



PENYELESAIAN MASALAH PEWARNAAN PADA GRAF DENGAN ALGORITMA GENETIKA

Skripsi

disusun sebagai salah satu syarat
untuk memperoleh gelar Sarjana Sains

Program Studi Matematika

UNNES

UNIVERSITAS NEGERI SEMARANG

oleh
Lana Aristya Anggraini

4111413002

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI SEMARANG
2017**

PERNYATAAN

Saya menyatakan skripsi ini bebas plagiat, dan apabila di kemudian hari terbukti terdapat plagiat dalam skripsi ini, maka saya bersedia menerima sanksi sesuai peraturan perundang-undangan.

Semarang, 12 Oktober 2017



Lana Aristya Anggraini

4111413002

UNNES
UNIVERSITAS NEGERI SEMARANG

PENGESAHAN

Skripsi yang berjudul

Penyelesaian Masalah Pewarnaan Graf dengan Algoritma Genetika

disusun oleh

Lana Aristya Anggraini

4111413002

telah dipertahankan di hadapan sidang Panitia Ujian Skripsi FMIPA UNNES pada tanggal 16 Oktober 2017.



Prof. Dr. Zenuri, S.E., M.Si. Akt.
NIP. 196412231988031001

Sekretaris

Drs. Arief Agoestanto, M.Si.
NIP. 196807221993031005

Ketua Penguji

Prof. Dr. St. Budi Waluyo, M. Si.
196809071993031002

Anggota Penguji/
Pembimbing I

Dr. Isnaini Rosyida, M.Si.
NIP. 197302191998022001

Anggota Penguji/
Pembimbing II

Dr. Tri Sri Noor Asih, M.Si.
197706142008122002

MOTTO DAN PERSEMBAHAN

MOTTO

- Jadikanlah sabar dan sholat sebagai penolongmu.
- Ketika kamu memudahkan urusan orang lain, maka Allah akan memudahkan urusanmu.
- Tidak melulu tentang hasil, tidak ada proses yang tidak mendewasakanmu.

PERSEMBAHAN

1. Ibu dan Bapak tercinta, Ibu Ida dan Bapak Bambang.
2. Kakak-kakakku tersayang.
3. Sahabat-sahabat Matematika Murni 2013
4. Teman teman yang selalu mensupport.

UNNES
UNIVERSITAS NEGERI SEMARANG

PRAKATA

Puji syukur senantiasa penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penyelesaian Masalah Graf dengan Algoritma Genetika”.

Penulis menyadari dalam penyusunan skripsi ini penulis telah mendapat banyak bantuan, bimbingan, dan dorongan dari berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Dr. Fathur Rokhman, M.Hum., Rektor Universitas Negeri Semarang.
2. Prof. Dr. Zaenuri, S.E., M.Si., Akt., Dekan FMIPA Universitas Negeri Semarang.
3. Drs. Arief Agoestanto, M.Si., Ketua Jurusan Matematika FMIPA Universitas Negeri Semarang.
4. Drs. Mashuri, M.Si., Koordinator Program Studi Matematika FMIPA Universitas Negeri Semarang.
5. Drs. Isnaini Rosyida, M.Si., selaku Dosen Pembimbing I yang telah memberikan bimbingan, pengarahan, nasehat, dan saran selama penyusunan skripsi ini.
6. Drs. Tri Sri Noor Asih, M.Si., selaku Dosen Pembimbing II yang telah memberikan bimbingan, pengarahan, nasehat, dan saran selama penyusunan skripsi ini.
7. Seluruh Dosen Matematika yang telah membimbing dan memberikan ilmunya kepada penulis.

8. Kedua Orang Tua, Ibu Ida dan Bapak Bambang yang senantiasa memberikan semangat serta doa yang tidak ada putusnya.
9. Kakak-kakakku, Reni Hapsari, Gilang Rangga, Wisnu Wirawan, Sasha Shakuntala dan Nesia Putri Marina yang selalu memberikan support di hidup saya.
10. Sahabat-sahabat saya, Ary Sulistya, Desca Nur, Tamara Arindita, Tiara Budi dan Irvan Kurniawan yang telah memberikan semangat, dorongan, bantuan, saran serta motivasi terkait penyusunan skripsi ini.
11. Sahabat dan pendengar setia, Suparmi, Windasari, Farida Rahmawati, yang selalu menguatkan selama mereka proses perkuliahan dan penyusunan skripsi.
12. Kawan-kawan Jurusan Matematika yang memberikan dorongan untuk selalu semangat dalam bimbingan skripsi ini.
13. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu terselesaikannya skripsi ini.

Penulis menyadari, bahwa masih banyak keterbatasan pengetahuan dan kemampuan yang penulis miliki. Penulis mengharapkan kritik dan saran yang bisa membangun penelitian-penelitian yang lain. Semoga skripsi ini dapat berguna dan bermanfaat bagi pembaca.

Semarang, 12 Oktober 2017

Penulis

ABSTRAK

Anggraini, Lana Aristya. 2017. Penyelesaian Masalah Pewarnaan Graf dengan Algoritma Genetika. Skripsi, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Semarang, Pembimbing 1 Drs. Isnaini Rosyida, M.Si. dan Pembimbing 2 Drs. Tri Sri Noor Asih, M.Si.

Kata Kunci : Pewarnaan Graf, Metode *heuristic*, Algoritma Genetika.

Pada penelitian ini, dijelaskan langkah-langkah matematis tentang penyelesaian masalah pewarnaan graf (*graph colouring*) dengan menggunakan Algoritma Genetika. Langkah – langkah tersebut meliputi konstruksi nilai *fitness*, proses *crossover*, dan proses mutasi pada Algoritma Genetika untuk masalah pewarnaan graf. Pewarnaan pada graf umumnya menggunakan Algoritma Welsh-Powell. Namun seiring berkembangnya ilmu pengetahuan, metode *heuristic* digunakan untuk mewarnai graf. Untuk menyelesaikan masalah pewarnaan graf dengan Algoritma Genetika, dilakukan pengkodean kromosom berbentuk *array*. Kemudian kromosom tersebut dikenakan operator seleksi dengan metode roda *roullet*, *crossover* satu titik dan mutasi satu gen sehingga menjadi populasi baru. Populasi baru yang terbentuk kemudian dievaluasi dengan konstruksi nilai *fitness* yang dibangun untuk meminimalisir kesalahan pewarnaan dan menemukan minimal warna. Proses tersebut dilakukan hingga didapatkan generasi yang memuat penyelesaian pewarnaan graf. Penyelesaian pewarnaan graf merupakan pelabelan titik dengan minimal warna dan nol kesalahan pewarnaan. Bilangan yang menyatakan minimal warna yang digunakan dalam pewarnaan titik graf disebut bilangan kromatik.



DAFTAR ISI

Halaman

PERNYATAAN.....	Error! Bookmark not defined.
PENGESAHAN	iii
MOTTO DAN PERSEMBAHAN	iv
PRAKATA	v
ABSTRAK	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN.....	xvi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	4
1.3 Tujuan Penelitian.....	4
1.4 Batasan Penelitian	4
1.5 Manfaat Penelitian.....	5
1.6 Sistematika Penulisan Skripsi	5
BAB II TINJAUAN PUSTAKA.....	8

2.1 Graf.....	8
2.2 Jenis – Jenis Graf.....	9
2.2.1 Graf Berdasar Ada Tidaknya Sisi Paralel atau Loop	10
2.2.1.1 Graf Sederhana	10
2.2.1.2 Graf Tak Sederhana (Unsimple Graph).....	10
2.2.2 Graf berdasarkan orientasi arah atau panah	11
2.2.2.1 Graf tak-berarah (undirected graph)	11
2.2.2.2 Graf Berarah	12
2.2.3 Graf berdasarkan jumlah titik pada suatu graf	12
2.2.3.1 Graf berhingga (limited graph).....	13
2.2.3.2 Graf tak-berhingga (unlimited graph).....	13
2.3 Istilah dalam Graf.....	13
2.3.1 Titik Terpencil (Isolated Vertex).....	14
2.3.2 Graf Kosong (Null Graph).....	14
2.3.3 Derajat (Degree)	15
2.4 Pewarnaan Graf.....	15
2.5 Algoritma Genetika	16
2.5.1 Komponen-Komponen Algoritma Genetika	19
2.5.2 Kontrol Parameter Algoritma Genetika	29

2.5.3 Penerapan Algoritma Genetika untuk Pencarian Nilai Maksimum	
Polinomial	29
2.6 MATLAB	37
2.7 Window-Window pada MATLAB.....	37
BAB III METODE PENELITIAN.....	39
3.1 Studi Pustaka	41
3.2 Penemuan Masalah.....	41
3.3 Rumusan Masalah	41
3.4 Pemecahan Masalah	42
3.5 Penarikan Kesimpulan.....	43
BAB 4 HASIL PENELITIAN DAN PEMBAHASAN	44
4.1 Konstruksi Fitness pada Algoritma Genetika untuk Pewarnaan Graf.....	44
4.2 Proses Seleksi, Crossover dan Mutasi.....	46
4.2.1 Seleksi	46
4.2.2 Crossover.....	46
4.2.3 Mutasi.....	47
4.3 Algoritma Genetika untuk Pewarnaan Graf	47
4.3.1 Simulasi Penyelesaian Graf dengan Algoritma Genetika I.....	49
4.3.2 Simulasi Penyelesaian Graf dengan Algoritma Genetika II.....	60

BAB 5 PENUTUP.....	71
5.1 Kesimpulan.....	71
5.2 Saran.....	72
DAFTAR PUSTAKA	73
LAMPIRAN.....	75



DAFTAR TABEL

Tabel	Halaman
Tabel 2.1 Populasi Awal	33
Tabel 2.2 Seleksi Roda Roulette	33
Tabel 2.3 Pemilihan Induk Crossover	34
Tabel 2.4 Proses Crossover Satu Titik	35
Tabel 2.5 Proses Mutasi Satu Gen	35
Tabel 2.6 Populasi Baru	35
Tabel 2.7 Data Hasil Pengujian Tujuh Generasi	36
Tabel 4.1 Seleksi Roda Roulette	46
Tabel 4.2 Populasi Pertama	50
Tabel 4.3 Proses Evaluasi Kromosom	50
Tabel 4.4 Proses Seleksi dengan Metode Roda Roulette	52
Tabel 4.5 Pemilihan Induk Crossover dengan Probabilitas 60%	53
Tabel 4.6 Proses Crossover	53
Tabel 4.7 Proses Mutasi dengan Probabilitas Mutasi sebesar 1%	54
Tabel 4.8 Proses Evaluasi Populasi Baru	55
Tabel 4.9 Populasi Baru Generasi Ketiga	56

