (*equilibrium*). Pada titik ini jumlah kelahiran dan kematian dianggap sama, sehingga grafiknya akan mendekati konstan (*zero growth*). Model pertumbuhan logistik dapat diturunkan dengan menggunakan asumsi:

- a. Laju pertumbuhan populasi  $\frac{1}{N} \frac{dN}{dt}$  pada saat  $N = 0$  adalah  $k$  (bernilai konstan).
- b. Laju pertumbuhan ini menurun secara linear dan bernilai 0 saat  $N = K$ .

Nilai  $k$  merupakan laju pertumbuhan intrinsik (*intrinsic growth rate*), yaitu nilai yang menggambarkan daya tumbuh suatu populasi. Dalam hal ini diasumsikan  $k > 0$ , mengingat setiap populasi memiliki potensi untuk berkembang biak. Berdasarkan asumsi di atas dapat

diturunkan suatu model pertumbuhan populasi yang disebut sebagai model pertumbuhan logistik, yaitu:

$$\frac{1}{N} \frac{dN}{dt} = k - \frac{k}{K} N$$

atau

$$\frac{dN}{dt} = kN \left(1 - \frac{N}{K}\right)$$

di mana  $K$  adalah *carrying capacity* atau kapasitas maksimum dari lingkungan untuk setiap spesies (Boyce dan DiPrima, 2000).

## 2.4 Metode Runge-Kutta

Metode Runge-Kutta merupakan metode satu langkah untuk menyelesaikan persamaan diferensial, karena metode tersebut hanya memerlukan satu titik sebelumnya untuk menghitung nilai yang baru. Metode Runge-Kutta yang populer adalah Runge-Kutta orde empat. Masalah syarat awal

$$x' = f(t, x) \quad x(t_0) = x_0$$

dapat diselesaikan menggunakan metode Runge-Kutta orde empat dengan rumus rekursif

$$\begin{aligned} t_{n+1} &= t_n + h \\ x_{n+1} &= x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), n = 0, 1, \dots \end{aligned} \quad (2.11)$$

di mana  $h$  adalah ukuran langkah dan

$$\begin{aligned} k_1 &= hf(t_n, x_n) \\ k_2 &= hf\left(t_n + \frac{h}{2}, x_n + \frac{1}{2}k_1\right) \\ k_3 &= hf\left(t_n + \frac{h}{2}, x_n + \frac{1}{2}k_2\right) \\ k_4 &= hf(t_n + h, x_n + k_3) \end{aligned} \quad (2.12)$$

Misalkan  $x_1(t), \dots, x_m(t)$  adalah solusi sistem yang terdiri dari  $m$  persamaan orde satu:

$$\begin{aligned} x_1(t) &= f_1(t, x_1, x_2, \dots, x_m), \\ x_2(t) &= f_2(t, x_1, x_2, \dots, x_m), \\ &\vdots \\ x_m(t) &= f_m(t, x_1, x_2, \dots, x_m), \end{aligned}$$

dan memenuhi syarat awal

$$x_1(t_0) = a_0, x_2(t_0) = a_2, \dots, x_m(t_0) = a_m.$$

Misalkan  $x_{n,i}$  merupakan taksiran untuk  $x_i(t_n)$  untuk  $i = 1, 2, \dots, m$  dan  $t_n = t_0 + nh$  untuk  $n = 0, 1, \dots$ , maka rumus rekursif metode Runge-Kutta orde empat untuk sistem persamaan diberikan oleh rumus (2.5) dan (2.6) dengan memandang  $x_n, k_1, k_2, k_3$ , dan  $k_4$  sebagai vektor yaitu  $x_n = (x_{n,1}, \dots, x_{n,m})$ ,  $k_1 = (k_{1,1}, \dots, k_{1,m})$ ,  $k_2 = (k_{2,1}, \dots, k_{2,m})$ , dan seterusnya. Oleh karena itu, bentuk umum metode Runge-Kutta orde empat untuk sistem persamaan diferensial adalah

$$\begin{aligned} t_{n+1} &= t_n + h, \\ x_{n+1,i} &= x_{n,i} + \frac{1}{6}(k_{1,i} + 2k_{2,i} + 2k_{3,i} + k_{4,i}), \end{aligned} \quad (2.13)$$

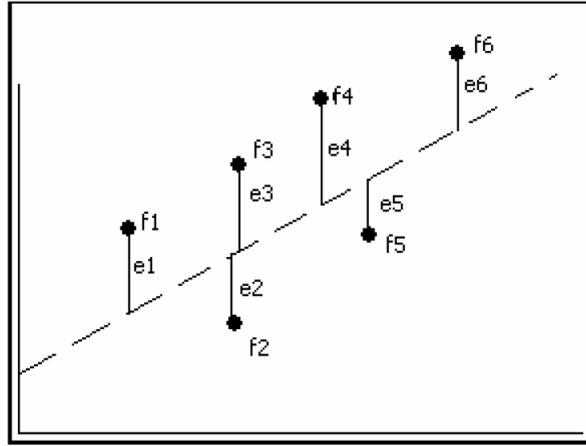
di mana  $h$  adalah ukuran langkah dan

$$\begin{aligned} k_{1,i} &= hf_i(t_n, x_{n,1}, x_{n,2}, \dots, x_{n,m}) \\ k_{2,i} &= hf_i\left(t_n + \frac{h}{2}, x_{n,1} + \frac{1}{2}k_{1,1}, x_{n,2} + \frac{1}{2}k_{1,2}, \dots, x_{n,m} + \frac{1}{2}k_{1,m}\right) \\ k_{3,i} &= hf_i\left(t_n + \frac{h}{2}, x_{n,1} + \frac{1}{2}k_{2,1}, x_{n,2} + \frac{1}{2}k_{2,2}, \dots, x_{n,m} + \frac{1}{2}k_{2,m}\right) \\ k_{4,i} &= hf_i(t_n + h, x_{n,1} + k_{3,1}, x_{n,2} + k_{3,2}, \dots, x_{n,m} + k_{3,m}) \end{aligned} \quad (2.14)$$

dengan  $n = 0, 1, \dots$  dan  $i = 1, 2, \dots, m$  (Chapra dan Raymond, 2010).

## 2.5 Metode Kuadrat Terkecil (*Least Squares Method*)

Metode kuadrat terkecil menggunakan asumsi bahwa kurva pendekatan terbaik adalah kurva yang mempunyai jumlah kuadrat kesalahan minimum (*least squares error*) terhadap data. Misal diberikan data  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  di mana  $x$  adalah variabel bebas dan  $y$  adalah variabel tak bebas. Gambar 2.1 merupakan ilustrasi jarak dari setiap titik terhadap garis yang diprediksi.



Gambar 2.1 Analisis kurva data pengamatan

Pencocokan kurva  $f(x)$  mempunyai deviasi  $e$  (*error*) pada setiap titik data, yaitu

$$e_1 = y_1 - f(x_1), e_2 = y_2 - f(x_2), \dots, e_n = y_n - f(x_n).$$

Kurva yang mendekati seperangkat data tertentu disebut jumlah kuadrat *error* dan dapat dinyatakan sebagai:

$$error = (e_1)^2 + (e_2)^2 + \dots + (e_n)^2 = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n [y_i - f(x_i)]^2, \quad (2.15)$$

semakin kecil nilai jumlah kuadrat *error*, kesesuaian antara kurva pendekatan terhadap data pengamatan akan semakin baik (Everitt dan Howell, 2005).

## 2.6 Optimasi

Optimasi merupakan masalah memaksimalkan atau meminimumkan suatu besaran tertentu yang disebut fungsi objektif. Fungsi objektif bergantung pada sejumlah variabel yang tidak saling berhubungan atau saling bergantung melalui satu atau lebih kendala. Optimasi terbagi menjadi dua, yaitu optimasi linear dan non linear. Suatu permasalahan optimasi disebut non linear jika fungsi tujuan dan kendalanya mempunyai bentuk non linear pada salah satu atau keduanya.

Optimasi non linear terbagi menjadi empat berdasarkan kriteria permasalahannya yaitu optimasi nonlinear satu variabel tanpa kendala, multi variabel tanpa kendala, satu variabel dengan kendala, dan multi variabel dengan kendala. Secara umum optimasi nonlinear multi variabel tanpa kendala untuk permasalahan minimum didefinisikan sebagai berikut:

$$\min f(x) \text{ dengan } x = (x_1, x_2, \dots, x_n) \in I \subset \mathbb{R}^n \quad (2.16)$$

di mana  $f: I \rightarrow \mathbb{R}$  dan  $I$  tertutup dan terbatas.

Tujuan penyelesaian optimasi secara umum adalah untuk memperoleh solusi optimum. Suatu titik dikatakan solusi permasalahan minimum jika titik tersebut menyebabkan nilai fungsi objektif minimum global (Bector dkk., 2004).

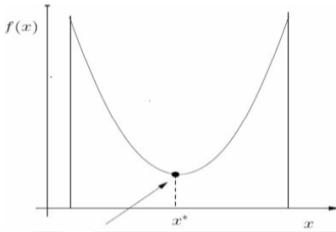
### **Definisi 2.6.1 (Minimum Lokal)**

Misalkan  $f: D \rightarrow \mathbb{R}$  fungsi sebarang di mana  $D \subseteq \mathbb{R}^n$ . Titik  $x^* \in D$  disebut minimum lokal dari  $f$ , jika terdapat persekitaran  $U$  dari  $x^*$  sehingga berlaku  $f(x^*) \leq f(x)$  untuk semua  $x \in D \cap U$  (Krumke, 2004).

### **Definisi 2.6.2 (Minimum Global)**

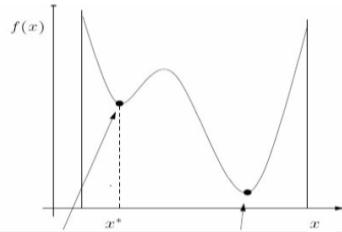
Misalkan  $f: D \rightarrow \mathbb{R}$  fungsi sebarang di mana  $D \subseteq \mathbb{R}^n$ . Titik  $x^* \in D$  disebut minimum global dari  $f$ , berlaku  $f(x^*) \leq f(x)$  untuk semua  $x \in D$  (Krumke, 2004).

Pada Gambar 2.2 ditampilkan fungsi yang memuat titik minimum lokal sekaligus minimum global dan fungsi yang memuat titik minimum lokal dan minimum global



Minimum global dan lokal

**a**



Minimum lokal

Minimum global

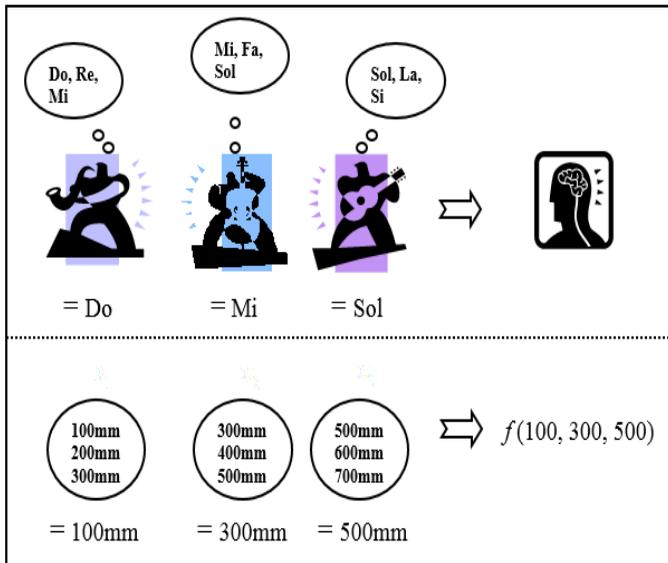
**b**

Gambar 2.2 (a). Fungsi yang memuat titik minimum lokal sekaligus minimum global. (b). Fungsi yang memuat titik minimum lokal dan minimum global

## 2.7 Algoritma *Harmony Search* (HS)

Algoritma *Harmony Search* diilhami dari proses perbaikan harmoni musik pada kelompok paduan musik jazz. Ketika kelompok paduan musik melakukan perbaikan pada harmoni musik yang dimainkan, ada beberapa cara yang digunakan antara lain memainkan harmoni musik terkenal berdasarkan ingatan mereka, memainkan harmoni musik yang serupa dengan harmoni musik yang terkenal namun ada sedikit penyesuaian, atau membuat harmoni musik baru. Ketiga komponen tersebut diformulasikan menjadi *Harmony Memory*, penyesuaian nada, dan proses pembangkitan secara random (Geem, 2009).

Dengan analogi tersebut, algoritma HS melakukan proses optimasi untuk mendapatkan keadaan terbaik dengan cara mengevaluasi fungsi objektif. Seperti halnya perkiraan estetika yang ditentukan menggunakan himpunan *pitches* yang dikeluarkan oleh alat musik, fungsi objektif pada algoritma HS dihitung menggunakan himpunan nilai-nilai pada setiap variabel keputusan (*decision variables*). Perbaikan nilai fungsi objektif pada algoritma HS menerapkan improvisasi yang terus ditingkatkan dari iterasi ke iterasi sama seperti perbaikan kualitas suara estetika yang diperbaiki dengan latihan demi latihan, Gambar 2.3 menunjukkan analogi antara improvisasi dan optimasi (Geem, 2009).



