

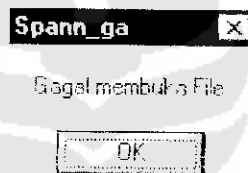
BAB VI

ANALISIS KINERJA PERANGKAT LUNAK

6.1 Analisis Proses

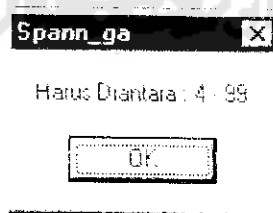
6.1.1 Proses Penginputan Data (Form Data)

Proses ini dijalankan di *Form Data* di mana *user* diminta untuk membuat file baru atau dapat membuka file yang telah di simpan sebelumnya. Jika File yang diBuka tidak berhasil maka akan muncul pesan seperti berikut



Gambar 6.1 Tampilan Pesan Kegagalan saat membuka File pada *Form Data*

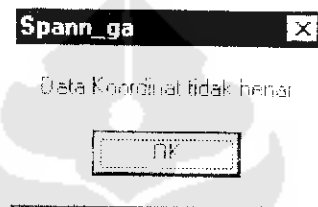
Pada saat mengisikan jumlah titik untuk membuat data baru, jika jumlah data yang diinputkan kurang dari 4 titik maka akan muncul pesan seperti berikut



Gambar 6.2 Tampilan Pesan Input Jumlah Titik pada *Form Data*

Pada saat pengisian data jarak beserta koordinatnya pada Grid, tombol yang diaktifkan hanya tombol angka 0 sampai 9, tombol Enter, tombol BackSpace, dan tombol Arrows. Sedangkan tombol yang lain tidak berfungsi atau dikunci. Hal ini dilakukan agar pengisian data terhindar dari data yang bukan angka.

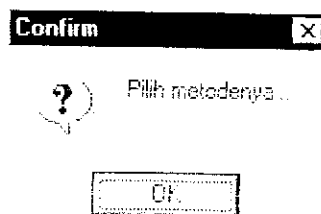
Jika data koordinat pada grid belum lengkap namun user sudah menekan *Tutup*, maka akan muncul pesan sebagai berikut



Gambar 6.3 Tampilan Pesan Data Koordinat pada Form *Data*

6.1.2 Proses Pencarian dengan Algoritma Genetik (Form *Genetik*)

Proses ini dijalankan di Form *Genetik* di mana *user* diminta untuk mengisikan inialisasi parameter genetik, dan memilih metode *CrossOver* dan Mutasi, kemudian menekan tombol *Proses*. Jika metode *CrossOver* atau Mutasi belum dipilih, maka akan muncul pesan sebagai berikut

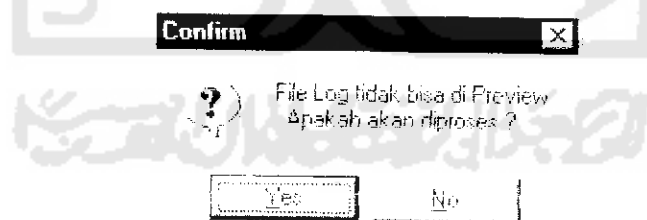


Gambar 6.3 Tampilan Konfirmasi jika ada metode yang belum dipilih pada Form *Genetik*

Beberapa parameter genetik telah mempunyai nilai *default*, misalnya pada Ukuran Populasi, secara *default range* nilai yang tertera ialah 10-10000. Pada probabilitas *CrossOver*, *range* nilai yang harus diisikan ialah 1-100, sedangkan pada Mutasi *range* nilainya ialah 0-100. Ini dibuat atas pertimbangan bahwa proses genetik di alam ini akan terjadi bila minimal telah mengalami proses persilangan. Jika proses pencarian selesai dijalankan maka *ProgressBar* akan menunjukkan pada keadaan maksimal.

