

PEMAMPATAN MATRIKS JARANG DENGAN METODE ALGORITMA GENETIKA MENGGUNAKAN PROGRAM PASCAL

YOSI PUTRI, NARWEN

*Program Studi Matematika,
Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Andalas,
Kampus UNAND Limau Manis Padang, Indonesia,
yosiiputrii@gmail.com*

Abstrak. Algoritma Genetika, yang pertama kali diperkenalkan secara terpisah oleh Holland dan De Jong pada tahun 1975, adalah teknik yang optimal untuk pencarian stokastik, dan berlandaskan prinsip dasar teori evolusi. Algoritma Genetika mempunyai aplikasi di berbagai bidang, antara lain pada pengaturan penjadwalan dari sederetan kegiatan, pencocokan kata secara acak, dan lain sebagainya. Tulisan ini bertujuan untuk mengkaji proses algoritma Genetika melalui pemampatan matriks jarang menggunakan Pascal. Faktor-faktor yang mempengaruhi ketepatan hasil dari pemampatan matriks ini juga diberikan.

Kata Kunci: Algoritma Genetika, matriks jarang, teknik optimisasi, Pascal

1. Pendahuluan

Array sebagai struktur data yang paling sederhana dapat berdimensi satu atau yang disebut dengan larik, serta dapat juga berdimensi dua atau yang disebut dengan matriks. Penyimpanan elemen matriks di dalam memori komputer akan menjadi boros jika banyak elemen dari matriks tersebut yang bernilai kosong (nol), salah satu solusinya adalah penyimpanan matriks tersebut dikonversi menjadi sebuah larik (*array* satu dimensi) atau yang dikenal dengan istilah pemampatan.

2. Matriks Jarang

Matriks merupakan *array* dua dimensi. *Array* adalah koleksi data yang tipenya sama, tersusun dalam bentuk barisan berurutan dan jumlah elemen atau datanya tidak berubah sesuai deklarasi awal. Berdasarkan konvensi, indeks pertama adalah baris, indeks kedua adalah kolom [2]. Definisi matriks jarang adalah matriks yang hanya mempunyai beberapa elemen tidak nol. Suatu matriks $n \times m$ dengan sejumlah k elemen tidak nol dikatakan sebagai matriks jarang bila k jauh lebih sedikit dibandingkan $n \times m$ [3].

3. Algoritma Genetika

Algoritma Genetika merupakan suatu algoritma pencarian yang berbasis pada mekanisme seleksi alam dan genetika. Sejak algoritma Genetika pertama kali dirin-

tis oleh John Holland dari Universitas Michigan pada tahun 1960-an, algoritma Genetika telah diaplikasikan secara luas pada berbagai bidang. John Holland menyatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika. Algoritma Genetika banyak digunakan dalam menyelesaikan masalah optimasi kompleks yang sulit dilakukan oleh metode konvensional [4].

Pada algoritma Genetika, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness*. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas dari kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah anak (*offspring*) terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator *crossover*.

Algoritma Genetika terdiri dari dua buah operasi yaitu operasi genetika dan operasi evolusi. Operasi genetika terdiri dari operator *crossover* dan operator mutasi. Operator *crossover* bertujuan menambah keanekaragaman *string* dalam satu populasi, sedangkan operator mutasi bertujuan untuk mengubah salah satu atau lebih bagian dalam kromosom. Pada operasi evolusi terdapat operator reproduksi, dimana reproduksi bertujuan untuk membentuk populasi dari generasi ke generasi. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai *fitness* dari kromosom induk (*parent*) dan nilai *fitness* dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan [1].

Fungsi evaluasi adalah penghubung antara algoritma Genetika dan masalah yang akan diselesaikan. Fungsi evaluasi merupakan dasar untuk melakukan proses reproduksi. Pada proses reproduksi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang dinamakan *fitness*.

4. Pemampatan Matriks Jarang dengan Metode Algoritma Genetika Menggunakan Program Pascal

4.1. Representasi Kromosom pada Matriks Jarang

Misalkan diberikan sebuah matriks jarang berukuran 10×10 . Matriks jarang tersebut diberikan dalam bentuk Tabel 1.