Gambar 2.3 Analogi algoritma HS

Pada Gambar 2.3, setiap pemain musik (*saxophonist*, *double bassist*, dan *guitarist*) merepresentasikan suatu *decision variabel* ( $x_1, x_2$ , dan  $x_3$ ). Kumpulan bunyi yang dihasilkan oleh setiap instrumen musik (*saxophone*={Do, Re, Mi}, *double bass*={Mi, Fa, Sol}, *guitar*={Sol, La, Si}) menyatakan rentang nilai variabel ( $x_1 = \{100, 200, 300\}$ ,  $x_2 = \{300, 400, 500\}$ ,  $x_3 = \{500, 600, 700\}$ ). Sebagai contoh, misalnya *saxophonist* mengeluarkan bunyi Do, *double bassist* mengeluarkan bunyi Mi, dan *guitarist* mengeluarkan bunyi Sol, maka ketiganya membangun suatu harmoni baru (Do, Mi, Sol). Jika harmoni ini lebih indah dibandingkan harmoni saat ini, maka harmoni baru ini dipertahankan. Harmoni yang diperoleh tersebut dalam dunia optimasi disebut dengan solusi yang direpresentasikan dalam bentuk dimensi vektor solusi (Geem, 2009).

Penggunaan *Harmony Memory* sangat penting karena *Harmony Memory* tersebut bisa menjamin bahwa harmoni yang bagus akan dipertimbangkan sebagai elemen-elemen dari vektor solusi yang baru. Agar *Harmony Memory* dapat digunakan secara efektif, algoritma *harmony search* mengadopsi sebuah parameter yang disebut *Harmony Memory Considering Rate (HMCR)*. Jika *HMCR* terlalu rendah, maka

hanya sedikit harmoni elit yang terpilih dan juga dapat menyebabkan proses konvergensi terlalu lambat. Jika  $HMCR$  terlalu besar, maka banyak nada-nada pada *Harmony Memory* yang terpakai dan tidak sempat mengeksplorasi nada lain, sehingga pada akhirnya sulit untuk mencapai solusi yang bagus. Oleh karena itu, biasanya digunakan  $0.7 \leq HMCR \leq 0.95$  (Geem, 2009).

Proses selanjutnya adalah penyesuaian nada, dalam proses penyesuaian nada terdapat beberapa parameter seperti *bandwidth* ( $bw$ ) dan *Pitch Adjusting Rate* ( $PAR$ ).  $PAR$  yang bernilai rendah dengan *bandwidth* yang sempit dapat menyebabkan proses konvergensi lambat dikarenakan keterbatasan eksplorasi pada ruang pencarian yang besar. Pada sisi lain,  $PAR$  yang tinggi dengan *bandwidth* yang lebar dapat menyebabkan solusi-solusi yang ada terlalu menyebar dari solusi optimal. Oleh karena itu, biasanya digunakan  $0.1 \leq PAR \leq 0.5$  (Geem, 2009).

Penyesuaian nada musik berarti pengubahan frekuensi nada. Hal itu berarti membangkitkan nilai yang sedikit berbeda pada algoritma HS. Berikut ini adalah formulasi penyesuaian nada:

$$x_{new} = x_{old} + bw \times \varepsilon$$

di mana  $x_{new}$  = nada baru setelah dilakukan penyesuaian nada  
 $x_{old}$  = nada yang tersimpan pada *Harmony Memory*  
 $bw$  = bandwidth  
 $\varepsilon$  = bilangan *random* dengan interval  $[-1,1]$

(Geem, 2009).

Berikut ini adalah langkah-langkah dari algoritma HS secara umum:

1. Inisialisasi masalah dan parameter algoritma.
2. Inisialisasi *Harmony Memory*.
3. Membangkitkan vektor solusi yang baru.
4. Perbaiki *Harmony Memory*.
5. Periksa kriteria pemberhentian.

Penjelasan mengenai langkah-langkah dari algoritma HS adalah:

1. Inisialisasi masalah dan parameter algoritma  
 Optimasi masalah berdasarkan fungsi tujuan, kemudian inisialisasi parameter algoritma yang terdiri dari:
  - a.  $HMS$  (*Harmony Memory Size*) adalah banyak solusi yang disimpan pada  $HM$  (*Harmony Memory*).

- b. *HMCR* (*Harmony Memory Considering Rate*) digunakan selama proses improvisasi untuk menentukan apakah variabel dari solusi tersebut dapat mengambil semua nilai pada HM. Nilai *HMCR* terletak dalam interval  $[0,1]$ .
- c. *PAR* (*Pitch Adjusting Rate*) juga digunakan selama proses improvisasi untuk menentukan apakah variabel dari solusi tersebut harus diganti ke suatu nilai tetangga. Nilai *PAR* terletak dalam interval  $[0,1]$ .

2. Inisialisasi *Harmony Memory*

Pada tahap ini dibangkitkan matriks *Harmony Memory* secara *random* yang berisi vektor-vektor solusi sebanyak *HMS*. Baris menunjukkan suatu himpunan solusi atau disebut vektor solusi  $X^i$ , sedangkan kolom menunjukkan variabel keputusan untuk tiap solusi. Setiap solusi  $X^i$  dapat dilihat sebagai salah satu susunan urutan. Pada Gambar 2.4 ditampilkan contoh matriks *Harmony Memory*:

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_N^1 \\ x_1^2 & x_2^2 & \dots & x_N^2 \\ \dots & \dots & \dots & \dots \\ x_1^{HMS-1} & x_2^{HMS-1} & \dots & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \dots & x_N^{HMS} \end{bmatrix}$$

Gambar 2.4 Matriks *harmony memory*

3. Membangkitkan vektor solusi yang baru

Pada langkah ini, suatu vektor baru  $X' = (x'_1, x'_2, \dots, x'_N)$  dibangkitkan berdasarkan tiga aturan, antara lain penggunaan *Harmony Memory*, penyesuaian nada, dan proses pembangkitan secara *random*. Dalam penggunaan *Harmony Memory*, nilai dari pemilihan pertama  $x'_1$  untuk vektor baru dipilih berdasarkan nilai yang tersedia pada *Harmony Memory* dari himpunan  $\{x_1^1, x_2^1, \dots, x_1^{HMS}\}$ , dengan kemungkinan  $HMCR \in [0,1]$ . Nilai keputusan dari  $x_2, x_3, \dots, x_N$  dipilih dengan cara yang sama. Jika nilai acak yang terpilih dengan kemungkinan  $1 - HMCR$ , maka nilai vektor solusi dipilih dalam *range* yang mungkin dari nilai tersebut.

$$x'_i = \begin{cases} x_i \in \{x_i^1, x_i^2, \dots, x_i^{HMS}\} & \text{dengan kemungkinan HMCR} \\ x_i \in X_i & \text{dengan kemungkinan } (1 - \text{HMCR}) \end{cases}$$

Setiap komponen berdasarkan pertimbangan *memory* diuji untuk menentukan apakah dapat dilakukan pencocokan nada. Langkah ini menggunakan parameter *PAR*, dengan pemilihan sebagai berikut:

$$x'_i = \begin{cases} x'_i + \text{rand}[-1,1] * bw & \text{dengan kemungkinan PAR} \\ x'_i & \text{dengan kemungkinan } (1-\text{PAR}) \end{cases}$$

di mana *bw* (*bandwidth*) adalah suatu nilai sebarang.

#### 4. Perbaiki *Harmony Memory*

Apabila nilai vektor solusi yang baru lebih baik daripada nilai vektor solusi terjelek pada *Harmony Memory* dilihat dari sudut pandang nilai fungsi tujuan, maka vektor terjelek akan dikeluarkan dan digantikan oleh vektor solusi baru. Jika nilai vektor solusi yang baru lebih buruk daripada nilai vektor solusi terjelek, maka tidak akan terjadi perubahan pada *Harmony Memory*.

#### 5. Memeriksa kriteria pemberhentian

Apabila kriteria pemberhentian telah tercapai maka iterasi dihentikan, apabila belum tercapai maka ulangi langkah tiga dan empat hingga kriteria pemberhentian dipenuhi. Terdapat beberapa macam kriteria pemberhentian seperti jumlah iterasi maksimal, tidak terjadi perubahan solusi setelah beberapa iterasi, dan lain-lain.

### 2.8 Imun non Spesifik

Sistem imun non spesifik adalah sistem imun bawaan (sudah ada) yang secara non selektif mempertahankan tubuh dari benda asing atau materi abnormal apapun jenisnya, bahkan meskipun baru pertama kali terpapar. Respon ini merupakan lini pertama pertahanan terhadap berbagai ancaman, termasuk agen infeksi iritan kimiawi dan cedera jaringan akibat trauma mekanis atau luka bakar. Komponen-komponen sistem imun bawaan selalu berada dalam keadaan siaga, siap melaksanakan tindakan-tindakan pertahanan yang terbatas dan relatif “kasar” terhadap semua penyerang. Dari berbagai sel efektor imun, neutrofil, dan makrofag merupakan spesialis fagositosis yang

sangat penting dalam pertahanan tubuh. Berbagai respon imun non spesifik diaktifkan sebagai tanggapan terhadap pola molekuler generik yang berkaitan dengan agen yang mengancam, misalnya karbohidrat yang biasanya ada di dinding sel bakteri tetapi tidak ditemukan di sel manusia (Timmreck, 2004).

Pertahanan pertama imun non spesifik mencakup beberapa hal berikut.

1. Peradangan, suatu respon non spesifik terhadap cedera jaringan di mana spesialis-spesialis fagositik, neutrofil dan makrofag berperan besar bersama dengan asupan suportif dari tipe sel imun lain.
2. *Interferon*, sekelompok protein yang secara non spesifik mempertahankan sel dari infeksi virus.
3. *Natural killer cells*, suatu kelompok khusus sel mirip limfosit yang secara spontan dan non spesifik melisis atau memecahkan dan menghancurkan sel pejamu yang terinfeksi virus dan sel kanker.
4. Sistem komplemen, sekelompok protein plasma inaktif yang jika diaktifkan secara berurutan akan merusak sel-sel asing dengan menyerang membran plasmanya.

## 2.9 Patogen

Patogen adalah agen biologis yang menyebabkan penyakit pada inangnya. Sebutan lain dari patogen adalah mikroorganisme parasit. Umumnya istilah ini diberikan untuk agen yang mengacaukan fisiologi normal hewan atau tumbuhan multiseluler dan manusia. Patogen diklasifikasikan ke dalam beberapa jenis yaitu virus, bakteri, fungi, dan nematoda. Patogenesis adalah perkembangan, produksi, atau proses pembentukan suatu penyakit. Konsep lain yang masih berkaitan adalah patogenisitas. Patogenik mengacu pada apa yang menjadi penyebab atau yang menimbulkan penyakit sementara patogenisitas menyatakan kemampuan, potensi, atau kekuatan yang dimiliki suatu substansi patogenik untuk menimbulkan penyakit. Mikroorganisme, substansi, atau *agens* yang hanya memiliki sedikit kemampuan atau bahkan tidak ada sama sekali untuk menimbulkan penyakit disebut non patogen (Timmreck, 2004).

Salah satu faktor yang membuat suatu organisme menjadi patogenik adalah kemampuannya untuk memasuki tubuh penjamu. Kemampuan untuk masuk dan menimbulkan penyakit di dalam tubuh penjamu disebut keinvasian, yaitu kapasitas suatu mikroorganisme

untuk masuk dan berkembang di dalam jaringan tubuh (Timmreck, 2004).

UNIVERSITAS BRAWIJAYA



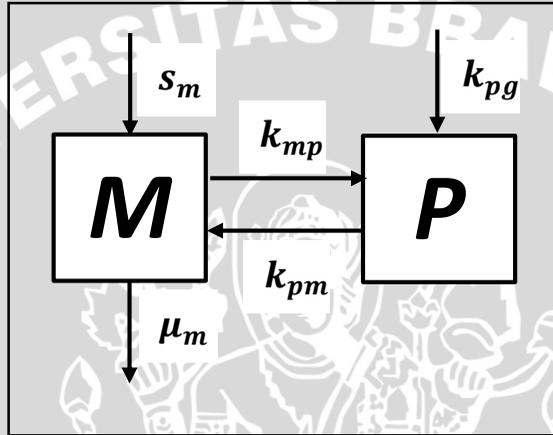
UNIVERSITAS BRAWIJAYA



## BAB III PEMBAHASAN

### 3.1 Struktur Model

Dalam mengkonstruksi model respon imun dan patogen, terdapat dua populasi yaitu, populasi imun ( $M$ ) dan populasi patogen ( $P$ ). Pada Gambar 3.1 disajikan diagram interaksi untuk mengkonstruksi model.



Gambar 3.1 Diagram interaksi

Dengan:

$M$  = populasi respon imun non spesifik pada waktu  $t$

$P$  = populasi patogen pada waktu  $t$

$S_m$  = sumber penghasil respon imun non spesifik

$\mu_m$  = laju kerusakan respon imun non spesifik

$k_{mp}$  = laju berkurangnya respon imun non spesifik oleh patogen

$k_{pg}$  = laju pertumbuhan patogen

$k_{pm}$  = laju respon imun non spesifik untuk membunuh patogen

## 3.2 Konstruksi Model Respon Imun dan Patogen

Model respon imun dan patogen diperoleh dengan memformulasikan diagram interaksi pada Gambar 3.1 ke dalam model matematika. Laju perubahan populasi imun dan patogen diperoleh dengan cara sebagai berikut.

