BAB IV
IMPLEMENTASI DAN PENGUJIAN

IV.1 Implementasi Simulasi

Tahap implementasi sistem merupakan tahap pembuatan perangkat lunak, tahap lanjut dari tahap perancangan sistem. Tahap yang dilakukan untuk menerjemahkan perancangan berdasarkan hasil analisis dalam bahasa yang dimengerti oleh komputer serta penerapan perangkat lunak pada keadaan yang sebenarnya.

IV.1.1 Implementasi Perangkat Keras

Spesifikasi perangkat keras (hardware) yang digunakan untuk membangun perangkat lunak simulasi pemilihan proyek ini tercantum dalam Tabel IV-1 Spesifikasi Implementasi Perangkat Keras berikut ini :

Tabel IV-1 Spesifikasi Implementasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Processor	Pentium Dual Core T4200 2.00 GHz
2	RAM	2 Gb
3	Harddisk	250 Gb
4	VGA	256 Mb
5	Monitor	1024x768

IV.1.2 Implementasi Perangkat Lunak

Spesifikasi perangkat keras (hardware) yang digunakan untuk membangun perangkat lunak simulasi ini tercantum dalam Tabel IV-2 Spesifikasi Implementasi Perangkat Lunak berikut ini :

Tabel IV-2 Spesifikasi Implementasi Perangkat Lunak

No	Perangkat Lunak	Keterangan
1	Windows 8	Sistem Operasi
2	Borland Delphi	Tool Pembangun Perangkat Lunak
3	Visio	Pemodelan Sistem

IV.1.3 Implementasi Antarmuka

Implementasi antarmuka dilakukan untuk setiap tampilan program yang dibangun dalam bentuk file code. Berikut ini adalah implementasi antarmuka untuk admin dan pengguna. Deskripsi implementasi antarmuka pada perangkat lunak simulasi dapat dilihat pada Tabel IV-3 Implementasi Antarmuka.

Tabel IV-3 Implementasi Antarmuka

No	Form Antarmuka	Proses	Nama File	Keterangan
1	Jumlah SDM di Perusahaan	Inputan jumlah SDM yang tersedia di perusahaan	Unit_SDM.dfm	Tampilan untuk menginputkan data jumlah SDM yang tersedia di perusahaan
2	Form Input Data Proyek	Proses inputan data proyek	UnitPemilihanKnapsack.dfm	Tampilan untuk menginputkan data proyek
3	Form Tampil Data Perhitungan	Proses tampil data	Unit_Hasil.dfm	Tampilan untuk menampilkan hasil

		perhitungan knapsack		perhitungan knapsack.

Gambar implementasi antarmuka dapat dilihat pada lampiran implementasi bagian B.

IV.2 Pengujian Simulasi

Pengujian dilakukan agar diketahui bahwa algoritma *knapsack* dalam menyelesaikan pemilihan proyek berhasil atau tidak. Pengujian dilakukan untuk mendapatkan hasil yang akurat serta algoritma yang digunakan akan diimplementasikan pada simulasi untuk mengetahui bagaimana performansi algoritma dalam mengeksekusi untuk mendapatkan solusi optimal.

Pengujian merupakan hal terpenting yang bertujuan untuk menemukan kesalahan-kesalahan atau kekurangan-kekurangan pada perangkat lunak yang akan diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

Pada tahap pengujian simulasi perangkat lunak ini terdiri dari beberapa bagian yaitu :

1. Rencana Pengujian
2. Pengujian *White Box*
3. Pengujian *Black Box*
4. Kesimpulan Pengujian

IV.2.1 Rencana Pengujian

Rencana pengujian perangkat lunak ini dibagi menjadi 2 bagian yaitu pengujian *black box* dan pengujian *white box.*

Pengujian *white box* dilakukan untuk menguji kesalahan logik dan asumsi yang tidak tepat pada kemungkinan eksekusi. Pengembangan kode-kode program selalu memungkinkan terjadinya alur program yang tidak dan tereksekusi dan

kesalahan *typography* yang sulit ditemukan kalau tidak dijalankan. Pada umumnya pengujian ini dilakukan hanya pada lingkungan developer.

Pengujian *black box* dilakukan untuk mencari fungsi-fungsi program yang tidak benar atau hilang. Kesalahan lain yang mungkin terjadi dalam pembuatan program adalah kesalahan dalam struktur data, kesalahan antar muka, inisialisasi dan akhir program dan kesalahan performansi.

