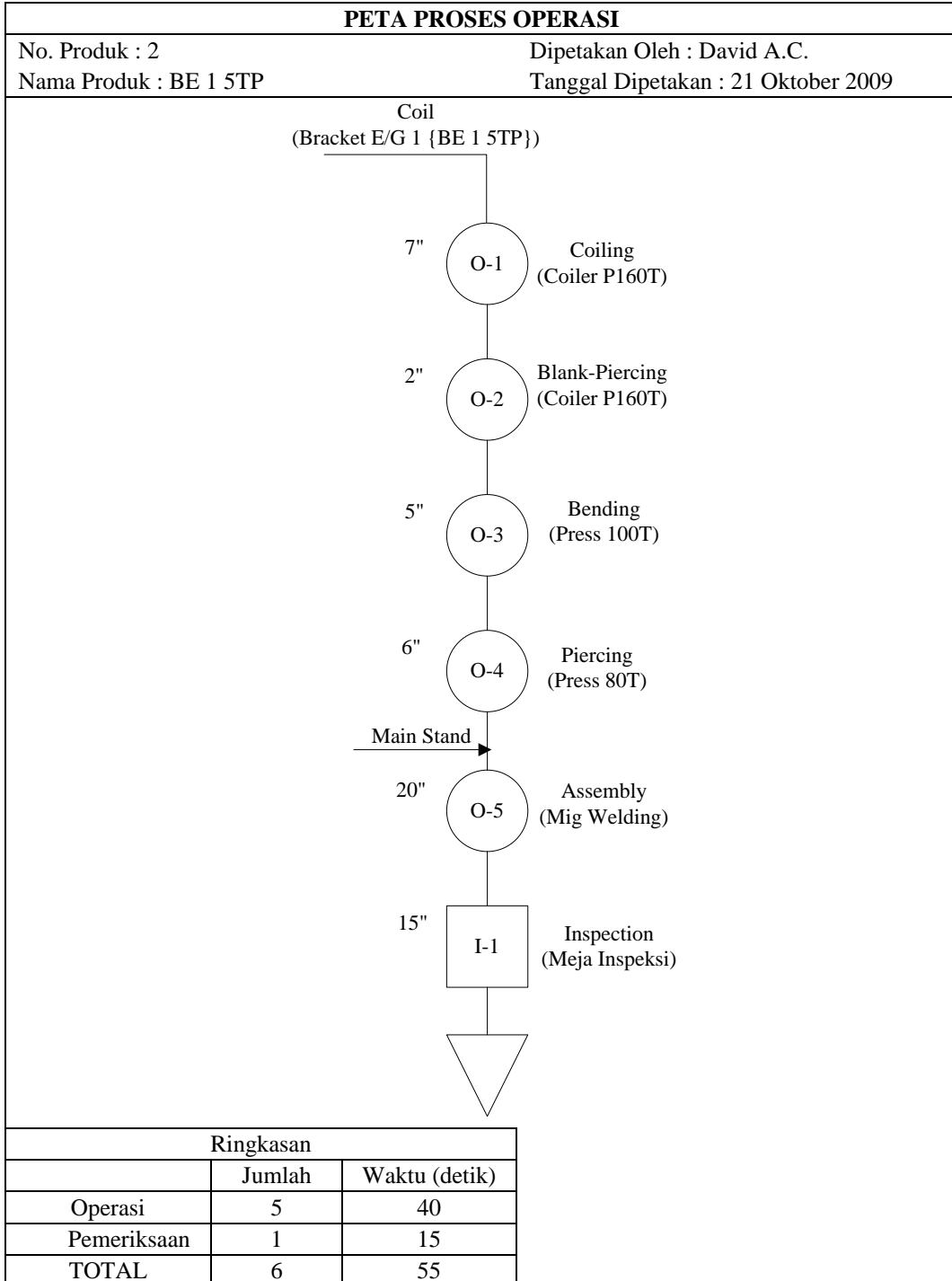

LAMPIRAN

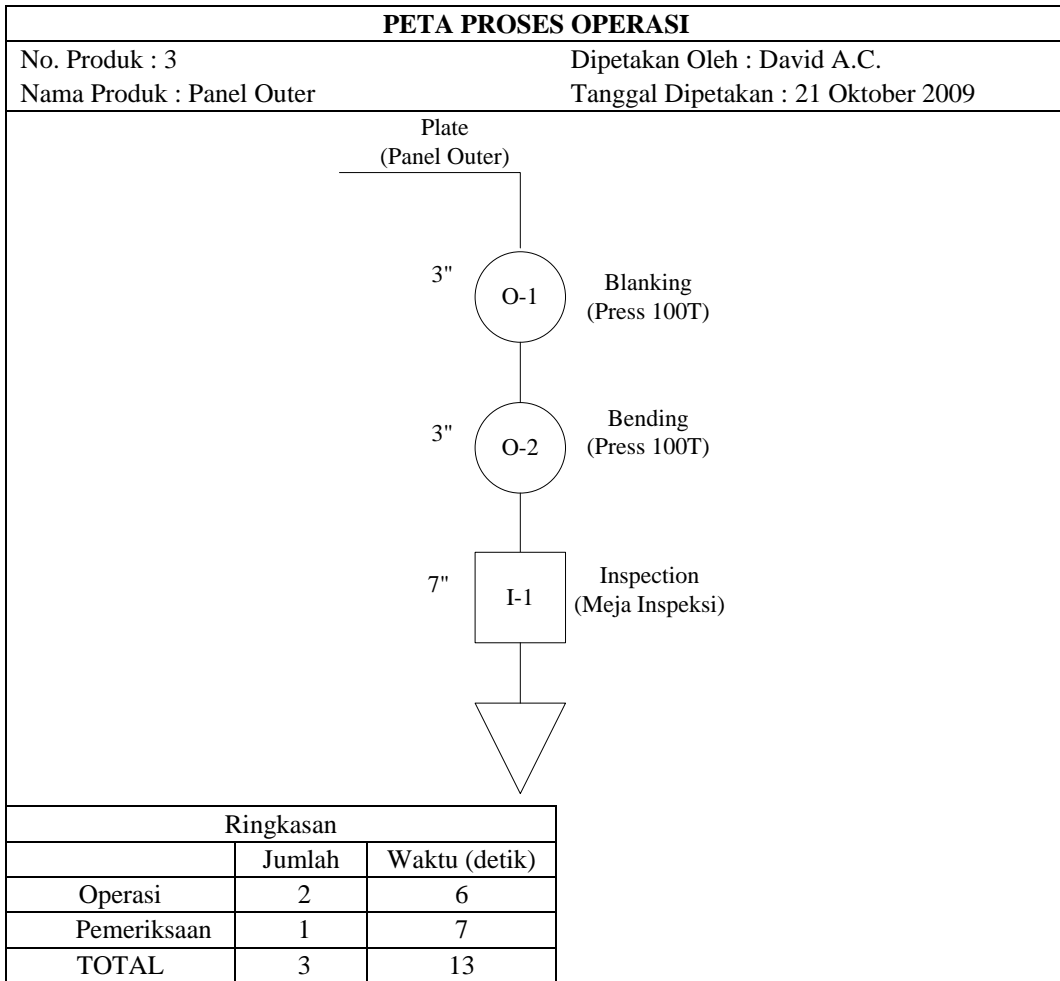
LAMPIRAN 1

- Peta Proses Operasi
- Struktur Produk
(Produk BE 1 5TP, Panel Outer, Panel Inner, dan Stay 5D9)