Tabel 4.10 Populasi Baru Generasi Keempat	57
Tabel 4.11 Populasi Baru Generasi Kelima	58
Tabel 4.12 Populasi Pertama.....	61
Tabel 4.13 Evaluasi Populasi Pertama	61
Tabel 4.14 Proses Seleksi dengan Metode Roda Roulette	63
Tabel 4.15 Pemilihan Induk Crossover dengan Probabilitas 60%.....	64
Tabel 4.16 Proses Crossover	64
Tabel 4.18 Evaluasi Populasi Baru	66
Tabel 4.19 Populasi Baru Generasi Ketiga	67
Tabel 4.20 Populasi Baru Generasi Keempat	68
Tabel 4.21 Populasi Baru Generasi Kelima	68
Tabel 4.22 Populasi Baru Generasi Keenam.....	69



DAFTAR GAMBAR

Gambar	Halaman
Gambar 2.1 Graf Bertetangga	9
Gambar 2.2 Graf Bersisian.....	9
Gambar 2.3 Graf sederhana.....	10
Gambar 2.4 Graf Ganda	11
Gambar 2.5 Graf Semu	11
Gambar 2.6 Graf tak berarah.....	12
Gambar 2.7 Graf berarah.....	12
Gambar 2.8 Graf Berhingga	13
Gambar 2.9 Graf Tak-Berhingga	13
Gambar 2.10 Graf dengan Titik Terpencil.....	14
Gambar 2.11 Graf Kosong	14
Gambar 2.12 Pewarnaan Titik.....	15
dengan 3 Warna.....	15
Gambar 2.13 Pewarnaan Titik.....	15
dengan 4 Warna.....	15
Gambar 2.15 Grafik fungsi $p(x)$	40
Gambar 3.1 Diagram Alir Penelitian	40
Gambar 4.1 Graf G	45

Gambar 4.2 Graf G pada Simulasi 1	49
Gambar 4.3 Pewarnaan Graf G pada Simulasi 1 dengan Tiga Warna	59
Gambar 4.4 Graf G pada Simulasi 2	60
Gambar 4.5 Pewarnaan Graf G pada Simulasi 2 dengan Empat Warna	70



DAFTAR LAMPIRAN

Lampiran	Halaman
1. Pewarnaan graf dengan Algoritma Genetika pada Simulasi 1.....	76
2. Pewarnaan graf dengan Algoritma Genetika pada Simulasi 2.....	85
3. Source code evaluasi fitness pada editor MATLAB.....	97
4. Source code crossover satu titik pada editor MATLAB.....	98
5. Source code mutasi pada editor MATLAB.....	99
6. Evaluasi fitness dengan bantuan MATLAB.....	99
7. Crossover dengan bantuan MATLAB.....	100
8. Mutasi dengan bantuan MATLAB.....	100



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Teori graf merupakan salah satu cabang dalam matematika diskrit yang menarik untuk dibahas karena berkaitan dengan permasalahan yang banyak ditemui dalam kehidupan sehari-hari (Wibisono, 2008). Teori graf mulai dikenalkan oleh seorang matematikawan bernama Leonhard Euler sekitar tahun 1736. Leonhard Euler mulai mengenalkan teori graf setelah menyelesaikan masalah jembatan Koinberg. Masalah tersebut kemudian dimodelkan Euler dalam bentuk graf dengan memisalkan daratan sebagai sebuah titik dan jembatan penghubungnya adalah sebuah sisi. Keunikan teori graf adalah kesederhanaan pokok bahasan yang dipelajarinya, karena dapat disajikan sebagai titik (*vertex*) dan sisi (*edge*) (Jusuf, 2009). Salah satu bagian dari teori graf adalah pewarnaan graf.

Pewarnaan graf adalah teknik pemberian warna pada elemen graf yang akan dijadikan subjek dalam memahami *constraint* permasalahan. Ada tiga macam persoalan pewarnaan graf (*graph colouring*), yaitu pewarnaan titik (*vertex*), pewarnaan sisi (*edge*), dan pewarnaan wilayah (*region*). Pada teknik pewarnaan graf, tidak hanya sekedar mewarnai titik dan sisi dengan warna yang berbeda. Tetapi diharapkan warna yang diperoleh adalah warna minimum.

Algoritma yang digunakan untuk mewarnai graf yaitu Algoritma Pewarnaan Barisan-Sederhana (*The Simple Sequential Colouring Problem*) dan Algoritma Pewarnaan Barisan-Besar Utama yang lebih dikenal dengan Algoritma Welch

Powell. Kedua algoritma tersebut hanyalah pendekatan dan tidak menjamin diperolehnya banyak warna dengan warna minimum (Budayasa, 2007:165). Langkah pengerjaan Algoritma Pewarnaan Barisan-Sederhana dimulai dengan melabeli titik-titik graf dengan v_1, v_2, \dots, v_n . Pewarnaan mulai dari titik v_1 dengan warna 1. Selanjutnya diwarnai dengan warna yang sama jika titik tidak bertetangga dan diwarnai dengan warna berbeda jika titik bertetangga. Pengerjaan dilakukan secara berurutan mulai dari pelabelan v_1, v_2, \dots, v_n . Perbedaannya dengan Algoritma Welch Powell adalah pengerjaan dilakukan urut dari derajat terbesar sampai yang terkecil. Kemudian mewarnai titik yang mempunyai derajat terbesar dengan warna 1 dan mewarnai titik dengan derajat yang lebih kecil dengan warna yang sama jika tidak bertetangga dan warna yang berbeda jika bertetangga. Dilakukan sampai derajat terkecil dan mendapatkan bilangan kromatik.

Seiring berkembangnya ilmu pengetahuan metode *heuristic* mulai digunakan untuk mewarnai graf. Metode *heuristic* merupakan suatu penyelesaian yang menggunakan konsep pendekatan. Pendekatan *heuristic* menggunakan suatu algoritma secara interaktif sehingga menghasilkan solusi yang mendekati optimal (Albar, 2013). Metode *heuristic* yang sudah digunakan untuk mewarnai graf adalah Algoritma Genetika, Tabu Search, dan Algoritma Semut (*Ant Colony*). Algoritma Genetika pernah digunakan oleh Gwee, Lim, dan Ho (1993) untuk mewarnai peta dengan menggunakan empat warna. Berikutnya Fleurent dan Ferland (1996) membandingkan Algoritma Genetika dengan Algoritma Hibrida pada pewarnaan graf. Choiritu dan Adriana (2002) meneliti penerapan Algoritma Genetika yang digabung dengan Tabu Search untuk menyelesaikan masalah pewarnaan graf. Glass

dan Bennet (2003) meneliti bilangan kromatik yang diperoleh dari Algoritma Genetika dan Algoritma Tabu Search. Shen (2003) meneliti pewarnaan graf dengan Genetika *Programming*. Hindi (2012) pewarnaan graf pada data DIMACS. Barod, Hawanna dan Jagtap (2014) membandingkan Algoritma Genetika dan Algoritma Memetika untuk pewarnaan graf. Algoritma Tabu Search digunakan oleh Hertz dan Werra (1987) untuk mencari teknik tabu search pada pewarnaan graf. Wulan (2015) meneliti aplikasi pewarnaan graf pada penjadwalan kereta api. Algoritma Semut digunakan oleh Costa dan Hertz (1997), dan Adeputra (2012) pada kasus pewarnaan graf.

Algoritma Genetika adalah suatu algoritma pencarian yang berbasis pada mekanisme seleksi alam dan genetika (Desiani, 2006:187). Pada Algoritma Genetika proses pencarian bersifat acak sehingga pemilihan operator yang digunakan sangat menentukan keberhasilan Algoritma Genetika dalam menemukan solusi optimum. Hindi (2012) telah meneliti Algoritma Genetika untuk pewarnaan pada Graf DIMACS Pada jurnal tersebut dijelaskan tentang Algoritma Genetika yang diterapkan pada program komputer. Hindi (2012) menjelaskan bahwa Algoritma Genetika sangat efisien untuk menyelesaikan masalah pewarnaan graf. Kekurangan pada jurnal tersebut adalah kurangnya penjelasan langkah-langkah untuk mewarnai suatu graf dengan Algoritma Genetika. Proses seleksi hingga pemilihan operator *crossover*, mutasi dan bagaimana evaluasi *fitness* tidak dijelaskan.

Pada penelitian ini penulis akan membahas mengenai langkah-langkah penerapan Algoritma Genetika untuk menyelesaikan masalah pewarnaan pada graf secara runtut dan pembuatan perancangan program untuk operator *crossover*, mutasi dan evaluasi *fitness* dengan bantuan *software* MATLAB.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, permasalahan yang akan diangkat pada penelitian ini adalah sebagai berikut.

1. Bagaimana mengkonstruksi *fitness* pada Algoritma Genetika dalam pewarnaan graf?
2. Bagaimana proses seleksi, *crossover*, dan mutasi pada Algoritma Genetika dalam pewarnaan graf?
3. Bagaimana langkah-langkah penyelesaian masalah pewarnaan graf dengan Algoritma Genetika?

1.3 Tujuan Penelitian

Adapun tujuan pada penelitian ini adalah sebagai berikut.

1. Mengkonstruksi *fitness* pada Algoritma Genetika dalam pewarnaan graf.
2. Mengetahui proses seleksi, *crossover*, dan mutasi pada Algoritma Genetika dalam pewarnaan graf.
3. Mengetahui langkah-langkah penyelesaian masalah pewarnaan graf dengan Algoritma Genetika.

1.4 Batasan Penelitian

Agar permasalahan tidak meluas, penulis membatasi ruang lingkup penelitian yaitu.

1. Pembahasan tentang penerapan Algoritma Genetika hanya untuk menyelesaikan pewarnaan pada graf.
2. *Crossover* yang dilakukan menggunakan *crossover* satu titik tanpa proses *random*.

