

INTEGRASI KROMOSOM BUATAN DINAMIS DAN *MULTI PARENT CROSSOVER* UNTUK MEMECAHKAN MASALAH KONVERGENSI PREMATUR PADA ALGORITMA GENETIKA

Oleh

MUHAMMAD RIKZAM KAMAL

P31.2012.01290



**PROGRAM PASCASARJANA
MAGISTER TEKNIK INFORMATIKA
UNIVERSITAS DIAN NUSWANTORO
SEMARANG**

2014

Daftar Isi

Daftar Isi.....	i
Daftar Gambar.....	iii
Daftar Tabel	iv
BAB I Pendahuluan	1
1.1 Latar Belakang Penelitian	1
1.2 Masalah Penelitian	4
1.3 Pertanyaan Penelitian	4
1.4 Tujuan Penelitian.....	4
1.5 Hubungan Antara Masalah Penelitian, Pertanyaan dan Tujuan	4
1.6 Kontribusi Penelitian	5
1.7 Sistematika Penulisan.....	5
BAB II Landasan Teori.....	7
2.1 Tinjauan Studi	7
2.1.1 Model Elsayed et al. (2014).....	7
2.1.2 Model Paves-Lazo et al. (2011)	9
2.1.3 <i>Summary of State-of-the-art on Genetic Algorithms</i>	11
2.2 Tinjauan Pustaka	12
2.2.1 Algoritma Genetika.....	12
2.2.1.1 Individu	14
2.2.1.2 Gen	15
2.2.1.3 <i>Fitness</i>	16
2.2.1.4 Populasi	17
2.2.2 Kromosom Buatan Dinamis	18
2.2.3 <i>Multi Parent Crossover (MPC)</i>	19
2.3 Kerangka Pemikiran Penelitian	20

BAB III METODE PENELITIAN.....	22
3.1 Perancangan Penelitian.....	22
3.2 Pengumpulan Data	23
3.3 Pengolahan Data Awal	24
3.4 Metode yang Diusulkan.....	24
3.5 Eksperimen dan Pengujian Model.....	26
3.6 Evaluasi dan Validasi	26
Referensi	27

Daftar Gambar

Gambar 1.1 Hubungan Masalah Penelitian, Pertanyaan Penelitian dan Kontribusi Penelitian.....	5
Gambar 2.1 Block diagram Elsyaed et al. model.....	8
Gambar 2.2 Block diagram Paves-Lazo et al.....	10
Gambar 2.3 <i>bit-string crossover</i>	13
Gambar 2.4 <i>Representation of Genotype and phenotype</i>	14
Gambar 2.5 Representasi kromosom	15
Gambar 2.6 Representasi gen.....	16
Gambar 2.7 Populasi	18
Gambar 2.8 Kerangka pemikiran penelitian	21
Gambar 3.1 Metode Penelitian.....	23
Gambar 3.2 Model yang diusulkan GA-DACMPC	25

Daftar Tabel

Tabel 1.1 Masalah Penelitian – Pertanyaan Penelitian – Tujuan Penelitian	4
Tabel 2.1 Hasil Perbandingan dengan Metode Lain	11
Tabel 2.2 <i>Summary of state-of-the-art on genetic algorithm</i>	12
Tabel 3.1 <i>Dataset Travelling Salesman Problem</i>	23
Tabel 3.2 Spesifikasi komputer yang digunakan	26