Strategi pengujian akan dilakukan dengan menggunakan strategi pengujian *Alpha*. Strategi pengujian alpha bertujuan untuk mengetahui kecocokkan fungsi-fungsi program apakah sudah berjalan sesuai kebutuhan atau tidak. Strategi pengujian ini dilakukan dengan cara mendampingi pengguna serta mencatat kesalahan-kesalahan maupun permasalahan dalam hal *usability* sebelum aplikasi di release ke pengguna. Deskripsi rencana pengujian dapat dilihat dalam Tabel IV-4.

Tabel IV-4 Rencana Pengujian

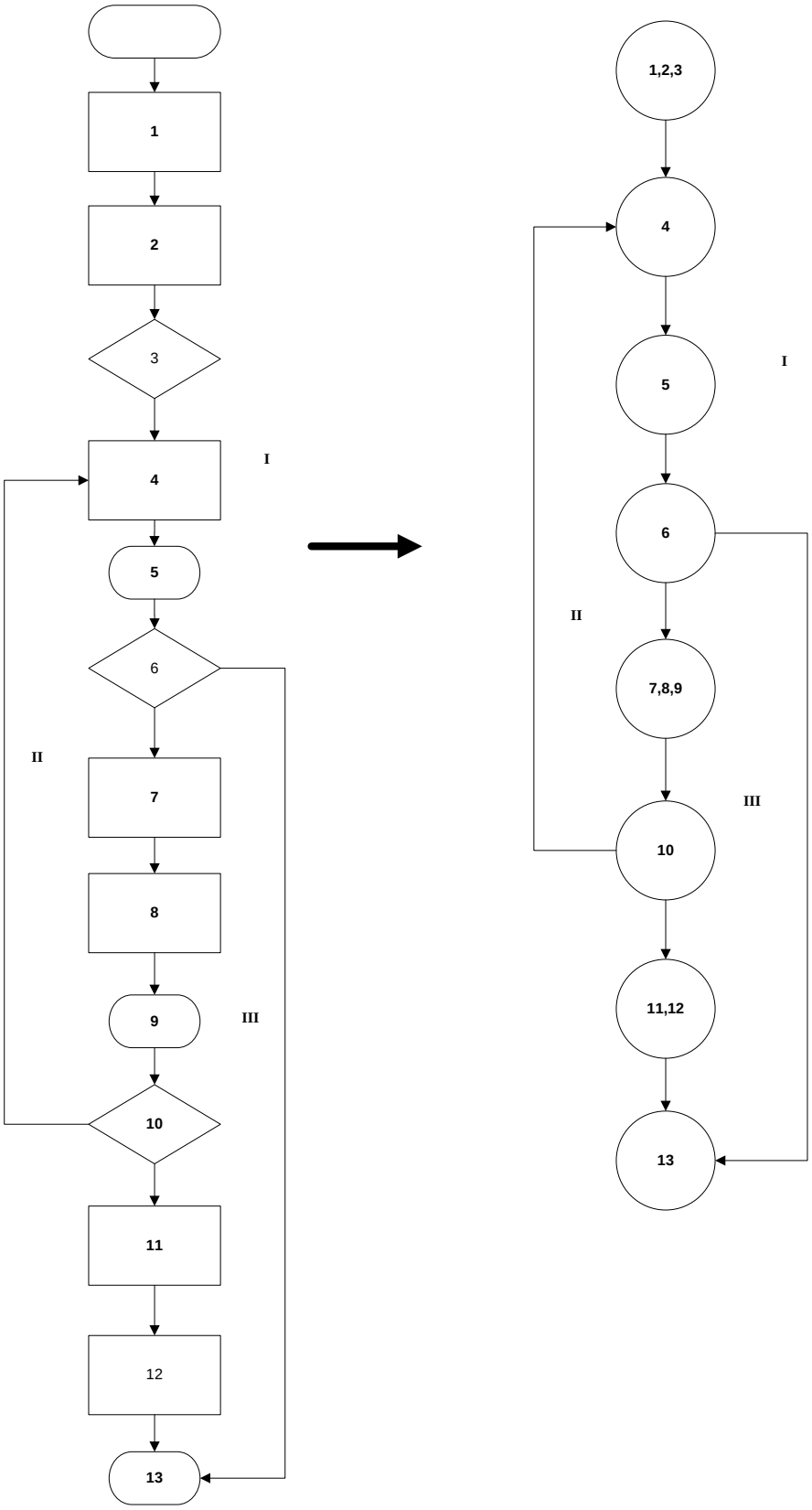
No	Item Uji	Detail Pengujian	Jenis Uji
1	Proses optimalisasi algoritma knapsack	Uji algoritma	White box
2	Validasi data masukkan profit dan SDM	Profit	Black Box
		Jumlah SDM yang dibutuhkan	
		Density	
3	Eksekusi data / pemilihan solusi optimum proyek	Status solusi optimal proyek	Black Box
		Total profit optimum	Black Box