1.5 Manfaat Penelitian

Adapun manfaat yang dapat diambil dari penelitian ini adalah sebagai berikut.

1. Bagi Peneliti

Manfaat yang bisa diambil bagi peneliti adalah peneliti mampu mengembangkan ilmunya, terutama dalam hal pewarnaan graf menggunakan Algoritma Genetika.

2. Bagi Universitas

Berdasarkan hasil penelitian manfaat bagi universitas adalah dapat menjadi bahan referensi yang berkaitan dengan teori graf.

1.6 Sistematika Penulisan Skripsi

Secara garis besar skripsi ini dibagi menjadi tiga bagian (bab) yaitu bagian awal skripsi, bagian isi skripsi, dan bagian akhir skripsi. Berikut ini dijelaskan masing-masing bagian skripsi.

1. Bagian awal skripsi

Bagian awal skripsi meliputi halaman judul, pernyataan keaslian tulisan, pengesahan, motto dan persembahan, prakata, abstrak, daftar isi, daftar gambar, daftar tabel, dan daftar lampiran.

2. Bagian isi skripsi

Bagian isi skripsi secara garis besar terdiri dari lima bab, yaitu sebagai berikut.

BAB 1. PENDAHULUAN

Bab ini berisi mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan skripsi.

BAB 2. TINJAUAN PUSTAKA

Bab ini berisi kajian teori yang mendasari dan berhubungan dengan pemecahan masalah. Teori-teori tersebut digunakan untuk memecahkan masalah yang diangkat dalam skripsi ini. Teori yang digunakan adalah Definisi Graf, Jenis-Jenis Graf, Istilah pada Graf, Pewarnaan Graf, dan Algoritma Genetika.

BAB 3. METODE PENELITIAN

Bab ini mengulas metode yang digunakan dalam penelitian yang berisi langkah-langkah yang dilakukan untuk memecahkan masalah yaitu studi pustaka, penemuan masalah, rumusan masalah, pemecahan masalah, kesimpulan dan saran.

BAB 4. HASIL PENELITIAN DAN PEMBAHASAN

Berisi hasil penelitian dan pembahasan sebagai jawaban atas permasalahan yang diungkapkan.

BAB 5. PENUTUP

Bab ini berisi tentang simpulan dari pembahasan dan saran yang berkaitan dengan simpulan.

3. Bagian akhir skripsi

Bagian akhir skripsi meliputi daftar pustaka yang memberikan informasi tentang buku sumber serta literatur yang digunakan dan lampiran-lampiran yang mendukung skripsi.



BAB II

TINJAUAN PUSTAKA

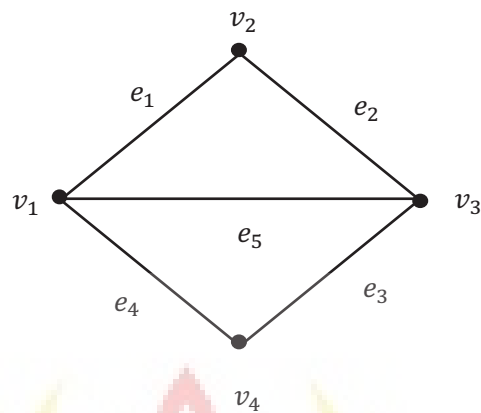
2.1 Graf

Sebuah graf G berisikan dua himpunan yaitu himpunan berhingga tak kosong $V(G)$ dari obyek-obyek yang disebut titik (vertex) dan himpunan berhingga (mungkin kosong) $E(G)$ yang elemen-elemennya disebut sisi (edge) sedemikian hingga setiap elemen e dalam $E(G)$ merupakan pasangan tak berurutan dari titik-titik di $V(G)$. Himpunan $V(G)$ disebut himpunan titik dan himpunan $E(G)$ disebut himpunan sisi (Budayasa, 2007: 1).

Cara umum yang digunakan untuk merepresentasikan sebuah graf adalah dengan diagram/gambar. Dalam diagram tersebut, titik-titik pada graf dinyatakan sebagai noktah sedangkan sisi dinyatakan sebagai ruas garis yang menghubungkan dua titik.

Pada graf terdapat istilah yang sering dijumpai, yaitu ketetanggaan (*adjacency*) dan bersisian (*incident*). Dua buah titik pada graf tak berarah G dikatakan bertetangga bila keduanya terhubung langsung dengan sebuah sisi. Dengan kata lain v_j bertetangga dengan v_k jika (v_j, v_k) adalah sebuah sisi pada graf G . Contoh graf bertetangga dapat dilihat pada Gambar 2.1. Titik v_1 bertetangga dengan titik v_2 dan v_4 , titik v_2 tidak bertetangga dengan titik v_4 .

Contoh.

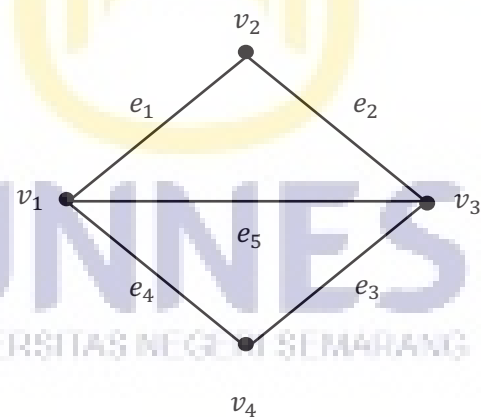


Gambar 2.1 Graf Bertetangga

Sedangkan, untuk sembarang sisi $e = (v_j, v_k)$ sisi e dikatakan bersisian dengan titik v_j dan titik v_k . Contoh konsep bersisian dapat dilihat pada Gambar 2.2.

Sisi e_1 bersisian dengan titik v_1 dan v_2 .

Contoh.



Gambar 2.2 Graf Bersisian

2.2 Jenis – Jenis Graf

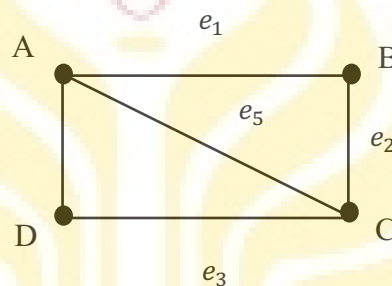
Graf dapat dikelompokkan berdasarkan ada tidaknya sisi yang paralel atau loop, jumlah titiknya, berdasarkan ada tidaknya arah pada sisinya, ada tidaknya bobot pada sisi nya, atau ada tidaknya hubungan dengan graf yang lain.

2.2.1 Graf Berdasar Ada Tidaknya Sisi Paralel atau Loop

Berdasarkan ada tidaknya sisi yang paralel atau loop pada suatu graf, graf dapat digolongkan sebagai berikut: graf sederhana dan graf tak sederhana.

2.2.1.1 Graf Sederhana

Graf sederhana adalah graf yang tidak mempunyai sisi ganda dan atau loop. Loop adalah sisi yang menghubungkan sebuah titik dengan dirinya sendiri. Berikut adalah contoh graf sederhana :



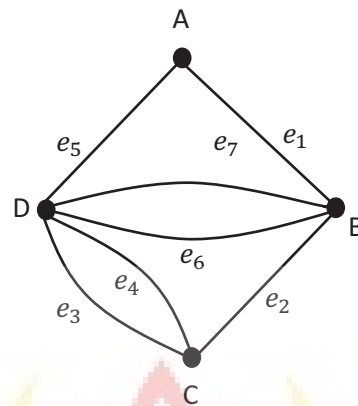
Gambar 2.3 Graf sederhana

2.2.1.2 Graf Tak Sederhana (*Unsimple Graph*)

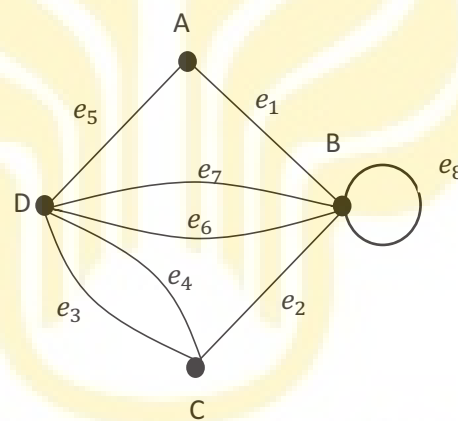
Graf tak sederhana adalah graf yang mengandung sisi ganda atau loop (Munir, 2005: 357). Graf tak sederhana dibagi dua macam, yaitu graf ganda (*multigraph*) dan graf semu (*pseudograph*).

- i. Graf ganda (*Multigraph*), adalah graf yang mengandung sisi ganda. Sisi ganda yang menghubungkan sepasang titik bisa lebih dari dua buah.
- ii. Graf semu (*Pseudograph*), adalah graf yang mempunyai loop (termasuk juga graf yang mempunyai loop dan sisi ganda). Graf semu lebih umum daripada graf ganda, karena graf semu sisi-nya dapat terhubung dengan dirinya sendiri.

Contoh.



Gambar 2.4 Graf Ganda



Gambar 2.5 Graf Semu

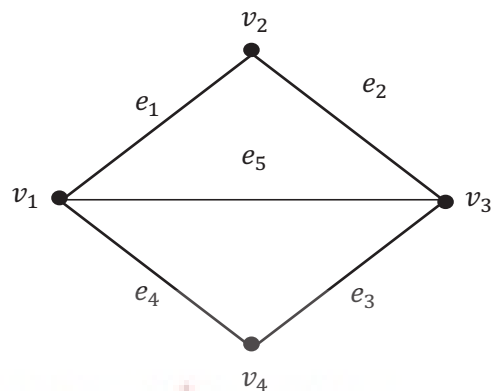
2.2.2 Graf berdasarkan orientasi arah atau panah

Selain berdasarkan ada tidaknya sisi yang paralel atau loop, graf dapat juga dikelompokkan berdasarkan orientasi arah atau panah, yaitu: graf berarah dan graf tak berarah.

2.2.2.1 Graf tak-berarah (*undirected graph*)

Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak berarah. Pada graf tak berarah, urutan pasangan titik yang dihubungkan oleh sisi tidak diperhatikan. Jadi, $(v_j, v_k) = (v_k, v_j)$.

Contoh.

