

Tingkat Vaksinasi Optimal dalam Model Epidemik Deterministik dengan Metode Algoritma Genetik

M. Iqbal, A. K. Supriatna, N. Anggriani

Departemen Matematika FMIPA Universitas Padjadjaran

Jl. Raya Bandung-Sumedang km 21 Jatinangor

Email: ak_supriatna@unpad.ac.id, n_anggriani@unpad.ac.id

ABSTRAK

Epidemik merupakan proses yang menjangkiti suatu populasi pada periode tertentu. Epidemik akan mulai menyebar apabila telah melebihi ambang batas tertentu. Agar penyebaran suatu penyakit tidak sampai melebihi ambang batas tertentu, harus dilakukan intervensi, salah satunya adalah dengan vaksinasi. Pencarian vaksinasi optimal dalam model epidemik deterministik telah dilakukan oleh beberapa peneliti. Dalam makalah ini akan dicari tingkat vaksinasi optimal dengan biaya yang minimum menggunakan metode algoritma genetik. Dalam perluasannya, dilakukan penambahan faktor demografi dan *host-vector* pada model epidemik tersebut. Pencarian tingkat vaksinasi dengan algoritma genetik memiliki kemampuan untuk memperoleh hasil dengan waktu yang lebih cepat dan *time horizon* yang lebih panjang dibandingkan dengan metode lain pada penelitian sebelumnya. Namun pada metode algoritma genetik memiliki kelemahan yaitu diperlukannya rangkaian percobaan dalam memperoleh tingkat vaksinasi optimal.

Kata Kunci: model epidemik, tingkat vaksinasi optimal, algoritma genetika

ABSTRACT

Epidemic is a process of the spreading of infectious disease which infect a population in a particular period. Epidemic will start to spread if it exceeds particular threshold. In order to stop the spreading of the disease to reach its maximum spread, an intervention is needed, e.g. by imposing vaccination. The study to find the optimal vaccination level from an epidemic model has been done by many authors. In this paper, we will find the level of optimal vaccination from an epidemic model with minimum cost by using genetic algorithm method. In the further development, we add demographic factors and host-vector to the epidemic model. The searching of the vaccination level with genetic algorithm has ability to have a faster time and longer time horizon than other method in previous researches. However it has a weakness such as the need of experiments in gaining the optimal vaccination level.

Keywords: epidemic model, optimal vaccination level, genetic algorithm method

1. Latar Belakang

Epidemik merupakan proses penyebaran wabah yang menjangkiti suatu populasi pada periode tertentu. Epidemik tersebut akan mulai menyebar apabila telah melebihi ambang batas tertentu. Dalam bidang matematika, kondisi seperti ini dapat dipelajari dalam sebuah model matematika. Model matematika banyak digunakan untuk menggambarkan penyebaran suatu epidemik (Zuleni, [9]). Dengan kondisi awal yang diberikan, sebuah model matematika dapat memprediksi penyebaran wabah, baik secara temporal maupun spasial. Salah satu bentuk tindakan pencegahan agar penyebarannya tidak semakin meluas adalah dengan memberikan vaksinasi. Akan tetapi, program vaksinasi sebagai bentuk pencegahan memerlukan biaya (*cost*) yang tidak sedikit. Oleh karena itu, harus diupayakan adanya pemberian tingkat vaksinasi yang optimal sehingga biaya yang harus dikeluarkan akan menjadi minimum.

Studi tentang model epidemik deterministik untuk pencarian vaksinasi optimal telah dilakukan oleh Hethcote dan Watman [4]. Penelitian tersebut menghasilkan perbandingan nilai-nilai parameter yang digunakan dalam model epidemik untuk *Time Horizon* yang pendek

($T=1.5$). Kemudian Widyaningsih [8] mengembangkan model Hethcote tersebut dengan menambahkan vektor penyebar (model *host-vector*) secara numerik dengan menggunakan efisiensi minimum sort. Dari penelitian tersebut terdapat beberapa kelemahan, yaitu waktu proses yang cukup lama dan *Time Horizon* yang terbatas (maksimum $T=0.8$ untuk waktu komputasi yang cukup panjang). Padahal dalam aplikasi di lapangan biasanya mempunyai *Time Horizon* yang panjang. Dari latar belakang tersebut maka penulis mencoba mencari tingkat vaksinasi optimal pada model epidemik deterministik *host-vector* tersebut dengan menggunakan metode algoritma genetik agar menghasilkan biaya minimum dalam waktu singkat dan *Time Horizon* yang lebih panjang.