BAB I

Pendahuluan

1.1 Latar Belakang Penelitian

Algoritma genetika (*Genetic Algorithm (GA)*) adalah bagian dari komputasi evolusioner yang merupakan daerah yang berkembang pesat dalam bidang kecerdasan buatan [1]. GA adalah metode adaptif yang digunakan untuk memecahkan masalah pencarian dan optimasi. GA didasarkan pada proses genetik organisme biologis. Dengan meniru prinsip evolusi alam, yaitu "*Survival of the Fittest*", GAs mampu mengembangkan solusi untuk masalah dunia nyata [2]. Sebelum GA dapat dijalankan, representasi atau pengkodean dari masalah harus dibuat. Inti dari GA adalah untuk mengkodekan satu set parameter (dikenal sebagai gen) dan gabungan dari gen-gen yang membentuk nilai tertentu dan menyatakan solusi yang mungkin dari suatu permasalahan disebut sebagai kromosom. Fungsi *fitness* juga diperlukan, yang memberikan nilai untuk setiap solusi yang didapatkan. *Fitness* setiap individu tergantung pada kromosom dan dievaluasi oleh fungsi *fitness*. Selama proses berjalan, orang tua harus dipilih untuk proses reproduksi dan digabungkan untuk menghasilkan keturunan. Orang tua secara acak dipilih dari populasi menggunakan skema yang menguntungkan individu. Setelah memilih orang tua, kromosom mereka digabungkan, biasanya dengan menggunakan mekanisme *crossover* dan mutasi untuk menghasilkan keturunan yang lebih baik, yang berarti solusi yang lebih baik. Proses iterasi ini terus berjalan sampai kriteria yang ditentukan telah tercapai.

GA sebagai metode pencarian dan optimisasi masalah sering digunakan dalam berbagai kasus seperti *Job-shop scheduling problem*, *timetabling* dan *travelling salesman problem* dengan cara menggunakan GA yang dikombinasikan dengan berbagai metode lain seperti *triangular crossover (TC)* [3], *unimodal distribution crossover (UNDX)* [4], yang digunakan untuk menyelesaikan permasalahan

optimasi. UNDX menggunakan beberapa orang tua (*parents*) untuk menciptakan solusi keturunan (*offspring*) disekitar pusat massa dari orang tua, sementara probabilitas kecil ditugaskan untuk solusi jauh dari pusat massa. Meskipun telah menunjukkan kinerja yang sangat baik untuk masalah yang sangat epistasis (ketika efek dari satu gen tergantung pada kehadiran satu atau lebih pengubah gen) [4], tapi itu mungkin tidak berhasil untuk menghasilkan keturunan dalam beberapa kasus seperti ketika ukuran populasi relatif terlalu kecil untuk ruang pencarian. UNDX juga memiliki kesulitan dalam menemukan solusi optimal pada daerah dekat batas-batas ruang pencarian. TC menggunakan tiga orang tua untuk *constrained problems* (masalah yang dibatasi), satu orang tua tidak layak dan dua orang tua harus layak, hal ini digunakan agar dapat menghasilkan satu set dari tiga keturunan, kemudian dari setiap keturunan yang dihasilkan sebagai kombinasi linear dari tiga orang tua. TC bekerja dengan baik dimana solusi optimal terletak diperbatasan daerah yang mungkin dari masalah yang memiliki daerah layak tunggal yang dibatasi dalam domain kontinyu.

GA banyak digunakan untuk memecahkan masalah optimasi, walaupun pada kenyataannya juga memiliki kemampuan yang baik untuk masalah-masalah selain optimasi. John Holland menyatakan bahwa setiap masalah yang berbentuk adaptasi dapat diformulasikan dalam terminologi genetika. Algoritma genetika adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom [5]. GA juga salah satu metode optimisasi global yang bekerja dengan baik dan efisien pada fungsi tujuan yang kompleks dalam hal nonlinier, tetapi GA juga mempunyai masalah yaitu jika keturunan yang dihasilkan didistribusikan lebih sempit daripada orang tua mereka, mereka mungkin akan kehilangan keragaman dan akan mengalami konvergensi prematur. Sebaliknya, jika keturunannya didistribusikan sangat luas, mereka mungkin terlalu beragam dan memakan waktu yang terlalu lama supaya konvergen ke optimalitas yang ingin dicapai [4][6][7][8][9][10], ini disebabkan karena bergantungnya seleksi operator pada kualitas individu dengan hasil informasi gen yang cenderung pada individu terbaik yang mendominasi karakter pada populasi.

