

**APLIKASI PENCARIAN RUTE TERPENDEK MENGGUNAKAN
ALGORITMA GENETIKA
(Studi Kasus: Pencarian Rute Terpendek untuk Pemadam Kebakaran di
Wilayah Kota Pontianak)**

^[1]Putri Yuli Utami, ^[2]Cucu Suhery, ^[3]Ilhamsyah

^[1] ^[2] ^[3]Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
Jl. Ahmad Yani, Pontianak
Telp./Fax.: (0561) 577963
e-mail:

^[1]putriyuliutami@gmail.com, ^[2]csuhery@gmail.com, ^[3]ilhamsm99@gmail.com

Abstrak

Pencarian rute terpendek (shortest path problem) adalah permasalahan untuk mencari rute minimum dari titik (node) awal ke titik (node) tujuan. Salah satu algoritma yang dapat digunakan untuk menyelesaikan masalah pencarian rute terpendek adalah Algoritma Genetika. Algoritma genetika digunakan untuk mencari rute terpendek yang akan dilalui oleh petugas pemadam kebakaran di wilayah Kota Pontianak dengan memperhatikan kondisi jalan yaitu waktu kemacetan dan panjang ruas jalan. Pencarian rute terpendek dilakukan berdasarkan titik awal, titik tujuan dan waktu keberangkatan. Pada penelitian ini, metode seleksi yang digunakan adalah roulette wheel dan elitisme. Metode crossover yang digunakan yaitu Partially Mapped Crossover (PMX) dengan probabilitas crossover 0,6 dan probabilitas mutasi 0,01. Berdasarkan hasil penelitian, rute terpendek merupakan rute tercepat tetapi jarak terpendek tidak berarti rute terpendek karena memperhitungkan faktor nilai bobot kemacetan jalan.

Kata Kunci: Algoritma Genetika, *Shortest Path Problem*, Crossover, Mutasi

1. PENDAHULUAN

Pencarian rute terpendek (*shortest path problem*) merupakan suatu permasalahan optimasi mencari rute minimum yang diperlukan untuk mencapai tempat tujuan berdasarkan beberapa jalur alternatif yang tersedia.

Salah satu permasalahan pencarian rute terpendek adalah rute pemadam kebakaran. Pada pencarian rute pemadam kebakaran dibutuhkan jarak tempuh yang pendek dan terhindar dari kemacetan agar segera tiba di tempat tujuan. Titik (*node*) awal adalah tempat pemadam kebakaran sedangkan titik (*node*) tujuan adalah tempat terjadi kebakaran. Untuk memecahkan permasalahan pencarian rute terpendek ini digunakan algoritma genetika. Algoritma genetika merupakan salah satu algoritma yang digunakan untuk menyelesaikan masalah optimasi.

Algoritma genetika meniru cara kerja proses genetika pada makhluk hidup, dimana terdapat proses seleksi, *crossover* dan mutasi untuk mendapatkan kromosom terbaik pada suatu generasi.

2. TINJAUAN PUSTAKA

2.1 Algoritma Genetika

Algoritma Genetika merupakan algoritma yang terinspirasi dari teori evolusi Darwin, yang menyatakan bahwa kelangsungan hidup suatu makhluk dipengaruhi oleh aturan yang kuat adalah yang menang. Berdasarkan teori evolusi, algoritma genetika dapat digunakan untuk mencari solusi dalam permasalahan optimasi (Widodo, 2012).

2.2 Komponen Algoritma Genetika

a. Pendefinisian Individu

Yaitu proses membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu.

b. Fungsi Evaluasi

Fungsi evaluasi merupakan tahapan memberikan nilai dengan menghitung nilai *fitness* setiap kromosom. Pada masalah optimasi, jika solusi yang dicari adalah memaksimalkan sebuah fungsi *h* (dikenal sebagai masalah maksimasi), maka nilai *fitness* yang digunakan adalah nilai dari fungsi *h* tersebut, $f = h$ (dimana *f* adalah nilai *fitness*). Tetapi jika masalahnya adalah meminimalkan fungsi *h* (masalah minimasi), maka fungsi *h* tidak bisa digunakan secara langsung. Hal ini disebabkan adanya aturan bahwa individu yang memiliki nilai *fitness* tinggi lebih mampu bertahan hidup pada generasi berikutnya. Oleh karena itu, nilai *fitness* yang bisa digunakan adalah $f = 1/h$, artinya semakin kecil nilai *h*, semakin besar nilai *f*. Pada masalah-masalah tertentu *h* bisa bernilai 0, yang mengakibatkan *f* dapat bernilai tak hingga. Untuk mengatasinya, *h* perlu ditambah sebuah bilangan yang dianggap sangat kecil sehingga nilai *fitness*nya menjadi: (Suyanto, 2005)