Berdasarkan penguraian latar belakang diatas, maka dalam paper ini ingin diketahui beberapa hal diantaranya:

1. Bagaimana merumuskan model penyebaran suatu epidemik dengan vaksinasi dalam metode algoritma genetik?
2. Bagaimana mencari tingkat vaksinasi optimal untuk suatu epidemik dengan metoda algoritma genetik agar didapat biaya minimum yang mendekati solusi eksaknya?

Untuk menjawab pertanyaan tersebut di atas akan dikembangkan sebuah model matematika, khususnya untuk menentukan tingkat vaksinasi optimal pada model penyebaran SIR dan perluasannya untuk penyakit yang ditularkan melalui vektor, di mana sebelumnya akan dibahas beberapa teori dan metode terkait yang diperlukan dalam analisis model.

2. Landasan Teori dan Metode

Salah satu model epidemik yang sering digunakan adalah model yang membagi populasi menjadi tiga kelas berbeda, yaitu populasi rentan (S), populasi terinfeksi (I), dan populasi yang dipindahkan karena pulih, kebal, diisolasi hingga pulih atau telah meninggal (R). Secara skematis, proses tersebut digambarkan sebagai $S \rightarrow I \rightarrow R$. Model ini seringkali dikenal sebagai model *SIR*. Jumlah kelas yang ingin dibedakan dapat dimodelkan bergantung pada karakteristik penularannya. Metode yang dipakai dalam analisis model adalah penyelesaian numerik dari sistem persamaan diferensial nonlinear dengan menggunakan metode Runge-Kutta orde empat yang diperluas (Djojodihardjo, [1]; Munir, [5]). Kemudian untuk memperoleh nilai optimal dari tingkat vaksinasi yang harus diberikan adalah dengan menggunakan algoritma genetika (Gen, [2]; Suyanto, [6]).

2.1 Metode Runge-Kutta Orde Empat

Metode Runge-Kutta yang terkenal dan banyak digunakan dalam praktek adalah metode Runge-Kutta orde empat. Orde empat berarti bahwa dalam metode ini fungsinya harus dihitung sebanyak empat kali. Misal diberikan persamaan $\frac{dy}{dt} = f(t, y)$, maka untuk menyelesaikan persamaan tersebut dapat menggunakan metode Runge-Kutta orde-4 dengan bentuk:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (1)$$

dengan

$$\begin{aligned}
 k_1 &= h \times f(t_n, y_n) \\
 k_2 &= h \times f\left(t_n + \frac{1}{2}, y_n + \frac{1}{2}k_1\right) \\
 k_3 &= h \times f\left(t_n + \frac{1}{2}, y_n + \frac{1}{2}k_2\right) \\
 k_4 &= h \times f(t_n + h, y_n + k_3)
 \end{aligned}$$

Metode Runge-Kutta orde empat yang telah diuraikan sebelumnya merupakan suatu metode untuk mencari solusi dari satu fungsi saja. Oleh karena itu, untuk menyelesaikan sistem persamaan diferensial biasa seperti:

$$\frac{dy^1}{dt} = f^1(t, y^1, y^2, \dots, y^z), \frac{dy^2}{dt} = f^2(t, y^1, y^2, \dots, y^z), \dots, \frac{dy^z}{dt} = f^z(t, y^1, y^2, \dots, y^z)$$

metode tersebut masih harus diperluas.

Perluasan metode Runge-Kutta untuk menyelesaikan sistem persamaan diferensial biasa di atas adalah dengan menggunakan rumus iteratif sebagai berikut:

$$\begin{aligned}
 y_{(n+1)}^1 &= y_{(n)}^1 + \frac{1}{6}(k_1^1 + 2k_2^1 + 2k_3^1 + k_4^1) \\
 y_{(n+1)}^2 &= y_{(n)}^2 + \frac{1}{6}(k_1^2 + 2k_2^2 + 2k_3^2 + k_4^2) \\
 &\vdots \\
 y_{(n+1)}^z &= y_{(n)}^z + \frac{1}{6}(k_1^z + 2k_2^z + 2k_3^z + k_4^z)
 \end{aligned}$$

dimana

$$\begin{aligned}
 k_1^i &= h \times f^i(t_n, y_n^1, y_n^2, \dots, y_n^z) \\
 k_2^i &= h \times f^i\left(t_n + \frac{h}{2}, y_n^1 + \frac{1}{2}k_1^1, y_n^2 + \frac{1}{2}k_1^2, \dots, y_n^z + \frac{1}{2}k_1^z\right) \\
 k_3^i &= h \times f^i\left(t_n + \frac{h}{2}, y_n^1 + \frac{1}{2}k_2^1, y_n^2 + \frac{1}{2}k_2^2, \dots, y_n^z + \frac{1}{2}k_2^z\right) \\
 k_4^i &= h \times f^i(t_n + h, y_n^1 + k_3^1, y_n^2 + k_3^2, \dots, y_n^z + k_3^z)
 \end{aligned}$$

Dengan $i \in \{1, 2, \dots, z\}$.