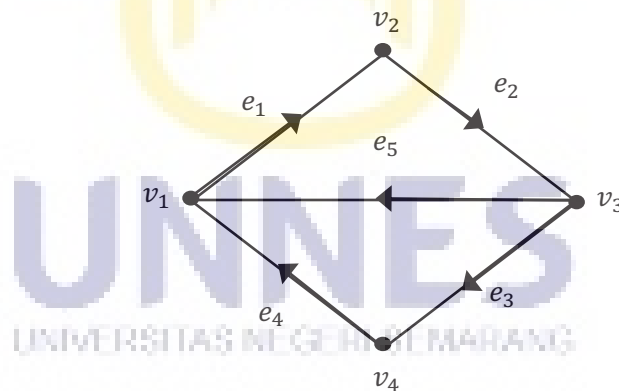


Gambar 2.6 Graf tak berarah

2.2.2.2 Graf Berarah

Graf yang setiap sisi nya memiliki orientasi arah atau panah. Pada graf berarah, (v_j, v_k) dan (v_k, v_j) menyatakan dua busur yang berbeda. Dengan kata lain $(v_j, v_k) \neq (v_k, v_j)$.

Contoh.



Gambar 2.7 Graf berarah

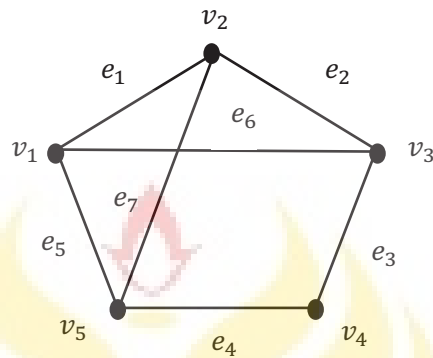
2.2.3 Graf berdasarkan jumlah titik pada suatu graf

Berdasarkan jumlah titik pada suatu graf, maka secara umum graf dapat digolongkan menjadi dua jenis, yaitu: graf berhingga dan graf tak berhingga.

2.2.3.1 Graf berhingga (*limited graph*)

Graf berhingga adalah graf yang jumlah titiknya n berhingga.

Contoh.



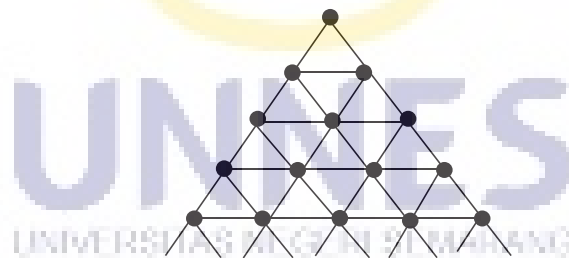
Gambar 2.8 Graf Berhingga

Pada Gambar 2.8, graf G memiliki titik sejumlah 5.

2.2.3.2 Graf tak-berhingga (*unlimited graph*).

Graf tak-berhingga adalah graf yang jumlah titiknya n tidak berhingga.

Contoh.



Gambar 2.9 Graf Tak-Berhingga

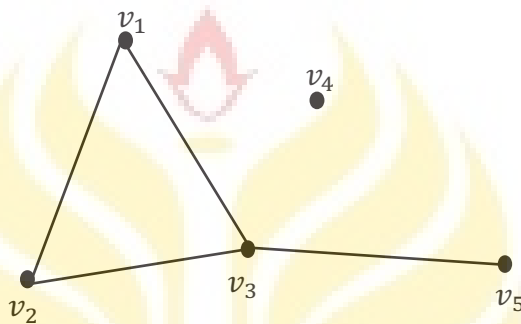
2.3 Istilah dalam Graf

Dalam pembahasan mengenai graf biasanya sering menggunakan terminologi (istilah) yang berkaitan dengan graf (Munir, 2005: 365). Terminologi yang berkaitan dengan graf adalah sebagai berikut.

2.3.1 Titik Terpencil (*Isolated Vertex*)

Titik yang tidak mempunyai sisi yang bersisian dengannya atau dapat juga dinyatakan bahwa titik terpencil adalah titik yang tidak satupun bertetangga dengan titik-titik lainnya. Contoh graf titik terpencil yaitu titik v_4 dapat dilihat pada Gambar 2.10.

Contoh.

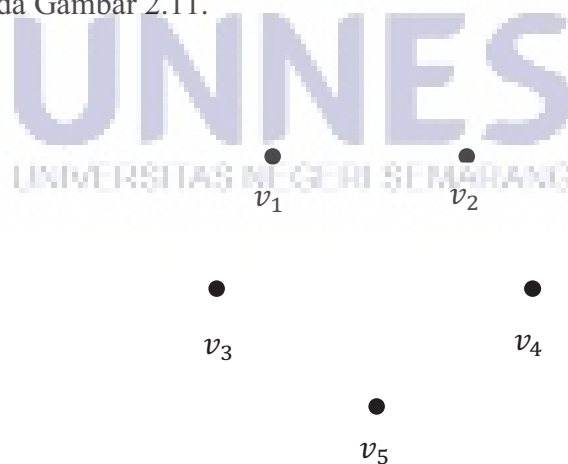


Gambar 2.10 Graf dengan Titik Terpencil

2.3.2 Graf Kosong (*Null Graph*)

Graf yang himpunan sisinya merupakan himpunan kosong disebut graf kosong dan ditulis sebagai N_n dengan n adalah jumlah titik. Contoh graf kosong dapat dilihat pada Gambar 2.11.

Contoh.



Gambar 2.11 Graf Kosong

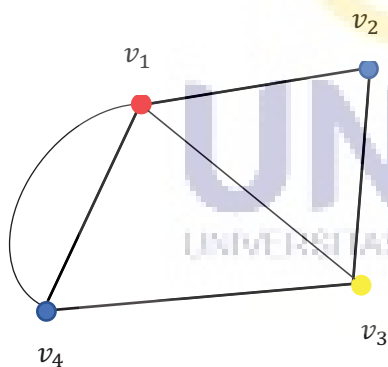
2.3.3 Derajat (*Degree*)

Derajat suatu titik pada graf tak berarah adalah jumlah sisi yang bersisian dengan titik tersebut. Titik yang mengandung loop menyumbang dua derajat.

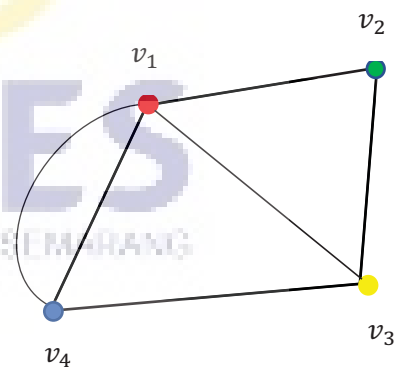
2.4 Pewarnaan Graf

Menurut Budayasa (2007: 151) ada dua macam pewarnaan graf, yaitu pewarnaan titik (*vertex*) dan pewarnaan sisi (*edge*). Misal G sebuah graf. Sebuah pewarnaan k dari G adalah pewarnaan semua titik G dengan menggunakan k warna sedemikian hingga dua titik G yang berhubungan langsung mendapat warna yang berbeda. Sebuah pewarnaan sisi pada graf G adalah pewarnaan semua sisi G sedemikian hingga setiap dua sisi yang terkait pada titik yang sama mendapatkan warna yang berbeda. Bilangan yang menyatakan banyaknya warna minimal yang digunakan dalam pewarnaan titik graf disebut bilangan kromatik (Rosyida, 2016).

Contoh.



Gambar 2.12 Pewarnaan Titik
dengan 3 Warna



Gambar 2.13 Pewarnaan Titik
dengan 4 Warna

Bilangan kromatik pada graf G adalah 3.

2.5 Algoritma Genetika

Algoritma Genetika menurut Desiani (2006) adalah suatu algoritma pencarian yang berbasis pada mekanisme, seleksi alam, dan genetika. Algoritma Genetika merupakan salah satu algoritma yang sangat tepat digunakan dalam menyelesaikan masalah optimasi kompleks, yang sulit dilakukan oleh metode konvensional.

Beberapa definisi penting dalam Algoritma Genetika yaitu:

1. Gen (*Genotype*) adalah sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom.
2. Allel adalah nilai dari gen.
3. Kromosom adalah gabungan gen-gen yang membentuk nilai tertentu.
4. Individu menyatakan suatu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
5. Populasi merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.
6. Generasi menyatakan satu satuan siklus proses evolusi.
7. Nilai *fitness* menyatakan seberapa baik nilai dari suatu individu atau solusi yang didapatkan.

Secara umum, Thiang, dkk (2001) mengemukakan bahwa struktur dari suatu Algoritma Genetika dapat didefinisikan dengan langkah-langkah sebagai berikut:

1. Membangkitkan populasi awal

Populasi awal dibangkitkan secara random sehingga didapatkan solusi awal. Populasi terdiri dari beberapa kromosom yang merepresentasikan solusi yang diinginkan.