$$f = \frac{1}{(a + h)} \dots\dots\dots(1)$$

dengan:

f = nilai *fitness*

a = bilangan yang dianggap sangat kecil dan bervariasi sesuai dengan masalah yang akan diselesaikan.

Pada penelitian ini, *h* merupakan nilai panjang ruas jalan berdasarkan nilai kemacetan setiap rute.

Sehingga, *h* dihitung dengan rumus:

$$h = \sum_{i=1}^n x_i y_i \dots\dots\dots(2)$$

dengan:

h = nilai fungsi evaluasi

x_i = panjang ruas jalan

y_i = nilai bobot kemacetan jalan

c. Seleksi

Seleksi merupakan metode untuk mendapatkan calon induk yang baik. Metode

seleksi induk yang akan digunakan adalah *roulette wheel selection* dan *elitisme*.

Algoritma seleksi *Roulette Wheel*:

1. Hitung total *fitness* (*f*):

$$\text{Total fitness} = \sum f_k ; k= 1, 2, \dots n$$

2. Hitung probabilitas *fitness* (*p_k*) tiap individu:

$$p_k = \text{fitness}(i) / \text{total_Fitness}$$

3. Hitung probabilitas kumulatif (*q_k*):

$$q_1 = p_1$$

$$q_k = q_{k-1} + p_k ; k= 2,3,$$

4. Pilih induk yang akan di *crossover* dengan cara:

Bangkitkan bilangan random *r*. Jika $q_k < r$ dan $q_{k+1} > r$, maka pilih kromosom ke (*k*+1) sebagai pilihan induk.

d. Kawin Silang (*Crossover*)

Crossover adalah proses kawin silang yang melibatkan dua induk untuk membentuk kromosom baru. *Crossover* tidak selalu dilakukan pada semua kromosom yang ada, kromosom dipilih secara acak untuk dilakukan *crossing* dengan probabilitas *crossover* antara 0,6 sampai 0,95. *Crossover* dilakukan dengan harapan kromosom-kromosom baru yang terbentuk dapat lebih baik dari kromosom lama.

e. Mutasi

Proses mutasi bertujuan untuk mengubah salah satu atau beberapa gen dalam satu kromosom. Parameter yang sangat penting dalam mutasi adalah probabilitas mutasi. Probabilitas mutasi menunjukkan persentase dari jumlah total gen dalam satu populasi yang akan mengalami mutasi. Umumnya, probabilitas mutasi diberi nilai yang kecil sebab, jika terlalu besar akan terlalu banyak gangguan acak, sehingga anak kehilangan sifat induknya (Kusumadewi, 2003).

3. PERANCANGAN SISTEM

3.1 Perancangan Algoritma Genetika

Pada penelitian ini perancangan algoritma genetika berisikan masukan, proses dan keluaran.

3.1.1 Masukan

Proses masukan yang diperlukan untuk pembuatan aplikasi pencarian rute terpendek, berupa data nama jalan, panjang ruas jalan, waktu kemacetan dan parameter dalam algoritma genetika.

a. Data nama jalan merupakan titik (*node*) yang menjelaskan tempat, persimpangan jalan atau belokan jalan. Nama jalan disimbolkan dengan urutan nomor, dimana terdapat 53 *node* yang digunakan. Jalan-jalan tersebut merupakan jalan yang dimungkinkan dapat dilewati menuju ke titik tujuan. Pada penelitian ini, ditetapkan empat titik awal yaitu Kantor Dinas Pemadam Kebakaran Kota Pontianak Jl. A. Yani, Yayasan Budi Pekerti Jl. Gajah Mada, Yayasan Panca Bhakti Jl. L. Suprpto dan Yayasan Bintang Timur Jl. Rasuna Said, serta 49 titik tujuan dari 53 total ruas jalan yang digunakan.