IV.2.2 Pengujian White Box

Pengujian *white box* bertujuan untuk mengetahui kinerja logika yang dibuat pada sebuah perangkat lunak apakah berjalan dengan baik atau tidak. Pengujian *white box* akan digunakan pada algoritma *fractional knapsack*, untuk mengukur kinerja logika berdasarkan *pseudocode* yang telah dibuat pada tahap analisis. Pengujian *white box* dilakukan dengan beberapa tahap yaitu :

1. Pengubahan pseucode menjadi *flowchart* kemudian menjadi *flowgraph.*

1	Opt_sol←0
2	i←1
3	for i←1 to n do
4	optimum[i] ← 0
5	Endfor
6	if SDM[i] > k
	then Eof
7	Else optimum[i] ← 1
8	k←k-SDM[i]
9	Endif
10	if i<= n
11	then optimum[i] ← k/SDM[i]
12/13	opt_sol←0 (asumsikan ada endif)



Gambar IV-1 Pengubahan Flowchart menjadi Flowgraph algoritma *Knapsack*

2. Cyclometic Complexity

   V(G) = E – N + 2

   V(G) = 14 – 13 + 2

   V(G) = 3

   E = Jumlah Busur pada flow graph

   N = Jumlah Simpul pada flow graph

3. Independent Path

   Path 1 : 1,2,3,4,5,6,7,8,9,10,11,12,13

   Path 2 : 1,2,3,4,5,6,13

   Path 3 : 1,2,3,4,5,6,7,8,9,10,4,5,6,13

4. Hitung Jumlah region

   Jumlah Region = Jumlah Kurva Tertutup + 1 (Region Terluar)

   Jumlah Region = 2 + 1

   Jumlah Region = 3

5. Hitung jumlah predicate Node

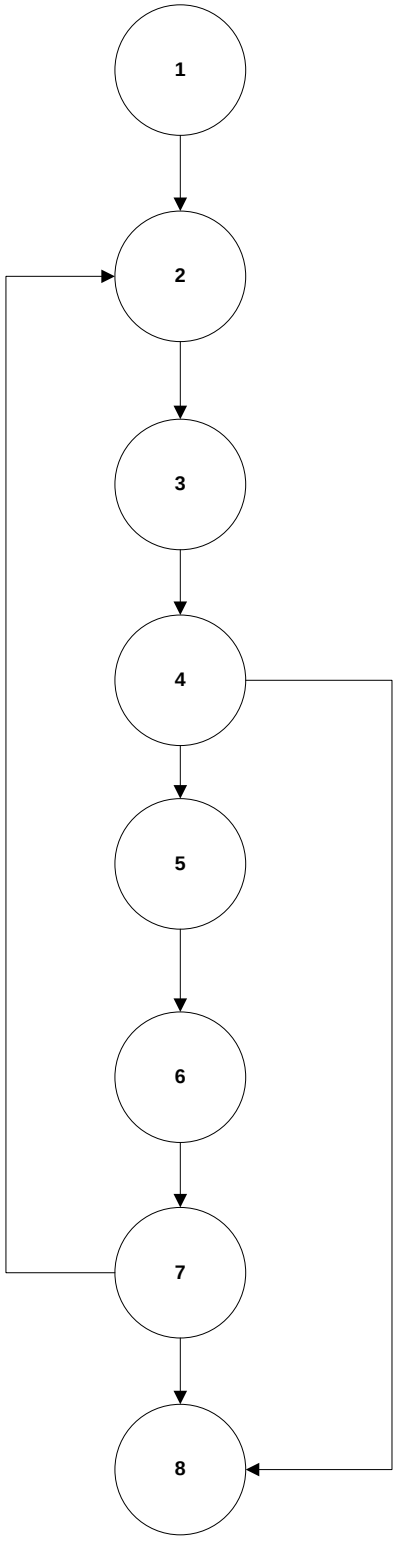
   Predicate Node = 6, 10

   V(G) = Jumlah Predicate Node + 1

   V(G) = 2 + 1

   V(G) = 3

6. Penomoran ulang flowgraph



Gambar IV-2 Penomoran Ulang Graph

7. Graph matriks

Node	1	2	3	4	5	6	7	8	Jumlah
1		1							0
2			1						0
3				1					0
4					1			1	1
5						1			0
6							1		0
7		1						1	1
8				1					0
Jumlah									2

V(g) = x + 1

V(g) = 2 + 1

V(g) = 3

8. Kesimpulan Pengujian White Box

Kesimpulan pengujian white box dilakukan setelah dilakukannya langkah uji kasus yang terakhir yaitu memeriksa kondisi alur algoritma. Pengujian kondisi alur algoritma akan diperlihatkan pada Tabel IV-5.