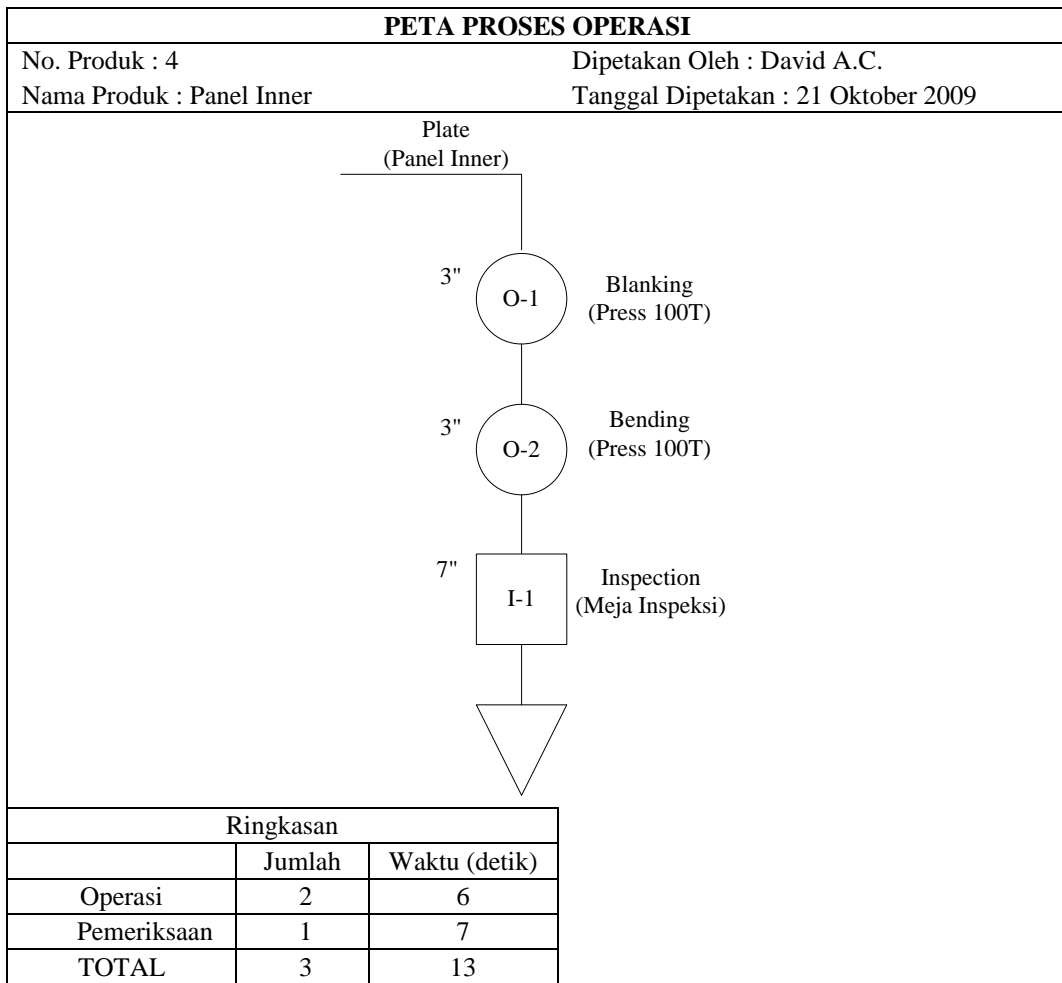
- Peta Proses Operasi Produk BE 1 5TP



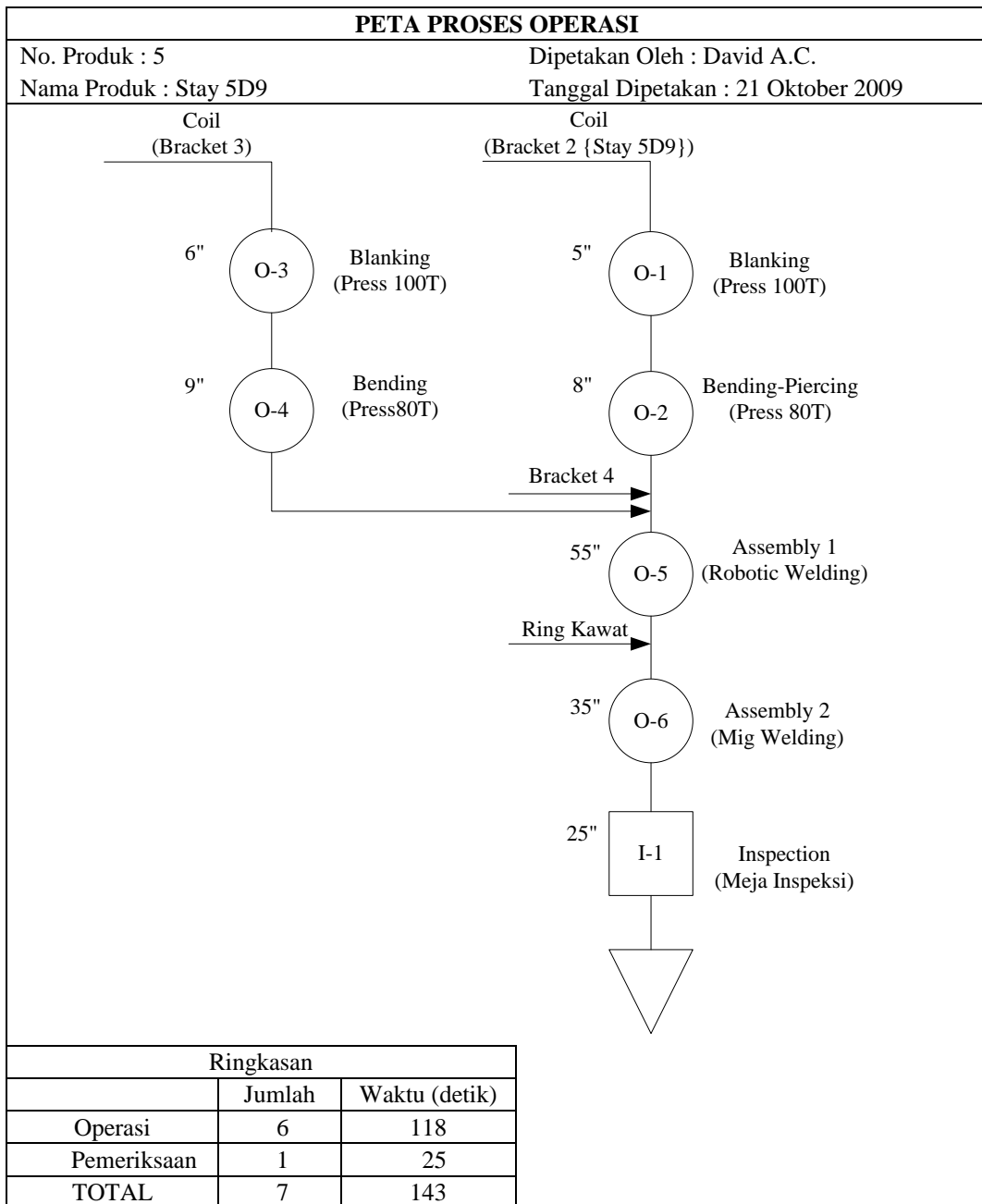
- Peta Proses Operasi Produk Panel Outer