b. Panjang ruas jalan merupakan ruas (*edge*) yang menjelaskan jarak antara *node* satu dengan *node* lainnya. Panjang ruas jalan dihitung menggunakan Aplikasi Google Maps.

c. Waktu kemacetan dibuat dalam tiga pengelompokan yaitu normal, padat dan macet. Waktu kemacetan berhubungan dengan hasil nilai evaluasi dari setiap rute yang terbentuk, untuk itu setiap kelompok kemacetan memiliki nilai. Normal memiliki nilai kemacetan 1, padat memiliki nilai kemacetan 1,5 dan macet memiliki nilai kemacetan 2.

d. Parameter yang digunakan dalam proses algoritma genetika ditunjukkan pada **Tabel 1** berikut.

Tabel 1 Spesifikasi Algoritma Genetika

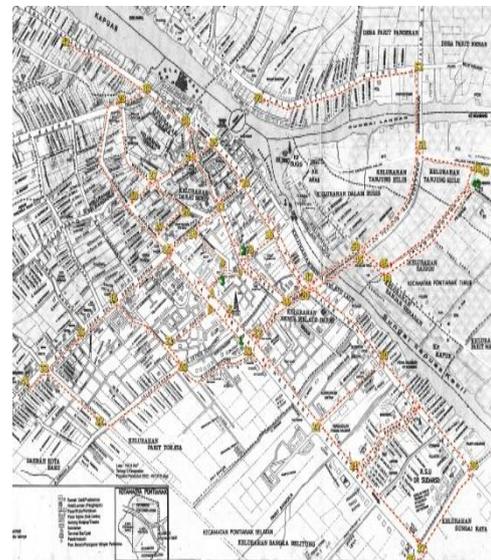
Parameter Awal	Nilai dan Metode
Titik awal	4 titik awal
Titik tujuan	49 titik tujuan
Jumlah node keseluruhan	53
Panjang kromosom maksimum	16
Metode seleksi yang digunakan	1. Seleksi <i>Roulette Wheel</i>

	2. Seleksi <i>Elitisme</i>
Jenis <i>Crossover</i>	<i>Partial Mapped Crossover</i> (PMX)
Peluang <i>crossover</i>	0,6
Peluang mutasi	0,01
Maksimum iterasi	1000

Tabel 1 merupakan spesifikasi algoritma genetika yang digunakan dalam pemrograman.

3.1.2 Proses

Proses algoritma genetika yang dilakukan yaitu perhitungan nilai evaluasi, perhitungan nilai *fitness*, seleksi, *crossover*, mutasi hingga penentuan iterasi maksimum. Rute pada penelitian ini ditunjukkan pada Gambar 1 berikut.



Gambar 1 Peta Kota Pontianak

Gambar 1 merupakan peta Kota Pontianak yang menunjukkan rute jalan.

3.1.2 Keluaran

Keluaran yang diperoleh yaitu rute terpendek.

3.2 Perancangan Aplikasi

Sistem pencarian rute terpendek dimulai ketika *user* memasukan titik awal, titik tujuan dan waktu keberangkatan pada aplikasi. Setelah menekan tombol “cari rute” maka proses algoritma genetika akan bekerja. Proses algoritma genetika

dimulai dari pembentukan populasi awal berisi sejumlah kromosom yang menjadi pilihan rute alternatif, kemudian proses evaluasi yaitu perhitungan nilai evaluasi kromosom dan nilai *fitness*. Selanjutnya tahapan seleksi nilai *fitness* untuk menghasilkan kromosom baru dengan nilai *fitness* terbaik. Kromosom baru yang dihasilkan akan dilanjutkan ke tahapan *crossover* dan mutasi. Proses ini akan terus berlangsung sampai kriteria penghentian *iterasi* tercapai dan dihasilkan rute terpendek.

3.3 Perancangan Antarmuka

Perancangan antarmuka dibuat pada tampilan *web browser* dengan program PHP. Aplikasi ini memiliki satu buah form utama, pada form utama ini *user* akan memasukan titik awal, titik tujuan dan waktu keberangkatan yang dibutuhkan untuk menjalankan sistem. Tampilan utama sistem ditunjukkan pada **Gambar 2** berikut.