Warna abu-abu pada Tabel 1 di kolom ke-1 merupakan *dummy*, yaitu elemen yang bersifat unik. *Dummy* ini berfungsi sebagai penanda kolom ke-1 untuk baris tertentu. Selanjutnya akan dibentuk generasi awal untuk setiap individu yang diberikan. Misalkan banyak individu setiap generasi adalah sepuluh, maka akan diberikan masing-masing dengan sepuluh gen. Secara acak diperoleh hasil populasi awal pada Gambar 1.

Selanjutnya akan dibangun matriks satu dimensi. Untuk mencari individu per-

	1	2	3	4	5	6	7	8	9	10
1	0	10	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	27	0	0
4	0	31	0	0	34	0	0	0	38	0
5	4	0	0	0	0	0	46	0	0	0
6	5	0	52	0	0	0	0	0	0	0
7	0	61	62	0	0	0	0	0	0	0
8	7	71	0	0	0	75	76	0	0	0
9	8	0	0	83	0	0	0	0	0	0
10	9	0	0	0	0	0	0	0	98	99

Tabel 1. Contoh matriks jarang 10×10

No	Individu									
1	9	8	4	5	6	7	1	3	2	10
2	1	3	5	6	4	8	10	7	2	9
3	5	7	8	6	2	9	10	1	3	4
4	5	7	6	2	3	8	4	1	9	10
5	3	6	8	9	2	7	10	1	5	4
6	1	3	5	9	6	10	8	2	7	4
7	6	3	4	10	1	5	7	2	8	9
8	8	1	3	6	7	9	2	10	5	4
9	6	9	3	5	4	8	2	7	10	1
10	1	5	3	7	2	9	10	4	6	8

Gambar 1. Hasil populasi awal

tama, misalkan secara acak diperoleh keterurutan seperti yang ditunjukkan pada Tabel 2.

Lokus	1	2	3	4	5	6	7	8	9	10
Allele	9	8	4	5	6	7	1	3	2	10

Tabel 2. Representasi kromosom untuk individu pertama

Karena ada sepuluh baris, ini berarti bahwa dalam satu kromosom terdapat sepuluh gen. Nilai-nilai gen pada kromosom menunjukkan operasi yang spesifik. Proses penempatan baris didasarkan pada nilai-nilai gen tersebut. Pada representasi di atas, lokus 1 berarti urutan penempatan ke-1, lokus 2 berarti urutan penempatan ke-2. Sehingga baris ke-9 memiliki urutan penempatan ke-1, baris ke-8 memiliki urutan penempatan ke-2, dan seterusnya hingga baris ke-10 yang memiliki urutan penempatan ke-10. Allele 9 berarti baris ke-9. Elemen dari baris ke-9 ditempatkan di *array* satu dimensi dan *array* posisi menyimpan posisi baris dari elemen tidak nol seperti yang dapat dilihat pada Gambar 2.

Allele 8 berarti baris ke-8. Elemen dari baris ke-8 ditempatkan di *array* satu dimensi dengan aturan bahwa bila di suatu kolom di *array* satu dimensi sudah

Array satu dimensi	0	0	0	0	0	0	0	0	0	0	0
Array posisi	0	0	0	0	0	0	0	0	0	0	0
(a)											
Baris 9	8	0	0	83	0	0	0	0	0	0	0
Array satu dimensi	8	0	0	83	0	0	0	0	0	0	0
Array posisi	9	0	0	9	0	0	0	0	0	0	0
(b)											

Gambar 2. (a) Kondisi awal *array* (b) Kondisi *array* setelah penempatan baris kesembilan dari individu pertama

ditempati oleh elemen tidak nol dari penempatan sebelumnya, kolom tersebut tidak dapat ditempati lagi oleh elemen tidak nol dari penempatan berikutnya. Untuk menempatkan baris ke-8, karena kolom ke-1 dan kolom ke-4 dari *array* satu dimensi sudah ditempati oleh elemen 8 dan 83 dari penempatan sebelumnya, maka elemen 7 tidak dapat ditempatkan lagi di kolom ke-1 tetapi ditempatkan pada kolom ke-2 seperti yang dapat dilihat pada Gambar 3.