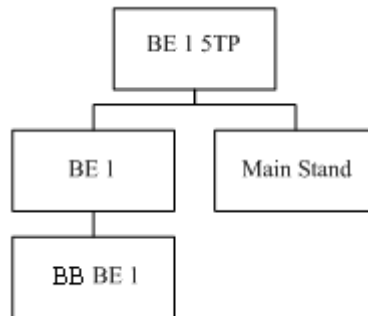
• Peta Proses Operasi Produk Panel Inner



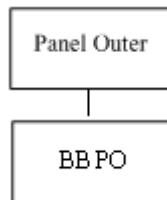
• Peta Proses Operasi Produk Stay 5D9



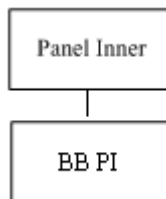
- Struktur Produk BE 1 5TP



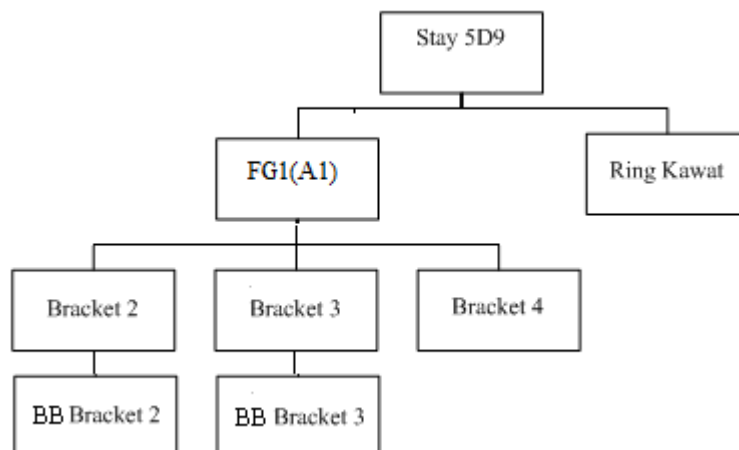
- Struktur Produk Panel Outer



- Struktur Produk Panel Inner



- Struktur Produk Stay 5D9



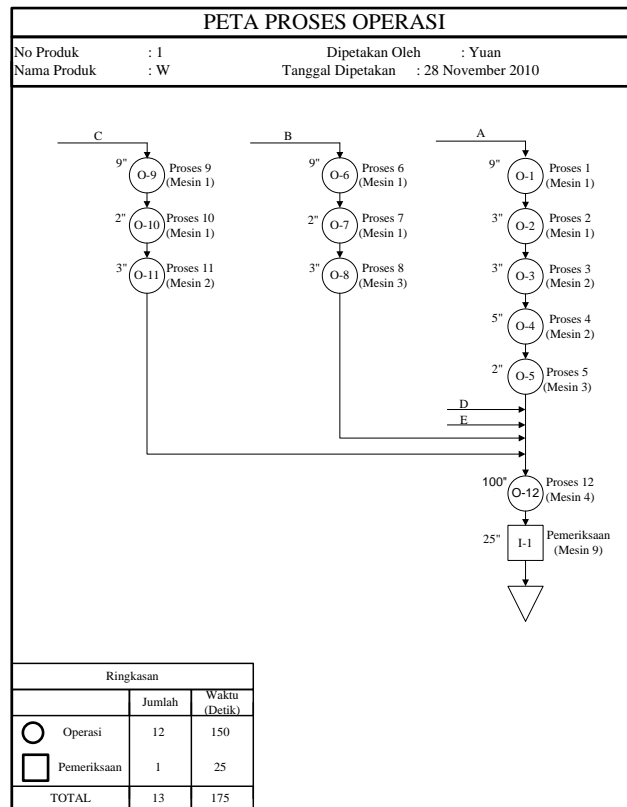
LAMPIRAN 2

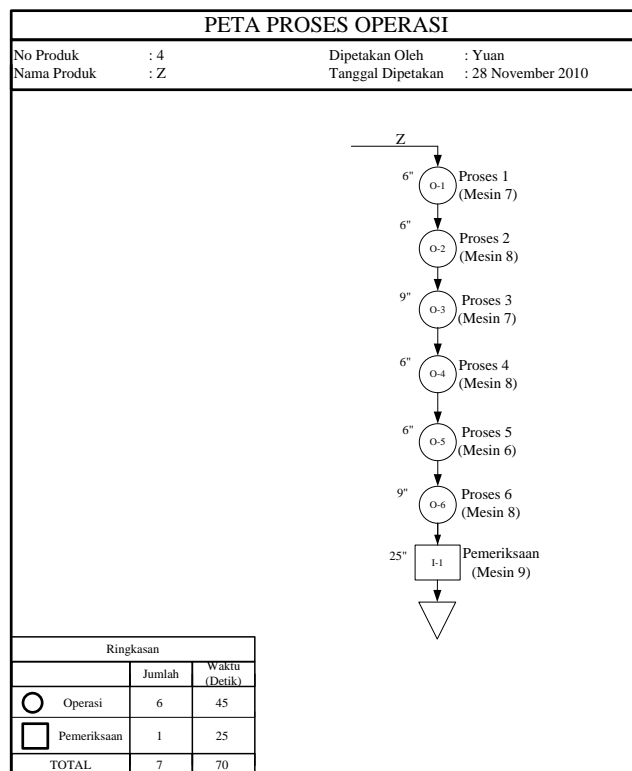
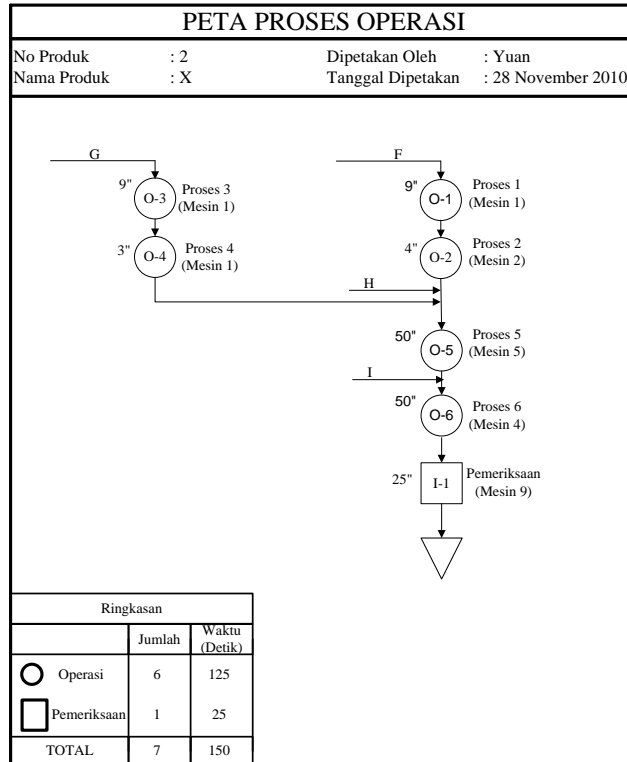
- Langkah-langkah Pembuatan Kasus Sederhana

Langkah-langkah pembuatan kasus sederhana yang akan dilakukan untuk validasi adalah sebagai berikut :

1. Pembuatan Peta Proses Operasi

Peta Proses Operasi (PPO / OPC) menunjukkan urutan proses suatu produk. Terdapat 3 (tiga) produk pada Kasus Sederhana ini, antara lain produk W, produk X, dan produk Z.