Tabel IV-5 Pengujian alur kondisi algoritma *knapsack*

No	Kasus Uji	Hasil yang Diharapkan	Hasil Sesuai Uji Kasus	Keterangan
1	Memasukkan nilai n=4 ke dalam kondisi : for i←1 to n do optimum[i] ← 0	Menampilkan hasil nilai array 0 untuk optimum[1] s.d optimum[4]	Nilai array untuk optimum[1] s.d optimum[4]yang dihasilkan adalah 0.	[✓] Alur Terlewati [ ] Alur Tidak Terlewati
2	Memasukkan nilai SDM[i]=2, k=5 ke dalam kondisi : if SDM[i] > k then Eof Else optimum[i] ← 1 k←k-SDM[i] Endif	Menampilkan hasil optimum[i]=1 dan k=5-2 k=3	Nilai yang dihasilkan adalah optimum[i]=1 dan k=5-2 k=3	[✓] Alur Terlewati [ ] Alur Tidak Terlewati
3	Memasukkan nilai i=2, n=3, k=2 dan SDM[i]=4 ke dalam kondisi : if i<= n then optimum[i]←k/SDM[i] opt_sol←0	Menampilkan hasil optimum[i]=2/4 optimum[i]=0,5 opt_sol=0	Nilai yang dihasilkan adalah : optimum[i]=2/4 optimum[i]=0,5 opt_sol=0	[✓] Alur Terlewati [ ] Alur Tidak Terlewati

Berdasarkan pengujian pada setiap metode, dihasilkan nilai Cyclomatic Complexity yang sama yaitu 3. Maka dapat disimpulkan bahwa pengujian *white box* pada algoritma *knapsack* berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

IV.2.3 Pengujian Black Box

Berdasarkan rencana pengujian yang telah disusun maka dapat dilakukan pengujian black box yang disajikan pada Tabel IV-6.

Tabel IV-6 Pengujian *Black Box*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Hasil yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Form data proyek berupa edit teks profit dan edit teks SDM diisi dengan : 12 dan 1. Kemudian tombol masukkan ke daftar ditekan	Data proyek =12 dan SDM = 1 masuk ke dalam list	Masuk data proyek=12 dan data SDM=1 ke dalam daftar proyek yang akan dieksekusi	[✓] Diterima [ ] Ditolak
Form jumlah SDM yang tersedia diisi dengan 2 kemudian tombol eksekusi ditekan	Data proyek optimal berupa status proyek dan total profit optimum akan muncul	Muncul data proyek optimal berupa status proyek dan total profit optimum	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Kosong)			
Data Masukan	**Hasil yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Edit text profit dan SDM dikosongkan secara bergiliran kemudian tombol masukkan ke daftar ditekan	Muncul pesan kesalahan dikarenakan data yang dimasukkan tidak valid	Dapat menampilkan kesalahan apabila user tidak memasukkan data yang valid	[✓] Diterima [ ] Ditolak

Edit text jumlah SDM yang tersedia dikosongkan ditekan kemudian tombol eksekusi ditekan	Muncul pesan kesalahan dikarenakan data yang dimasukkan tidak valid	Dapat menampilkan kesalahan apabila data yang dimasukkan tidak valid	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Hasil yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Edit text profit dan *Edit text* SDM dimasukkan nilai yang salah, secara bergiliran kemudian tombol masukkan ke daftar ditekan	Muncul pesan kesalahan dikarenakan data yang dimasukkan tidak valid	Dapat menampilkan kesalahan apabila user tidak memasukkan data yang valid	[✓] Diterima [ ] Ditolak
Edit text jumlah SDM yang tersedia diisi dengan nilai yang bukan integer kemudian tombol eksekusi ditekan	Muncul pesan kesalahan dikarenakan data yang dimasukkan tidak valid	Dapat menampilkan kesalahan apabila data yang dimasukkan tidak valid	[✓] Diterima [ ] Ditolak