Beberapa peneliti telah mencoba melakukan uji coba menggunakan beberapa algoritma untuk menyelesaikan masalah konvergensi prematur di dalam GA, diantaranya yaitu dengan *parent centric crossover* (PCX) yang memungkinkan kemungkinan besar menciptakan solusi dekat setiap orang tua, bukan di dekat pusat orang tua, di mana untuk setiap keturunan satu orang tua dipilih dan vektor perbedaan dihitung antara orang tua dan orang tua m yang terpilih. PCX menerapkan pendekatan adaptif diri dimana vektor solusi baru terus bergerak menuju optimum [6]. Ketika PCX diterapkan dengan GA, dibutuhkan waktu yang lebih lama dibandingkan dengan operator *crossover* yang lain, dan menemukan kesulitan dalam memecahkan masalah multimodal yang dapat dipisahkan. *Deterministic annular crossover* [9] yang digunakan untuk memperkaya keturunan (*offspring*) hasil dari proses *crossover*, dengan operator seleksi deterministik, keragaman yang lebih besar antara individu-individu dari populasi dapat diperoleh karena menggunakan informasi genetik dari individu terburuk dengan probabilitas yang sama, sehingga melengkapi karakteristik individu-individu terbaik dari populasi. *Multi Parents Crossover* (MPC) [6][3] yang menggunakan tiga orang tua dalam proses *crossover* untuk menghindari keturunan (*offspring*) yang kurang beragam daripada orang tuanya [6].

Pada penelitian ini akan mengadopsi ide dari Ursem, et al.'s (1999) yaitu dengan mengurangi dan meningkatkan keragaman operator untuk mengontrol keragaman pada populasi. Oleh karena itu juga, melihat dari Mauldin (1984), keseimbangan yang tepat antara eksplorasi dan eksploitasi pencarian dapat dipertahankan dengan mengendalikan tingkat keragaman populasi [8]. Mekanisme kontrol dapat dibangun ke dalam GA, dan idenya adalah untuk mengontrol keragaman populasi dengan menyuntikkan *dynamic artificial chromosomes* (DAC) ke dalam sistem sampai ukuran keragaman mencapai tingkat tertentu kemudian berhenti. Selain itu juga akan digunakan *multi parent crossover* (MPC) untuk lebih memperkaya jenis individu yang akan diproses dalam *crossover* [6] sehingga bisa keluar dari lokal optimum (konvergensi prematur).

1.2 Masalah Penelitian

Berdasarkan latar belakang penelitian di atas maka masalah penelitian (*Research Problem (RO)*) ini adalah algoritma genetika sebagai salah satu algoritma yang bisa mengatasi masalah optimisasi secara global, tetapi mempunyai suatu masalah yaitu jika keturunan yang dihasilkan didistribusikan lebih sempit daripada orang tua mereka, mereka mungkin akan kehilangan keragaman dan akan mengalami konvergensi prematur.

1.3 Pertanyaan Penelitian

Pertanyaan penelitian (*Research Question (RQ)*) pada penelitian ini adalah bagaimana kromosom buatan dinamis (*dynamic artificial chromoshomes*) dan *multi parent crossover* mampu mengatasi masalah konvergensi prematur pada algoritma genetika?

1.4 Tujuan Penelitian

Tujuan penelitian (*Research Objective (RO)*) ini adalah mengintegrasikan kromosom buatan dinamis dan *multi parent crossover* untuk mengatasi masalah konvergensi prematur pada algoritma genetika.

1.5 Hubungan Antara Masalah Penelitian, Pertanyaan dan Tujuan

Berdasarkan pada gambaran pernyataan masalah sebelumnya, penelitian ini ditunjukkan di Tabel 1.1 yang terdiri dari 1 Masalah Penelitian, 1 Pertanyaan Penelitian, 1 Tujuan Penelitian.

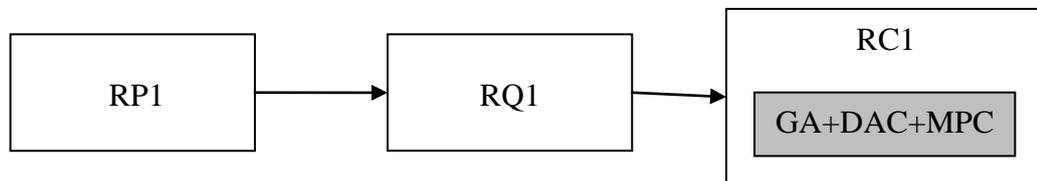
Tabel 1.1 Masalah Penelitian – Pertanyaan Penelitian – Tujuan Penelitian

Masalah Penelitian		Pertanyaan Penelitian		Tujuan Penelitian	
RP	Sulitnya mengatasi masalah konvergensi prematur pada algoritma genetika.	RQ	Bagaimana kromosom buatan dinamis dan <i>multi parent crossover</i> mampu mengatasi masalah konvergensi prematur pada algoritma genetika?	RO	Mengintegrasikan kromosom buatan dinamis dan <i>multi parent crossover</i> untuk mengatasi masalah konvergensi prematur pada algoritma genetika.