2.2 Algoritma Genetik

Menurut Goldberg [3], algoritma genetik dapat didefinisikan sebagai suatu metode pencarian yang didasari oleh mekanisme dari genetika ilmiah dan seleksi alam. Struktur umum dari suatu algoritma genetika dapat didefinisikan dengan langkah-langkah sebagai berikut:

1. Membangkitkan populasi awal, populasi awal ini dibangkitkan secara random sehingga didapatkan solusi awal.
2. Populasi itu sendiri terdiri dari sejumlah kromosom yang mempresentasikan solusi yang diinginkan.

3. Membentuk generasi baru, dalam membentuk generasi baru digunakan tiga operator yaitu reproduksi/seleksi, crossover dan mutasi.
4. Evaluasi solusi, proses ini akan mengevaluasi setiap populasi dengan menghitung nilai fitness setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Apabila kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah 2.

Dalam algoritma genetik, terdapat tiga operator yaitu reproduksi, *crossover*, dan mutasi.

1. Reproduksi: Proses pembentukan kromosom baru berdasarkan kromosom yang ada atau pada reproduksi sebuah individu lama pada suatu populasi menjelma sebagai dirinya sendiri pada populasi baru.
2. *Crossover*: Dalam proses *crossover*, pertama-tama yang harus dilakukan adalah penentuan secara random, kromosom mana yang akan menjadi parent. Terdapat beberapa cara dalam *crossover*. Yang paling sederhana adalah *one-point crossover*. Suatu titik potong dipilih secara random, kemudian bagian pertama dari parent 1 digabungkan dengan bagian kedua dari parent 2.
3. Mutasi: Mutasi adalah proses untuk menghasilkan keturunan yang jauh berbeda dari mayoritas individu dalam populasi. Proses mutasi biasanya dibutuhkan untuk menghindari algoritma genetik dari kondisi statis (*stuck*).

3. Penerapan Metode Runge-Kutta Dan Algoritma Genetik Pada Model SIR

Model penyebaran suatu epidemik dengan penambahan faktor vaksinasi dapat dirumuskan sebagai berikut:

$$S'(t) = -\beta IS - \alpha \quad (2)$$

$$I'(t) = \beta IS - \gamma I \quad (3)$$

$$R'(t) = \gamma I \quad (4)$$

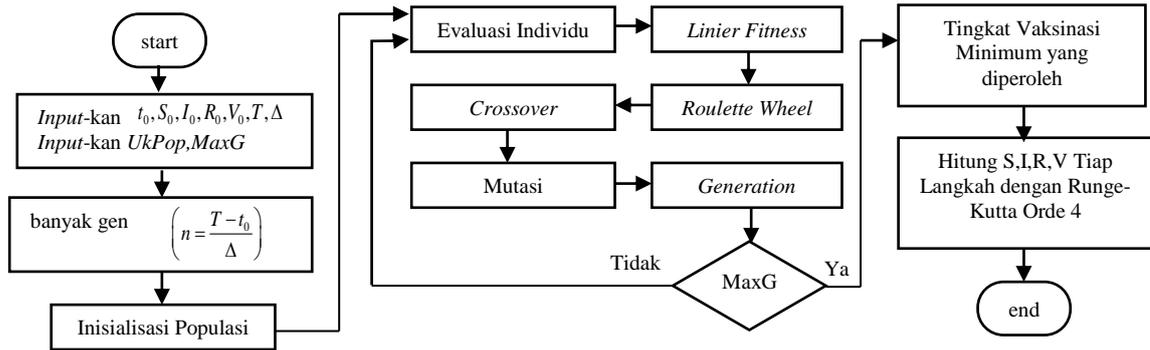
$$V'(t) = \alpha \quad (5)$$

Dengan mengikuti Hethcote dan Watman [4], masalah yang dihadapi dalam pencegahan epidemik ini dapat dirumuskan sebagai berikut: diberikan fungsi biaya $C(\alpha)$, A, B, T adalah suatu konstanta, nilai yang konstan untuk S_0, I_0, A, B, T dan jarak titik tiap kelas dari fungsi diskrit $\alpha(t)$.