IV.3.2.1. Pengujian Performansi

Pada pengujian performansi ini akan dilakukan pencocokan perhitungan matematis dengan percobaan pemilihan proyek optimal dengan menggunakan perangkat lunak simulasi. Data uji yang digunakan berasal dari PT. GITS Indonesia. Data uji yang didapatkan nilai profitnya disamarkan dan diasumsikan

menjadi rasio profit, namun perbandingan nilai antara tiap profit proyek adalah hasil perhitungan dari data nyata. Pengujian akan dilakukan dua kali yaitu dengan data bulan pertama dan data bulan kedua. Diketahui jumlah SDM yang terdapat di perusahaan adalah sebanyak 10 orang. Kemudian pengujian sample data pertama dapat dilihat pada Tabel IV-7

Tabel IV-7 Pegujian Sample Data Bulan Pertama, Pemilihan Proyek dengan Algoritma Fractional Knapsack Menggunakan Pendekatan Greedy By Density

Properti Proyek				***Fractional Knapsack With Greedy by density***		**Kesimpulan**
i	w_i	p_i	p_i/w_i	**Solusi Optimal Dengan Perhitungan Matematis**	**Pemilihan dengan Simulasi**	
1	2	50000	25000	1	1	[✓] Diterima [ ] Ditolak
2	3	51000	17000	0	0	[✓] Diterima [ ] Ditolak
3	2	45000	22500	1	1	[✓] Diterima [ ] Ditolak
4	4	123000	30750	1	1	[✓] Diterima [ ] Ditolak
5	2	40000	20000	0	0	[✓] Diterima [ ] Ditolak
6	2	56000	28000	1	1	[✓] Diterima [ ] Ditolak
Total SDM Terpakai				**10**	**10**	[✓] Diterima [ ] Ditolak
Total keuntungan				**274000**	**274000**	[✓] Diterima [ ] Ditolak

Waktu Eksekusi Perangkat Lunak Simulasi	**1197 milisecond**	
Kebutuhan Memori Saat Eksekusi	**7.468 KB**	

Berdasarkan dari tabel Tabel IV-7 dapat dilihat bahwa pada pengujian data sample bulan pertama berhasil menghasilkan solusi proyek yang optimal. Solusi optimal untuk data diatas adalah (1, 0, 1, 1, 0, 1) dimana nilai 1 (satu) berarti proyek ke-i terpilih sebagai solusi optimal. Nilai 0 (nol) memiliki arti proyek ke-i tidak terpilih sebagai solusi optimal.

Pengujian data bulan kedua akan dilakukan dengan jumlah sample data sebanyak lima proyek. pengujian dapat dilihat pada Tabel IV-8.

Tabel IV-8 Pegujian Sample Data Bulan Kedua, Pemilihan Proyek dengan Algoritma Fractional Knapsack Menggunakan Pendekatan Greedy By Density

Properti Proyek				***Fractional Knapsack With Greedy by density***		**Kesimpulan**
i	w_i	p_i	p_i/w_i	**Solusi Optimal Dengan Perhitungan Matematis**	**Pemilihan dengan Simulasi**	
1	2	10000	5000	1	1	[✓] Diterima [ ] Ditolak
2	3	45000	15000	1	1	[✓] Diterima [ ] Ditolak
3	2	37250	18625	1	1	[✓] Diterima [ ] Ditolak
4	3	12500	4166,667	0	0	[✓] Diterima [ ] Ditolak
5	3	41600	13866,67	1	1	[✓] Diterima

						[ ] Ditolak
Total SDM Terpakai				**10**	**10**	[✓] Diterima [ ] Ditolak
Total keuntungan				**133850**	**133850**	[✓] Diterima [ ] Ditolak
Waktu Eksekusi Perangkat Lunak Simulasi				**1312 milisecond**		
Kebutuhan Memori Saat Eksekusi				**7.440 KB**		

Pengujian data proyek bulan kedua lagi-lagi menghasilkan solusi optimal. Solusi optimal untuk data uji kedua ini adalah (0, 1, 1, 1, 0, 1). Total keuntungan yang dihasilkan adalah sebesar 133850.

IV.2.4 Kesimpulan Pengujian

Berdasarkan hasil pengujian terhadap perangkat lunak simulasi dapat ditarik kesimpulan perangkat lunak simulasi dapat berfungsi sesuai dengan yang diharapkan serta perangkat lunak simulasi yang dibangun telah cukup memenuhi tujuan awal pembangunan dan algoritma *knapsack* berhasil menampilkan hasil pemilihan proyek yang optimal.