Baris 8	7	71	0	0	0	75	76	0	0	0	0
Array satu dimensi	8	7	71	83	0	0	75	76	0	0	0
Array posisi	9	8	8	9	0	0	8	8	0	0	0

Gambar 3. Kondisi *array* setelah penempatan baris kedelapan dari individu pertama

Seterusnya hingga allele yang terakhir yaitu allele 10, penempatan elemen pertama barisnya pada *array* satu dimensi dimulai pada kolom ke-23. Seperti yang dapat dilihat pada Gambar 4 yang sekaligus merupakan proses akhir dari pengurutan gen untuk individu pertama.

Baris 10																												9	0	0	0	0	0	0	0	98	99
Array satu dimensi	8	7	71	83	0	31	75	76	34	4	1	5	38	52	2	46	0	61	62	0	10	27	9	0	0	0	0	0	0	98	99						
Array posisi	9	8	8	9	4	4	8	8	4	5	2	6	4	6	3	5	7	7	7	1	1	3	10	0	0	0	0	0	0	10	10						

Gambar 4. Kondisi *array* setelah penempatan baris kesepuluh dari individu pertama

Selanjutnya untuk individu kedua, misalkan secara acak diperoleh keterurutan seperti yang ditunjukkan pada Tabel 3.

Dengan cara yang sama, dilakukan proses pengurutan gen dengan hasil akhir seperti yang dapat dilihat pada Gambar 5. Seterusnya hingga individu kesepuluh dengan hasil akhir pengurutan gennya seperti yang dapat dilihat pada Gambar 6.

Lokus	1	2	3	4	5	6	7	8	9	10
Allele	1	3	5	6	4	8	10	7	2	9

Tabel 3. Representasi kromosom untuk individu kedua

Baris 9																												8	0	0	83	0	0	0	0	0	0
Array satu dimensi	0	10	2	5	4	52	1	0	31	27	46	34	7	71	9	38	0	75	76	0	61	62	98	99	8	0	0	83	0	0	0	0	0	0			
Array posisi	1	1	3	6	5	6	2	4	4	3	5	4	8	8	10	4	0	8	8	7	7	7	10	10	9	0	0	9	0	0	0	0	0	0			

Gambar 5. Kondisi *array* setelah penempatan baris keempat dari individu kedua

Baris 8																												7	71	0	0	0	75	76	0	0	0
Array satu dimensi	0	10	4	2	0	61	62	1	46	8	27	9	83	0	31	0	5	34	52	98	99	38	7	71	0	0	0	75	76	0	0	0					
Array posisi	1	1	5	3	7	7	7	2	5	9	3	10	9	4	4	0	6	4	6	10	10	4	8	8	0	0	0	8	8	0	0	0					

Gambar 6. Kondisi *array* setelah penempatan baris kedelapan dari individu kesepuluh

4.2. Perhitungan Nilai Fitness

Nilai *fitness* akan dihitung dengan cara membagi banyaknya elemen tidak nol dengan panjang *array* satu dimensi. Dari perhitungan yang dilakukan, diperoleh data populasi awal individu dengan nilai *fitness* seperti yang tercantum pada Gambar 7.

No	Individu	Nilai Fitness
1.	9 8 4 5 6 7 1 3 2 10	<--> 0,69
2.	1 3 5 6 4 8 10 7 2 9	<--> 0,65
3.	5 7 8 6 2 9 10 1 3 4	<--> 0,69
4.	5 7 6 2 3 8 4 1 9 10	<--> 0,71
5.	3 6 8 9 2 7 10 1 5 4	<--> 0,61
6.	1 3 5 9 6 10 8 2 7 4	<--> 0,67
7.	6 3 4 10 1 5 7 2 8 9	<--> 0,67
8.	8 1 3 6 7 9 2 10 5 4	<--> 0,61
9.	6 9 3 5 4 8 2 7 10 1	<--> 0,63
10.	1 5 3 7 2 9 10 4 6 8	<--> 0,69

Gambar 7. Nilai *fitness* masing-masing individu

4.3. Reproduksi pada Array Satu Dimensi

Reproduksi dilakukan dengan menggunakan metode *roulette wheel selection*, untuk memilih individu induk (*parent*) dilakukan dengan menghitung persentase *fitness* masing-masing individu. Selanjutnya akan dihitung persentase dari nilai *fitness* masing-masing individu dengan jumlah total nilai *fitness* adalah 6,62. Dari perhitungan yang telah dilakukan, diperoleh persentase nilai individu seperti pada Gambar 8.