### 3.2.1 Laju Perubahan Populasi Imun

Perubahan populasi imun bertambah dari sumbernya dengan laju  $s_m$ . Laju perubahan  $M$  pada waktu  $t$  berbanding lurus dengan banyak sel baru yang dihasilkan:

$$\frac{dM}{dt} = s_m. \quad (3.1)$$

Laju kerusakan alami  $\mu_m$  mengakibatkan jumlah imun berkurang, sehingga laju perubahan  $M$  karena kerusakan alami menjadi:

$$\frac{dM}{dt} = -\mu_m M. \quad (3.2)$$

Jumlah imun yang rusak karena patogen ditentukan oleh interaksi yang terjadi antara imun dan patogen. Patogen menyerang imun dengan laju  $k_{mp}$ , sehingga laju perubahan  $M$  karena interaksi dengan  $P$  menjadi:

$$\frac{dM}{dt} = -k_{mp}MP. \quad (3.3)$$

Berdasarkan persamaan (3.1), (3.2) dan (3.3) dapat dibuat model laju perubahan  $M$  terhadap waktu  $t$  yaitu:

$$\frac{dM}{dt} = s_m - \mu_m M - k_{mp}MP. \quad (3.4)$$

### 3.2.2 Laju Perubahan Populasi Patogen

Perubahan populasi patogen diasumsikan bertambah menggunakan pertumbuhan logistik dan akan mendekati daya dukung lingkungan yaitu  $p_\infty$ , sehingga laju populasi  $P$  yang ada tidak dapat melebihi daya dukung lingkungan:

$$\frac{dP}{dt} = k_{pg}P - k_{pg} \frac{P^2}{p_\infty}. \quad (3.5)$$

Jumlah patogen yang rusak karena imun ditentukan oleh interaksi yang terjadi antara imun dan patogen. Imun menyerang patogen dengan laju  $k_{pm}$ , sehingga laju perubahan  $P$  karena interaksi dengan  $M$ :

$$\frac{dP}{dt} = -k_{pm}MP. \quad (3.6)$$

Berdasarkan persamaan (3.5) dan (3.6) dapat dibuat model laju perubahan  $P$  terhadap waktu  $t$  yaitu:

$$\frac{dP}{dt} = k_{pg}P - k_{pg}\frac{P^2}{P_\infty} - k_{pm}MP. \quad (3.7)$$

Selanjutnya, diasumsikan respon imun mencapai kondisi stabil, yaitu ketika  $\frac{dM}{dt} = 0$  dan substitusikan

$$M = \frac{S_m}{\mu_m + k_{mp}P}$$

ke persamaan (3.7), maka diperoleh model laju perubahan  $P$  terhadap waktu  $t$  yaitu:

$$\frac{dP}{dt} = k_{pg}P - k_{pg}\frac{P^2}{P_\infty} - \frac{S_mk_{pm}P}{\mu_m + k_{mp}P}. \quad (3.8)$$

### 3.2.3 Model Respon Imun dan Patogen

Berdasarkan persamaan (3.4) dan (3.8) diperoleh model respon imun dan patogen sebagai berikut:

$$\begin{aligned} \frac{dM}{dt} &= S_m - \mu_m M - k_{mp}MP, \\ \frac{dP}{dt} &= k_{pg}P - k_{pg}\frac{P^2}{P_\infty} - \frac{S_mk_{pm}P}{\mu_m + k_{mp}P}. \end{aligned} \quad (3.9)$$

### 3.3 Titik Kesetimbangan

Titik kesetimbangan sistem persamaan (3.9) diperoleh jika

$$\frac{dM}{dt} = \frac{dP}{dt} = 0,$$

sehingga diperoleh

$$S_m - \mu_m M - k_{mp}MP = 0, \quad (3.10a)$$

$$k_{pg}P - k_{pg}\frac{P^2}{P_\infty} - \frac{S_mk_{pm}P}{\mu_m + k_{mp}P} = 0. \quad (3.10b)$$

Berdasarkan persamaan (3.10a) dan (3.10b) diperoleh

$$\begin{aligned} M &= \frac{S_m}{\mu_m + k_{mp}P}, \\ P &= 0 \end{aligned}$$

atau

$$P = \frac{d-e+f}{g},$$

atau

$$P = \frac{-d+e+f}{-g},$$

dengan

$$\begin{aligned} d &= k_{pg} p_{\infty} k_{mp}, \\ e &= k_{pg} \mu_m, \end{aligned}$$

$$\begin{aligned} f &= \sqrt{k_{pg}^2 p_{\infty}^2 k_{mp}^2 + 2 k_{pg}^2 p_{\infty} k_{mp} \mu_m + k_{pg}^2 \mu_m^2 - 4 k_{pg} k_{mp} k_{pm} S_m p_{\infty}}, \\ g &= 2 k_{pg} k_{mp}. \end{aligned}$$

Berdasarkan perhitungan, diperoleh tiga titik kesetimbangan. Pada simulasi estimasi parameter hanya digunakan satu titik kesetimbangan yaitu  $E_1(M, P)$ . Sedangkan titik kesetimbangan yang lain dapat dilakukan secara analog.

### 3.3.1 Titik Kesetimbangan $E_1(M, P)$

Titik kesetimbangan pertama yaitu  $E_1(M, P)$  dengan

$$\begin{aligned} M &= \frac{S_m}{\mu_m}, \\ P &= 0. \end{aligned}$$

### 3.3.2 Titik Kesetimbangan $E_2(M, P)$

Titik kesetimbangan kedua yaitu  $E_2(M, P)$  dengan

$$\begin{aligned} M &= \frac{S_m}{\mu_m + k_{mp} \left( \frac{d-e+f}{g} \right)}, \\ P &= \frac{d-e+f}{g}, \end{aligned}$$

### 3.3.3 Titik Kesetimbangan $E_3(M, P)$

Titik kesetimbangan ketiga yaitu  $E_3(M, P)$  dengan

$$M = \frac{S_m}{\mu_m + k_{mp} \left( \frac{-d+e+f}{-g} \right)},$$

$$P = \frac{-d+e+f}{-g},$$

### 3.4 Analisis Kestabilan Titik Kesetimbangan

Sebelum menentukan kestabilan setiap titik kesetimbangan terlebih dahulu dilakukan perhitungan matriks Jacobi sistem persamaan agar diperoleh nilai eigen yang digunakan untuk menentukan kestabilan titik kesetimbangan. Matriks Jacobi sistem persamaan (3.9) adalah

$$J(M, P) = \begin{bmatrix} -\mu_m - k_{mp}P & -k_{mp}M \\ 0 & k_{pg} - 2k_{pg} \frac{P}{P_\infty} - \frac{\mu_m s_m k_{pm}}{(\mu_m + k_{mp}P)^2} \end{bmatrix}. \quad (3.11)$$

#### 3.4.1 Analisis Kestabilan Titik Kesetimbangan $E_1(M, P)$

Jika titik kesetimbangan  $E_1$  disubstitusikan ke dalam persamaan (3.11), maka diperoleh matriks  $J(E_1)$ :

$$J(E_1) = \begin{bmatrix} -\mu_m & -\frac{k_{mp}s_m}{\mu_m} \\ 0 & k_{pg} - \frac{s_m k_{pm}}{\mu_m} \end{bmatrix}.$$

Nilai eigen untuk  $J(E_1)$  adalah

$$\lambda_1 = k_{pg} - \frac{s_m k_{pm}}{\mu_m}, \quad (3.12)$$

$$\lambda_2 = -\mu_m.$$

Nilai  $\lambda_2 = -\mu_m < 0$  dan nilai  $\lambda_1$  akan bernilai negatif apabila

$$k_{pg} - \frac{s_m k_{pm}}{\mu_m} < 0,$$

$$k_{pg} < \frac{s_m k_{pm}}{\mu_m}.$$

Berdasarkan nilai eigen ini, jenis kestabilan titik kesetimbangan  $E_1$  bersifat stabil asimtotik dengan syarat yang harus dipenuhi adalah

$$k_{pg} < \frac{s_m k_{pm}}{\mu_m}.$$

#### 3.4.2 Analisis Kestabilan Titik Kesetimbangan $E_2(M, P)$

Jika titik kesetimbangan  $E_2$  disubstitusikan ke dalam persamaan (3.11), maka diperoleh matriks  $J(E_2)$ :

$$\begin{bmatrix} -\mu_m - \frac{k_{mp}(d-e+f)}{g} & -k_{mp} \left( \frac{sm}{\mu_m + \frac{k_{mp}(d-e+f)}{g}} \right) \\ 0 & k_{pg} - \frac{2k_{pg}(d-e+f)}{g.P_\infty} - \frac{\mu_m s m k_{pm}}{\left( \mu_m + \frac{k_{mp}(d-e+f)}{g} \right)^2} \end{bmatrix}$$

Nilai eigen  $J(E_2)$  adalah

$$\begin{aligned} \lambda_1 &= -\mu_m - \frac{k_{mp}(d-e+f)}{g}, \\ \lambda_2 &= k_{pg} - \frac{2k_{pg}(d-e+f)}{g.P_\infty} - \frac{\mu_m s m k_{pm}}{\left( \mu_m + \frac{k_{mp}(d-e+f)}{g} \right)^2}. \end{aligned} \quad (3.13)$$

Berdasarkan nilai eigen ini, jenis kestabilan titik kesetimbangan  $E_2$  bersifat stabil asimtotik dengan syarat yang harus dipenuhi adalah

$$\begin{aligned} -\mu_m - \frac{k_{mp}(d-e+f)}{g} &< 0, \\ k_{pg} - \frac{2k_{pg}(d-e+f)}{g.P_\infty} - \frac{\mu_m s m k_{pm}}{\left( \mu_m + \frac{k_{mp}(d-e+f)}{g} \right)^2} &< 0. \end{aligned}$$

### 3.4.3 Analisis Kestabilan Titik Kesetimbangan $E_3(M, P)$

Jika titik kesetimbangan  $E_3$  disubstitusikan ke dalam persamaan (3.11), maka diperoleh matriks  $J(E_3)$ :

$$\begin{bmatrix} -\mu_m - \frac{k_{mp}(-d+e+f)}{-g} & -k_{mp} \left( \frac{sm}{\mu_m + \frac{k_{mp}(-d+e+f)}{-g}} \right) \\ 0 & k_{pg} - \frac{2k_{pg}(-d+e+f)}{-g.P_\infty} - \frac{\mu_m s m k_{pm}}{\left( \mu_m + \frac{k_{mp}(-d+e+f)}{-g} \right)^2} \end{bmatrix}$$

Nilai eigen  $J(E_3)$  adalah

$$\begin{aligned} \lambda_1 &= -\mu_m - \frac{k_{mp}(-d+e+f)}{-g}, \\ \lambda_2 &= k_{pg} - \frac{2k_{pg}(-d+e+f)}{-g.P_\infty} - \frac{\mu_m s m k_{pm}}{\left( \mu_m + \frac{k_{mp}(-d+e+f)}{-g} \right)^2}. \end{aligned} \quad (3.14)$$

Berdasarkan nilai eigen ini, jenis kestabilan titik kesetimbangan  $E_3$  bersifat stabil asimtotik dengan syarat yang harus dipenuhi adalah

$$-\mu_m - \frac{k_{mp}(-d+e+f)}{-g} < 0,$$

$$k_{pg} - \frac{2k_{pg}(-d+e+f)}{-g.P_{\infty}} - \frac{\mu_m s_m k_{pm}}{\left(\mu_m + \frac{k_{mp}(-d+e+f)}{-g}\right)^2} < 0.$$

### 3.5 Estimasi Parameter Model Respon Imun dan Patogen (Kasus Satu)

#### 3.5.1 Estimasi Parameter Menggunakan Data yang Dibangkitkan

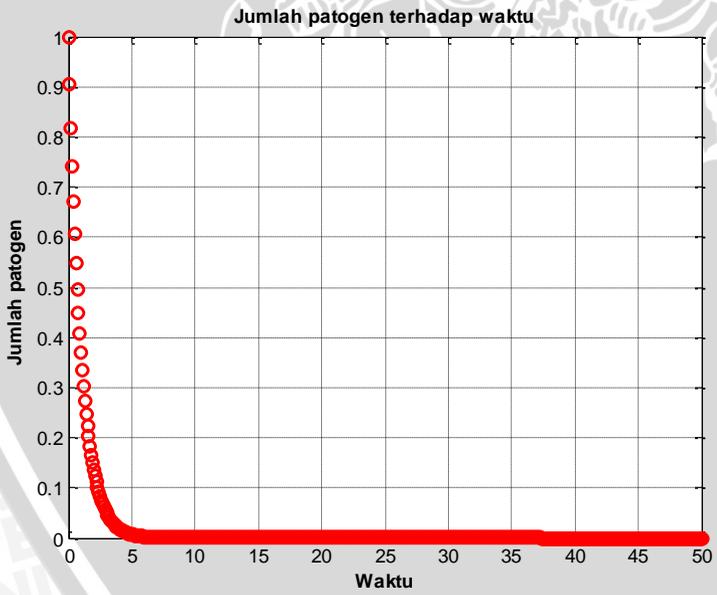
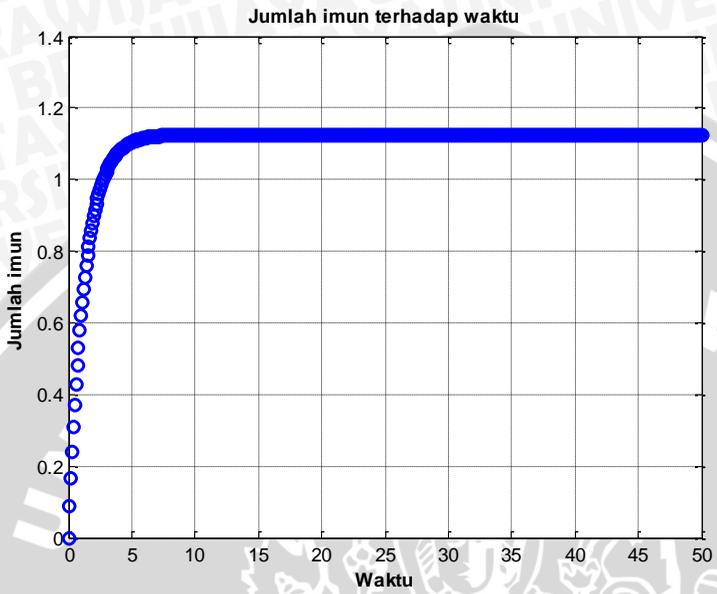
Secara garis besar proses simulasi estimasi parameter model dilakukan dengan langkah-langkah sebagai berikut.