1.6 Kontribusi Penelitian

Mengikuti perbaikan metode yang dilakukan dapat mempertimbangkan Kontribusi Penelitian (*Research Contribution* (RC)) pada penelitian ini adalah memperbaiki kelemahan GA yaitu lokal optima (konvergensi prematur) dengan menggunakan DAC dan MPC ke dalam GA.

Gambar 1.1 menunjukkan hubungan antara Masalah Penelitian, Pertanyaan Penelitian dan Kontribusi Penelitian pada penelitian ini.



Gambar 1.1 Hubungan Masalah Penelitian, Pertanyaan Penelitian dan Kontribusi Penelitian

1.7 Sistematika Penulisan

Secara garis besar penulisan akan dibagi dalam beberapa bagian sebagai berikut.

BAB I : Pendahuluan

Menggambarkan latarbelakang penelitian, masalah penelitian, tujuan penelitian, kontribusi penelitian dan sistematika penulisan pada tesis.

BAB II : Landasan Teori

Menggambarkan hubungan landasan teori pada algoritma genetika, *dynamic artificial chromosomes* dan *multi parent crossover*.

BAB III : Metodologi Penelitian

Menggambarkan metodologi penelitian dan data set yang digunakan pada penelitian ini.

BAB IV : Hasil Percobaan

Menggambarkan diskusi pada hasil percobaan di penelitian ini.

BAB V : Kesimpulan

Bab ini membahas tentang kesimpulan dari pembahasan pada bab-bab sebelumnya dan saran-saran untuk penelitian selanjutnya.

BAB II

Landasan Teori

2.1 Tinjauan Studi

2.1.1 Model Elsayed et al. (2014)

GA dengan multi-parent crossover (GAMPC) [6] metode yang digunakan oleh Elsayed et al. untuk mengatasi masalah pada *constrained optimization problem* (COP). Didalam penelitian ini, diusulkan juga *diversity operator* untuk lebih lanjut membuat variasi pada pembangkitan *offspring*. Saat menerapkan *diversity operator*, dua poin harus dipertimbangkan: proporsi populasi menjalani operasi tersebut (berdasarkan probabilitas keragaman), dan skala kontribusi operator (berdasarkan informasi yang digunakan untuk menghasilkan keturunan akhir).

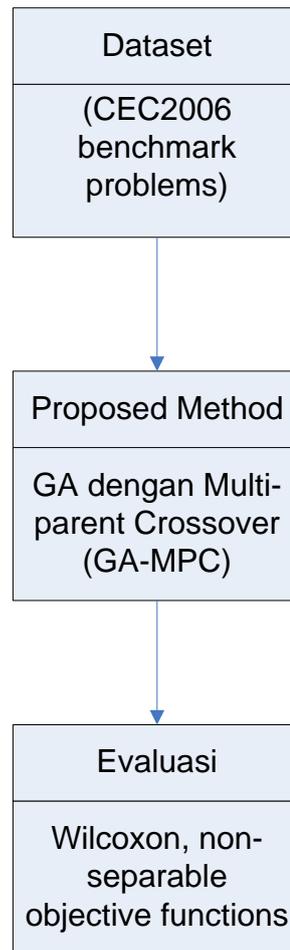
Dalam melaksanakan *diversity operator* yang diusulkan, digunakan informasi dari keturunan yang dihasilkan dan dari arsip kolam seleksi (A) yang berisi terbaik m individu dalam generasi sekarang. Keturunan akhir $o_i = \{o_1^i, o_2^i, \dots, o_j^i, \dots, o_i^D\}$ dihasilkan berdasarkan operator keragaman sebagai berikut dari pseudocode:

```
For  $i \in \{1;2;\dots;PS\}$ s
    For  $j \in \{1;2;\dots;D\}$ 
        If  $\text{rand}(0,1) < p$ 
            Select a random individual from the archive pool ( $A_{\text{arch}}$ ).
                
$$o_i^j = x_{\text{arch}}^j$$

        End
    End
End
```

Dimana p adalah *diversity probability*, o_i ke- i *offspring*, PS ukuran populasi dan $arch \in \{1,2,\dots, m\}$. Perhatikan bahwa untuk menghasilkan o_j akhir i , kita dapat menggunakan nilai yang berbeda untuk lengkungan, hal ini karena kita tidak ingin menggunakan informasi dari vektor tunggal sebagai yang dapat menyebabkan terjebak dalam *local minimum*.