Gambar 2 Tampilan Utama Aplikasi

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Perangkat lunak yang dibangun pada penelitian ini diimplementasikan menggunakan program PHP. Adapun implementasi algoritma genetika yang dilakukan terdapat 6 tahapan, yaitu sebagai berikut:

a. Inisialisasi Kromosom

Pada tahap inisialisasi membangkitkan populasi awal secara random berdasarkan masukan titik awal dan titik tujuan

oleh user. Untuk membangkitkan populasi awal dibuat matriks dua dimensi yang akan membentuk rute jalan. Berikut data matriks inisialisasi rute jalan awal.

```
$tempRute = "";  
$rute[1][4] = 4;  
$rute[1][5] = 5;  
$rute[1][6] = 6;  
$rute[1][15] = 15;  
$rute[2][7] = 7;  
$rute[2][8] = 8;  
$rute[2][10] = 10;  
$rute[3][4] = 4;  
$rute[3][7] = 7;  
.....
```

b. Evaluasi

Pada tahap evaluasi mencari nilai evaluasi dan nilai *fitness* setiap kromosom. Nilai evaluasi menghitung panjang ruas jalan dan nilai kemacetan berdasarkan waktu keberangkatan. Pada program rumus evaluasi dan *fitness* dituliskan sebagai berikut:

```
// menghitung nilai evaluasi  
$hasilEval += ($data2[0]*$val);  
// rumus fitness  
$hasilfit = 1 / (1+$paramFit);
```

c. Seleksi

Proses seleksi yaitu membandingkan nilai probabilitas kumulatif dengan nilai random *roulette wheel* kemudian dipilih nilai probabilitas kumulatif yang lebih besar untuk dijadikan kromosom baru sebagai hasil seleksi. Proses seleksi dituliskan sebagai berikut:

```
//membandingkannilaikomulatifdengannilai roulette wheel  
if(($arrKomulatif[($j-1)]<$arrRandomRW[$i])and ($arrRandomRW[$i]<$arrKomulatif[($j)]))
```

c. Crossover

Proses *crossover* membangkitkan bilangan random sebanyak jumlah kromosom dalam satu populasi, kemudian membandingkannya dengan probabilitas

crossover dan dipilih bilangan random yang lebih kecil dari 0,6. Kromosom yang terpilih akan mengalami *crossover* dengan metode *partially mapped crossover*, yaitu menentukan dua posisi substring kromosom pada aturan acak. Kemudian menukarkan dua bagian *substring* induk untuk menghasilkan anak. Selanjutnya, mengecek kromosom ke-2 ada yang sama atau tidak, jika ada yang sama maka gen akan ditukarkan. Hasil *crossover* akan terbentuk populasi baru untuk proses selanjutnya, yaitu proses mutasi.

Pada program proses *crossover* dituliskan sebagai berikut:

```
//perulangan sebanyak $jlhRute
for($i=0;$i<$jlhRute;$i++){ $arrCO[] =
nilacak();
//membandingkan dengan Probabilitas
crossover
if($arrCO[$i]<= $probCrossover){
// menentukan dua posisi substring
kromosom pada aturan acak
$jlhSubAcak = rand(2, $batas-2);
$awalIndSubAcak = rand(1, ($batas-1-
$jlhSubAcak));
$akhirIndSubAcak= $awalIndSubAcak+
($jlhSubAcak-1);
//menukarkan dua bagian substring induk
untuk menghasilkan anak
$subA[] = $arrRuteA[$i];
$subB[] = $arrRuteB[$i];
$temp = $arrRuteA[$i];
$arrRuteA[$i] = $arrRuteB[$i];
$arrRuteB[$i] = $temp;
// mengecek kromo-som ke-2 ada yang
sama atau tidak,
$sama = array();
for($i=0;$i<(count($arrRuteB));$i++){
for($j=( $i+1);$j<(count($arrRuteB));$j++)
{
if($arrRuteB[$i]== $arrRuteB[$j]){
$sama[] = $j
```

d. Mutasi

Proses mutasi dimulai dengan menghitung jumlah gen keseluruhan dalam satu populasi. Kemudian menghitung jumlah gen yang akan mengalami mutasi. Selanjutnya, membuat bilangan

acak sebanyak jumlah gen keseluruhan dan dipilih bilangan acak yang lebih kecil dari probabilitas mutasi 0,01. Kemudian nilai dari gen yang terpilih akan saling ditukarkan. Hasil mutasi akan terbentuk populasi baru. Setiap populasi baru hasil mutasi, akan digunakan sebagai populasi awal untuk generasi berikutnya.