2. Pembuatan *Routing Sheet*

Routing Sheet dibuat untuk mengetahui jumlah mesin yang dibutuhkan untuk setiap proses.

ROUTING SHEET							
Nomor Produk : 1							
Nama Produk : W							
Jam Kerja Efektif : 40 jam/minggu = 144000 detik/minggu							
Jumlah Produksi : 3600 unit/ming							
No	Item	No Operasi	Nama Operasi	Mesin	Waktu Proses (detik)	Kapasitas Mesin (unit/minggu)	Jumlah Mesin (F)
1	A	O-1	Proses 1	Mesin 1	9	16,000	0.225
		O-2	Proses 2	Mesin 1	3	48,000	0.075
		O-3	Proses 3	Mesin 2	3	48,000	0.075
		O-4	Proses 4	Mesin 2	5	28,800	0.125
		O-5	Proses 5	Mesin 3	2	72,000	0.050
2	B	O-6	Proses 6	Mesin 1	9	16,000	0.225
		O-7	Proses 7	Mesin 1	2	72,000	0.050
		O-8	Proses 8	Mesin 2	3	48,000	0.075
3	C	O-9	Proses 9	Mesin 1	9	16,000	0.225
		O-10	Proses 10	Mesin 1	2	72,000	0.050
		O-11	Proses 11	Mesin 3	3	48,000	0.075
4	Perakitan	O-12	Proses 12	Mesin 4	100	1,440	2.500
5	Pemeriksaan	I-1	Pemeriksaan	Mesin 9	25	5,760	0.625

ROUTING SHEET							
Nomor Produk : 2							
Nama Produk : X							
Jam Kerja Efektif : 40 jam/minggu = 144000 detik/minggu							
Jumlah Produksi : 4500 unit/ming							
No	Item	No Operasi	Nama Operasi	Mesin	Waktu Proses (detik)	Kapasitas Mesin (unit/minggu)	Jumlah Mesin (F)
1	F	O-1	Proses 1	Mesin 1	9	16,000	0.281
		O-2	Proses 2	Mesin 2	4	36,000	0.125
2	G	O-3	Proses 3	Mesin 1	9	16,000	0.281
		O-4	Proses 4	Mesin 1	3	48,000	0.094
3	Perakitan 1	O-5	Proses 5	Mesin 5	50	2,880	1.563
4	Perakitan 2	O-6	Proses 6	Mesin 4	50	2,880	1.563
5	Pemeriksaan	I-1	Pemeriksaan	Mesin 9	25	5,760	0.781

ROUTING SHEET							
Nomor Produk : 3							
Nama Produk : Z							
Jam Kerja Efektif : 40 jam/minggu = 144000 detik/minggu							
Jumlah Produksi : 4200 unit/ming							
No	Item	No Operasi	Nama Operasi	Mesin	Waktu Proses (detik)	Kapasitas Mesin (unit/minggu)	Jumlah Mesin (F)
1	Z	O-1	Proses 1	Mesin 7	9	16,000	0.263
2		O-2	Proses 2	Mesin 8	6	24,000	0.175
3		O-3	Proses 3	Mesin 7	9	16,000	0.263
4		O-4	Proses 4	Mesin 8	6	24,000	0.175
5		O-5	Proses 5	Mesin 6	6	24,000	0.175
6		O-6	Proses 6	Mesin 8	9	16,000	0.263
7		I-1	Pemeriksaan	Mesin 9	25	5,760	0.729

Contoh Perhitungan (Produk Z, Operasi O-1) :

- Kapasitas Mesin = $\frac{\text{Jam Kerja Efektif}}{\text{Waktu Proses}} = \frac{144,000 \text{ detik/minggu}}{9 \text{ detik/unit}}$
= 16000 unit/minggu
- Jumlah Mesin = $\frac{\text{Jumlah Produksi(unit/minggu)}}{\text{Kapasitas Mesin(unit/minggu)}} = \frac{18,000 \text{ unit/minggu}}{16,000 \text{ unit/minggu}} = 1.125$

3. Perhitungan Kebutuhan Mesin

Perhitungan jumlah mesin dilakukan untuk menghitung total mesin yang disediakan untuk memproduksi keseluruhan item atau produk, sedangkan perhitungan fraktal dilakukan untuk menghitung jumlah fraktal yang dibentuk untuk mengalokasikan mesin-mesin yang ada.

Mesin	Produk			Kebutuhan Mesin	Jumlah Disediakan
	1	2	3		
Mesin 1	0.850	0.656		1.506	2
Mesin 2	0.275	0.125		0.400	1
Mesin 3	0.125			0.125	1
Mesin 4	2.500	1.563		4.063	5
Mesin 5		1.563		1.563	2
Mesin 6			0.175	0.175	1
Mesin 7			0.525	0.525	1
Mesin 8			0.613	0.613	1
Mesin 9	0.625	0.781	0.729	2.135	3
	Total Mesin				17

Contoh Perhitungan (Mesin 4) :

- Kebutuhan Mesin = $2.500 + 1.563 = 4.063$ unit
- Jumlah Disediakan = $4.063 \approx 5$ unit.

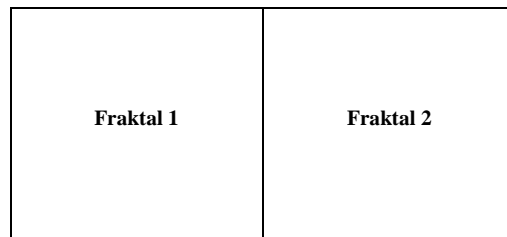
Jumlah mesin yang disediakan selalu dibulatkan ke atas.

4. Penentuan Jumlah Fraktal

Selanjutnya, mesin-mesin tersebut (total mesin = 17 mesin) akan dialokasikan ke dalam yang terbentuk. Perhitungan jumlah fraktal yang terbentuk adalah sebagai berikut :

$$\begin{aligned} \text{Jumlah fraktal} &= \text{Total mesin} / \text{jumlah jenis mesin} \\ &= 17 / 9 \\ &= 1.889 \approx 2 \text{ fraktal} \end{aligned}$$

Jumlah fraktal yang terbentuk dibulatkan ke atas ($1.889 \approx 2$) agar pengalokasian mesin seimbang (dilihat berdasarkan jumlah masing-masing mesin).



5. Mengalokasikan Mesin pada Tiap Fraktal
6. Penentuan Jarak antar Mesin

LAMPIRAN 3

- Decoding Kromosom

Lampiran 3

Decoding Kromosom Offspring Mutasi 2

Table with columns: Item, From, To, Jarak (m), Kapasitas Tersedia (detik/minggu), Jumlah Produksi (unit/minggu), Waktu Proses (detik/unit), Kapasitas Produksi (unit/minggu), Sisa Produksi (unit/minggu), Sisa Kapasitas (detik/minggu), Unit diproduksi (unit/minggu), Kebutuhan Material (unit/minggu), Alat/MH, Kapasitas MH (Volume, Berat), Frekuensi Terpilih, Cost (rupiah), and OMH (rupiah).