1. Menentukan parameter yang akan dipakai kembali pada algoritma HS, meliputi  $HMS$ ,  $HMCR$ ,  $PAR$ ,  $bw$ , batas atas, dan batas bawah solusi masing-masing parameter, dan iterasi maksimum.
2. Membangkitkan vektor solusi secara *random* untuk masing-masing parameter yang akan diestimasi yaitu  $s_m, \mu_m, k_{mp}, k_{pg}, k_{pm}$ , kemudian dibentuk matriks *Harmony Memory*.
3. Menyelesaikan model secara numerik menggunakan metode Runge-Kutta orde empat berdasarkan parameter yang diperoleh dari estimasi awal.
4. Menghitung nilai fungsi objektif yang berupa jumlah kuadrat *error* antara data dengan data prediksi (penyelesaian sistem persamaan diferensial secara numerik dengan Runge-Kutta orde empat) menggunakan metode kuadrat terkecil,

$$h = error(s_m, \mu_m, k_{mp}, k_{pg}, k_{pm}) = \sum_{i=1}^2 \sum_{j=1}^n [y_{i,j} - y_{i,j}^*]^2.$$

5. Meminimumkan fungsi  $h$  menggunakan algoritma HS.

Data yang digunakan untuk estimasi parameter adalah data yang dibangkitkan dari model dengan menggunakan nilai parameter  $s_m = 0.9, \mu_m = 0.8, k_{mp} = 0.001, k_{pg} = 0.022$ , dan  $k_{pm} = 0.9$ , di mana grafik solusi numerik dapat dilihat pada Gambar 3.2. Berdasarkan data bangkitan, dilakukan proses estimasi menggunakan algoritma HS, dengan nilai parameter  $s_m, \mu_m, k_{mp}, k_{pg}, k_{pm}$  terletak pada interval  $[0,1]$ ,  $HMS = 10, HMCR = 0.8, PAR = 0.2$ , dan  $bw = 0.001$ .



Gambar 3.2 Solusi numerik  $M$  dan  $P$  terhadap waktu  $t$

### 3.5.2 Solusi Numerik Menggunakan Metode Runge-Kutta Orde Empat

Andaikan persamaan (3.9) mempunyai nilai parameter  $s_m = 0.9$ ,  $\mu_m = 0.8$ ,  $k_{mp} = 0.001$ ,  $k_{pg} = 0.022$ ,  $k_{pm} = 0.9$ ,  $P_\infty = 2$ ,  $M = 0$ ,  $P = 1$ , maka sistem persamaan diferensial dapat diselesaikan menggunakan metode Runge-Kutta orde empat. Dengan menggunakan program Runge-Kutta Orde empat pada software Matlab (*source code* dapat dilihat pada Lampiran 1), diperoleh kurva solusi populasi respon imun dan populasi patogen. Berdasarkan titik kesetimbangan  $E_1$ , solusi numerik konvergen ke titik  $M = \frac{s_m}{\mu_m} = \frac{0.9}{0.8} = 1.125$  dan  $P = 0$ . Kurva solusi model respon imun dan patogen dapat dilihat pada Gambar 3.3. Berdasarkan Gambar 3.3 terlihat bahwa solusi mulai stabil di titik  $t = 10$ , sehingga untuk simulasi cukup diambil  $t \in [0,10]$ .





### 3.5.3 Implementasi Algoritma *Harmony Search* (HS)

Pada bagian ini dijelaskan bagaimana implementasi algoritma HS sebagai metode optimasi untuk mencari nilai parameter yang optimal (dengan jumlah kuadrat *error* minimum). Fungsi objektif yang digunakan adalah jumlah kuadrat *error* (*least squares method*), dengan  $s_m, \mu_m, k_{mp}, k_{pg}, k_{pm} \in [0,1]$ . Langkah-langkah algoritma HS sebagai berikut (*source code* dapat dilihat pada Lampiran 2).

#### 1. Inisialisasi masalah dan parameter algoritma

Pada tahap ini diperkenalkan fungsi yang akan diminimalkan. Fungsi berupa jumlah kuadrat selisih antara data bangkitan dengan solusi numerik data hasil estimasi yang dinyatakan sebagai:

$$h = error = \sum_{i=1}^2 \sum_{j=1}^n [y_{i,j} - y_{i,j}^*]^2,$$

dengan  $y_{i,j}$  adalah solusi persamaan diferensial data bangkitan dan  $y_{i,j}^*$  adalah solusi persamaan diferensial data hasil estimasi. Selanjutnya parameter untuk algoritma HS ditentukan sebagai berikut:

$$HMS = 10.$$

$$HMCR = 0.8.$$

$$PAR = 0.2.$$

$$x_U = 1, x_L = 0.$$

$$bw = 0.001.$$

Untuk tahap awal, kriteria pemberhentian yang digunakan adalah maksimum iterasi=500 dan *galat* < 0.00001.

#### 2. Inisialisasi *Harmony Memory*

Pada tahap ini dibangkitkan vektor solusi secara random. Vektor solusi pada algoritma HS merepresentasikan nilai prioritas aktivitas parameter. Kemudian pada setiap iterasi, nilai prioritas aktivitas akan diperbaiki melalui mekanisme algoritma HS. Vektor solusi yang merepresentasikan nilai prioritas aktivitas parameter dinotasikan sebagai berikut.

$$X_i = [x_1^i \quad x_2^i \quad \dots \quad x_{N-1}^i \quad x_N^i],$$

di mana  $X$  = vektor solusi,  $i$  = indeks vektor solusi,  $x$  = nilai prioritas aktivitas, dan  $N$  = banyaknya aktivitas.

Vektor solusi dibangkitkan sebanyak  $HMS$ , apabila vektor solusi telah dibangkitkan maka akan terbentuk matriks *Harmony Memory* seperti pada Gambar 2.4.

Pada estimasi parameter, proses pembangkitan vektor solusi sebanyak  $HMS$  dilakukan dengan membangkitkan vektor solusi secara random, di mana daerah interval  $s_m, \mu_m, k_{mp}, k_{pg}, k_{pm} \in [0,1]$ .

Vektor solusi pertama untuk  $s_m = [0.8147, 0.0975, 0.1576, 0.1419, 0.6557, 0.7577, 0.7060, 0.8235, 0.4387, 0.4898]$ .

Vektor solusi kedua untuk  $\mu_m = [0.9058, 0.2785, 0.9706, 0.4218, 0.0357, 0.7431, 0.0318, 0.6948, 0.3816, 0.4456]$ .

Vektor solusi ketiga untuk  $k_{mp} = [0.1270, 0.5469, 0.9572, 0.9157, 0.8491, 0.3922, 0.2769, 0.3171, 0.7655, 0.6463]$ .

Vektor solusi keempat untuk  $k_{pg} = [0.9134, 0.9575, 0.4864, 0.7922, 0.9340, 0.6555, 0.0462, 0.9502, 0.7952, 0.7094]$ .

Vektor solusi kelima untuk  $k_{pm} = [0.6324, 0.9649, 0.8003, 0.9595, 0.6787, 0.1712, 0.0971, 0.0344, 0.1869, 0.7547]$ .

Berdasarkan semua vektor solusi yang telah dibangkitkan maka akan terbentuk matriks *Harmony Memory* sebagai berikut:

0.8147	0.9058	0.1270	0.9134	0.6324	$h = 78.7126$
0.0975	0.2785	0.5469	0.9575	0.9649	$h = 366.0027$
0.1576	0.9706	0.9572	0.4854	0.8003	$h = 319.4244$
0.1419	0.4218	0.9157	0.7922	0.9595	$h = 354.8761$
0.6557	0.0357	0.8491	0.9340	0.6787	$h = 51.3063$
0.7577	0.7431	0.3922	0.6555	0.1712	$h = 248.5756$
0.7060	0.0318	0.2769	0.0462	0.0971	$h = 491.9067$
0.8235	0.6948	0.3171	0.9502	0.0344	$h = 331.1240$
0.4387	0.3816	0.7655	0.7952	0.1869	$h = 341.8054$
0.4898	0.4456	0.6463	0.7094	0.7547	$h = 133.3888$

### 3. Membangkitkan vektor solusi yang baru

Proses pembangkitan vektor solusi yang baru melalui tiga aturan, antara lain:

- Apabila bilangan *random* yang dibangkitkan lebih besar dari  $HMCR$ , maka akan dibangkitkan variabel keputusan yang baru secara *random*

$$x'(i) = \text{random}[x_L(i), x_U(i)], \quad (3.15)$$

di mana  $x'$  = variabel keputusan baru,  
 $x_L$  = batas bawah variabel keputusan,  
 $x_U$  = batas atas variabel keputusan.

- b. Misalkan  $int(x)$  menyatakan bilangan bulat yang lebih besar atau sama dengan  $x$ . Apabila bilangan *random* yang dibangkitkan lebih kecil dari *HMCR* dan pembangkitan bilangan *random* berikutnya lebih besar dari *PAR*, maka variabel keputusan ke- $i$  akan diambil dari *Harmony Memory*

$$x1' = int[1 + (HMS - 1)random], \quad (3.16)$$

$$x2' = HM(x1', i), \quad (3.17)$$

$$x'(i) = x2', \quad (3.18)$$

di mana  $x1'$  = nilai yang menyatakan pemilihan lokasi pada *Harmony Memory* secara *random*,

$int$  = bilangan bulat,

$x2'$  = nilai variabel keputusan yang diambil dari *Harmony Memory*.

- c. Apabila bilangan *random* yang dibangkitkan lebih kecil dari *HMCR* dan pembangkitan bilangan *random* berikutnya lebih kecil dari *PAR*, maka terdapat penyesuaian terhadap variabel keputusan ke- $i$  yang telah diambil dari *Harmony Memory*

$$x3' = x2' + bw \times \varepsilon, \quad (3.19)$$

$$x'(i) = x3', \quad (3.20)$$

di mana  $x3'$  = nilai variabel keputusan setelah dilakukan penyesuaian,

$bw$  = *bandwidth*,

$\varepsilon$  = bilangan *random* dengan interval  $[-1,1]$ .

Kemudian setelah dilakukan proses penyesuaian akan dilakukan pengecekan terhadap batas atas dan batas bawah variabel keputusan. Apabila  $x3' < x_L$ , maka  $x3' = x_L$  selanjutnya apabila  $x3' > x_U$ , maka  $x3' = x_U$ . Batas bawah semua variabel keputusan adalah 0 dan batas atasnya adalah 1.

Pada estimasi parameter, proses improvisasi tersebut direpresentasikan sebagai berikut.

- i. Nilai  $x'_1$

Bilangan *random* yang dibangkitkan misalnya  $r_1 = 0.2760$ .