Gambar 2.1 menunjukkan dari GAMPC yang telah diusulkan oleh Elsayed et al.



Gambar 2.1 Block diagram Elsayed et al. model

Pada penelitian ini Elsayed et al. menggunakan metode *multi parents crossover* (MPC), yaitu menggunakan 3 *parents* pada salah satu proses di dalam GA, yaitu pada proses *crossover*. Dimana pada saat proses *crossover* tersebut Elsayed et al. menggunakan 3 *parents* untuk membuat hasil individu yang diproses di dalam proses *crossover* menjadi keturunan (*offspring*) yang lebih beragam. Setelah dilakukan pengukuran dengan menggunakan metode pengukuran wilcoxon dan *non-separable objective functions*, menunjukkan bahwa GA-MPC dibandingkan dengan SBX-NU, PCX-NU dan APF-GA menghasilkan hasil yang melebihi dari metode tersebut, tetapi untuk GA-MPC dibandingkan dengan MDE dan ECHTEP-2 menghasilkan hasil yang relatif sama, tidak ada perbedaan yang signifikan.

2.1.2 Model Paves-Lazo et al. (2011)

Pada penelitian ini diusulkan *deterministic annular crossover* algoritma genetika (DACGA) sebagai metode untuk menyelesaikan salah satu masalah pada GA yaitu sulitnya mengatasi masalah konvergensi prematur pada algoritma genetika didalam *unit commitment problem* (UCP) [9].

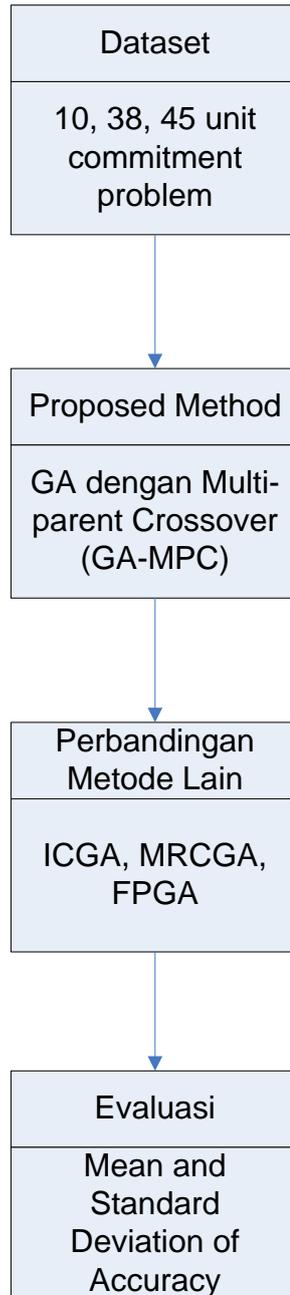
Deterministic berbasis seleksi digunakan untuk mencari nilai kecocokan (*fitness*) individual dengan kecocokan yang lebih tinggi, yang kemudian mempunyai kemungkinan yang lebih tinggi pula pada proses *crossover* untuk menggantikan individu yang buruk dengan individu yang lebih baik.

Annular crossover digunakan untuk melakukan proses pertukaran informasi genetik antara dua individu dengan operator *crossover* yang direpresentasikan dalam bentuk cincin. *Annular crossover* dilakukan dengan mendefinisikan sejumlah C_1 yang menunjukkan *crossover locus*. Jumlah ini adalah dalam rentang $[1, L-1]$, di mana L adalah panjang dari kromosom. Selanjutnya nomor harus ditetapkan untuk menetapkan panjang semi-ring C_s yang dipertukarkan selama *crossover*. Istilah semi-ring mengacu pada sektor cincin dengan panjang dalam rentang $[1, L/2]$. Untuk pertukaran informasi genetik untuk menjadi layak, panjang semi-ring harus sama di kedua kromosom[9].