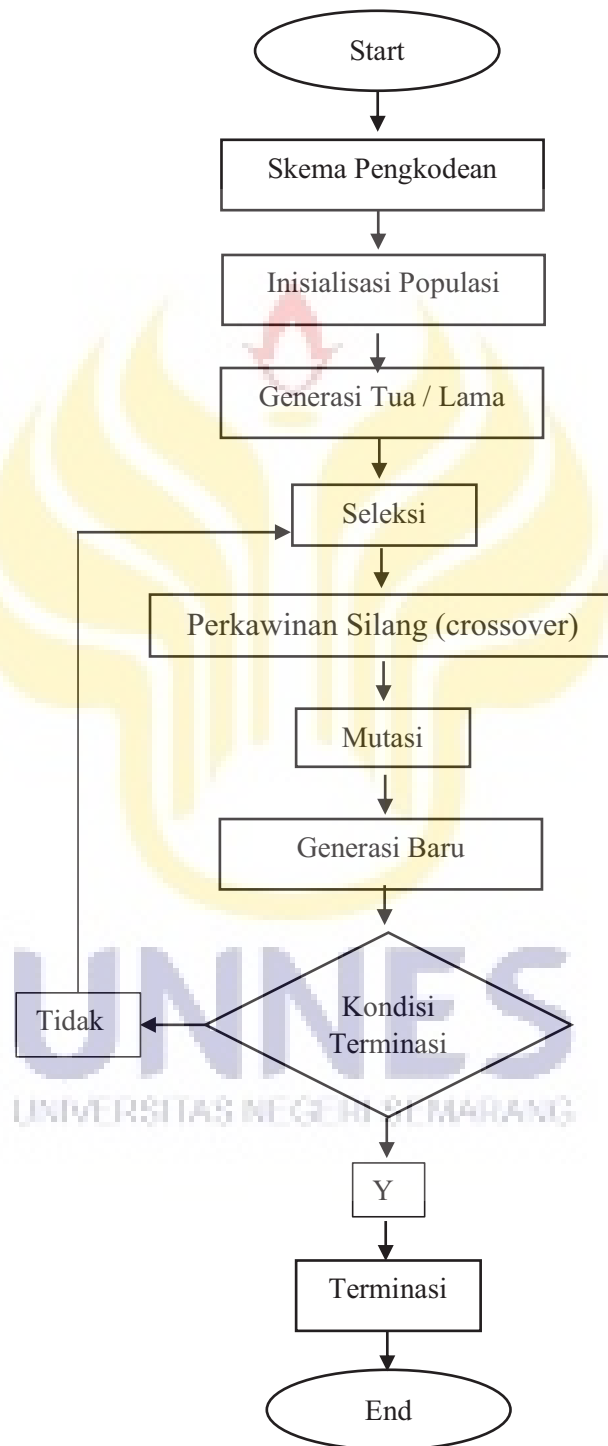
2. Membentuk generasi baru

Dalam membentuk generasi baru, digunakan operator seleksi, *crossover* (kawin silang) dan mutasi. Proses ini dilakukan berulang-ulang hingga didapatkan jumlah kromosom yang cukup untuk membentuk generasi baru di mana generasi baru ini merupakan representasi dari solusi baru. Generasi baru ini dikenal dengan istilah anak (*offspring*).

3. Evaluasi solusi

Pada tiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang dinamakan *fitness*. Nilai *fitness* suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut. Proses ini mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dan mengevaluasinya sampai kriteria terpenuhi. Bila kriteria belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah 2.

Struktur Umum Algoritma Genetika dapat diilustrasikan dalam diagram sebagai berikut:



Gambar 2.14 Flowchart Struktur Algoritma Genetika

Menurut Desiani (2006), Algoritma Genetika merupakan proses pencarian yang *heuristic* dan acak sehingga penekanan pemilihan operator yang digunakan sangat menentukan keberhasilan Algoritma Genetika dalam menemukan solusi optimum.

2.5.1 Komponen-Komponen Algoritma Genetika

Komponen-komponen utama pada Algoritma Genetika (Kusumadewi, 2003).

1. Skema Pengkodean

Pengkodean adalah suatu teknik untuk menyatakan populasi awal sebagai calon solusi suatu masalah ke dalam suatu kromosom sebagai suatu kunci pokok persoalan ketika menggunakan Algoritma Genetika.

Teknik pengkodean ini meliputi pengkodean gen dan kromosom. Gen merupakan bagian dari kromosom. Satu gen bisa mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk string bit, pohon, *array* bilangan real, daftar aturan, elemen permutasi, elemen program, atau representasi lainnya yang dapat diimplementasikan untuk operator Genetika.

2. Prosedur Inisialisasi Populasi

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran popuasi ditentukan, kemudian harus dilakukan inisialisasi kromosom secara acak, namun tetap harus memperhatikan domain solusi dan kendala pada permasalahan yang ada. Ukuran populasi yang sering digunakan oleh peneliti yang sudah ada adalah antara 20 sampai 30 (Desiani, 2006). Tetapi, beberapa penelitian menunjukkan bahwa

ukuran popuasi yang terbik tergantung dari barisan yang dienkodekan. Artinya jika terdapat ukuan kromosom 16 bit makan ukuran populasi adalah 16 kromosom.

3. Seleksi

Seleksi bertujuan memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling baik. Langkah pertama dalam seleksi ini adalah pencarian nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam wadah seleksi tersebut. Nilai *fitness* inilah yang nantinya akan digunakan pada tahap-tahap seleksi berikutnya (Kusumadewi, 2003).

Ada beberapa metode bagaimana memilih kromosom, antara lain:

a. *Rank-Based Fitness*

Pada *Rank-Based fitness*, populasi diurutkan menurut nilai objektifnya. Nilai *fitness* tiap-tiap individu tersebut dalam urutan, dan tidak dipengaruhi oleh nilai objektifnya.

b. *Roulette Whell Selection*

Metode seleksi dengan mesin *roulette* ini merupakan metode yang paling sederhana dan sering dikenal dengan nama *stochastic sampling with replacement*.

Cara kerja metode ini adalah sebagai berikut.

- 1) Menghitung nilai *fitness* dari masing-masing individu (f_i , dimana i adalah individu ke-1 sampai dengan ke- n).
- 2) Menghitung total *fitness* semua individu.
- 3) Menghitung probabilitas masing-masing individu.

- 4) Dari probabilitas tersebut dihitung jatah masing-masing individu pada angka 1 sampai 100.
- 5) Membangkitkan bilangan random antara 1 sampai 100.
- 6) Dari bilangan random yang dihasilkan, menentukan individu mana yang terpilih dalam proses seleksi.

Individu 1: *fitness* = 10%

Individu 2: *fitness* = 25%

Individu 3: *fitness* = 40%

Individu 4: *fitness* = 15%

Individu 5: *fitness* = 10%

Jatah untuk individu 1: 1-10

Jatah untuk individu 2: 11-35

Jatah untuk individu 3: 36-75

Jatah untuk individu 4: 76-90

Jatah untuk individu 5: 91-100

Dibangkitkan bilangan random antara 1-100 sebanyak 5 kali

Jika bilangan random yang dibangkitkan 30 maka individu yang terpilih adalah individu 2.

c. *Stochastic Universal Sampling*

Pada metode ini, individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen individu memiliki ukuran yang sama dengan ukuran *fitness* seperti halnya pada seleksi roda roulette. Kemudian diberikan sejumlah pointer sebanyak individu yang ingin diseleksi

pada garis tersebut. Andaikan n adalah jumlah individu yang akan diseleksi, maka jarak antar pointer adalah $1/n$ dan posisi pointer pertama diberikan secara acak pada range $[1, 1/n]$

d. *Trauncation Selection*

Seleksi ini biasanya digunakan oleh populasi yang jumlahnya sangat besar. Pada metode ini, individu-individu yang terbaik saja yang dipilih sebagai induk. Parameter yang digunakan dalam metode ini adalah suatu nilai ambang trunk yang mengindikasikan ukuran populasi yang akan diseleksi sebagai induk yang berkisar antar 50%-100%. Individu-individu yang ada di bawah nilai ambang ini tidak akan menghasilkan kromosom.

e. *Tournament Selection*

Pada metode seleksi dengan turnamen, ditetapkan suatu nilai tour untuk individu-individu yang dipilih secara acak dari suatu populasi. Individu-individu yang terbaik dalam klompok ini akan diseleksi sebagai induk. Parameter yang digunakan pada metode ini adalah ukuran yang bernilai antara 2 sampai n (jumlah individu dalam suatu populasi).

4. Evaluasi *fitness*

Evaluasi *fitness* merupakan dasar untuk proses seleksi. Langkah-langkahnya yaitu sting dikonversi ke parameter fungsi, fungsi objektifnya dievaluasi, kemudian mengubah fungsi objektif tersebut ke dalam fungsi *fitness*, dimana untuk maksimasi problem, *fitness* sama dengan fungsi objektifnya. Output dari fungsi *fitness* dipergunakan sebagai dasar menyeleksi induk pada generasi berikutnya.

Untuk permasalahan minimalisasi, nilai *fitness* adalah inversi dari nilai maksimal yang diharapkan. Proses inversi dapat dilakukan dengan rumusan *fitness* $= \frac{1}{f(x)}$ dengan x merupakan kromosom (Basuki, 2003:17).

$$fitness = A - f(x) \text{ atau } fitness = \frac{A}{f(x) + \varepsilon}$$

Keterangan:

A : konstanta yang ditentukan

x : individu (kromosom)

$f(x)$: nilai fungsi kromosom

ε : bilangan kecil yang ditentukan untuk menghindari pembagi nol atau $f(x) = 0$

5. Perkawinan silang (*Crossover*)

Crossover (perkawinan silang) bertujuan menambah keanekaragaman string dalam satu populasi dengan penyilangan antar string yang diperoleh dari reproduksi sebelumnya. Beberapa jenis *crossover* antara lain:

a. *Crossover* Diskret

Proses *crossover* dilakukan dengan menukar nilai variabel antar kromosom induk. Misalkan ada 2 induk dengan 3 variabel, yaitu:

Induk 1: 12 25 5

Induk 2: 123 4 34

Untuk tiap-tiap variabel induk yang menyumbangkan variabelnya ke anak dipilih secara acak dengan probabilitas yang sama.

Sampel 1: 2 2 1

Sampel 2: 1 2 1

Variabel pada sampel diatas dipilih secara acak. Disimbolkan dengan angka 1 dan 2 yang dapat diartikan 1 adalah gen dari induk 1 sedangkan 2 gen dari induk 2.

Kromosom baru yang terbentuk:

Anak 1: 123 4 5

Anak 2: 12 4 5

b. *Crossover* Satu Titik

Crossover dilakukan dengan memisahkan suatu string menjadi dua bagian dan selanjutnya salah satu bagian dipertukarkan dengan salah satu bagian dari string yang lain yang telah dipisahkan dengan cara yang sama. Proses yang demikian dinamakan operator *crossover* satu titik seperti diperlihatkan pada gambar berikut:

Induk 1: 1 1 0 0 1 | 0 0 1

Induk 2: 1 0 0 0 1 | 1 1 0

Anak 1 : 1 1 0 0 1 1 1 0

Anak 2 : 1 0 0 0 1 0 0 1

c. *Crossover* Seragam

Crossover seragam hampir sama dengan *crossover* satu titik. Hanya saja *crossover* seragam menyalin bit-bit secara acak dari kromosom induk.