No	Individu	Nilai Fitness	Persentase
1.	9 8 4 5 6 7 1 3 2 10	<--> 0,69	<--> 10
2.	1 3 5 6 4 8 10 7 2 9	<--> 0,65	<--> 9
3.	5 7 8 6 2 9 10 1 3 4	<--> 0,69	<--> 10
4.	5 7 6 2 3 8 4 1 9 10	<--> 0,71	<--> 10
5.	3 6 8 9 2 7 10 1 5 4	<--> 0,61	<--> 9
6.	1 3 5 9 6 10 8 2 7 4	<--> 0,67	<--> 10
7.	6 3 4 10 1 5 7 2 8 9	<--> 0,67	<--> 10
8.	8 1 3 6 7 9 2 10 5 4	<--> 0,61	<--> 9
9.	6 9 3 5 4 8 2 7 10 1	<--> 0,63	<--> 9
10.	1 5 3 7 2 9 10 4 6 8	<--> 0,69	<--> 10

Gambar 8. Persentase *fitness* masing-masing individu

Dari persentase nilai *fitness* yang terdapat pada data-data di atas, dapat diketahui individu pertama mempunyai persentase nilai *fitness* 10, maka individu pertama mempunyai lebar 10. Akibatnya individu pertama menempati nilai $1, 2, \dots, 10$. Seterusnya individu kesepuluh mempunyai persentase nilai *fitness* 10, maka individu kesepuluh mempunyai lebar 10. Akibatnya individu kesepuluh menempati nilai $87, 88, \dots, 96$ sehingga diperoleh data yang tercantum pada Gambar 9.

No	Individu	Persentase	Range
1.	9 8 4 5 6 7 1 3 2 10	<--> 10	1, 2, ..., 10
2.	1 3 5 6 4 8 10 7 2 9	<--> 9	11, 12, ..., 19
3.	5 7 8 6 2 9 10 1 3 4	<--> 10	20, 21, ..., 29
4.	5 7 6 2 3 8 4 1 9 10	<--> 10	30, 31, ..., 39
5.	3 6 8 9 2 7 10 1 5 4	<--> 9	40, 41, ..., 48
6.	1 3 5 9 6 10 8 2 7 4	<--> 10	49, 50, ..., 58
7.	6 3 4 10 1 5 7 2 8 9	<--> 10	59, 60, ..., 68
8.	8 1 3 6 7 9 2 10 5 4	<--> 9	69, 70, ..., 77
9.	6 9 3 5 4 8 2 7 10 1	<--> 9	78, 79, ..., 86
10.	1 5 3 7 2 9 10 4 6 8	<--> 10	87, 88, ..., 96

Gambar 9. Range masing-masing individu

Dengan menggunakan bahasa pemrograman Pascal, dibangkitkan bilangan acak dari 1 sampai 96 sebanyak individu awal yaitu 10, dari proses pengacakan pada program pascal tersebut misalkan pengacakan pertama terpilih bilangan 28, karena 28 berada pada *range* individu ketiga, maka individu ketiga menjadi individu yang terpilih pertama. Sehingga diperoleh data individu yang terpilih pada proses reproduksi seperti yang dapat dilihat pada Gambar 10.

4.4. Pindah Silang (Crossover) pada Array Satu Dimensi

Metode pindah silang (*crossover*) yang digunakan adalah *Mixed-type Partial Matching Crossover* (MPMX) [6]. Prosedur awal untuk memilih induk mana yang akan mengalami proses pindah silang (*crossover*) adalah menentukan probabilitas *crossover*. Probabilitas *crossover* berada antara 0,6 sampai 0,9 [5]. Selanjutnya dihitung lagi nilai *fitness* masing-masing individu, kemudian dibandingkan nilai *fit-*