Nilai tersebut kemudian disesuaikan dengan parameter *HMCR*. Karena bilangan *random* yang dibangkitkan lebih

kecil dari nilai  $HMCR$ , maka variabel keputusan dipilih secara acak dengan aturan (3.16)

$$x1' = \text{int}[1 + (HMS - 1)\text{random}],$$

di mana  $\text{random} = 0.6797$  dan  $HMS = 10$ , sehingga diperoleh

$$x1' = \text{int}[1 + (10 - 1)0.6797],$$

$$x1' = \text{int}[7.1173],$$

$$x1' = 8,$$

$$x'_1 = HM(x1', 1),$$

$$x'_1 = HM(8, 1),$$

$$x'_1 = 0.8235.$$

Selanjutnya dilakukan pembangkitan bilangan  $\text{random}$  untuk melakukan penyesuaian dengan nilai  $PAR$ . Bilangan yang dibangkitkan secara  $\text{random}$  pada tahap  $PAR$  adalah  $r_2 = 0.6551$ . Karena bilangan  $\text{random}$  tersebut lebih besar dari nilai  $PAR$ , maka nilai  $x'_1 = 0.8235$  tetap dipertahankan.

ii. Nilai  $x'_2$

Bilangan  $\text{random}$  yang dibangkitkan misalnya  $r_1 = 0.1626$ . Nilai tersebut kemudian disesuaikan dengan parameter  $HMCR$ . Karena bilangan  $\text{random}$  yang dibangkitkan lebih kecil dari nilai  $HMCR$ , maka variabel keputusan dipilih secara acak dengan aturan (3.16)

$$x1' = \text{int}[1 + (HMS - 1)\text{random}],$$

di mana  $\text{random} = 0.1190$  dan  $HMS = 10$ , sehingga diperoleh

$$x1' = \text{int}[1 + (10 - 1)0.1190],$$

$$x1' = \text{int}[2.0710],$$

$$x1' = 3,$$

$$x'_2 = HM(x1', 2),$$

$$x'_2 = HM(3, 2),$$

$$x'_2 = 0.9706.$$

Selanjutnya dilakukan pembangkitan bilangan  $\text{random}$  untuk melakukan penyesuaian dengan nilai  $PAR$ . Bilangan yang dibangkitkan secara  $\text{random}$  pada tahap  $PAR$  adalah  $r_2 = 0.4984$ . Karena bilangan  $\text{random}$  tersebut lebih besar dari nilai  $PAR$ , maka nilai  $x'_2 = 0.9706$  tetap dipertahankan.

iii. Nilai  $x'_3$

Bilangan *random* yang dibangkitkan misalnya  $r_1 = 0.9597$ . Nilai tersebut kemudian disesuaikan dengan parameter *HMCR*. Karena bilangan *random* yang dibangkitkan lebih besar dari nilai *HMCR*, maka variabel keputusan dipilih secara acak dengan aturan (3.15). Variabel keputusan yang terbentuk adalah  $x'_3 = 0.3404$ .

iv. Nilai  $x'_4$

Bilangan *random* yang dibangkitkan misalnya  $r_1 = 0.5853$ . Nilai tersebut kemudian disesuaikan dengan parameter *HMCR*. Karena bilangan *random* yang dibangkitkan lebih kecil dari nilai *HMCR*, maka variabel keputusan dipilih secara acak dengan aturan (3.16)

$$x1' = \text{int}[1 + (HMS - 1)\text{random}],$$

di mana  $\text{random} = 0.2238$  dan  $HMS = 10$ , sehingga diperoleh

$$x1' = \text{int}[1 + (10 - 1)0.2238],$$

$$x1' = \text{int}[3.0142],$$

$$x1' = 4,$$

$$x'_4 = HM(x1', 2),$$

$$x'_4 = HM(4, 4),$$

$$x'_4 = 0.7922.$$

Selanjutnya dilakukan pembangkitan bilangan *random* untuk melakukan penyesuaian dengan nilai *PAR*. Bilangan yang dibangkitkan secara *random* pada tahap *PAR* adalah  $r_2 = 0.7513$ . Karena bilangan *random* tersebut lebih besar dari nilai *PAR*, maka nilai  $x'_4 = 0.7922$  tetap dipertahankan.

v. Nilai  $x'_5$

Bilangan *random* yang dibangkitkan misalnya  $r_1 = 0.2551$ . Nilai tersebut kemudian disesuaikan dengan parameter *HMCR*. Karena bilangan *random* yang dibangkitkan lebih kecil dari nilai *HMCR*, maka variabel keputusan dipilih secara acak dengan aturan (3.16)

$$x1' = \text{int}[1 + (HMS - 1)\text{random}],$$

di mana  $\text{random} = 0.5060$  dan  $HMS = 10$ , sehingga diperoleh

$$x1' = \text{int}[1 + (10 - 1)0.5060],$$

$$x1' = \text{int}[5.5540],$$

$$x1' = 6,$$

$$x'_5 = \text{HM}(x1', 2),$$

$$x'_5 = \text{HM}(6, 5),$$

$$x'_5 = 0.1712.$$

Selanjutnya dilakukan pembangkitan bilangan *random* untuk melakukan penyesuaian dengan nilai *PAR*. Bilangan yang dibangkitkan secara *random* pada tahap *PAR* adalah  $r_2 = 0.6991$ . Karena bilangan *random* tersebut lebih besar dari nilai *PAR*, maka nilai  $x'_5 = 0.1712$  tetap dipertahankan.

Setelah proses selesai dilakukan maka diperoleh suatu vektor solusi baru  $x^{new} = [0.8235, 0.9706, 0.3404, 0.7922, 0.1712]$ . Berdasarkan vektor solusi baru, dihitung nilai fungsi objektifnya sehingga diperoleh  $h(x^{new}) = 274.9885$

#### 4. Perbaiki *Harmony Memory*

Langkah perbaikan *Harmony Memory* dengan melakukan pengecekan terhadap vektor solusi baru. Apabila nilai vektor solusi baru lebih baik daripada vektor solusi terjelek pada *Harmony Memory* dilihat dari sudut pandang nilai fungsi tujuan, maka vektor solusi yang baru akan dimasukkan ke dalam *Harmony Memory* dan vektor solusi terjelek pada *Harmony Memory* akan dikeluarkan. Apabila nilai vektor solusi yang baru lebih buruk daripada nilai vektor solusi terjelek pada *Harmony Memory*, maka tidak akan terjadi perubahan pada *Harmony Memory*. Karena  $h(x^{new}) = 274.9885$  lebih baik dari vektor terjelek pada *Harmony Memory*  $h(x^7) = 491.9067$ , maka vektor solusi baru dimasukkan ke dalam *Harmony Memory* menggantikan vektor solusi terjelek.

#### 5. Memeriksa kriteria pemberhentian

Kriteria pemberhentian menentukan banyaknya proses pembangkitan solusi baru dan meperbarui HM. Apabila kriteria pemberhentian terpenuhi maka proses iterasi akan berhenti. Karena untuk tahap awal digunakan maksimum iterasi 500 dan  $galat < 0.00001$ , maka iterasi terus dilanjutkan.

Pada saat iterasi ke-1, hasil estimasi yang diperoleh adalah nilai minimum (jumlah kuadrat *error*) sebesar 51.306340, dengan parameter ditampilkan pada Tabel 3.1 dan waktu komputasi selama 3.732619 detik.

Tabel 3.1 Parameter hasil estimasi iterasi ke-1

	$S_m$	$\mu_m$	$k_{mp}$	$k_{pg}$	$k_{pm}$
Data Bangkitan	0.9	0.8	0.001	0.022	0.9
Data Estimasi	0.6557	0.0357	0.8491	0.9340	0.6781

Gambar 3.4 merepresentasikan perbandingan antara solusi numerik yang dihasilkan dan data yang dibangkitkan. Pada saat iterasi ke-50, terlihat kurva solusi mulai mendekati data, hasil estimasi yang diperoleh adalah nilai minimum (jumlah kuadrat *error*) sebesar 2.399683, dengan parameter tertera pada Tabel 3.2 dengan waktu komputasi selama 16.085471 detik.

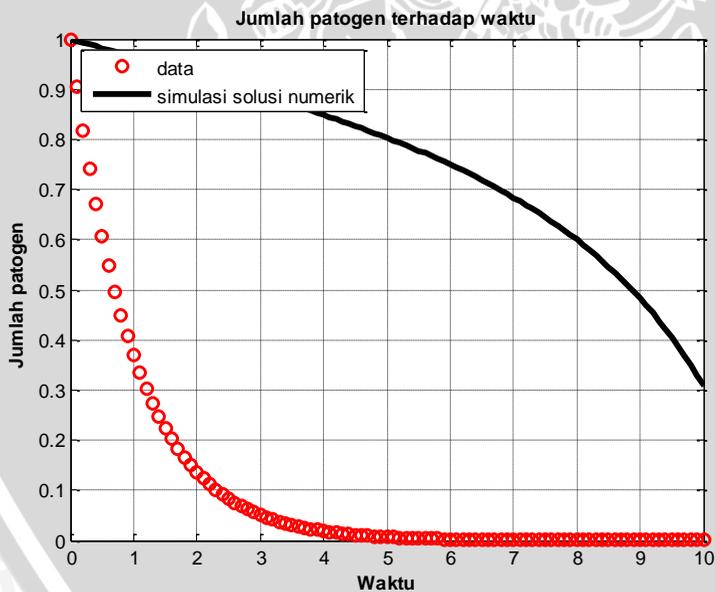
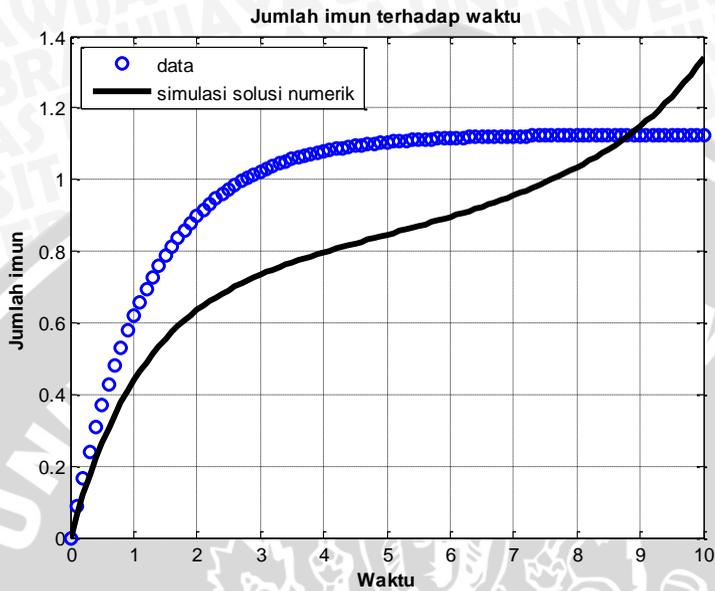
Tabel 3.2 Parameter hasil estimasi iterasi ke-50

	$S_m$	$\mu_m$	$k_{mp}$	$k_{pg}$	$k_{pm}$
Data Bangkitan	0.9	0.8	0.001	0.022	0.9
Data Estimasi	0.5752	0.4466	0.2543	0.0782	0.6790

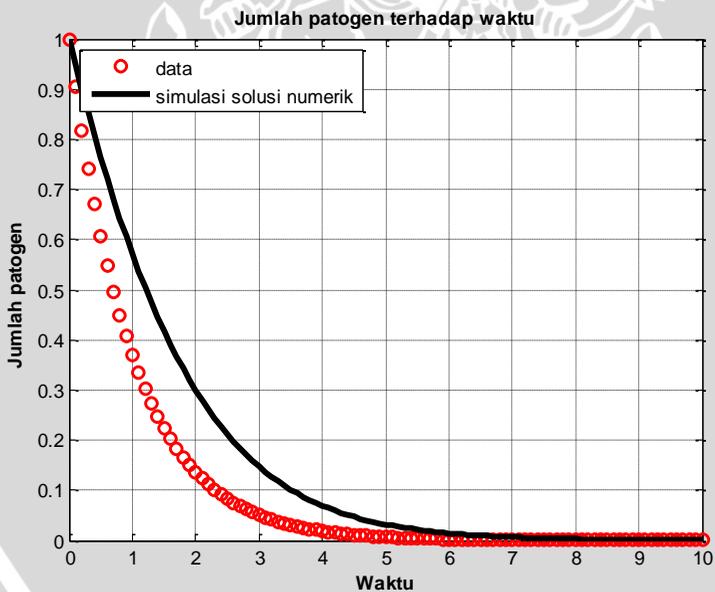
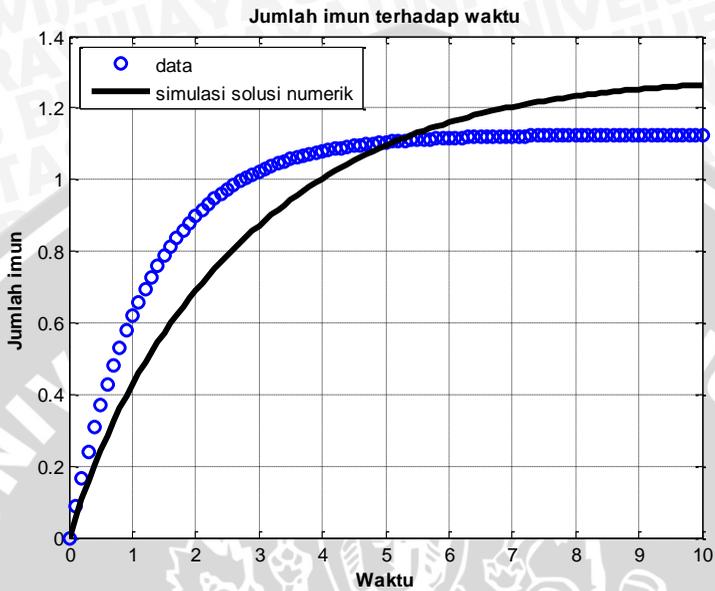
Gambar 3.5 ditampilkan perbandingan antara solusi numerik yang dihasilkan dan data yang dibangkitkan. Setelah mencapai iterasi ke-500, kurva solusi sudah mulai konvergen seperti pada Gambar 3.6, dapat dijelaskan bahwa nilai minimum (jumlah kuadrat *error*) sebesar 0.867083 dengan waktu komputasi selama 133.517081 detik dan parameter ditampilkan pada Tabel 3.3. Grafik konvergensi algoritma HS setelah iterasi ke-500 tertera pada Gambar 3.7.