Akhirnya, DACGA diusulkan diterapkan pada UCP menggunakan sistem uji 10, 38 dan 45 unit pembangkit. Penjadwalan waktu horizon dipilih sebagai salah satu hari dengan 24 interval setiap satu jam.

Gambar 2.2 menunjukkan secara garis besar penelitian dari Paves-Lazo et al. Dengan dataset yang digunakan adalah (10, 38, 45) *Unit Commitment Problem* (UCP), yang didalam penelitian yang dilakukan oleh Pavez-Lazo et al. ini menggunakan metode GA yang dikombinasikan dengan *deterministic annular crossover* untuk membuat keturunan yang lebih beragam pada saat proses *crossover* dengan cara menggunakan sistem seleksi agar nilai *fitness* pada individu yang ada di dalam *crossover* adalah nilai *fitness* yang paling tinggi. Pada Tabel 2.1 menunjukkan penelitian Pavez-Lazo et al. juga dibandingkan dengan hasil dari metode yang diusulkan, yaitu GA-MPC dengan metode penelitian lain sebelumnya, yaitu metode

Integer-Coded Genetic Algorithm (ICGA), Matrix Real-Coded Genetic Algorithm (MRCGA) dan Floating-Point Genetic Algorithm (FPGA). Dimana dari hasil evaluasi atau perbandingan yang dilakukan dengan menggunakan *Mean and Standard Deviation of Accuracy* menghasilkan bahwa nilai total biaya pada GA-MPC mendapatkan hasil paling rendah dibandingkan dengan metode ICGA, MRGA dan FPGA.



Gambar 2.2 Block diagram Paves-Lazo et al.

Tabel 2.1 Hasil Perbandingan dengan Metode Lain

	Total cost (\$)	Convergence (generations)
<i>10-Unit System</i>		
ICGA (Damousis et al., 2004)	566,404	300
MRCGA (Sun et al., 2006)	564,244	500
FPGA (Dang & Li, 2007)	564,094	>3000
DACGA	563,987	<300
<i>38 Unit System</i>		
MRGA (Sun et al., 2006)	206,000	1900
DACGA	195,042	<1500
<i>45- Unit System</i>		
PRGA (Arroyo & conejo, 2002)	1,029,557	200
DACGA	1,029,100	<2000

2.1.3 Summary of State-of-the-art on Genetic Algorithms

Pada Tabel 2.2 terdapat *State-of-the-art* penelitian yang sudah pernah dilakukan oleh para peneliti sebelumnya. Dalam penelitian sebelumnya, GA yang mengalami masalah di dalam mengatasi solusi lokal optima (konvergensi prematur) dilakukan dengan cara menambahkan jumlah *parents* pada proses *crossover* sehingga membuat keturunan (*offspring*) menjadi lebih beragam [6]. Selain itu juga ada yang menggunakan metode lain seperti *deterministic annular crossover genetic algorithm* yang menggunakan sistem semi-ring di dalam proses *crossover* untuk melakukan pertukaran informasi antara dua individu.

Tabel 2.2 *Summary of state-of-the-art on genetic algorithm*

Model	Research Problems	Proposed Method	Dataset	Measurement	Result
Elsayed et al. Model (2014)	GA mempunyai masalah, yaitu jika keturunan yang dihasilkan didistribusikan lebih sempit daripada orangtua mereka, mereka mungkin akan kehilangan keragaman dan akan mengalami konvergensi prematur	Multi-parent crossover (MPC)	CEC2006	Wilcoxon	(Parent/before/after diversity(%)) P1/12/34 P2/0.35/36.9 P3/34/85/4
Pavel-Lazo et al. Model (2011)	<i>One of the disadvantages of GA is premature convergence</i>	<i>deterministic annular crossover genetic algorithm (DACGA)</i>	10, 38, 45 unit commitment problem	<i>Mean and Standard Deviation of Accuracy</i>	<i>Mean and Standard Deviation of Accuracy: 83.12±4.82</i>
Rikzam Model (2014)	Sulitnya mengatasi masalah konvergensi prematur pada algoritma genetika	Kromosom buatan dinamis dan <i>multi parent crossover</i>	<i>Travelling salesman problem</i> (eil51, eil76, eil101, kroA100)	Wilcoxon	?