Pilih fungsi $\alpha(t)$ seperti yang ditulis pada persamaan (2) - (5) dengan kondisi awal S_0, I_0 sehingga menghasilkan solusi $C(\alpha) = \text{minimum}$ dengan ketentuan:

- i. $R(T) + I(T) \leq A$
- ii. $\text{maks}_{[0,T]} I(t) \leq B$
- iii. $C(\alpha) = \sum_{t=0,1}^T C(\alpha(t)) = \sum_{t=0}^{n-1} (M_i d\Delta)^2$

Untuk menemukan tingkat vaksinasi dengan biaya yang minimum, maka harus dicari solusi yang memenuhi kendala-kendala tersebut (lihat Gambar 1).

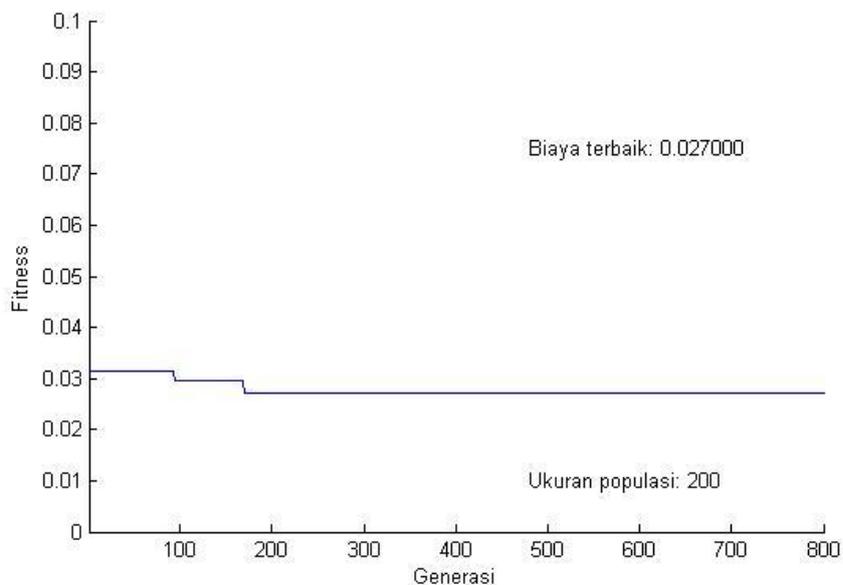


Gambar 1. Langkah-langkah yang dilakukan dalam pencarian tingkat vaksinasi pada model algoritma genetik

Berikut ini disajikan pencarian tingkat vaksinasi yang optimal dengan menggunakan Matlab 7.6, kemudian dilanjutkan dengan analisis solusi dan perbandingan solusi terhadap metode lain seperti pada Widyaningsih [8].

3.1 Tingkat Vaksinasi Optimal

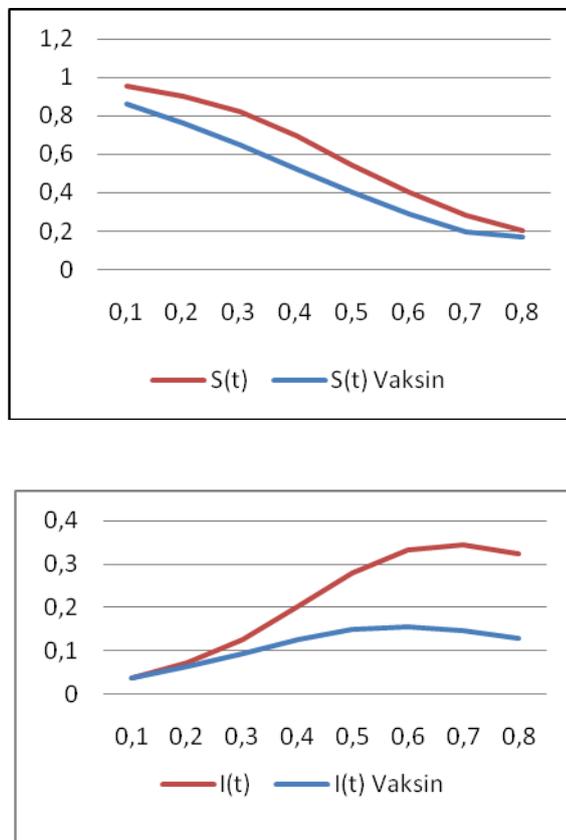
Dalam contoh ini diberikan kondisi awal $(S_0, I_0, R_0, V_0) = (0.98, 0.02, 0, 0)$ dengan waktu awal $t = 0$, waktu akhir $T = 0.8$ dan $\Delta = 0.1$ (Hethcote dan Waltman [4]) dan diberikan ukuran populasi sebesar 200 dan maksimum generasi sebesar 800. Dalam komputasi program dengan vaksinasi dalam selang waktu $[0;0.8]$ didapatkan tingkat vaksinasi dengan biaya yang minimum, yaitu sebesar 0.0297 dengan waktu eksekusi program 45.7 detik sebagaimana ditunjukkan oleh snapshot Gambar 2-3 dan Tabel 1.