No	Terpilih	Susunan Kromosom
1.	<---> 10	<---> 1 5 3 7 2 9 10 4 6 8
2.	<---> 10	<---> 1 5 3 7 2 9 10 4 6 8
3.	<---> 1	<---> 9 8 4 5 6 7 1 3 2 10
4.	<---> 7	<---> 6 3 4 10 1 5 7 2 8 9
5.	<---> 9	<---> 6 9 3 5 4 8 2 7 10 1
6.	<---> 1	<---> 9 8 4 5 6 7 1 3 2 10
7.	<---> 2	<---> 1 3 5 6 4 8 10 7 2 9
8.	<---> 9	<---> 6 9 3 5 4 8 2 7 10 1
9.	<---> 6	<---> 1 3 5 9 6 10 8 2 7 4
10.	<---> 3	<---> 5 7 8 6 2 9 10 1 3 4

Gambar 10. Susunan kromosom individu terpilih

ness yang terpilih dengan probabilitas *crossover*. Misalkan diperoleh individu yang mengalami *crossover* seperti yang dapat dilihat pada Gambar 11.

No	Susunan Kromosom Anak									
1.	2	5	3	8	7	9	10	4	1	6
2.	6	5	3	2	1	9	10	4	8	7
3.	9	8	4	5	6	7	1	3	2	10
4.	6	3	4	10	1	5	7	2	8	9
5.	6	9	3	5	4	8	2	7	10	1
6.	9	8	4	5	6	7	1	3	2	10
7.	1	3	5	6	4	8	10	7	2	9
8.	6	9	3	5	4	8	2	7	10	1
9.	1	3	5	9	6	10	8	2	7	4
10.	5	7	8	6	2	9	10	1	3	4

Gambar 11. Hasil *crossover* individu

4.5. Mutasi pada Array Satu Dimensi

Metode mutasi yang digunakan adalah *remove and insert*. Pada metode ini penempatan gen hanya melihat tempat kosong pada gen sebelumnya, bukan dua atau tiga gen sebelumnya [6]. Prosedur awal yang dilakukan untuk mengetahui gen mana yang dimutasi adalah menentukan probabilitas mutasi. Probabilitas mutasi berada antara 0,001 sampai 0,01 [5]. Selanjutnya dibangkitkan bilangan acak 0 sampai dengan 1 sebanyak jumlah individu awal yaitu 10, kemudian dibandingkan bilangan acak yang terpilih dengan probabilitas mutasi. Misalkan diperoleh individu yang mengalami mutasi seperti yang dapat dilihat pada Gambar 12.

4.6. Membentuk Individu Baru

Anak hasil perkawinan silang (*crossover*) dan mutasi menjadi generasi baru untuk dilakukan proses regenerasi. Pada generasi berikutnya, individu terbaik (yang memiliki nilai *fitness* terbesar) dapat dipertahankan dengan proses elitisme. Proses elitisme adalah individu pada generasi selanjutnya yang memiliki nilai *fitness* teren-

No	Susunan Kromosom Anak									
1.	2	5	8	7	9	10	3	4	1	6
2.	6	5	3	2	1	9	10	4	8	7
3.	9	8	4	5	6	7	1	3	2	10
4.	3	4	10	1	5	7	2	8	9	6
5.	6	9	3	5	4	8	2	7	10	1
6.	9	8	4	5	7	1	6	3	2	10
7.	1	3	5	6	4	8	10	7	2	9
8.	6	9	3	5	4	8	2	7	10	1
9.	1	3	5	9	6	10	8	2	4	7
10.	5	7	6	2	9	8	10	1	3	4

Gambar 12. Hasil mutasi individu

dah dilakukan penggantian dengan individu yang memiliki nilai *fitness* terbesar. Misalkan diperoleh generasi selanjutnya dari individu seperti pada Gambar 13.