Pada program proses mutasi dituliskan sebagai berikut:

```
// menghitung jumlah gen
$jlhseluruh += $jlhdatarute;
//menghitung jumlah gen yang akan
mengalami mutasi
$jlhgendimutasi =
ceil($probMutasi*$jlhseluruh);
//bilangan acak
$arrRandomSeluruh[] = nilacak();
if($arrRandomSeluruh[$i]<=
$probMutasi){
//menukarkan nilai gen yang terpilih
$stempTukar = $genTukar1;
$arrTukar1[$indeksTukar1] =
$genTukar2;
$arrTukar2[$indeksTukar2] =
$stempTukar;
```

e. Penghentian Iterasi

Proses algoritma genetika akan berhenti setelah mencapai batas maksimum yaitu 1000 iterasi. Proses penghentian *iterasi* pada program ditunjukkan sebagai berikut:

```
//batasmaksimumiterasi
for($sturunan=1;$sturunan<=1000;$sturunan
++){}
```

4.2 Pengujian

1. Pengujian pertama dari titik awal 1 yang berada di Jl. A.Yani, ketitik tujuan 42 yang berada di Jl. Kom. Yos. Sudarso. Hasil pengujian ditunjukkan **Tabel 2** berikut.

Tabel 2. Hasil Pengujian

Waktu Keberangkatan	Fitness Terbaik	Nilai Evaluasi	Jarak (km)	Rute
00.00-05.59	0.1626	5.15	5.15	1-15-19-28-39-42
06.00-07.59	0.1136	7.8	5.2	1-15-28-39-42
08.00-10.59	0.132	6.575	5.25	1-15-19-27-28-39-42
11.00-13.59	0.1356	6.375	5.25	1-15-19-27-28-39-42
14.00-15.59	0.132	6.575	5.25	1-15-19-27-28-39-42
16.00-17.59	0.1136	7.8	5.2	1-15-28-39-42
18.00-20.59	0.127	6.875	5.25	1-15-19-27-28-39-42
21.00-23.59	0.1626	5.15	5.15	1-15-19-28-39-42

2. Pengujian kedua dari titik awal 2 yang berada di Jl. Gajah Mada, ke titik tujuan 42 yang berada di Jl. Kom. Yos. Sudarso. Hasil pengujian ditunjukkan **Tabel 3** berikut.

Tabel 3. Hasil Pengujian

Waktu Keberangkatan	Fitness Terbaik	Nilai Evaluasi	Jarak (km)	Rute
00.00-05.59	0.1786	4.6	4.6	2-7-11-34-38-39-42
06.00-07.59	0.1205	7.3	5.0	2-7-11-29-35-38-39-42
08.00-10.59	0.1399	6.15	5.0	2-7-11-29-35-38-39-42
11.00-13.59	0.146	5.85	4.6	2-7-11-34-38-39-42
14.00-15.59	0.1399	6.15	5.0	2-7-11-29-35-38-39-42
16.00-17.59	0.1205	7.3	5.0	2-7-11-34-38-39-42
18.00-20.59	0.1282	6.8	5.0	2-7-11-29-35-38-39-42
21.00-23.59	0.1786	4.6	4.6	2-7-11-34-38-39-42

3. Pengujian ketiga dari titik awal 3 yang berada di Jl. L. Suprpto ke titik tujuan 42 yang berada di Jl. Kom. Yos. Sudarso. Hasil pengujian ditunjukkan **Tabel 4** berikut.