Gambar 2. Pencarian vaksinasi dengan algoritma genetik

Tabel 1: Hasil komputasi tanpa vaksinasi (kiri) dan dengan vaksinasi (kanan)

Iterasi	t	$\alpha(t)$	Tanpa Vaksinasi				Dengan Vaksinasi			
			$I(t)$	$S(t)$	$R(t)$	$V(t)$	$I(t)$	$S(t)$	$R(t)$	$V(t)$
1	0.1	0.9	0.039	0.9525	0.0085	0	0.0373	0.8643	0.0084	0.09
2	0.2	0.6	0.0731	0.902	0.0249	0	0.0624	0.7644	0.0232	0.15
3	0.3	0.6	0.1283	0.8171	0.0545	0	0.0939	0.6496	0.0465	0.21
4	0.4	0.6	0.2029	0.6932	0.1039	0	0.1252	0.5253	0.0795	0.27
5	0.5	0.6	0.2793	0.5442	0.1765	0	0.1474	0.4019	0.1207	0.33
6	0.6	0.6	0.3312	0.3998	0.269	0	0.1541	0.2896	0.1663	0.39
7	0.7	0.6	0.3444	0.2843	0.3713	0	0.1451	0.1933	0.2116	0.45
8	0.8	0	0.3247	0.203	0.4723	0	0.1288	0.1686	0.2527	0.45



Gambar 3. Time-series populasi sehat (atas) dan terinfeksi (bawah), untuk perhitungan tanpa vaksinasi (merah) dan dengan vaksinasi (biru)

Gambar 3 memperlihatkan jumlah populasi sehat dan terinfeksi sebagai output program pada Gambar 2. Hasil dari komputasi program dengan vaksinasi diperoleh I maksimum sebesar 0.1541, sedangkan komputasi program tanpa vaksinasi diperoleh I maksimum sebesar 0.344. Dapat disimpulkan bahwa kondisi *infected* dapat dikendalikan dengan pemberian vaksinasi sehingga didapat nilai I maksimum yang lebih rendah dibandingkan tanpa pemberian vaksinasi.

3.2 Analisis Solusi

Seperti yang telah kita ketahui bahwa metode algoritma genetika merupakan metode pendekatan. Artinya solusi yang dihasilkan tidak harus optimal, tetapi cukup 'bagus' atau dapat diterima. Oleh karena itu perlu dilakukan beberapa percobaan agar solusi yang optimal diperoleh.

Tabel 2 memperlihatkan output dari beberapa percobaan yang dilakukan. Dari hasil tabel tersebut diperoleh nilai fitness terkecil pada percobaan ke 8 dengan kromosom 3 2 2 2 2 1 1 1 diperoleh biaya minimum sebesar 0.0252, sehingga diperoleh tingkat vaksinasi $\alpha(t) = \{0.9; 0.6; 0.6; 0.6; 0.6; 0.3; 0.3; 0.3\}$.

Tabel 3 memperlihatkan perbedaan kinerja algoritma genetika dengan metode lain (Efisiensi sort pada Widyaningsih [8]). Berdasarkan tabel tersebut, metode efisiensi sort lebih baik dalam mencari tingkat vaksinasi, namun *time horizon*-nya hanya terbatas pada waktu akhir $T = 0.8$. Pada penelitian yang sedang dilakukan, dapat dicari tingkat vaksinasi yang tidak terbatas pada waktu akhir $T = 0.8$ dan dengan waktu eksekusi program yang jauh lebih singkat.