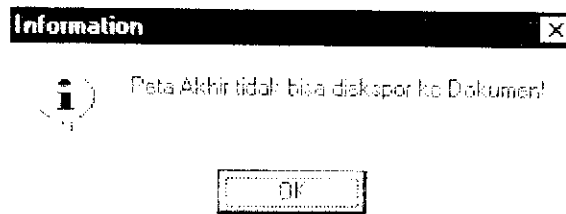
6.1.3 Proses Keluaran Solusi (Form Keluaran)

Proses ini dijalankan di Form *Keluaran* di mana *user* diminta untuk memilih jenis keluaran yang akan dicetak atau didokumentasi. Jika tombol *Cetak* ditekan pada pilihan cetak *FileLog* maka akan muncul pesan bahwa *FileLog* tidak bisa dicetak dikarenakan besarnya file tersebut, sehingga hanya dapat langsung diproses atau diekspor ke Document Word untuk diedit.



Gambar 6.4 Tampilan Konfirmasi jika tombol *Cetak* ditekan pada pilihan cetak *FileLog* pada Form *Keluaran*

Jika tombol *Dokumentasi* ditekan pada pilihan dokumentasi *Peta Akhir* maka akan muncul pesan bahwa *Peta Akhir* tidak bisa diekspor ke Dokumen, sehingga hanya dapat langsung dicetak.



Gambar 6.5 Tampilan Informasi jika tombol *Dokumentasi* ditekan pada pilihan *Peta Akhir* pada Form *Keluaran*

6.2 Analisis terhadap Antarmuka Perangkat Lunak

Antarmuka menjadi hal yang sangat penting dalam suatu perangkat lunak. Perangkat lunak yang mempunyai (*interface*) yang baik harus bersifat ramah pengguna (*user friendly*), artinya mudah digunakan tanpa melalui proses belajar yang rumit, tidak membingungkan pengguna serta tidak menghabiskan banyak waktu untuk menyelesaikan persoalan *minimum spanning tree* dengan perangkat lunak ini. Hal ini dapat dilihat dari beberapa segi berikut ini.

6.2.1 Level Keahlian Pemakai

Perangkat lunak ini dibuat untuk dapat digunakan oleh kalangan yang berhubungan dengan persoalan *minimum spanning tree*. Untuk mengoperasikan perangkat lunak ini hanya dibutuhkan pengetahuan tentang *KeyBoard* dan *Mouse*, dan tidak memerlukan keahlian khusus dibidang komputer. Selain itu yang dibutuhkan hanyalah konsep dasar tentang *minimum spanning tree*.

6.2.2 Interaksi Manusia dan Komputer

Perangkat lunak ini sangat komunikatif, artinya mudah dimengerti oleh pengguna. Dari sisi tampilan, penggunaan objek sudah sangat familier bagi mereka yang sudah mengenal Microsoft Windows, seperti objek *OpenDialog* dan *SaveDialog* (terdapat hampir di semua aplikasi di Windows), *Grid* (seperti di Microsoft Excel). Dari sisi pesan-pesan kesalahan, ataupun informasi juga sangat mudah dipahami oleh *user* jika melakukan kesalahan ataupun terjadi kegagalan sistem.

6.2.3 Input Data maupun Proses

Penginputan data maupun proses yang lebih sedikit serta pembatasan input atau proses (semua objek dibatasi penggunaannya baik dengan penguncian *keyboard* ataupun dengan objek-objek yang bersifat *optional* seperti *ComboBox*) menjadi perhatian dalam perancangan lunak ini. Hal ini diharapkan dapat mengurangi kesalahan oleh *user*.