LAMPIRAN 4

- Penugasan Kromosom terpilih

- Penugasan Kromosom Terpilih Kasus Sederhana, dengan total OMH Rp 47,700.70.

Item	Material	Kebutuhan	Produksi	Volume	Weight	Dari	Ke	Kapasitas	Waktu	JumProduksi	JumPindah	Frekuensi	Jarak	Handling	Cost	Total OMH
FG3(A1)	BB Z	4200	4200	13	26	0-0-0	2-Mesin 7-1	144000	9	4200	2100	162	16	Pallet	1.98	5132.16
FG3(A1)	FG3(A1)	4200	4200	26	13	2-Mesin 7-1	2-Mesin 8-1	144000	6	4200	4200	324	5	Keranjang	0.42	680.4
FG3(A1)	FG3(A1)	4200	4200	26	13	2-Mesin 8-1	2-Mesin 7-1	106200	9	4200	4200	324	5	Keranjang	0.42	680.4
FG3(A1)	FG3(A1)	4200	4200	26	13	2-Mesin 7-1	2-Mesin 8-1	118800	6	4200	4200	324	5	Keranjang	0.42	680.4
FG3(A1)	FG3(A1)	4200	4200	26	13	2-Mesin 8-1	2-Mesin 6-1	144000	6	4200	4200	324	6	Keranjang	0.42	816.48
FG3(A1)	FG3(A1)	4200	4200	26	13	2-Mesin 6-1	2-Mesin 8-1	93600	9	4200	4200	324	6	Keranjang	0.42	816.48
FG3(A1)	FG3(A1)	4200	4200	26	13	2-Mesin 8-1	1-Mesin 9-2	144000	25	4200	4200	324	5	Keranjang	0.42	680.4
FG3(A1)	FG3(A1)	4200	4200	30	26	1-Mesin 9-2	1-Warehouse-1	1000000000	0	4200	4200	162	11	Pallet	1.98	3528.36
C	BB C	3600	3600	2	1	0-0-0	2-Mesin 1-1	144000	11	3600	10	10	16	Pallet	1.98	316.8
C	C	3600	3600	13	12	2-Mesin 1-1	1-Mesin 3-1	144000	3	3600	3600	300	10	Keranjang	0.42	1260
B	BB B	3600	3600	2	1	0-0-0	2-Mesin 1-1	104400	11	3600	17	17	16	Pallet	1.98	538.56
B	B	3600	3600	27	28	2-Mesin 1-1	1-Mesin 2-1	144000	3	3600	3600	134	12	Keranjang	0.42	675.36
A	BB A	3600	3600	1	1	0-0-0	1-Mesin 1-1	144000	12	3600	45	45	10	Pallet	1.98	891
A	A	3600	3600	25	23	1-Mesin 1-1	1-Mesin 2-1	133200	8	3600	3600	157	5	Keranjang	0.42	329.7
A	A	3600	3600	25	23	1-Mesin 2-1	1-Mesin 3-1	133200	2	3600	3600	157	5	Keranjang	0.42	329.7
FG1(A1)	A	3600	3600	25	23	1-Mesin 3-1	1-Mesin 4-3	144000	100	1440	1440	63	6	Keranjang	0.42	158.76
FG1(A1)	B	3600	3600	27	28	1-Mesin 2-1	1-Mesin 4-3	144000	100	1440	1440	54	6	Keranjang	0.42	136.08
FG1(A1)	C	3600	3600	13	12	1-Mesin 3-1	1-Mesin 4-3	144000	100	1440	1440	120	6	Keranjang	0.42	302.4
FG1(A1)	D	3600	3600	13	12	0-0-0	1-Mesin 4-3	144000	100	1440	1440	120	16	Keranjang	0.42	806.4
FG1(A1)	E	3600	3600	13	12	0-0-0	1-Mesin 4-3	144000	100	1440	1440	120	16	Keranjang	0.42	806.4
FG1(A1)	A	3600	3600	25	23	1-Mesin 3-1	1-Mesin 4-1	144000	100	1440	1440	63	5	Keranjang	0.42	132.3
FG1(A1)	B	3600	3600	27	28	1-Mesin 2-1	1-Mesin 4-1	144000	100	1440	1440	54	6	Keranjang	0.42	136.08
FG1(A1)	C	3600	3600	13	12	1-Mesin 3-1	1-Mesin 4-1	144000	100	1440	1440	120	5	Keranjang	0.42	252
FG1(A1)	D	3600	3600	13	12	0-0-0	1-Mesin 4-1	144000	100	1440	1440	120	12	Keranjang	0.42	604.8
FG1(A1)	E	3600	3600	13	12	0-0-0	1-Mesin 4-1	144000	100	1440	1440	120	12	Keranjang	0.42	604.8
FG1(A1)	A	3600	3600	25	23	1-Mesin 3-1	2-Mesin 4-1	144000	100	720	720	32	10	Keranjang	0.42	134.4
FG1(A1)	B	3600	3600	27	28	1-Mesin 2-1	2-Mesin 4-1	144000	100	720	720	27	10	Keranjang	0.42	113.4
FG1(A1)	C	3600	3600	13	12	1-Mesin 3-1	2-Mesin 4-1	144000	100	720	720	60	10	Keranjang	0.42	252
FG1(A1)	D	3600	3600	13	12	0-0-0	2-Mesin 4-1	144000	100	720	720	60	18	Keranjang	0.42	453.6
FG1(A1)	E	3600	3600	13	12	0-0-0	2-Mesin 4-1	144000	100	720	720	60	18	Keranjang	0.42	453.6
FG1(A1)	FG1(A1)	3600	3600	13	12	1-Mesin 4-3	2-Mesin 9-1	144000	25	3600	1440	120	8	Keranjang	0.42	403.2
FG1(A1)	FG1(A1)	3600	3600	13	12	1-Mesin 4-1	2-Mesin 9-1	144000	25	3600	1440	120	7	Keranjang	0.42	352.8
FG1(A1)	FG1(A1)	3600	3600	13	12	2-Mesin 4-1	2-Mesin 9-1	144000	25	3600	720	60	5	Keranjang	0.42	126
FG1(A1)	FG1(A1)	3600	3600	24	23	2-Mesin 9-1	1-Warehouse-1	1000000000	0	3600	3600	157	10	Pallet	1.98	3108.6
F	BB F	4500	4500	1	1	0-0-0	1-Mesin 1-1	100800	9	4500	53	53	10	Pallet	1.98	1049.4
F	F	4500	4500	9	11	1-Mesin 1-1	1-Mesin 2-1	104400	4	4500	4500	500	5	Keranjang	0.42	1050
G	BB G	4500	4500	1	1	0-0-0	1-Mesin 1-1	60300	12	4500	94	94	10	Pallet	1.98	1861.2
FG2(A1)	F	4500	4500	9	11	1-Mesin 2-1	1-Mesin 5-1	144000	50	2880	2880	320	10	Keranjang	0.42	1344
FG2(A1)	G	4500	4500	30	29	1-Mesin 1-1	1-Mesin 5-1	144000	50	2880	2880	100	9	Keranjang	0.42	378
FG2(A1)	H	4500	4500	23	27	0-0-0	1-Mesin 5-1	144000	50	2880	2880	126	17	Keranjang	0.42	899.64
FG2(A1)	F	4500	4500	9	11	1-Mesin 2-1	2-Mesin 5-1	144000	50	1620	1620	180	15	Keranjang	0.42	1134
FG2(A1)	G	4500	4500	30	29	1-Mesin 1-1	2-Mesin 5-1	144000	50	1620	1620	56	16	Keranjang	0.42	376.32
FG2(A1)	H	4500	4500	23	27	0-0-0	2-Mesin 5-1	144000	50	1620	1620	71	20	Keranjang	0.42	596.4
FG2(A2)	FG2(A1)	4500	4500	23	17	1-Mesin 5-1	1-Mesin 4-3	0	50	0	0	0	7	Keranjang	0.42	0
FG2(A2)	I	4500	4500	23	17	0-0-0	1-Mesin 4-3	0	50	0	0	0	16	Keranjang	0.42	0
FG2(A2)	FG2(A1)	4500	4500	23	17	1-Mesin 5-1	1-Mesin 4-2	144000	50	2880	2880	170	5	Keranjang	0.42	357
FG2(A2)	I	4500	4500	23	17	0-0-0	1-Mesin 4-2	144000	50	2880	2880	170	14	Keranjang	0.42	999.6
FG2(A2)	FG2(A1)	4500	4500	23	17	2-Mesin 5-1	2-Mesin 4-1	72000	50	1440	1440	85	5	Keranjang	0.42	178.5
FG2(A2)	I	4500	4500	23	17	0-0-0	2-Mesin 4-1	72000	50	1440	1440	85	18	Keranjang	0.42	642.6
FG2(A2)	FG2(A1)	4500	4500	23	17	2-Mesin 5-1	2-Mesin 4-2	144000	50	180	180	11	10	Keranjang	0.42	46.2
FG2(A2)	I	4500	4500	23	17	0-0-0	2-Mesin 4-2	144000	50	180	180	11	19	Keranjang	0.42	87.78
FG2(A2)	FG2(A2)	4500	4500	22	22	1-Mesin 4-2	1-Mesin 9-2	39000	25	1560	1560	71	6	Keranjang	0.42	178.92
FG2(A2)	FG2(A2)	4500	4500	22	22	1-Mesin 4-2	2-Mesin 9-1	54000	25	2160	1320	60	7	Keranjang	0.42	176.4
FG2(A2)	FG2(A2)	4500	4500	22	22	2-Mesin 4-1	2-Mesin 9-1	54000	25	2160	840	39	5	Keranjang	0.42	81.9
FG2(A2)	FG2(A2)	4500	4500	22	22	2-Mesin 4-1	1-Mesin 9-1	144000	25	780	600	28	10	Keranjang	0.42	117.6
FG2(A2)	FG2(A2)	4500	4500	22	22	2-Mesin 4-2	1-Mesin 9-1	144000	25	780	180	9	10	Keranjang	0.42	37.8
FG2(A2)	FG2(A2)	4500	4500	41	42	1-Mesin 9-2	1-Warehouse-1	1000000000	0	4500	1560	39	11	Pallet	1.98	849.42
FG2(A2)	FG2(A2)	4500	4500	41	42	2-Mesin 9-1	1-Warehouse-1	1000000000	0	4500	2160	53	10	Pallet	1.98	1049.4
FG2(A2)	FG2(A2)	4500	4500	41	42	1-Mesin 9-1	1-Warehouse-1	1000000000	0	4500	780	20	13	Pallet	1.98	514.8
Total OMH																40700.7