2.2 Tinjauan Pustaka

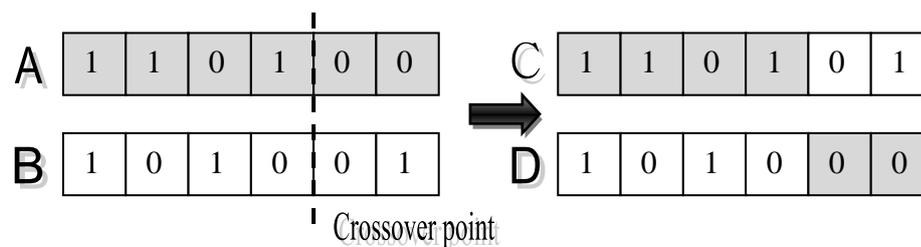
2.2.1 Algoritma Genetika

Algoritma Genetika (*Genetic algorithm (GA)*) adalah teknik yang paling populer dalam penelitian komputasi evolusi [5]. Tahap pertama untuk mendefinisikan GA adalah untuk menghubungkan antara dunia nyata ke dunia GA. Objek membentuk sebuah solusi yang mungkin dalam konteks masalah asli yang disebut sebagai *phenotype*, sedangkan pengkodean mereka disebut *genotype*. Dalam masalah kita, *phenotype* adalah vektor bilangan asli dengan angka mati untuk setiap baris[11]. Dalam algoritma genetika tradisional, representasi yang digunakan adalah bit string *fixed-length*. Setiap posisi dalam string diasumsikan untuk mewakili fitur tertentu

dari sebuah individu. Biasanya string dievaluasi sebagai kumpulan fitur struktural solusi yang memiliki sedikit atau tidak ada interaksi. Analogi dapat ditarik langsung ke gen dalam organisme biologis. Setiap gen merupakan entitas yang secara struktural independen dari gen lain.

Operator reproduksi utama yang digunakan adalah *bit-string crossover*, dimana dua string digunakan sebagai orang tua dan individu baru dibentuk dengan menukar sub-urutan antara dua string. Operator lain yang populer adalah *bit-flipping* mutasi, dimana satu bit dalam string membalik untuk membentuk keturunan string baru.

Gambar 2.3 berikut adalah gambar *bit-string crossover* pada *parents* a dan b ke *offspring* c & d.



Gambar 2.3 *bit-string crossover*

Kebanyakan orang melakukan pencarian terhadap solusi terbaik dalam serangkaian solusi tertentu. Ruang semua solusi (himpunan solusi di antaranya solusi yang diinginkan berada) layak disebut ruang pencarian (ruang keadaan). Setiap titik dalam ruang pencarian merupakan kemungkinan untuk menjadi salah satu solusi. Oleh karena itu setiap solusi yang mungkin dapat "ditandai" dengan nilai *fitness*-nya, tergantung definisi pada masalah algoritma genetika, dengan mencari satu solusi terbaik diantara sejumlah solusi yang mungkin diwakili oleh satu titik dalam ruang pencarian yaitu; GAs digunakan untuk mencari ruang pencarian untuk solusi terbaik misalnya minimal. Kesulitan dalam kemudahan ini adalah minima lokal dan titik awal dari pencarian.

Algoritma genetika menimbulkan beberapa fitur penting [5]. Pertama adalah *stochastic algorithm*, yaitu keacakan sebagai peran penting dalam algoritma genetika. Kemudian, seleksi dan reproduksi membutuhkan prosedur acak. Kedua adalah bahwa algoritma genetika selalu mempertimbangkan populasi solusi. Menjaga

memori lebih dari satu solusi pada setiap iterasi menawarkan banyak keuntungan. Algoritma dapat bergabung dengan solusi yang berbeda untuk mendapatkan solusi yang lebih baik dan sebagainya, dapat menggunakan manfaat dari keragaman populasi. Sebuah algoritma dasar populasi juga sangat memungkinkan untuk paralelisasi. Ketahanan algoritma juga harus disebutkan sebagai sesuatu yang penting bagi keberhasilan algoritma, karena mengacu pada kemampuan untuk melakukannya secara konsisten.