6.3 Analisis Kinerja dan Pengujian Perangkat Lunak

Kinerja yang dihasilkan oleh persoalan *minimum spanning tree* menggunakan algoritma Genetik ini dapat dilihat dari beberapa sisi yaitu

6.3.1 Analisis Kinerja untuk Data Masukan (*Input*)

Perangkat lunak ini menerima masukan (*input*) dari file yang berekstensi *.txt*. Perangkat lunak ini mampu menangani jumlah data hingga 99 untuk letak

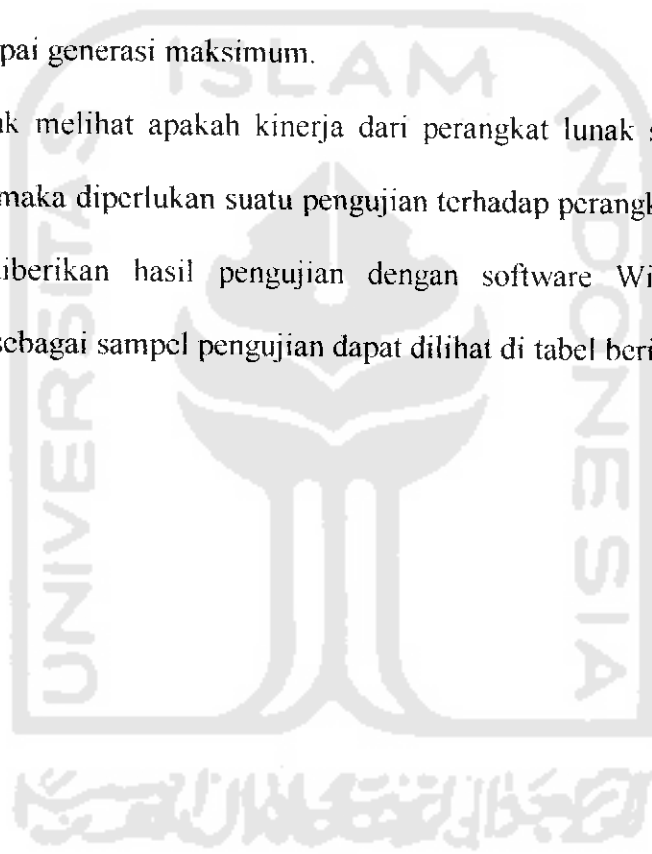
titik pada peta. Pada penginputan jarak, data yang dapat dimasukkan berupa bilangan bulat untuk titik yang terhubung. Sedangkan untuk titik yang tidak terhubung user dapat memasukkan angka 0 (nol)

6.3.2 Analisis Kinerja Pemrosesan dengan Algoritma Genetik (*Proses*)

Di dalam pemrosesan data masukan, perangkat lunak ini mampu menangani iterasi sebanyak 1000 kali dengan ukuran populasi maksimum 100. Untuk waktu pemrosesan, grafiknya ialah linier (berbanding lurus) untuk setiap jumlah data (panjang kromosom) dan parameter genetik (ukuran populasi dan banyak generasi). Sebelum melakukan optimasi, ditentukan terlebih dahulu model pengkodean sistem (pengkodean gen-gen), dalam hal ini yang digunakan ialah deret (barisan) angka (himpunan angka) sepanjang 1 hingga panjang kromosom. Langkah selanjutnya adalah pembangkitan generasi awal populasi dengan pencuplikan acak gen-gen yang membentuk kromosom. Kromosom-kromosom yang terbentuk selanjutnya dievaluasi dengan penghitungan *fitness* yang didasarkan dengan fungsi jarak. Di dalam penghitungan ini juga dilakukan penghitungan peluang untuk memperoleh tekanan seleksi yang seragam. Pemilihan induk dilakukan dengan menggunakan roda rolet, di mana semua individu ditempatkan ke dalam suatu lingkaran. Persentase luas dari setiap induk pada lingkaran roda rolet didasarkan nilai *fitness*-nya. Semakin besar nilai *fitness* semakin luas pula bagiannya pada roda rolet. Perbaikan generasi dilakukan dengan mekanisme seleksi, *CrossOver* dan mutasi. Dua induk (atau kelipatannya) dipilih untuk melakukan *CrossOver*. Sementara mutasi dilakukan dengan