Tabel 2: Perbandingan barisan vaksinasi dari hasil percobaan yang berbeda

Percobaan	Kromosom (Profil Vaksinasi)	Fitness (biaya)
1	3 2 2 2 2 2 2 0	0.0297
2	3 4 2 2 1 2 0 1	0.0351
3	3 2 2 2 2 2 2 2	0.0333
4	4 2 2 2 2 2 0 2	0.0360
5	4 3 1 1 1 1 1 3	0.0351
6	3 3 2 3 1 2 1 2	0.0369
7	6 1 1 1 1 1 1 1	0.0387
8	3 2 2 2 2 1 1 1	0.0252
9	3 2 2 2 3 1 2 1	0.0324
10	3 2 2 2 2 2 0 0	0.0261
11	3 2 3 2 2 1 1 1	0.0297
12	3 2 2 2 2 2 0 2	0.0297
13	3 2 2 2 3 2 0 0	0.0306
14	3 2 3 3 1 1 0 1	0.0306
15	6 1 1 1 1 1 1 1	0.0387

Fitness pada tabel diatas diperoleh dari rumus $C(\alpha) = \sum_{t=0,1}^T C(\alpha(t)) = \sum_{t=0}^{n-1} (M_i d \Delta)^2$ dengan $\Delta = 0.1$; $d = 0.3$ dan M_i adalah elemen ke I dari kromosom.

Tabel 3 Perbandingan kinerja antara algoritma genetika dengan metode lain

Perbandingan	Efisiensi Min Sort	Algoritma Genetika
<i>Software</i>	Borland C++ 5.02	Matlab 7.6
Metode pencarian	Minimum Sort	Algoritma Genetik
Ketelitian program	Analitik	Pendekatan
<i>Time Horizon</i>	[0;0.8]	> 0.8
Eksekusi waktu program (T=0.8)	481 detik	45.7 detik
Percobaan	Tidak	Ya
untuk selang waktu [0;0.8] dengan nilai awal S,I,R,V yang sama, jalur vaksinasi yang diperoleh	3 2 2 2 1 1 0	3 2 2 2 2 1 1 1
Biaya vaksinasi	0.0243	0.0252

4. Penerapan Metode Runge-Kutta dan Algoritma Genetik pada Model SIR yang diperluas (Host-Vector)

Berikut ini adalah sistem persamaan diferensial dari model epidemik SIR yang diperluas dengan penambahan faktor demografi dan penambahan vektor perantara (sistem *host-vector*) yang dikembangkan penulis:

$$S'_H(t) = B - \beta_H I_V S_H - \alpha - \mu_H S_H \tag{6}$$

$$I'_H(t) = \beta_H I_V S_H - I_H (\gamma_H + \mu_H) \tag{7}$$

$$R'_H(t) = \gamma_H I_H - \mu_H R_H \tag{8}$$

$$V'_H(t) = \alpha - \mu_H V_H \tag{9}$$

$$S'_V(t) = B_V - \beta_V I_H S_V - \mu_V S_V \tag{10}$$

$$I'_V(t) = \beta_V I_H S_V - \mu_V I_V \tag{11}$$

dan

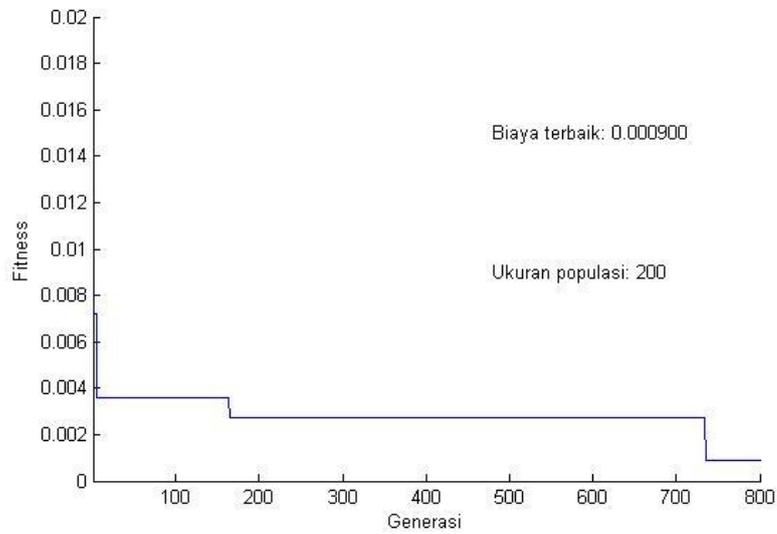
$$S_H(t) + I_H(t) + R_H(t) + V_H(t) = N_H \quad \text{dan} \quad S_V(t) + I_V(t) = N_V$$

dengan B adalah laju kelahiran individu baru, μ_H adalah laju kematian individu, B_V adalah laju kelahiran *vector* baru, μ_V adalah laju kematian *vector*.