LAMPIRAN 5

- *List Program*

Lampiran 5

```
unit Proses;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, ComCtrls;

type
  TFormProses = class(TForm)
    Panel2: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    TGenerasi: TEdit;
    TPopulasi: TEdit;
    TPc: TEdit;
    TPm: TEdit;
    TParameter: TEdit;
    PButton: TButton;
    Panel1: TPanel;
    Label6: TLabel;
    Label7: TLabel;
    Label10: TLabel;
    TItem: TEdit;
    TLokasi: TEdit;
    TCase: TEdit;
    CButton: TButton;
    Label8: TLabel;
    LCase: TLabel;
    BProses: TButton;
    BExit: TButton;
    Splitter1: TSplitter;
    Panel3: TPanel;
    Label9: TLabel;
    TCost: TEdit;
    Shape1: TShape;
    Label12: TLabel;
    Shape2: TShape;
    Shape3: TShape;
    Label13: TLabel;
    None: TLabel;
    Mesin1: TEdit;
    Mesin2: TEdit;
    Mesin3: TEdit;
    Mesin4: TEdit;
    Qty1: TEdit;
    Qty2: TEdit;
    Qty3: TEdit;
    Qty4: TEdit;
    No1: TEdit;
    No2: TEdit;
    No3: TEdit;
  end;
end;
```

Lampiran 5

```
No4: TEdit;
SB: TScrollBar;
Shape4: TShape;
Label16: TLabel;
Item1: TEdit;
Item2: TEdit;
Item3: TEdit;
Item4: TEdit;
Label11: TLabel;
TReplikasi: TEdit;
Label14: TLabel;
Label20: TLabel;
PBReplikasi: TProgressBar;
PBGenerasi: TProgressBar;
Label23: TLabel;
TIterasi: TEdit;
Label24: TLabel;
THandling: TEdit;
Label15: TLabel;
Shape5: TShape;
Cost4: TEdit;
Cost3: TEdit;
Cost2: TEdit;
Cost1: TEdit;
Label17: TLabel;
Shape6: TShape;
Handling1: TEdit;
Handling2: TEdit;
Handling3: TEdit;
Handling4: TEdit;
procedure TCaseChange(Sender: TObject);
procedure TParameterChange(Sender: TObject);
procedure CButtonClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure TGenerasiChange(Sender: TObject);
procedure TPopulasiChange(Sender: TObject);
procedure TPcChange(Sender: TObject);
procedure TPmChange(Sender: TObject);
procedure PButtonClick(Sender: TObject);
procedure BExitClick(Sender: TObject);
procedure BProsesClick(Sender: TObject);
procedure SBChange(Sender: TObject);
procedure TCaseKeyPress(Sender: TObject; var Key: Char);
procedure TParameterKeyPress(Sender: TObject; var Key: Char);
private
  { Private declarations }
public
  { Public declarations }
end;
```

Type

```
RParameter = Record
  Generasi : Integer;
  Populasi : Integer;
  Pc : Real;
  Pm : Real;
```