Tabel 3.3 Parameter hasil estimasi iterasi ke-500

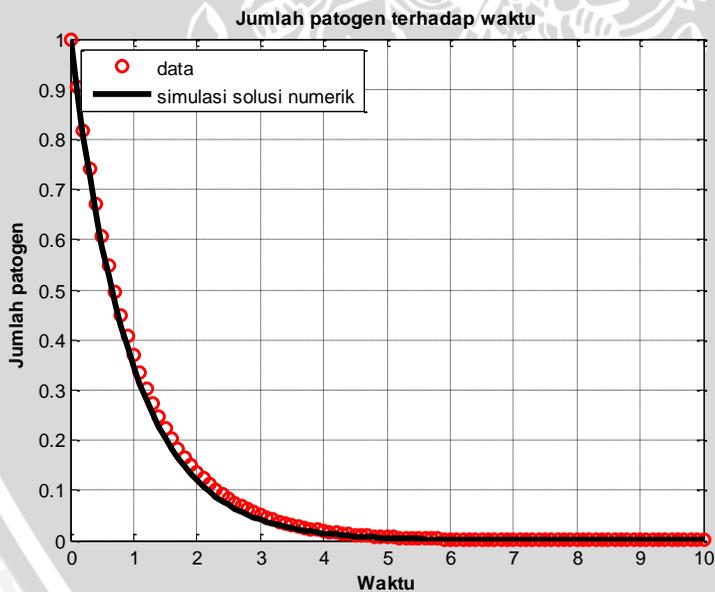
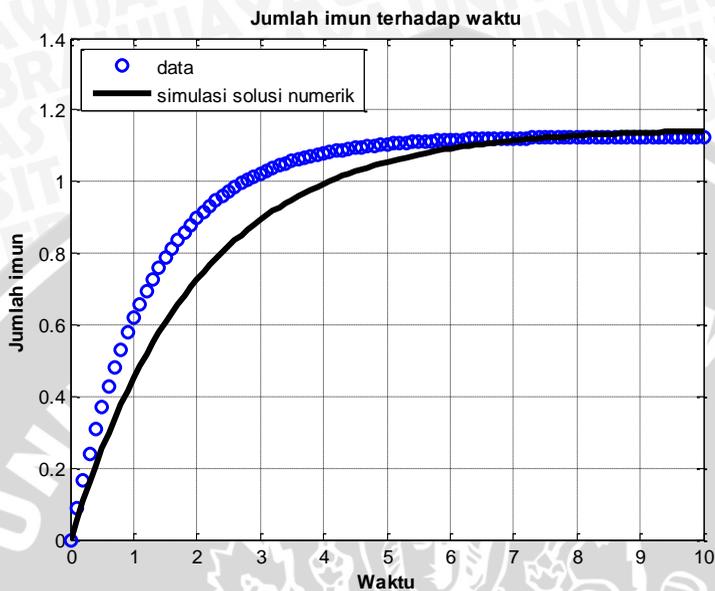
	$S_m$	$\mu_m$	$k_{mp}$	$k_{pg}$	$k_{pm}$
Data Bangkitan	0.9	0.8	0.001	0.022	0.9
Data Estimasi	0.5750	0.5005	0.0004	0.0319	0.9447



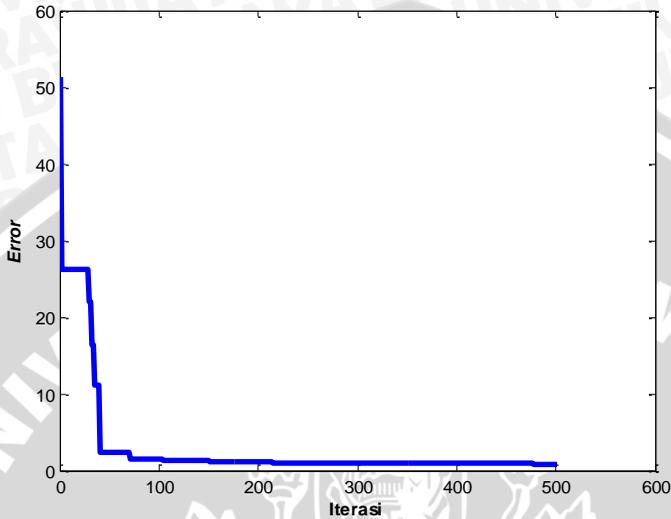
Gambar 3.4 Perbandingan antara data dan solusi numerik pada iterasi ke-1



Gambar 3.5 Perbandingan antara data dan solusi numerik pada iterasi ke-50



Gambar 3.6 Perbandingan antara data dan solusi numerik pada iterasi ke-500



Gambar 3.7 Grafik konvergensi algoritma HS

### 3.5.4 Uji Perubahan Parameter Algoritma

Uji perubahan parameter algoritma dilakukan untuk mengetahui efek perubahan parameter algoritma terhadap solusi yang dihasilkan dan performa algoritma HS pada estimasi parameter. Uji perubahan parameter algoritma dilakukan dengan cara melakukan kombinasi parameter  $HMCR$  dan  $PAR$ . Pada setiap kombinasi digunakan  $HMCR$  sebesar 0.7, 0.8, dan 0.9,  $PAR$  sebesar 0.1, 0.3, dan 0.5,  $HMS$  sebesar 10, 50 dan 100,  $bw$  sebesar 0.001, dan iterasi sebanyak 500, 1000, 3000, 5000. Berdasarkan hasil simulasi, diperoleh nilai jumlah kuadrat  $error$  yang disajikan seperti pada Tabel 3.4 untuk  $HMCR = 0.7$ , Tabel 3.6 untuk  $HMCR = 0.8$  dan pada Tabel 3.7 untuk  $HMCR = 0.9$ .

Tabel 3.4 *Output* uji perubahan parameter algoritma bagian I

HMS	HMCR	PAR	Error				Waktu Komputasi (detik)			
			Iterasi				Iterasi			
			500	1000	3000	5000	500	1000	3000	5000
10	0.7	0.1	0.009910	0.007918	0.002380	0.002110	115.578040	279.655367	683.599470	1124.570048
10		0.3	0.010890	0.001339	0.000989	0.000583	118.487086	234.792809	675.170391	1106.348995
10		0.5	1.052271	0.221978	0.046187	0.034396	141.876579	236.067759	690.971013	1152.930343
50		0.1	0.044772	0.039674	0.001787	0.001787	130.931074	245.466342	712.917453	1234.608719
50		0.3	0.183859	0.015205	0.003812	0.003812	132.484724	260.396278	728.640280	1270.933266
50		0.5	0.516973	0.109893	0.038934	0.031529	142.095996	267.115046	786.468003	1287.548804
100		0.1	0.392674	0.118899	0.048081	0.031926	153.565819	280.348658	780.090075	1284.019148
100		0.3	1.075947	0.047815	0.021804	0.009304	156.148028	279.306329	692.576527	1131.585700
100		0.5	0.205754	0.086972	0.011924	0.011924	144.169236	264.779205	1020.415339	1656.620408

Pada iterasi ke-5000 untuk Tabel 3.4, dengan ukuran  $HMS$  sebesar 10,  $HMCR$  sebesar 0.7 dan  $PAR$  sebesar 0.3 merupakan nilai yang paling optimal dengan nilai jumlah kuadrat  $error$  sebesar 0.000583 dan waktu komputasi selama 1106.348995 detik. Sedangkan pada Tabel 3.6, nilai paling optimal diperoleh ketika ukuran  $HMS$  sebesar 50,  $HMCR$  sebesar 0.8 dan  $PAR$  sebesar 0.5 dengan nilai jumlah kuadrat  $error$  sebesar 0.001481 dan waktu komputasi selama 1215.346918 detik. Jika ukuran  $HMCR$  sebesar 0.9 seperti yang tertera pada Tabel 3.7, nilai jumlah kuadrat  $error$  terkecil sebesar 0.000015 dan waktu komputasi selama 1175.461494 detik dengan  $HMS$  sebesar 10 dan  $PAR$  sebesar 0.3. Nilai parameter yang diperoleh ditampilkan pada Tabel 3.5

Tabel 3.5 Parameter hasil estimasi iterasi ke-5000

	$S_m$	$\mu_m$	$k_{mp}$	$k_{pg}$	$k_{pm}$
Data Bangkitan	0.9	0.8	0.001	0.022	0.9
Data Estimasi	0.9037	0.8030	0.0141	0.0543	0.9310

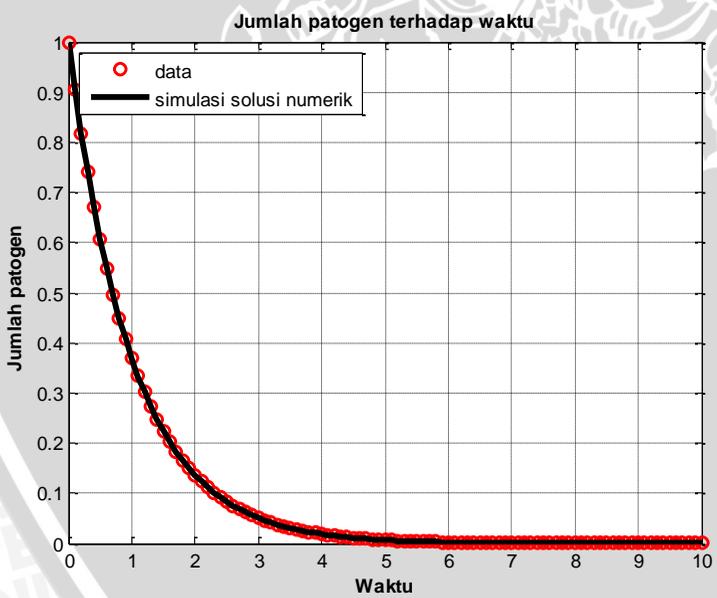
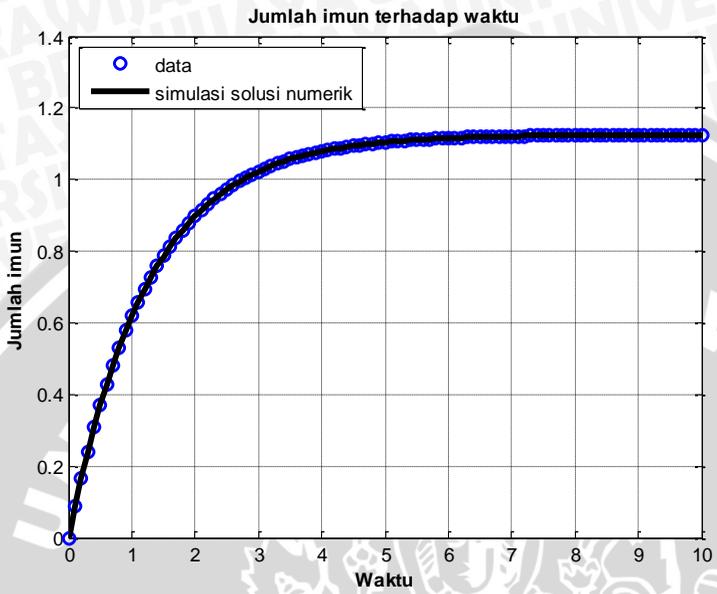
Berdasarkan nilai parameter hasil estimasi dengan jumlah kuadrat  $error$  terkecil antara Tabel 3.4, Tabel 3.6 dan Tabel 3.7, pencocokan penyelesaian sistem persamaan diferensial (3.9) terhadap data tampak seperti pada Gambar 3.8. Terlihat data dan solusi numerik sudah saling berhimpit. Sedangkan hasil pencocokan kurva solusi antara parameter data bangkitan dan parameter hasil estimasi dengan nilai awal sebesar  $M = 0.2, P = 0.8$  tertera pada Tabel 3.5 dan kurvanya ditampilkan pada Gambar 3.9. Tampak bahwa hasil estimasi parameter cukup baik untuk mendekati solusi dari data bangkitan.