Tabel 4. Hasil Pengujian

Waktu Keberangkatan	Fitness Terbaik	Nilai Evaluasi	Jarak (km)	Rute
00.00-05.59	0.1852	4.4	4.4	3-7-11-34-38-39-42
06.00-07.59	0.1235	7.1	4.8	3-7-11-29-35-38-39-42
08.00-10.59	0.1439	5.95	4.8	3-7-11-29-35-38-39-42
11.00-13.59	0.1504	5.65	4.8	3-7-11-34-38-39-42
14.00-15.59	0.1439	5.95	4.8	3-7-11-29-35-38-39-42
16.00-17.59	0.1235	7.1	4.8	3-7-11-29-35-38-39-42
18.00-20.59	0.1316	6.6	4.8	3-7-11-29-35-38-39-42
21.00-23.59	0.1852	4.4	4.4	3-7-11-34-38-39-42

4. Pengujian keempat dari titik awal 43 yang berada di Jl. Rasuna Said, ke titik tujuan 42 yang berada di Jl. Kom. Yos. Sudarso. Hasil pengujian ditunjukkan **Tabel 5** berikut.

Tabel 5. Hasil Pengujian

Waktu Keberangkatan	Fitness Terbaik	Nilai Evaluasi	Jarak (km)	Rute
00.00-05.59	0.1085	8.22	8.22	43-44-45-18-29-35-38-39-42
06.00-07.59	0.0715	12.99	8.22	43-44-45-18-29-35-38-39-42
08.00-10.59	0.0918	9.89	8.22	43-44-45-18-29-35-38-39-42
11.00-13.59	0.1037	8.64	8.22	43-44-45-18-29-35-38-39-42
14.00-15.59	0.0794	11.59	8.22	43-44-45-18-29-35-38-39-42
16.00-17.59	0.0702	13.24	8.22	43-44-45-18-29-35-38-39-42
18.00-20.59	0.0817	11.24	8.22	43-44-45-18-29-35-38-39-42
21.00-23.59	0.1085	8.22	8.22	43-44-45-18-29-35-38-39-42

Berdasarkan tabel pengujian setiap rute memiliki nilai *fitness* dan nilai evaluasi yang berbeda walaupun rutanya sama. Hal ini karena nilai bobot kemacetan jalan berpengaruh terhadap nilai evaluasi.

Hasil analisis terhadap tabel pengujian, diperoleh nilai evaluasi terendah terdapat diantara jam 00.00-05.59 dan 21.00-23.59 karena dalam rentang waktu tersebut nilai bobot kemacetan jalan normal yaitu dalam keadaan lalu lintas lancar. Sedangkan nilai evaluasi tertinggi terdapat diantara jam 06.00-07.59 dan 16.00-17.59 karena dalam rentang

waktu tersebut nilai bobot kemacetan jalan macet yaitu dalam keadaan lalu lintas sangat padat.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pengujian dapat diambil kesimpulan sebagai berikut:

1. Hasil penelitian ini, perhitungan untuk mencari rute terpendek tidak hanya menggunakan faktor jarak tetapi juga menggunakan faktor nilai bobot kemacetan, maka hasil yang diperoleh adalah rute terpendek tidak selalu merupakan jarak terpendek tetapi dapat pula jarak yang lebih panjang namun memiliki faktor nilai bobot kemacetan yang kecil.

2. Nilai bobot kemacetan jalan berpengaruh terhadap nilai evaluasi, sehingga berdasarkan hasil analisis terhadap keseluruhan table pengujian, diperoleh nilai evaluasi terendah terdapat diantara jam 00.00-05.59 dan 21.00-23.59 karena dalam rentang waktu tersebut nilai bobot kemacetan jalan normal yaitu dalam keadaan lalu lintas lancar. Sedangkan nilai evaluasi tertinggi terdapat diantara jam 06.00-07.59 dan 16.00-17.59 karena dalam rentang waktu tersebut nilai bobot kemacetan jalan macet yaitu dalam keadaan lalu lintas sangat padat.

5.2 Saran

Pada penelitian selanjutnya dapat menggunakan metode seleksi dan metode *crossover* yang berbeda, serta membandingkan metode mana yang lebih baik digunakan. Aplikasi juga dapat ditambahkan titik tujuan dengan memperhitungkan nilai panjang ruas jalan dan nilai bobot kemacetan jalan.

DAFTAR PUSTAKA

[1] Kusumadewi, Sri. (2003). *Artificial Intelligence*. Yogyakarta: Graha Ilmu.

[2] Suyanto. (2005). *Algoritma Genetika dalam Matlab*. Yogyakarta: Andi Offset .

[3] Widodo, Thomas. S. (2012). *Komputasi Evolusioner Algoritma Genetik, Pemrograman Genetik, dan Pemrograman Evolusioner*. Yogyakarta: Graha.