Lampiran 5

```
End;

ROperasi = Record
  Lokasi : String[30];
  LokasiKe : Integer;
  Waktu : Real;
  Handling : String[30];
  HandlingKe : Integer;
  Volume : Real;
  Weight : Real;
End;

RItem = Record
  Nama : String[30];
  Material : String[30];
  Lambang : String[30];
  Produksi : Real;
  Kebutuhan : Real;
  Pendahulu : String[30];
  PendahuluKe : Integer;
  Level : Integer;
  JumOperasi : Integer;
  Operasi : Array[1..100] of ROperasi;
End;

RCase = Record
  JumItem : Integer;
  Item : Array[1..50] of RItem;
End;

RHandling2 = Record
  Nama : String[30];
  Cost : Real;
End;

RHandling = Record
  JumHandling : Integer;
  Handling : Array[1..100] of RHandling2;
End;

RLokasi2 = Record
  Nama : String[30];
  Lambang : String[30];
  Kapasitas : Real;
  Fraktal : Array[1..6] of Integer;
End;

RLokasi = Record
  JumLokasi : Integer;
  Lokasi : Array[1..100] of RLokasi2;
End;

ROptions = Record
  FraktalMaks : Integer;
  JumlahMaks : Integer;
  SProduksi : String[30];
  SKapasitas : String[30];
End;

RJarak = Record
```

Lampiran 5

```
Dari : String[30];
Ke : String[30];
Jarak : Real;
DataKe : Integer;
End;

RJarak2 = Record
  DariMesinKe : Integer;
  DariFraktalKe : Integer;
  DariNoMesinKe : Integer;
  KeMesinKe : Integer;
  KeFraktalKe : Integer;
  KeNoMesinKe : Integer;
  Available : Integer;
  Jarak : Real;
End;

RMesin = Record
  Dari : Integer;
  Ke : Integer;
End;

RPredec = Record
  ItemKe : Integer;
  ItemKeSort : Integer;
End;

ROpTotal2 = Record
  Item : String[30];
  ItemKe : Integer;
  Level : Integer;
  Produk : String[30];
  ProdukKe : Integer;
  TotalCost : Real;
  JumPredec : Integer;
  Predec : Array[1..20] of RPredec;
  JumMesin : Integer;
  Mesin : Array[1..50] of RMesin;
End;

ROpTotal = Record
  Item : Array[1..101] of ROpTotal2;
  JumItem : Integer;
End;

RGen = Record
  Pm : Real;
  ItemKe : Integer;
  Produk : String[30];
  ProdukKe : Integer;
  OperasiKe : Integer;
  MesinKe : Integer;
  FraktalKe : Integer;
  NoMesinKe : Integer;
  Produksi : Real;
  Keluar : Real;
  Cost : Real;
End;

RKromosom = Record
```

Lampiran 5

```
JumGen : Integer;
Pc : Real;
TotalCost : Real;
Gen : Array[1..3501] of RGen;
End;

RGen2 = Record
  Item : String[30];
  Mesin : String[60];
  Handling : String[30];
  Produksi : String[30];
  Cost : String[30];
End;
RTampilan = Record
  JumGen : Integer;
  Gen : Array[1..3501] of RGen2;
End;

RSeleksi = Record
  Urutan : Integer;
  TotalCost : Real;
End;

RLocator4 = Record
  Kapasitas : Real;
  Sisa : Real;
End;
RLocator3 = Record
  JumMesin : Integer;
  MesinKe : Array[1..50] of RLocator4;
End;
RLocator2 = Record
  Nama : String[30];
  Lambang : String[30];
  Fraktal : Array[1..6] of RLocator3;
End;
RLocator = Record
  JumLokasi : Integer;
  Lokasi : Array[1..100] of RLocator2;
End;

var
  FormProses: TFormProses;
  Parameter : RParameter;
  FParameter : File Of RParameter;
  Handling : RHandling;
  FHandling : File of RHandling;
  Lokasi : RLokasi;
  FLokasi : File of RLokasi;
  Kasus : RCase;
  FKasus : File of RCase;
  Options : ROptions;
  FOptions : File of ROptions;
  Jarak : RJarak2;
  FJarak, FJarak2 : File of RJarak2;
  JarakDummy : RJarak;
```

Lampiran 5

```
FJarakDummy,FJarakDummy2 : File of RJarak;
OpTotal : ROpTotal;
FOpTotal : File of ROpTotal;
Kromosom : RKromosom;
FKromosom : File of RKromosom;

Genetic,Genetic2,GeneticBest,GeneticKsg : RKromosom;
FGenetic : File of RKromosom;

Locator, LocatorTemp : RLocator;

Parent1,Parent2, Anak1, Anak2 : RKromosom;

GenSeleksi : Array[1..5001] of RSeleksi;

Tampilan : RTampilan;

i,j,k,l,m : integer;
IterasiJalan,PopulasiJalan : integer;
Replikasi,ReplikasiJalan : Integer;

CostRata2,CostGood : Real;

CBPilih : Integer;

Click1,Click2 : Integer;
DumyString : String;
Test : Text;
WaktuMulai, WaktuSelesai : TDateTime;
WaktuMulai1, WaktuSelesai1 : String;

implementation

{$R *.dfm}

procedure TFormProses.BExitClick(Sender: TObject);
begin
    Close;
end;

procedure TFormProses.BProsesClick(Sender: TObject);
procedure Cleansing_Layar;
Begin
    SB.Position:=1;
    SB.Enabled:=False;
    No1.Text:='';
    No2.Text:='';
    No3.Text:='';
    No4.Text:='';
    Item1.Text:='';
    Item2.Text:='';
    Item3.Text:='';
    Item4.Text:='';
    Mesin1.Text:='';
    Mesin2.Text:='';
    Mesin3.Text:='';
```

Lampiran 5

```
Mesin4.Text:='';
Qty1.Text:='';
Qty2.Text:='';
Qty3.Text:='';
Qty4.Text:='';
Cost1.Text:='';
Cost2.Text:='';
Cost3.Text:='';
Cost4.Text:='';
{Handling1.Text:='';
Handling2.Text:='';
Handling3.Text:='';
Handling4.Text:=''; }
TIterasi.Text:='';
TCost.Text:='';
End;
procedure Create_File;
Begin
    WaktuMulai:=GetTime;
    WaktuMulail:=DateTimetoStr(now);
    str(ReplikasiJalan,DumyString);
    System.Assign(Test,'Solusi\Summary'+DumyString+'.txt');
    System.Rewrite(Test);
    System.Reset(Test);
    System.Append(Test);
    System.Writeln(Test,'Generasi;Populasi;Total OMH;Detail');
    System.Close(Test);

    System.Assign(FGenetic,'Temp\Genetic.gen');
    System.Rewrite(FGenetic);
    System.Close(FGenetic);

    System.Assign(Test,'Solusi\Best'+DumyString+'.txt');
    System.Rewrite(Test);
    System.Append(Test);
    System.Writeln(Test,'Hasil Akhir untuk Kasus :
',LCase.Caption);
        System.Writeln(Test,'Jumlah Item           :
',Kasus.JumItem);
        System.Writeln(Test,'Jumlah Handling       :
',Handling.JumHandling);
        System.Writeln(Test,'Jumlah Lokasi         :
',Lokasi.JumLokasi);
        System.Writeln(Test,'Parameter Generasi    :
',Parameter.Generasi);
        System.Writeln(Test,'Parameter Populasi    :
',Parameter.Populasi);
        System.Writeln(Test,'Parameter Pc           :
',Parameter.Pc:0:4);
        System.Writeln(Test,'Parameter Pm           :
',Parameter.Pm:0:4);
    System.Writeln(Test);
    System.Close(Test);

    System.Assign(Test,'Solusi\MeanEfi'+DumyString+'.txt');
    System.Rewrite(Test);
```