4.1 Tingkat Vaksinasi Optimal

Dalam contoh ini diberikan kondisi awal $(S_0, I_0, R_0, V_0) = (0.98, 0.02, 0, 0)$ dengan $t = 0, T = 0.7$, dan $\Delta = 0.1$ (Hethcote dan Waltman, [4]) dan diberikan parameter faktor demografi $\mu_H = 1, B_H = 0.8, \mu_V = 0.5, B_V = 0.5$ serta parameter *host-vector* $S_{V0} = 1, I_{V0} = 0$. Untuk parameter algoritma genetik diberikan ukuran populasi 200 dan maksimum generasi 800. Dari hasil program dengan waktu eksekusi 20 detik untuk kasus waktu pencapaian sama dengan 7 serta $\mu_H = 1, B_H = 0.8, \mu_V = 0.5, B_V = 0.5, \mu_h = 1, B_h = 0.8, \mu_v = 0.5, B_v = 0.5$ dan kondisi awal yang

diberikan (lihat snapshot pada Gambar 4), maka pemberian vaksinasi dengan tingkat $\{0,3;0;0;0;0;0;0\}$ disimpulkan dapat menjaga kondisi populasi *infected* agar tidak meningkat (Tabel 4). Tingkat vaksinasi tersebut merupakan tingkat yang memenuhi kendala $\max_{[0,T]} I(t) \leq B$ dan $R(T) + I(T) \leq A$ sehingga biaya minimum yang akan dikeluarkan adalah sebesar 0.000900. Tabel 5 memperlihatkan perbandingan kinerja algoritma genetika dengan metode lainnya (efisiensi sort pada Widyaningsih [8]). Berdasarkan tabel diatas, kedua metode pencarian tingkat vaksinasi menghasilkan jalur dan biaya vaksinasi yang sama. Artinya kedua metode tersebut tidak memiliki perbedaan pada tingkat vaksinasi. Namun untuk melakukan penelitian pada *time horizon* $T > 0.7$, metode efisiensi minimum sort tidak dapat lagi digunakan disebabkan kinerja program yang hanya mampu mencari tingkat vaksinasi hanya sampai *time horizon* $T = 0.7$.



Gambar 4. Profile vaksinasi yang dihasilkan

Tabel 4: Perbandingan barisan vaksinasi dari hasil percobaan yang berbeda

Percobaan	Kromosom (Profil Vaksinasi)	Fitness (biaya)
1	1 1 2 1 2 2 1	0.0144
2	0 1 1 1 0 0 3	0.0108
3	1 1 0 0 1 1 0	0.0036
4	1 1 1 1 1 1 1	0.0063
5	0 0 0 0 1 1 0	0.0018
6	1 0 1 1 0 0 0	0.0027
7	0 1 0 1 0 0 1	0.0027
8	3 11100 0	0.0108
9	1 1 1 1 1 1 1	0.0063
10	1 0 0 0 0 0 0	0.0009
11	0 1 0 0 0 1 1	0.0027
12	0 0 1 0 0 0 1	0.0018
13	0 0 0 0 1 0 1	0.0018
14	1 0 1 1 0 2 1	0.0072
15	1 1 10000	0.0027
16	0 0 0 0 3 0 0	0.0081
17	1 0 1 0 1 1 0	0.0036
18	1 2 1 1 1 1 1	0.0090
19	1 1 1 1 1 1 1	0.0063
20	0 1 1 1 0 0 0	0.0027

Tabel 5: Perbandingan kinerja antara algoritma genetika dengan metode lain.

Perbandingan	Efisiensi Min Sort	Algoritma Genetik
<i>Software</i>	Borland C++ 5.02	Matlab 7.6
Metode pencarian	Efisiensi Minimum Sort	Algoritma Genetik
Ketelitian program	Analitik	Pendekatan
<i>Time Horizon</i>	[0;0.7]	> 0.7
Percobaan	Tidak	Ya
Contoh Masalah 1		
untuk selang waktu [0;0.7] dengan nilai awal S,I,R,V,Sv,Iv yang sama, jalur vaksinasi yang diperoleh	1 0 0 0 0 0	1 0 0 0 0 0
Biaya vaksinasi	0.0009	0.0009
Eksekusi waktu program	78 detik	38.9 detik
Contoh Masalah 2		
untuk selang waktu [0;2] dengan nilai awal S,I,R,V,Sv,Iv yang sama, jalur vaksinasi yang diperoleh	Solusi Tidak Diperoleh	2 3 3 2 2 2 2 2 2 2 1 1 2 1 0 1 1 0 1 1
Biaya vaksinasi	-	0.0549
Eksekusi waktu program	-	276.5 detik