Induk 1 : 1 1 0 0 1 0 1 1

Induk 2 : 1 1 0 1 1 1 1 1

Anak : 1 1 0 1 1 1 1 1

d. *Crossover Intermediate* (menengah)

Crossover menengah merupakan metode *crossover* yang hanya dapat digunakan untuk variabel real. Nilai variabel anak dipilih disekitar dan antara nilai-nilai variabel induk. Anak dihasilkan menurut aturan sebagai berikut:

$$\text{Anak} = \text{induk 1} + \alpha (\text{induk2} - \text{induk1})$$

Dengan alpha adalah faktor skala yang dipilih secara random pada interval $[-d, 1+d]$, biasanya $d=0,25$. Tiap-tiap variabel pada anak merupakan hasil *crossover* variabel-variabel menurut aturan di atas dengan nilai alpha dipilih ulang untuk tiap variabel.

Misalkan ada 2 individu dengan 3 variabel, yaitu:

Induk 1: 12 25 5

Induk 2: 123 4 34

Misalkan nilai alpha yang terpilih adalah:

Sampel 1: 0,5 1,1 0,1

Sampel 2: 0,1 0,8 0,5

Kromosom baru yang terbentuk

Anak 1: 67,5 1,9 2,1

Anak 2: 23,1 8,2 19,5

e. *Crossover* Garis

Pada dasarnya *crossover* garis ini sama dengan *crossover* menengah, hanya saja nilai alpha untuk semua variabel sama. Misalkan ada 2 individu dengan 3 variabel, yaitu:

Induk 1: 12 25 5

Induk 22: 123 4 34

Alpha yang dipilih adalah:

Sampel 1: 0,5

Sampel 2: 0,1

f. *Order Crossover*

Order crossover merupakan cara *crossover* dengan menukar kromosom dengan tetap menjaga urutan gen yang bukan bagian dari kromosom tersebut.

Contoh *order crossover* adalah:

Misalkan ada 3 kromosom induk yang akan dilakukan *crossover*:

Kromosom [1] = [ABCD]

Kromosom [2] = [BACD]

Kromosom [3] = [ACDB]

Proses *crossover*:

Kromosom [1] = Kromosom [1] $\times\times$ Kromosom [2]

= [ABCD] $\times\times$ [BACD]

= [ACBD]

Kromosom [2] = Kromosom [2] $\times\times$ Kromosom [3]

= [BACD] $\times\times$ [BCDA]

= [BCDA]

Kromosom [3] = Kromosom [3] $\times\times$ Kromosom [1]

= [BCDA] $\times\times$ [ABCD]

= [BCAD]

6. Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Selain untuk sebagai penekanan selektif yang lebih efisien, operator mutasi dapat digunakan untuk menghindari terjadinya konvergensi prematur tersebut dan tetap menjaga perbedaan kromosom dalam populasi. Beberapa cara operasi mutasi yang diterapkan dalam Algoritma Genetika adalah:

a. Mutasi dalam pengkodean biner.

Proses yang dilakukan pada mutasi pengkodean biner adalah menginversi nilai bit pada posisi tertentu yang dipilih secara acak (atau dengan menggunakan skema tertentu) pada kromosom.

Contoh mutasi pengkodean biner:

Kromosom sebelum mutasi : 1 0 0 1 0 1 1 1

Kromosom sebelum mutasi : 1 0 0 1 0 0 1 1

b. Mutasi dalam pengkodean permutasi.

Proses mutasi yang dilakukan dengan cara memilih dua posisi (locus) dari kromosom dan kemudian nilainya saling dipertukarkan.

Contoh mutasi dalam pengkodean permutasi:

Kromosom sebelum mutasi : 1 2 3 4 5 6 7 8 9

Kromosom setelah mutasi : 1 2 7 4 6 5 8 3 9

c. Mutasi dalam pengkodean nilai.

Proses mutasi pada pengkodean nilai dapat dilakukan dengan berbagai cara, salah satunya yaitu dengan memilih sembarang posisi gen pada kromosom. Nilai

yang ada tersebut kemudian ditambahkan atau dikurangkan dengan suatu nilai kecil tertentu yang diambil secara acak.

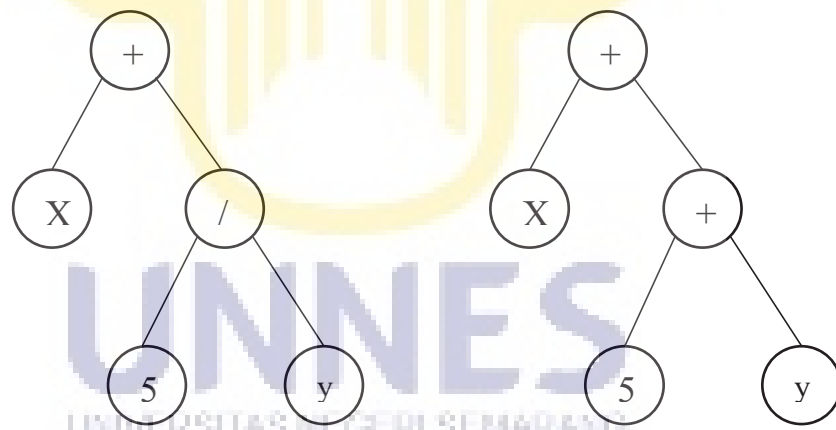
Contoh mutasi dalam pengkodean nilai real dengan nilai yang ditambahkan atau dikurangkan dengan 0,1.

Kromosom sebelum mutasi : 1,43 1,09 4,51 9,11 6,94

Kromosom setelah mutasi : 1,43 1,19 4,51 9,01 6,94

d. Mutasi dalam pengkodean pohon.

Mutasi dalam pengkodean pohon dapat dilakukan antara lain dengan cara mengubah operator (+, -, *, /) atau dapat juga dilakukan dengan memilih dua verex dari pohon dan saling mempertukarkan operator atau nilainya. Contoh mutasi pada pengkodean pohon:



e. *Swapping Mutation.*

Proses mutasi dengan cara ini dilakukan dengan menentukan jumlah kromosom yang akan mengalami mutasi dalam satu populasi melalui parameter mutation rate (pm). Proses mutasi dilakukan dengan cara menukar gen yang telah dipilih secara acak dengan gen sesudahnya, jika gen tersebut berada di akhir kromosom, maka ditukar dengan gen yang pertama.

Pertama hitung panjang total gen yang ada pada suatu populasi: Populasi total gen = Gen dalam 1 kromosom * jumlah kromosom

Untuk memilih posisi gen yang akan mengalami mutasi dilakukan dengan membangkitkan bilangan acak antar 1 sampai panjang total kromosom untuk dilakukan proses mutasi (Kusumadewi, 2003:14).

2.5.2 Kontrol Parameter Algoritma Genetika

Menurut Desiani (2006), dua parameter dasar Algoritma Genetika yaitu probabilitas *crossover* dan probabilitas mutasi. Probabilitas *crossover* menyatakan seberapa sering proses *crossover* akan terjadi antara dua kromosom orang tua. Hasil penelitian yang sudah pernah dilakukan oleh praktisi Algoritma Genetika menunjukkan bahwa angka probabilitas *crossover* sebaiknya cukup tinggi, yaitu antara 80% sampai 90%. Untuk beberapa masalah tertentu probabilitas 60% sudah memberikan hasil yang cukup baik (Marek, 1998). Sedangkan probabilitas mutasi menyatakan seberapa sering bagian-bagian kromosom dimutasikan. Probabilitas mutasi sebaiknya diberikan nilai yang kecil karena tujuan mutasi hanya untuk menghindari perbedaan kromosom.

2.5.3 Penerapan Algoritma Genetika untuk Pencarian Nilai Maksimum

Polinomial

Anggap bahwa polinomial yang akan dicari nilai maksimumnya adalah polinomial $p(x)$ sebagai berikut:

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0$$

Secara matematis, masalah optimasi untuk mencari nilai maksimum bagi polinomial sebagaimana persamaan di atas untuk jangkauan $x_a \leq x \leq x_b$ dapat dinyatakan sebagai: $\max p(x)$ untuk $x_a \leq x \leq x_b$ (Zukhri : 2014).

Langkah-langkah yang harus dilakukan dalam penerapan Algoritma Genetika untuk menentukan titik maksimum dari polinomial di atas adalah:

a. Pembentukan populasi awal.

Populasi awal dibentuk dari sejumlah kromosom. Representasi kromosom yang paling sederhana untuk menyelesaikan masalah ini adalah dalam bentuk kode biner. Banyak bit yang diinginkan berdasarkan tingkat ketelitian yang diinginkan. Jika ukuran kromosom ditentukan N bit, maka akan terdapat kemungkinan kromosom yang merupakan representasi dari nilai x dengan jangkauan $x_a \leq x \leq x_b$.

Secara matematis representasi kromosom (v) dengan kode biner N bit dapat dinyatakan dalam persamaan berikut:

$$v_1 = [b_1, b_2, \dots, b_N]$$

dengan $1 \leq i \leq \text{Ukuran Populasi}$ dan $b_i \in \{0, 1\}$

Secara umum, pengkodean kromosom menjadi penyelesaian yang berbentuk bilangan riil ini berdasarkan persamaan:

$$x = 1 + 4 \times \left(\frac{\sum_{j=1}^N b_j \times 2^{N-j}}{2^N - 1} \right)$$

Banyak bit bilangan biner (N) menunjukkan tingkat ketelitian yang diinginkan.

b. Proses evaluasi

Dalam proses evaluasi ini perlu dilakukan perhitungan-perhitungan sebagai berikut:

- i. Dekode setiap representasi kromosom v_1 menjadi x_1
- ii. Hitung nilai *fitness* untuk setiap kromosom.
- c. Proses seleksi

Proses seleksi yang paling mudah digunakan adalah proses seleksi dengan pendekatan roda *roulette*. Proses ini dapat dilakukan dengan pemilihan secara acak menggunakan bilangan riil dengan langkah-langkah:

- i. Bangkitkan bilangan random yang bernilai antara 0 sampai 1.
- ii. Jika r kurang dari sama dengan probabilitas kromosom pertama, maka pilih kromosom pertama, jika kurang dari sama dengan probabilitas kromosom i , maka pilih kromosom ke i .
- iii. Ulangi langkah diatas sebanyak kromosom dalam sebuah populasi.
- d. Penerapan operator penyilangan dan mutasi.