Individu Generasi ke- 2											
No	Susunan Kromosom Anak										
1.	10	5	6	7	3	4	9	1	8	2	
2.	10	4	6	5	3	1	9	2	7	8	
3.	2	6	9	10	7	8	3	4	5	1	
4.	9	5	4	3	7	8	6	2	10	1	
5.	8	9	1	2	10	7	5	4	3	6	
6.	1	5	8	7	6	9	3	2	10	4	
7.	1	9	6	4	2	7	8	5	3	10	
8.	3	1	10	6	2	5	8	4	7	9	
9.	10	5	6	7	9	1	2	4	3	8	
10.	9	5	4	3	7	8	6	2	10	1	
Individu: 9 5 4 3 7 8 6 2 10 1 <--> Nilai Fitness = 0,76											

Gambar 13. Generasi ke-2

Individu yang terdapat pada generasi ke-2 dijadikan sebagai individu awal untuk dilakukan reproduksi, *crossover*, dan mutasi. Setelah melakukan serangkaian proses algoritma Genetika tersebut diperoleh data seperti pada Gambar 14.

Individu Generasi ke- 3											
No	Susunan Kromosom Anak										
1.	4	7	6	3	10	1	9	5	8	2	
2.	10	5	7	6	3	4	9	1	8	2	
3.	4	7	1	3	9	6	2	5	10	8	
4.	5	9	8	1	7	2	4	3	10	6	
5.	1	9	2	6	7	4	8	5	3	10	
6.	7	3	5	1	2	10	4	8	6	9	
7.	1	4	2	10	5	9	7	3	8	6	
8.	3	9	1	8	10	7	5	4	2	6	
9.	2	10	3	9	7	8	4	1	6	5	
10.	4	7	1	3	9	6	2	5	10	8	
Individu: 4 7 1 3 9 6 2 5 10 8 <--> Nilai Fitness = 0,78											

Gambar 14. Generasi ke-3

Dan seterusnya hingga generasi ke-50 yang dapat dilihat pada Gambar 15 serta diperoleh satu individu yang merupakan hasil pemampatan matriks jarang yang paling optimal seperti yang dapat dilihat pada Gambar 16.

Individu Generasi ke- 50													
No	Susunan Kromosom Anak												
1.	7	3	5	10	2	1	8	6	4	9			
2.	7	3	9	8	2	5	1	10	4	6			
3.	6	2	8	3	1	9	5	7	4	10			
4.	4	7	6	9	2	8	5	10	1	3			
5.	7	8	5	6	1	3	2	9	10	4			
6.	1	7	10	2	6	8	5	3	4	9			
7.	10	2	5	7	9	8	6	3	4	1			
8.	8	1	2	5	7	6	3	4	9	10			
9.	1	8	10	2	4	3	7	5	9	6			
10.	7	8	5	6	1	3	2	10	4	9			
Individu: 7 8 5 6 1 3 2 10 4 9 <--> Nilai Fitness = 0,81													

Gambar 15. Generasi ke-50

Array satu dimensi	0	61	62	7	71	4	0	10	75	76	5	46	52	2	1	9	8	0	31	83	27	34	0	98	99	38	0
Array posisi	7	7	7	8	8	5	1	1	8	8	6	5	6	3	2	10	9	4	4	9	3	4	0	10	10	4	0

Gambar 16. Matriks satu dimensi dan matriks posisi hasil pemampatan matriks jarang

5. Ucapan Terima kasih

Penulis mengucapkan terima kasih kepada Ibu Lyra Yulianti, Ibu Susila Bahri, Bapak Zulakmal yang telah memberikan masukan dan saran sehingga paper ini dapat diselesaikan dengan baik.

Daftar Pustaka

- [1] Basuki, Achmad. 2003. *Algoritma Genetika Suatu Alternatif Penyelesaian Permasalahan Searching, Optimasi dan Machine Learning*. Politeknik Elektronika Negeri Surabaya. Surabaya
- [2] Black, Paul E. <http://www.nist.gov/dads/HTML/matrix.html>
- [3] Black, Paul E. <http://www.nist.gov/dads/HTML/sparsematrix.html>
- [4] Kusumadewi, Sri. 2005. *Pengantar Kecerdasan Buatan*. Jakarta
- [5] Sakawa, Masatoshi. 2002. *Genetic Algorithms and Fuzzy Multiobjective Optimization*. Kluwer Academic Publishers. USA. Hal 11-30
- [6] Saputro, N. dan J. Arifonang. 2005. *Mengukur Kinerja Algoritma Genetika pada Pemampatan Matriks Jarang*. Jurusan Ilmu Komputer, FMIPA, Universitas Katolik Parahyangan. Bandung