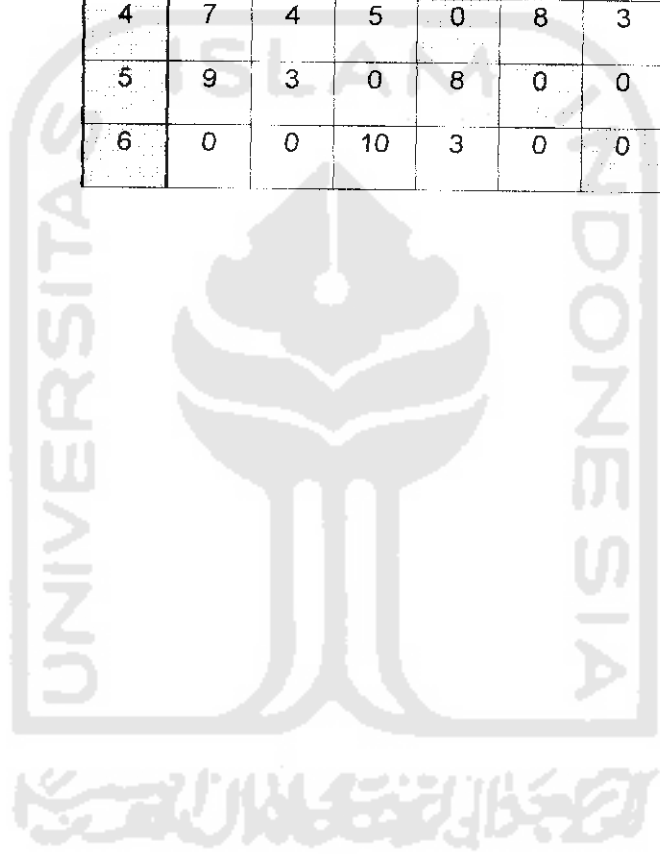
peloncatan-peloncatan gen-gen induk-induk baru (hasil *CrossOver* induk-induk generasi sebelumnya), tujuannya adalah untuk mempertahankan perbedaan diantara populasi dan mencegah terjadinya konvergensi *premature* (dini). Setelah itu proses optimasi dimulai dari awal lagi dengan populasi yang baru (hasil operasi algoritma genetik generasi sebelumnya). Proses akan berhenti jika iterasi telah mencapai generasi maksimum.

Untuk melihat apakah kinerja dari perangkat lunak sesuai dengan yang diharapkan maka diperlukan suatu pengujian terhadap perangkat lunak ini. Selain itu, juga diberikan hasil pengujian dengan software WinQSB. Data yang digunakan sebagai sampel pengujian dapat dilihat di tabel berikut :



Tabel 6.1 Jarak antar titik

Titik	1	2	3	4	5	6
1	0	1	5	7	9	0
2	1	0	6	4	3	0
3	5	6	0	5	0	10
4	7	4	5	0	8	3
5	9	3	0	8	0	0
6	0	0	10	3	0	0



Tabel 6.2 Posisi koordinat titik pada peta awal

Titik	X	Y
1	2	6
2	4	10
3	6	6
4	5	2
5	10	8
6	10	4

Nilai – nilai parameter genetik yang digunakan saat pengujian adalah :

1. Ukuran Populasi = 50
2. Banyak Generasi = 500
3. Probabilitas Persilangan = 0.4
4. Probabilitas Mutasi = 0.01
5. Metode Persilangan = 5 metode diuji
6. Metode Mutasi = 4 metode diuji

Tabel 6.3 Hasil Pengujian persoalan *minimum spanning tree* dengan algoritma genetik