Algoritma genetika ini baik digunakan untuk berbagai jenis masalah. Karena tidak ada persyaratan khusus dalam menggunakan GA, sehingga dapat diterapkan untuk menyelesaikan masalah [5]. Algoritma genetika banyak digunakan untuk memecahkan masalah optimasi, walaupun pada kenyataannya juga memiliki kemampuan yang baik untuk masalah-masalah selain optimasi. John Holland menyatakan bahwa setiap masalah yang berbentuk adaptasi dapat diformulasikan dalam terminologi genetika. GA adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom.

2..1.1 Individu

Individu adalah satu buah solusi. Kelompok individu yang bersama-sama, mempunyai dua bentuk solusi seperti yang diberikan di bawah ini:

1. Kromosom, yang merupakan dasar informasi 'genetik' (*genotype*). Kromosom merupakan gabungan dari gen-gen yang membentuk nilai tertentu dan menyatakan solusi yang mungkin dari suatu permasalahan.
2. *Phenotype*, yang merupakan ekspresif kromosom dalam hal model.

Solution set Phenotype				
Factor 1	Factor 2	Factor 3	Factor N

Gambar 2.4 Representation of Genotype and phenotype

GA dimulai dengan set awal solusi acak untuk masalah yang sedang dipertimbangkan. Set solusi ini dikenal sebagai populasi. Individu-individu dari populasi disebut sebagai kromosom. Kromosom dari populasi dievaluasi sesuai dengan fungsi *fitness* yang telah ditetapkan [12]. Sebuah kromosom dibagi menjadi gen. Gen adalah representasi GA tentang faktor tunggal untuk faktor kontrol. Setiap

faktor dalam himpunan solusi sesuai dengan gen dalam kromosom. Pada Gambar 2.4 menunjukkan representasi dari *genotipe* dan *phenotype*.

Sebuah kromosom harus dalam beberapa cara yang berisi informasi tentang solusi yang mewakili. Fungsi morfogenesis asosiasi masing-masing *genotipe* dengan *phenotype*. Ini hanya berarti bahwa setiap kromosom harus menentukan satu solusi yang unik, tetapi tidak berarti bahwa setiap solusi dikodekan oleh tepat satu kromosom. Memang, fungsi morfogenesis tidak perlu bijective, dan bahkan kadang-kadang tidak mungkin (terutama dengan representasi biner). Namun demikian, fungsi morfogenesis setidaknya harus subyektif. Memang, semua solusi kandidat masalah harus sesuai dengan setidaknya satu kromosom yang mungkin, untuk memastikan bahwa seluruh ruang pencarian dapat dieksplorasi. Ketika fungsi morfogenesis yang mengaitkan setiap kromosom satu solusi adalah tidak injective, yaitu kromosom yang berbeda dapat mengkodekan solusi yang sama, representasi dikatakan merosot. Sedikit degenerasi tidak begitu mengkhawatirkan, bahkan jika ruang di mana algoritma ini mencari solusi optimal pasti diperbesar. Tapi jika degenerasi dianggap terlalu penting bisa menjadi masalah yang lebih serius. Hal ini akan mempengaruhi perilaku GA, terutama karena jika beberapa kromosom dapat mewakili *phenotype* yang sama, arti dari setiap gen akan jelas tidak sesuai dengan karakteristik spesifik dari solusi. Selain itu hal ini bisa saja akan menambahkan beberapa masalah yang lain, yaitu akan bertambahnya suatu jenis kebingungan dalam proses pencarian.

Pada Gambar 2.5 ini merepresentasikan sebuah kromosom yang dikodekan oleh bit string, dimana nantinya dari hasil bit string ini bisa diproses menjadi suatu nilai yang akan menjadi sebuah solusi.

1 0 1 0 1 0 1 1 1 1 0 1 0 1 1 0

Gambar 2.5 Representasi kromosom

2..1.2 Gen

Gen adalah dasar "instruksi" untuk membangun generik algoritma. Kromosom adalah suatu urutan gen. Gen mungkin menggambarkan kemungkinan solusi untuk masalah, tanpa benar-benar menjadi solusi. Sebuah gen adalah string bit panjang