Tabel 3.6 *Output* uji perubahan parameter algoritma bagian II

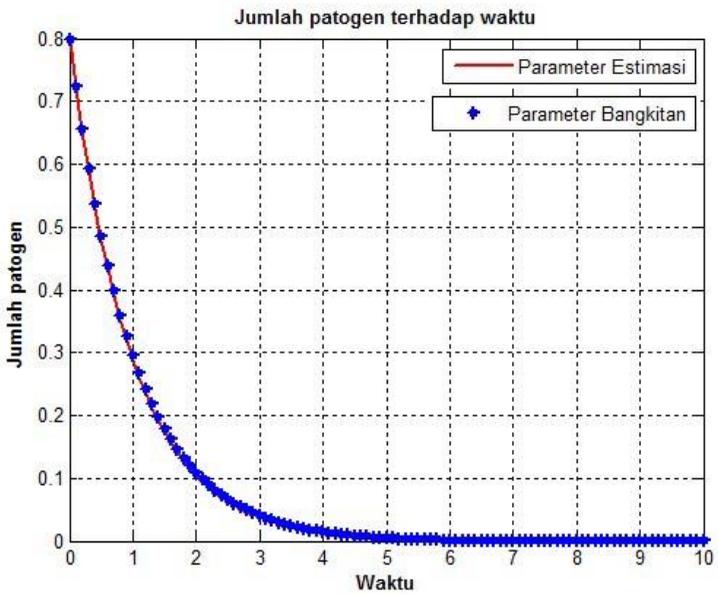
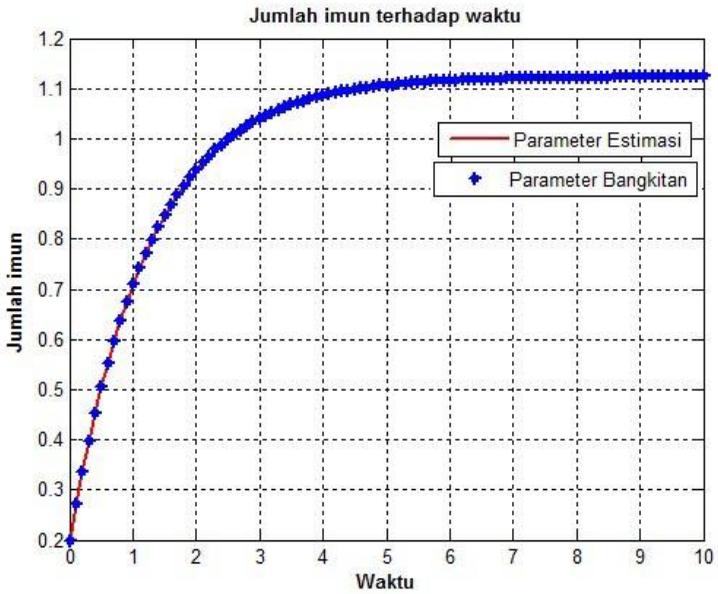
HMS	HMCR	PAR	Error				Waktu Komputasi (detik)			
			Iterasi				Iterasi			
			500	1000	3000	5000	500	1000	3000	5000
10	0.8	0.1	0.018790	0.007127	0.003474	0.002743	193.507164	319.910114	956.006014	1344.415531
10		0.3	0.026521	0.024243	0.014228	0.007341	115.620794	228.183100	691.252065	1147.842794
10		0.5	0.983126	0.080708	0.045709	0.004906	145.871860	258.321717	789.482666	1237.630224
50		0.1	0.267188	0.031987	0.031987	0.029528	142.623580	267.635180	750.443548	1226.823885
50		0.3	0.102651	0.012266	0.002483	0.001803	131.348390	250.309856	719.595914	1194.932049
50		0.5	0.560804	0.206182	0.004172	0.001481	139.878512	262.711758	765.461991	1215.346918
100		0.1	0.869999	0.229731	0.037447	0.037447	141.856839	263.382522	1036.064860	1788.894399
100		0.3	0.023403	0.023403	0.006033	0.006033	208.724038	401.326159	1129.726782	1706.622744
100		0.5	0.338605	0.059466	0.033347	0.013922	171.059462	322.772942	1015.514705	1323.926394

Tabel 3.7 *Output* uji perubahan parameter algoritma bagian III

HMS	HMCR	PAR	Error				Waktu Komputasi (detik)			
			Iterasi				Iterasi			
			500	1000	3000	5000	500	1000	3000	5000
10	0.9	0.1	0.005304	0.003230	0.001659	0.000839	115.770906	238.719068	724.074342	1183.273118
10		0.3	0.474875	0.430672	0.309568	0.000015	128.274690	246.628701	693.800253	1175.461494
10		0.5	0.054001	0.025542	0.006336	0.000732	122.022155	243.129874	701.675134	1165.184679
50		0.1	0.404335	0.069774	0.060816	0.002752	127.540612	241.164285	701.227656	1173.003497
50		0.3	0.241396	0.173589	0.143264	0.124579	143.951961	264.092066	747.377299	1262.862269
50		0.5	0.279837	0.194321	0.139620	0.114601	136.140933	259.518919	728.957854	1218.399616
100		0.1	0.089381	0.089381	0.057747	0.026615	144.951250	267.133737	732.139132	1191.100914
100		0.3	0.135832	0.135832	0.120654	0.113243	146.417064	266.641359	764.035212	1257.304504
100		0.5	0.103136	0.034998	0.000628	0.000628	143.484921	263.247173	765.704866	1265.937443



Gambar 3.8 Perbandingan antara data dan solusi numerik pada iterasi ke-5000



Gambar 3.9 Perbandingan kurva solusi antara parameter data bangkitan dan parameter hasil estimasi

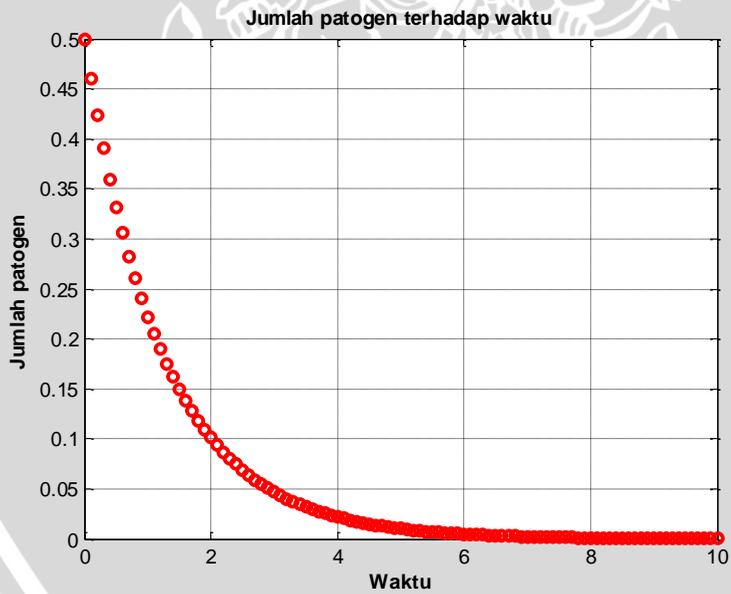
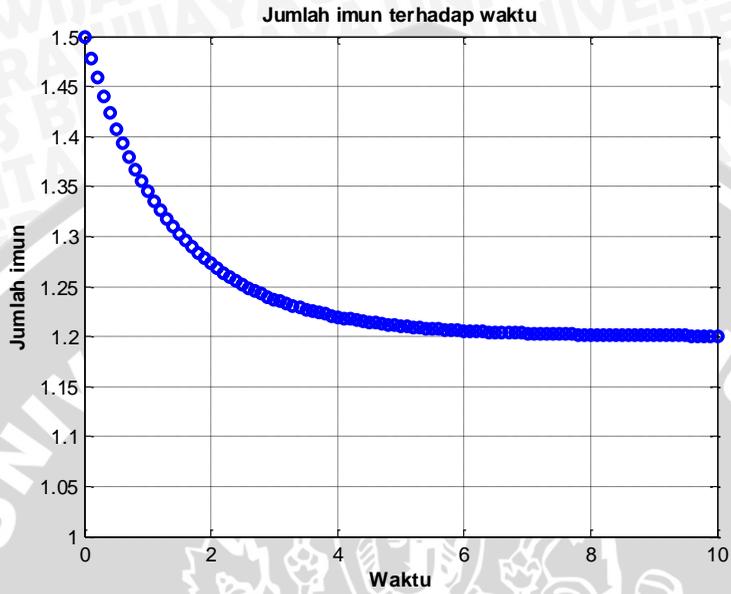
### 3.6 Estimasi Parameter Model Respon Imun dan Patogen (Kasus dua)

Pada kasus dua, data dibangkitkan menggunakan nilai parameter  $s_m = 0.6$ ,  $\mu_m = 0.5$ ,  $k_{mp} = 0.1$ ,  $k_{pg} = 0.3$ ,  $k_{pm} = 0.8$ ,  $P_\infty = 2$ ,  $M = 1.5$ ,  $P = 0.5$ . Kurva solusi populasi respon imun dan populasi patogen diperoleh dengan menggunakan program Runge-Kutta orde empat pada software Matlab seperti pada Gambar 3.10. Pada subbab ini, untuk mengestimasi parameter model digunakan parameter algoritma HS terbaik berdasarkan percobaan sebelumnya, yaitu *HMCR* sebesar 0.9, *PAR* sebesar 0.3, *HMS* sebesar  $bw$  sebesar 0.001, dan iterasi sebanyak 5000. Kurva solusi sudah mulai konvergen setelah mencapai iterasi ke-5000 seperti yang tertera pada Gambar 3.11, dengan nilai minimum (jumlah kuadrat *error*) sebesar 0.000151 dan waktu komputasi selama 1517.288559 detik. Nilai parameter disajikan pada Tabel 3.8.

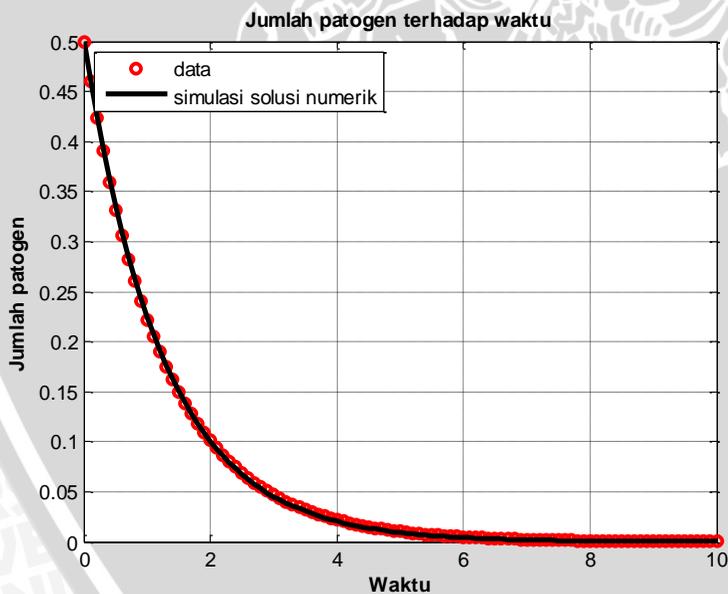
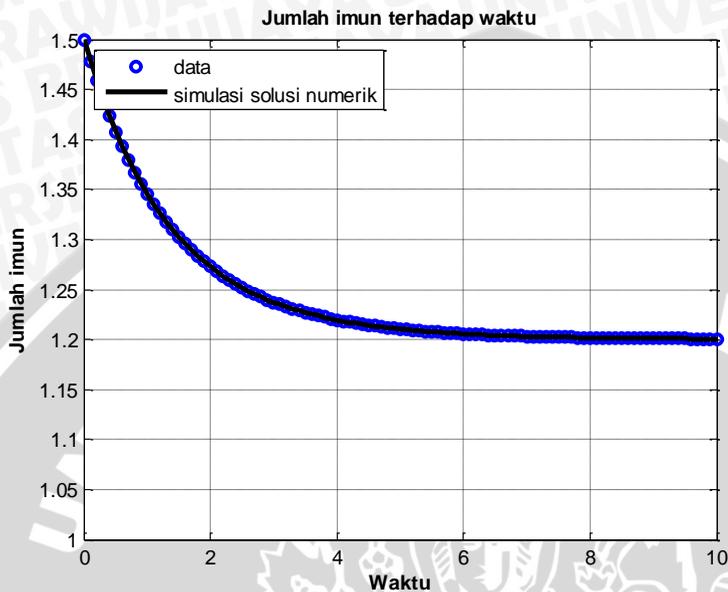
Tabel 3.8 Parameter hasil estimasi iterasi ke-5000

	$s_m$	$\mu_m$	$k_{mp}$	$k_{pg}$	$k_{pm}$
Data Bangkitan	0.6	0.5	0.1	0.3	0.8
Data Estimasi	0.6486	0.5405	0.0799	0.3344	0.9440

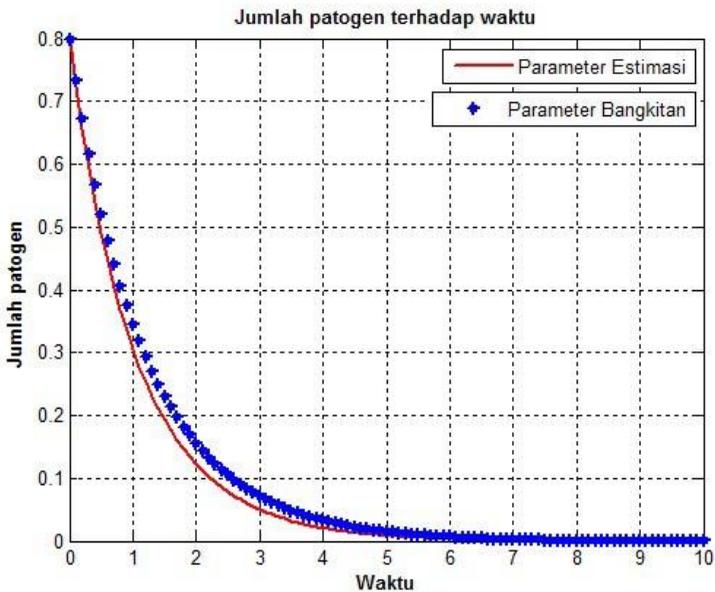
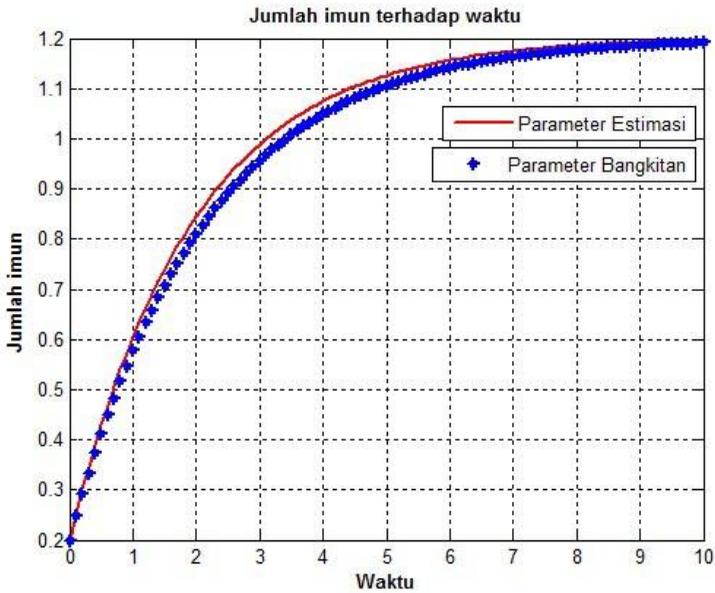
Berdasar Tabel 3.8, kurva solusi antara parameter data bangkitan dan parameter hasil estimasi dengan nilai awal  $M = 0.2$ ,  $P = 0.8$  dicocokkan dan Gambar 3.12 menampilkan kurva hasilnya. Terlihat bahwa parameter hasil estimasi dapat digunakan untuk mewakili sifat model meskipun pada Gambar 3.12 masih terdapat selisih solusi persamaan diferensial yang cukup besar.