5. Kesimpulan

Berdasarkan uraian dan pembahasan yang telah dilakukan, maka dapat ditarik kesimpulan bahwa Algoritma Genetik dapat diterapkan dalam pencarian tingkat vaksinasi yang optimal pada model epidemik deterministik. Proses pencarian tingkat vaksinasi dengan metode algoritma genetik membutuhkan 9 tahapan utama yang meliputi : *input*-kan kondisi awal, menentukan jumlah gen, inialisasi populasi awal, evaluasi individu, *linier fitness ranking*, *roulette wheel*, *crossover*, mutasi, *generation replacement*. Dalam menentukan tingkat vaksinasi yang memenuhi kendala dilakukan pada tahap inialisasi individu dan evaluasi individu. Dimana pada tahap inialisasi individu dilakukan dengan membangkitkan suatu populasi awal yang memenuhi kendala $\max I_{\alpha}(t) \leq B$ dan $R(t)+I(t) \leq A$. Dan evaluasi individu bertujuan untuk menghitung biaya yang dibutuhkan untuk setiap solusi yang dihasilkan pada tahap inialisasi individu.

Metode algoritma genetik merupakan metode pendekatan. Artinya solusi yang dihasilkan tidak harus optimal, tetapi cukup 'bagus' atau dapat diterima. Oleh karena itu untuk mendapatkan tingkat vaksinasi yang optimal dengan biaya minimum perlu dilakukan beberapa percobaan eksekusi program. Metode algoritma genetik tidak hanya dapat diterapkan dalam pencarian tingkat vaksinasi optimal pada model epidemik deterministik, namun dapat juga diterapkan untuk mencari tingkat vaksinasi optimal pada model epidemik *host-vector*. Dalam mencari tingkat vaksinasi optimal penelitian Widyaningsih (2008) menghasilkan jalur dan biaya yang lebih baik, namun *time horizon*-nyahanya terbatas pada waktu akhir $T = 0.8$. Pada penelitian yang sedang dilakukan, dapat dicari tingkat vaksinasi yang tidak terbatas pada waktu akhir $T = 0.8$ dan dengan waktu eksekusi program yang jauh lebih singkat. Pada perbandingan kinerja program antara metode efisiensi minimum sort

dengan algoritma genetik untuk model epidemik tanpa vektor diperoleh selisih 1 tingkat jalur vaksinasi yang menyebabkan perbedaan biaya sebesar 0.0009. Namun pada perbandingan kinerja program untuk model epidemik *host-vector* tidak memiliki perbedaan pada tingkat vaksinasi.

Pernyataan: Tulisan ini merupakan intisari dari skripsi S1 penulis pertama di Jurusan Matematika FMIPA Unpad pada tahun 2009 dan ditulis secara independen dengan tidak mengacu pada Widayani dkk [7].

Daftar Pustaka

1. Djojodihardjo, H. 2000. *Metode Numerik*. Jakarta: PT. Gramedia Pustaka Umum.
2. Gen, M. dan R. Cheng. 1997. *Genetic Algorithms And Engineering*. United States Of America: John Wiley and Sons.
3. Goldberg, D.E. 1989. *Genetic algorithms in search optimization and machine learning*. United State of America: Addison Westley.
4. Hethcote, W.H. dan Waltman, P. 1973. *Optimal Vaccination Schedules in a Deterministic Epidemic Model*. *Mathematical Biosciences* (18): 365-381.
5. Munir, R. 2003. *Metode Numerik*. Bandung: Informatika.
6. Suyanto. 2005. *Algoritma Genetik Dalam Matlab*. Yogyakarta: Andi Yogyakarta.
7. Widayani, H., M. Kalista, N. Nuraini dan M. Sari. 2014. *Optimization model of vaccination strategy for dengue transmission*. *Amer. Ins. Phys. Conf. Proc.* 1587, 127 (2014); doi: 10.1063/1.4866548.
8. Widyaningsih, S.. 2008. *Penjadwalan Vaksinasi Optimal Pada Model Epidemik Deterministik Dengan Efisiensi Minimal Sort*. Skripsi tidak diterbitkan. Bandung: Program Sarjana UNPAD.
9. Zuleni, L. 2008. *Model Matematika Penyebaran Penyakit Menular Melalui Host Vector*. Skripsi tidak diterbitkan. Bandung: Program Sarjana UNPAD.