Uji Ke	Metode	Rentang	Total Sisi	Generasi
1	<i>OX - Inv</i>	3 4 5 2 1 6 0 1 2 4 2 4	18	4
2	<i>OX - Ins</i>	3 4 5 2 1 6 0 1 2 4 2 4	18	4
3	<i>OX - Dis</i>	4 3 5 2 1 6 0 1 2 4 2 4	16	21
4	<i>OX - Rec</i>	3 4 5 2 1 6 0 1 2 4 2 4	18	4
5	<i>OBX - Inv</i>	1 2 5 3 4 6 0 1 2 4 5 4	20	11
6	<i>OBX - Ins</i>	3 5 1 2 4 6 0 1 2 3 2 4	23	13
7	<i>OBX - Dis</i>	5 2 1 6 3 4 0 1 3 4 2 5	23	4
8	<i>OBX - Rec</i>	5 2 1 6 3 4 0 1 3 4 2 3	20	4
9	<i>PBX - Inv</i>	1 3 2 6 5 4 0 1 4 4 2 3	20	16
10	<i>PBX - Ins</i>	5 2 1 6 3 4 0 1 3 4 2 5	23	4
11	<i>PBX - Dis</i>	5 2 1 6 3 4 0 1 3 4 2 5	23	4
12	<i>PBX - Rec</i>	5 2 1 6 3 4 0 1 3 4 2 5	23	4
13	<i>CX - Inv</i>	3 4 5 2 1 6 0 1 2 3 2 4	20	4
14	<i>CX - Ins</i>	3 4 5 2 1 6 0 1 2 3 2 4	20	4
15	<i>CX - Dis</i>	3 4 5 2 1 6 0 1 2 4 2 4	18	3
16	<i>CX - Rec</i>	3 4 5 2 1 6	20	4

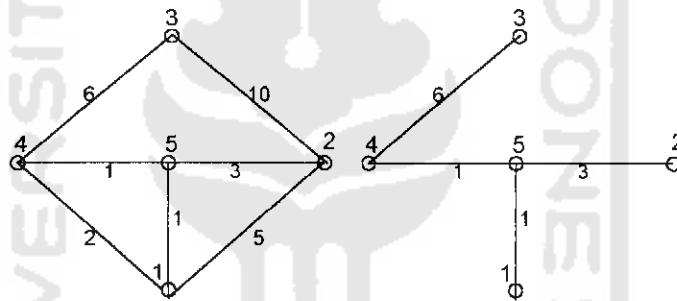
		0 1 2 3 2 4		
17	<i>PMX - Inv</i>	3 4 5 2 1 6 0 1 2 4 2 4	18	4
18	<i>PMX - Ins</i>	3 2 5 1 4 6 0 1 2 3 5 4	20	11
19	<i>PMX - Dis</i>	5 2 1 6 3 4 0 1 3 4 2 5	23	4
20	<i>PMX - Rec</i>	4 3 5 2 1 6 0 1 2 3 2 4	18	9



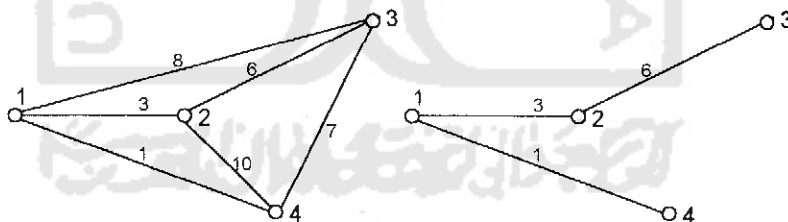
Tabel 6.4 Hasil Pengujian dengan software WinQSB.

10-02-2004	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
1	Node1	Node2	1	4	Node2	Node5	3
2	Node1	Node3	5	5	Node4	Node6	3
3	Node2	Node4	4				
	Total	Minimal	Connected	Distance	or Cost	=	16

Berikut ini diberikan data tambahan untuk penyelesaian persoalan *minimum spanning tree* dengan algoritma genetik.



Gambar 6.6 Graf dan *tree* 5 titik



Gambar 6.7 Graf dan *tree* 4 titik

Dari hasil pengujian dengan data 6 titik persoalan *minimum spanning tree* dengan algoritma genetik dapat memberikan solusi alternatif optimum. Pada gambar 6.6 dan 6.7 terlihat bahwa data-data tersebut juga dapat memberikan solusi alternatif optimum. Pada saat proses *tree* yang dihasilkan menggunakan metode *CrossOver*, mutasi dan parameter genetik yang berbeda-beda.