Gambar 3.10 Solusi numerik  $M$  dan  $P$  terhadap waktu  $t$



Gambar 3.11 Perbandingan antara data dan solusi numerik pada iterasi ke-5000



Gambar 3.12 Perbandingan kurva solusi antara parameter data bangkitan dan parameter hasil estimasi

UNIVERSITAS BRAWIJAYA



## BAB IV PENUTUP

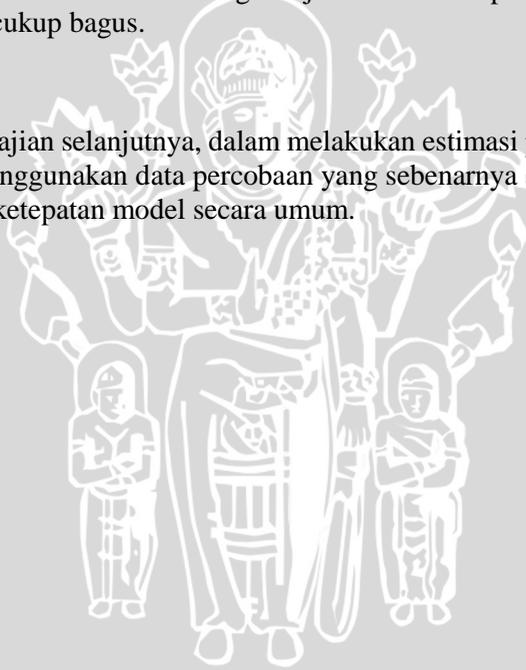
### 4.1 Kesimpulan

Berdasarkan tujuan pengerjaan skripsi ini yang diuraikan pada Bab I dan hasil pembahasan pada Bab III, maka dapat diambil kesimpulan sebagai berikut.

1. Model interaksi respon imun dan patogen berupa sistem otonomus nonlinear dengan dua variabel dan lima parameter.
2. Pada skripsi ini telah diperlihatkan kinerja dari algoritma *harmony search* untuk mengestimasi nilai parameter dengan menggunakan data bangkitan berupa solusi model respon imun dan patogen dengan parameter tertentu sebagai rujukan dan mampu mencapai hasil yang cukup bagus.

### 4.2 Saran

Pada pengkajian selanjutnya, dalam melakukan estimasi parameter diharapkan menggunakan data percobaan yang sebenarnya agar lebih menunjukkan ketepatan model secara umum.



UNIVERSITAS BRAWIJAYA



## DAFTAR PUSTAKA

- Bector C. R., S. Chandra, dan J. Dutta. 2004. *Principles of Optimization Theory*. Narosa Publishers: New Delhi.
- Boyce, W. E. dan R. C. Diprima. 2000. *Elementary Differential Equation and Boundary Value Problems*. Seventh edition. John Wiley and Sons: New York.
- Chapra, S. C. dan R. P. Canale. 2010. *Numerical Methods for Engineers*. The McGraw-Hill Companies, Inc: New York.
- Everitt, B. S. dan D. C. Howell. 2005. *Least Squares Estimation*. John Wiley and Sons: Chicester.
- Geem, Z. W. 2010. *Recent Advances in Harmony Search Algorithm*. Springer-Verlag Berlin Heidelberg: USA.
- Kartikowati. 2009. *Estimasi Parameter Model HIV Menggunakan Algoritma Genetika*. Universitas Brawijaya: Malang.
- Krumke, S. O. 2004. *Nonlinear Optimization*. Technical University of Kaiser-Slautern: Germany.
- Reynolds, A., J. Rubin, G. Clermont, dkk. 2006. A Reduced Mathematical Model of the Acute Inflammatory Response: I. Derivation of Model and Analysis of Anti Inflammation. *Journal of Theoretical Biology* 242 220-236.
- Ross, C. C. 2004. *Differential Equation an Introduction with Mathematics*. Second edition. Springer: USA.
- Setiawan, A. dan B. Santosa. 2010. *Penerapan Algoritma Harmony Search dalam Penyelesaian Resource-Constrained Project Scheduling Problem*. ITS Library: Surabaya.
- Timmreck, C. T. 2004. *Epidemiolgi*. Buku Kedokteran EGC: Jakarta.

UNIVERSITAS BRAWIJAYA



## LAMPIRAN

### Lampiran 1. Program RK4

```
%Mencari solusi numerik menggunakan nilai awal
%File name: "RK4"
clear all; clc;
f=inline('0.9-0.8*x-0.001*x*y','t','x','y');
g=inline('(0.022*y)-(0.022*y^2/(2))-
(0.9*0.9*y/0.8+0.001*y)','t','x','y');
jumlahimun=0;
jumlahpatogen=1;
h=0.1;
t=0:h:10;
n=length(t)-1;
x=zeros(1,n);x(1)=jumlahimun;
y=zeros(1,n);y(1)=jumlahpatogen;

for i = 1:n
    k1=h*f(t(i),x(i),y(i));
    l1=h*g(t(i),x(i),y(i));
    k2=h*f(t(i)+(h/2),x(i)+(k1/2),y(i)+(l1/2));
    l2=h*g(t(i)+(h/2),x(i)+(k1/2),y(i)+(l1/2));
    k3=h*f(t(i)+(h/2),x(i)+(k2/2),y(i)+(l2/2));
    l3=h*g(t(i)+(h/2),x(i)+(k2/2),y(i)+(l2/2));
    k4=h*f(t(i)+h,x(i)+k3,y(i)+l3);
    l4=h*g(t(i)+h,x(i)+k3,y(i)+l3);
    x(i+1)=x(i)+(1/6)*(k1+2*k2+2*k3+k4);
    y(i+1)=y(i)+(1/6)*(l1+2*l2+2*l3+l4);
end
figure(1)
plot(t',x,'o','color','b','LineWidth',2)
title('Jumlah imun terhadap
waktu','FontWeight','bold')
xlabel('Waktu','FontWeight','bold')
ylabel('Jumlah imun','FontWeight','bold')
grid on
hold on
figure(2)
plot(t',y,'o','Color','r','LineWidth',2)
title('Jumlah patogen terhadap
waktu','FontWeight','bold')
xlabel('Waktu','FontWeight','bold')
```

```
ylabel('Jumlah patogen','FontWeight','bold')
grid on
hold on
ynum=[x' y'];clear x y;
```

UNIVERSITAS BRAWIJAYA



## Lampiran 2. Program Algoritma *Harmony Search* untuk Estimasi Parameter

```
%Mencari error
%Nama file: "error"
function [galat]=error1(ynum, rk)
galat=0;
n=length(ynum);
for i=1:2
    for j=1:n
        galat=galat+((ynum(j,i))-rk(j,i))^2;
    end
end

%Fungsi model respon imun dan patogen
%Nama file: "fungsi"
function [f1,f2] = fungsi(hm)
% urutannya adalah sm, miun, kmp, kpg, kpm
f1=[num2str(hm(1)) '-' num2str(hm(2)) '*x1-'
num2str(hm(3)) '*x1*x2'];
f2=[num2str(hm(4)) '*x2-' num2str(hm(4))
'*x2^2/(2)-' num2str(hm(5)) '*'+...
num2str(hm(1)) '*x2' '/' num2str(hm(2)) '+'
num2str(hm(3)) '*x2)'];
end

%Proses Algoritma Harmony Search
%Nama file: "harmony"
function [xbest,fbest,rkbest] =
harmony(x0, f, xL, xU, HSparams, n, ynum, h)

HMS      = HSparams.HMS;
MaxImp   = HSparams.MaxImp;
HMCR     = HSparams.HMCR;
PAR      = HSparams.PAR;
b        = HSparams.b;

N        = length(xL);
HM       = zeros(HMS, N);
F        = zeros(HMS, 1);
xnew     = zeros(1, N);
```

```

for j = 1:HMS
    HM(j,:) = xL + (xU-xL).*rand(1,N);
    [f1,f2] = fungsi(HM(j,:));
    rk=rk14(x0,xL(1),xU(1),n,char(f1),char(f2),h);
    F(j) = f(ynum,rk);
end
HM
F

eps=0.00001;
gal=1;
iter=1;
while gal>eps & iter<=MaxImp
    for i = 1:N
        if rand < HMCR
            xnew(i) = HM(ceil(1+(HMS-1)*rand),i);
            if rand < PAR
                xnew(i) = xnew(i) + (2*rand-
1)*b(i);
                xnew(i) =
min(max(xnew(i),xL(i)),xU(i));
            end
            else
                xnew(i) = xL(i) +rand*(xU(i)-xL(i));
            end
        end
        [f1,f2] = fungsi(xnew);
        rk=rk14(x0,xL(1),xU(1),n,char(f1),char(f2));
        fnew = f(ynum,rk);
        [fworst,idxworst] = max(F);
        if fnew < fworst
            HM(idxworst,:) = xnew;
            F(idxworst) = fnew;
        end
        gal=min(F);
        fbest(iter)=min(F);
        iter=iter+1;
    end
    disp(iter);
    [fbest(iter),idxbest] = min(F);
    xbest = HM(idxbest,:);

```

```
[f1,f2] = fungsi(xbest);
rkbest=rk14(x0,xL(1),xU(1),n,char(f1),char(f2));
HM
F
```

```
%Mencari solusi numerik untuk hasil Algoritma HS
%Nama file: "rk14"
function rk=rk14(x0,a,b,n,f,g,h)
f1=inline(f,'x1','x2');
g1=inline(g,'x1','x2');
del=(b-a)/n;
h=0.1;
rk(1,:)=x0;
for i=1:n
    k1=h*f1(rk(i,1),rk(i,2));
    l1=h*g1(rk(i,1),rk(i,2));
    k2=h*f1(rk(i,1)+k1/2,rk(i,2)+l1/2);
    l2=h*g1(rk(i,1)+k1/2,rk(i,2)+l1/2);
    k3=h*f1(rk(i,1)+k2/2,rk(i,2)+l2/2);
    l3=h*g1(rk(i,1)+k2/2,rk(i,2)+l2/2);
    k4=h*f1(rk(i,1)+k3,rk(i,2)+l3);
    l4=h*g1(rk(i,1)+k3,rk(i,2)+l3);
    rk(i+1,1)=rk(i,1)+(k1+2*k2+2*k3+k4)/6;
    rk(i+1,2)=rk(i,2)+(l1+2*l2+2*l3+l4)/6;
end
```

```
%Program utama
clc;
tic;
RK4;
rand('twister',5489);
f = @error;
xL = [0 0 0 0 0];
xU = [1 1 1 1 1];
x0 =[0 1];
h=0.1;
n=length(t)-1;
% Pemilihan parameter harmony
HSparams.HMS = 10;
HSparams.MaxImp =500;
HSparams.HMCR = 0.8;
HSparams.PAR = 0.2;
```

```
HSparams.b = 0.001;
```

```
[xbest,fbest,rkbest] =  
harmony(x0,f,xL,xU,HSparams,n,ynum,h);  
fprintf('galat=%f\n',fbest(end));  
[f1,f2]=fungsi(xbest);  
rkbest=rk14(x0,xL(1),xU(1),n,f1,f2);  
figure(1)  
plot(t',rkbest(:,1),'-'  
, 'Color','black','LineWidth',3)  
legend('data','simulasi solusi numerik',2)  
figure(2)  
plot(t',rkbest(:,2),'-'  
, 'Color','black','LineWidth',3)  
legend('data','simulasi solusi numerik',2)  
figure(3)  
plot(fbest,'LineWidth',3)  
xlabel('Iterasi','FontWeight','bold')  
ylabel('Error','FontWeight','bold','FontAngle',  
'Italic')  
toc;
```