Pada masalah ini operator yang digunakan adalah *crossover* satu titik dan mutasi pada pengkodean biner. Dengan menentukan probabilitas *crossover* dan probabilitas mutasi terlebih dahulu.

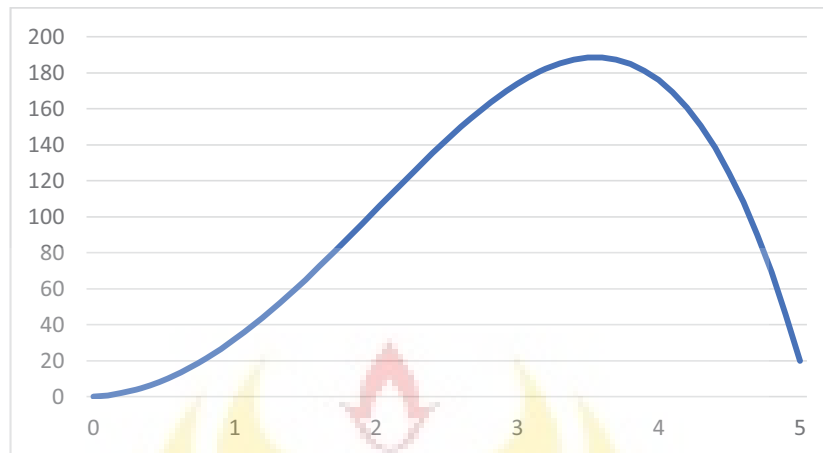
- e. Mengulangi langkah b sampai e sebanyak generasi yang diinginkan dan pilih kromosom terbaik pada setiap generasi.

Contoh Soal

Dipunyai fungsi sebagai berikut.

$$p(x) = -x^4 - x^3 + 30x^2 + 4x$$

Dengan grafik:



Gambar 2.15 Grafik fungsi $p(x)$

Tentukan nilai optimal dari fungsi $p(x)$?

Penyelesaian:

Dalam penyelesaian ini digunakan 7 bit bilangan biner maka ada 128 (2^7) kemungkinan kombinasi. Tujuan pembahasan ini memberi gambaran penyelesaian dengan Algoritma Genetika. Dengan $x_{\min} = 1$, $x_{\max} = 5$, dan lebar bit = 7.

Diperoleh rumus pengkodean sebagai berikut:

$$x = 1 + 4 \times \left(\frac{\sum_{j=1}^7 b_j \times 2^{7-j}}{127} \right)$$

Dengan b_j adalah bit dari kromosom yang dibangkitkan

1. Dengan menggunakan bantuan Microsoft Excel, populasi awal dibangkitkan dengan bilangan biner 0 dan 1. Variabel x_i merupakan konversi nilai biner dari kromosom yang terbentuk. $P(x_i)$ merupakan hasil polinomial dengan x_i yang diperoleh dengan pengkodean acak. Proses pembangkitan kromosom untuk menjadi populasi awal ditunjukkan pada Tabel 2.1.

Tabel 2.1 Populasi Awal

i	Kromosom							x_i	$P(x_i)$	$\frac{P(x_i)}{\sum P(x_j)}$	Probabilitas
1	0	0	0	1	1	1	1	1,47244	63,039279	0,0628316	0,0628316
2	1	1	1	0	0	1	0	4,590551	109,74151	0,10938	0,1722117
3	1	1	0	0	1	0	0	4,149606	165,220397	0,164676	0,3368879
4	0	0	1	1	1	1	1	1,976377	102,110354	0,101774	0,4386619
5	0	1	1	1	0	0	1	2,795275	162,695194	0,162159	0,6008213
6	0	0	0	0	0	1	0	1,062992	35,672613	0,035555	0,6363764
7	0	0	1	1	0	0	0	1,755905	84,599783	0,084321	0,7206975
8	0	0	0	0	1	1	0	1,188976	43,48658	0,043343	0,7640409
9	1	1	0	0	0	1	0	4,086614	170,206889	0,169646	0,9336872
10	1	1	1	1	0	0	1	4,811023	66,531916	0,066312	1
Nilai terbaik								170,206889			
rata-rata <i>fitness</i>								166,11014			

2. Seleksi roda *roulette*

Langkah awal pada seleksi dengan metode roda *roulette* adalah membangkitkan bilangan secara acak pada Microsoft Excel dengan rumus =RAND() dan akan muncul bilangan acak desimal antara 0 sampai 1. Setelah bilangan acak dibangkitkan pilih x_i dengan r kurang dari atau sama dengan probabilitas yang di dapat dengan rumus $\frac{P(x_i)}{\sum P(x_j)}$, kemudian dijumlahkan untuk mencari probabilitas pada seleksi *roulette*.

Tabel 2.2 Seleksi Roda *Roulette*

Seleksi roda <i>roulette</i>	
r	Hasil Seleksi
0,618451006	6
0,51180434	5
0,010896902	1
0,741268363	8
0,754354521	8

0,858019656	9
0,67184715	7
0,540261012	5
0,816440278	9
0,818580731	9

Setelah seleksi dilakukan diperoleh 6 kromosom induk. Dapat dilihat dari proses seleksi kita dapat menghemat 4 kromosom untuk dibangkitkan menjadi generasi baru.

3. Penerapan operator penyilangan dan mutasi

Kromosom-kromosom induk yang terpilih kemudian dikenakan operator *crossover* dengan probabilitas 50 % dan operator mutasi dengan probabilitas 10%. Sebelum melakukan *crossover*, dibangkitkan bilangan random 0-1. Kemudian induk yang dipilih adalah induk yang memiliki probabilitas kurang dari 50%. Induk dengan probabilitas kurang dari 10% yang dipilih untuk dimutasi.

Tabel 2.3 Pemilihan Induk Crossover

Pemilihan Induk <i>Crossover</i> dengan Probabilitas 50%		
<i>i</i>	<i>r</i>	Kromosom Induk
1	0,026817709	v
2	0,631248618	x
3	0,778008978	x
4	0,153228906	v
5	0,829471845	x
6	0,113835032	v
7	0,523870079	x
8	0,998482061	x
9	0,819517558	x
10	0,219958968	v

Kromosom induk di *crossover* satu titik dan diperoleh hasil sebagai berikut.

Tabel 2.4 Proses *Crossover* Satu Titik

i	Hasil Seleksi	Induk	Kromosom Induk							Hasil Crossover						
1	6	v	0	0	0	0	0	1	0	0	0	0	0	0	1	0
2	5	x	0	1	1	1	0	0	1	0	1	1	1	0	0	1
3	1	x	0	0	0	1	1	1	1	0	0	0	1	1	1	1
4	8	v	0	0	0	0	1	1	0	0	0	0	0	0	1	0
5	8	x	0	0	0	0	1	1	0	0	0	0	0	1	1	0
6	9	v	1	1	0	0	1	0	1	1	1	0	0	1	1	0
7	7	x	0	0	1	1	0	0	0	0	0	1	1	0	0	0
8	5	x	0	1	1	1	0	0	1	0	1	1	1	0	0	1
9	9	x	1	1	0	0	0	1	0	1	1	0	0	0	1	0
10	9	v	1	1	0	0	0	1	0	1	1	0	0	0	1	0

Setelah di *crossover*, kromosom di mutasi dengan probabilitas mutasi 10%.

Tabel 2.5 Proses Mutasi Satu Gen

i	r	Induk	Kromosom induk							Hasil Mutasi						
1	0,323820531	x	0	0	0	0	0	1	0	0	0	0	0	0	1	0
2	0,490741694	x	0	1	1	1	0	0	1	0	1	1	1	0	0	1
3	0,605137892	x	0	0	0	1	1	1	1	0	0	0	1	1	1	1
4	0,990455249	x	0	0	0	0	1	1	0	0	0	0	0	1	1	0
5	0,58827241	x	0	0	0	0	1	1	0	0	0	0	0	1	1	0
6	0,951971576	x	1	1	0	0	0	1	0	1	1	0	0	0	1	0
7	0,086654252	v	0	0	1	1	0	0	0	1	1	1	0	0	0	0
8	0,565209871	x	0	1	1	1	0	0	1	0	1	1	1	0	0	1
9	0,167787989	x	1	1	0	0	0	1	0	1	1	0	0	0	1	0
10	0,036667366	v	1	1	0	0	0	1	0	1	1	0	1	0	1	0

Diperoleh populasi baru sebagai berikut:

Tabel 2.6 Populasi Baru

i	Kromosom							Sigma bj	X	p(xi)
1	0	0	0	0	0	1	0	2	1,062992126	35,67261384
2	0	1	1	1	0	0	1	57	2,795275591	162,6951941
3	0	0	0	1	1	1	1	15	1,472440945	63,03927964
4	0	0	0	0	1	1	0	6	1,188976378	43,486588
5	0	0	0	0	1	1	0	6	1,188976378	43,486588
6	1	1	0	0	0	1	0	98	4,086614173	170,2068899

7	0	1	1	1	0	0	0	56	2,763779528	160,7520342
8	0	1	1	1	0	0	1	57	2,795275591	162,6951941
9	1	1	0	0	0	1	0	98	4,086614173	170,2068899
10	1	1	0	1	0	1	0	106	4,338582677	146,0702687
									jumlah <i>fitness</i>	1158,31154
									rata-rata <i>fitness</i>	115,831154

Pada generasi kedua diperoleh nilai terbaik 170,2069

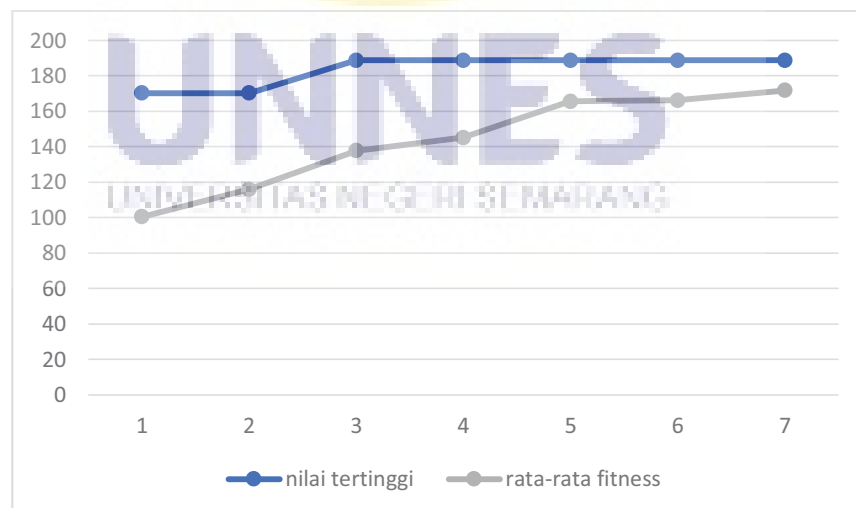
Mengulangi proses evaluasi, *crossover* dan mutasi sampai generasi ketujuh.

Data hasil pengujian 7 generasi:

Tabel 2.7 Data Hasil Pengujian Tujuh Generasi

	nilai tertinggi	rata-rata <i>fitness</i>
1	170,2068899	100,3305
2	170,2068899	115,831154
3	188,6602521	137,7615306
4	188,6602521	145,0821363
5	188,6602521	165,574619
6	188,6602521	166,1101408
7	188,6602521	171,7360509

Grafik hasil persamaan polinomial sebanyak tujuh generasi:



Gambar 2.16 Grafik Nilai Terbaik dan Rata-Rata *Fitness* Tujuh Generasi

Dapat dilihat dari Grafik Nilai Terbaik dan Rata-Rata *Fitness* Tujuh Generasi pada Gambar 2.16, dalam tujuh generasi nilai maksimum dapat ditemukan dengan rata rata *fitness* yang meningkat setiap generasinya. Dapat disimpulkan bahwa penggunaan Algoritma Genetika mengurangi banyaknya pemeriksaan terhadap semua kemungkinan penyelesaian masalah.

2.6 MATLAB

MATLAB (Matrix Laboratory) adalah sebuah program untuk analisis dan komputasi numerik dan merupakan suatu bahasa pemrograman matematika lanjutan yang dibentuk dengan dasar pemikiran menggunakan sifat dan bentuk matriks. Pada awalnya, program ini merupakan *interface* untuk koleksi rutin-rutin numerik dari proyek LINPACK dan EISPACK, dan dikembangkan menggunakan bahasa FORTRAN namun sekarang merupakan produk komersial dari perusahaan Mathworks, Inc. yang dalam perkembangan selanjutnya dikembangkan menggunakan bahasa C++ dan assembler (utamanya untuk fungsi-fungsi dasar MATLAB). Di dalam MATLAB setiap variabel dipandang sebagai matriks sehingga mempermudah MATLAB untuk memecahkan masalah Algoritma Genetika (Suyanto, 2006).

2.7 Window-Window pada MATLAB

Ada beberapa macam window yang tersedia dalam MATLAB, yang dapat dijelaskan sebagai berikut:

a. MATLAB Command window/editor MATLAB

Command window/editor merupakan window yang dibuka pertama kali setiap kali MATLAB dijalankan.

b. MATLAB Editor/Debugger (Editor M-File/Pencarian Kesalahan)

Window ini merupakan tool yang disediakan oleh Matlab 5 keatas. Berfungsi sebagai editor script Matlab (M-file). Walaupun sebenarnya script ini untuk pemrograman Matlab dapat saja menggunakan editor yang lain seperti notepad, wordpad bahkan word.

c. Figure Windows

Window ini adalah hasil visualisasi dari script Matlab. Namun Matlab memberi kemudahan bagi programer untuk mengedit window ini sekaligus memberikan program khusus untuk itu. Sehingga window ini selain berfungsi sebagai visualisasi output dapat juga sekaligus menjadi media input yang interaktif.

d. MATLAB help window

MATLAB menyediakan sistem *help* yang dapat diakses dengan perintah *help*.

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan dapat disimpulkan:

1. Konstruksi nilai *fitness* pada Algoritma Genetika dalam pewarnaan graf adalah minimum banyaknya warna dan nol kesalahan dalam pemberian warna. Nilai *fitness* pada pewarnaan graf mempunyai bobot lebih besar pada banyaknya kesalahan pewarnaan karena itu merupakan tujuan pewarnaan. Sehingga rumus nilai *fitness* adalah:

$$eval(v) = \frac{1}{f(v) + N * p(v)}$$

dengan $f(v)$ merupakan fungsi objektif, N merupakan banyaknya titik yang akan diwarnai, dan $p(v)$ merupakan banyaknya kesalahan pewarnaan. Dalam kasus pewarnaan fungsi objektif merupakan minimal warna pada kromosom.

2. Pada penerapan Algoritma Genetika untuk pewarnaan graf, proses seleksi yang digunakan adalah seleksi roda *roulette*, proses *crossover* menggunakan *crossover* satu titik dan proses mutasi menggunakan mutasi satu gen.
3. Langkah-langkah penyelesaian masalah pewarnaan graf dengan Algoritma Genetika yaitu: membangkitkan populasi awal secara random, menyeleksi induk dengan metode seleksi roda *roulette*, membentuk generasi baru dengan operator *crossover* dan operator mutasi, mengevaluasi solusi baru yang terbentuk, dan proses terhenti jika sudah mendapatkan solusi pewarnaan

dengan banyaknya warna minimal dan nol kesalahan pewarnaan. Berdasarkan hasil pewarnaan graf tersebut dapat diketahui bilangan kromatik yaitu jumlah warna minimum berbeda yang dibutuhkan untuk mewarnai graf G tanpa melanggar kendala *adjacency*, yang dinotasikan dengan $X(G)$. Jika $k = \{1, 2, 3 \dots\}$ dan $P(G, k)$ adalah jumlah solusi untuk mewarnai graf G dengan k , maka

$$x(G) = \min(k: P(G, k) > 0)$$

5.2 Saran

Berdasarkan hasil penelitian maka saran yang dapat disampaikan untuk penelitian berikutnya adalah sebagai berikut.

1. Penyelesaian masalah pewarnaan graf dengan Algoritma Genetika menggunakan proses *crossover* random.
2. Pembuatan program untuk jumlah n yang lebih besar.

DAFTAR PUSTAKA

- Adeputra, A. 2016. *Pemanfaatan Algoritma Semut untuk Penyelesaian Masalah Pewarnaan Graf*. Skripsi. Bandung: Institut Teknologi Bandung.
- Albar, M. A. 2013. *Algoritma Genetik Tabu Search dan Memetika pada Permasalahan Penjadwalan Kuliah*. Skripsi. Yogyakarta: Universitas Gajah Mada.
- Barod, P., V. Hawanna, & V. Jagtap. 2014. *Genetic and Memetic Algorithm on Graph Colouring*. Scientific Journal of Impact Factor(SJIF). ISSN(P): 2348-6406.
- Basuki, A. 2003. *Algoritma Genetika, Suatu Alternatif Penyelesaian Permasalahan Searching, Optimasi dan Machine Learning*. Surabaya: ITS.
- Budayasa, I. K. 2007. *Teori Graph dan Aplikasinya*. Surabaya: Unesa University Press.
- Costa, D. & A. Hertz. 1997. *Ants can colour graphs*, *Journal of the Operational Research Society* 48, 295-305.
- Croitoru, C. & A. Adriana. 2002. *A New Genetic Graph Coloring Heuristic*. In *Proceedings of The Computational Symposium on Graph Coloring and its Generalizations*, 63-74. Ithaca, New York, USA.
- Desiani, A. & M. Arhani. 2006. *Konsep Kecerdasan Buatan*. Yogyakarta: Andi Offset.
- Fleurent, C. & J.A. Ferland. 1996. "Genetic and hybrid algorithms for graph coloring," *Annals of Operations Research*, vol. 63, pp. 437-461.
- Glass, C. A. & A. P. Bennett. 2003. *Genetic algorithm for graph coloring: Exploration of Galinier and Hao's algorithm*. *Journal of Combinatorial Optimization* 3: 229-236
- Gwee, B. H., M, H. Lim, & J. S. Ho. 1993. *Solving fourcolouring map problem using genetic algorithm*. In *Proceedings of First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, 332-333. New Zealand.
- Hertz, D. A. 1987. *Using Tabu Search technique for Graph Coloring*. Volume 39. Issue 4. Pp 345-351.

- Hindi, M. M. & R. V. Yampolskiy. 2012. *Genetic Algorithm Applied to the Graph Coloring Problem*. Kentucky : Computer Engineering and Computer Science J.B Speed School of Engineering Louisville.
- Hopgood, A. A. 2001. *Intelligent Systems for Engineers an Scientists*. Boca Raton: CRC Press LLC.
- Jusuf, H. 2009. *Pewarnaan Graph Pada Simpul Untuk Mendeteksi Konflik Penjadwalan Kuliah*. Seminar Nasional Aplikasi Teknologi Informasi. ISSN:1907-5022.
- Kusumadewi, S. 2003. *Artificial Intelligence: Teknik dan Aplikasinya*. Yogyakarta: Graha Ilmu.
- Munir, R. 2005. *Matematika Diskrit*. Bandung : Informatika.
- Marek, O. 1998. Introduction to Genetic Algorithm. <http://www.obitko.com/tutorials/genetic-algorithms/>.
- Rosyida, I. 2016. *Pengembangan Metode Pewarnaan Titik dan Bilangan kromatik pada Graf-Graf tak Deterministik*. Yogyakarta: Universitas Gajah Mada.
- Shen, J. W. 2003. *Solving the Graph Coloring Problem using Genetic Programming*, in *Genetic Algorithms and Genetic Programming*. Stanford 187-196.
- Suyanto. 2006. *Algoritma Genetika dalam Matlab*. Yogyakarta: Andi Offset.
- Thiang., R. Kurniawan, & H. Ferdinando. 2001. *Implementasi Algoritma Genetika pada Mikrokontroler MCS51 Untuk Mencari Rute Terpendek*. Surabaya: Universitas Kristen Putra.
- Wibisono, S. 2008. *Matematika Diskrit (2nd ed.)*. Yogyakarta : Graha Ilmu.
- Wulan, R. 2015. *Aplikasi Graph Colouring dengan Algoritma Tabu Search dalam menyelesaikan Masalah Penjadwalan*. Yogyakarta: FMIPA Universitas Komputer Indonesia.
- Zukhri, Z. 2014. *Algoritma Genetika: Metode Komputasi Evolusioner untuk Menyelesaikan Masalah Optimasi*. Yogyakarta: Andi Offset.