Lampiran 5

```
System.Append(Test);
System.WriteLine(Test, 'Generasi;OMH Terbaik;Rata-Rata OMH');
System.Close(Test);
End;
procedure Total_Cost;
Var JumProd, JumKeb : Real;
    JumProd2, SisaProd2, Approved2 : Real;
    SisaProd, SisaKeb : Real;
    SisaKap, Approved : Real;
    Smaller : Real;
    Frek : Integer;
    Distance : Integer;
Begin
    Locator:=LocatorTemp;
    for i := 1 to Genetic.JumGen do
    Begin
        Genetic.Gen[i].Produksi:=0;
        Genetic.Gen[i].Keluar:=0;
        Genetic.Gen[i].Cost:=0;
    End;
    Genetic.TotalCost:=0;

    for i := 1 to OpTotal.JumItem do
        for j := 1 to OpTotal.Item[i].JumMesin do
            if (j=1) and (OpTotal.Item[i].JumPredec=0) then
                Begin
                    JumProd:=Kasus.Item[OpTotal.Item[i].ItemKe].Produksi;
SisaProd:=JumProd;
                    JumKeb:=Kasus.Item[OpTotal.Item[i].ItemKe].Kebutuhan;
SisaKeb:=JumKeb;
                    for k := OpTotal.Item[i].Mesin[j].Dari to
OpTotal.Item[i].Mesin[j].Ke do
                        Begin

SisaKap:=Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa;
                            if
SisaKap>(SisaProd*Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu) then Approved:=SisaProd
                            Else
                                Begin

Approved:=Round(SisaKap/Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu);
                                    if
Approved>SisaKap/Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu then
                                        Approved:=Approved-1;
                                    End;

Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa:=Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa-
Approved*Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu;
                                    SisaProd:=SisaProd-Approved;
                        End;
                End;
            End;
        End;
    End;
End;
```

Lampiran 5

```
Genetic.Gen[k].Produksi:=Approved;
Genetic.Gen[k].Keluar:=0;
{Hitung Total Cost Jarak}
if
Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Volume<Kasus.Item[Op
Total.Item[i].ItemKe].Operasi[j].Weight then

Smaller:=Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Volume
else
Smaller:=Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Weight;
Frek:=Round(((Approved/JumProd)*JumKeb)/Smaller);
if Frek<(((Approved/JumProd)*JumKeb)/Smaller) then
Frek:=Frek+1;
Distance:=(Genetic.Gen[k].MesinKe-
1)*OpTions.FraktalMaks*OpTions.JumlahMaks;
Distance:=Distance+((Genetic.Gen[k].FraktalKe-
1)*OpTions.JumlahMaks);
Distance:=Distance+Genetic.Gen[k].NoMesinKe;
System.Assign(FJarak,'Case\''+LCase.Caption+'.jrk');
System.Reset(FJarak);
System.Seek(FJarak,Distance-1);
System.Read(FJarak,Jarak);
System.Close(FJarak);

Genetic.Gen[k].Cost:=Jarak.Jarak*Frek*Handling.Handling[Kasus.Item
[OpTotal.Item[i].ItemKe].Operasi[j].HandlingKe].Cost;

Genetic.TotalCost:=Genetic.TotalCost+Genetic.Gen[k].Cost;
if SisaProd<=0 then Break;
End;
End
Else if (j=1) and (OpTotal.Item[i].JumPredec>0) then
Begin
JumProd:=Kasus.Item[OpTotal.Item[i].ItemKe].Produksi;
SisaProd:=JumProd;
JumKeb:=Kasus.Item[OpTotal.Item[i].ItemKe].Kebutuhan;
SisaKeb:=JumKeb;
for k := OpTotal.Item[i].Mesin[j].Dari to
OpTotal.Item[i].Mesin[j].Ke do
Begin

SisaKap:=Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Ge
n[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa;
if
SisaKap>(SisaProd*Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Wa
ktu) then Approved:=SisaProd
Else
Begin

Approved:=Round(SisaKap/Kasus.Item[OpTotal.Item[i].ItemKe].Operasi
[j].Waktu);
if
Approved>SisaKap/Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Wak
tu then
Approved:=Approved-1;
End;
```

Lampiran 5

```
Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa:=Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa-
Approved*Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu;
    SisaProd:=SisaProd-Approved;
    Genetic.Gen[k].Produksi:=Approved;
    Genetic.Gen[k].Keluar:=0;
    {Hitung Total Cost Jarak}
    for l := 1 to OpTotal.Item[i].JumPredec do
Begin

JumProd2:=Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Produksi;
    SisaProd2:=Approved/JumProd*JumProd2;
    if
OpTotal.Item[OpTotal.Item[i].Predec[l].ItemKeSort].JumMesin<=0
then
        Begin

JumProd2:=Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Kebutuhan;
    SisaProd2:=Approved/JumProd*JumProd2;
    Approved2:=SisaProd2;
    if
Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Operasi[1].Volume<Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Operasi[1].Weight then

Smaller:=Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Operasi[1].Volume
        else
Smaller:=Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Operasi[1].Weight;
        Frek:=Round(Approved2/Smaller);
        if Frek<(Approved2/Smaller) then Frek:=Frek+1;
        Distance:=(Genetic.Gen[k].MesinKe-
1)*OpTions.FraktalMaks*OpTions.JumlahMaks;
        Distance:=Distance+((Genetic.Gen[k].FraktalKe-
1)*OpTions.JumlahMaks);
        Distance:=Distance+Genetic.Gen[k].NoMesinKe;

System.Assign(FJarak, 'Case\' +LCase.Caption+'.jrk');
    System.Reset(FJarak);
    System.Seek(FJarak,Distance-1);
    System.Read(FJarak, Jarak);
    System.Close(FJarak);

Genetic.Gen[k].Cost:=Genetic.Gen[k].Cost+(Jarak.Jarak*Frek*Handling.Handling[{Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].HandlingKe}Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Operasi[1].HandlingKe].Cost);
        End
    Else for m :=
OpTotal.Item[OpTotal.Item[i].Predec[l].ItemKeSort].Mesin[OpTotal.Item[OpTotal.Item[i].Predec[l].ItemKeSort].JumMesin].Dari to
OpTotal.Item[OpTotal.Item[i].Predec[l].ItemKeSort].Mesin[OpTotal.Item[OpTotal.Item[i].Predec[l].ItemKeSort].JumMesin].Ke do
```


Lampiran 5

```
        if (Genetic.Gen[m].Produksi-
Genetic.Gen[m].Keluar)>0 then
        Begin
            if (Genetic.Gen[m].Produksi-
Genetic.Gen[m].Keluar)>=SisaProd2 then
                Approved2:=SisaProd2
            else Approved2:=Genetic.Gen[m].Produksi-
Genetic.Gen[m].Keluar;

Genetic.Gen[m].Keluar:=Genetic.Gen[m].Keluar+Approved2;
        SisaProd2:=SisaProd2-Approved2;

        if
Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Operasi[Kasus.Item[Op
Total.Item[i].Predec[l].ItemKe].JumOperasi+1].Volume<Kasus.Item[Op
Total.Item[i].Predec[l].ItemKe].Operasi[Kasus.Item[OpTotal.Item[i]
.Predec[l].ItemKe].JumOperasi+1].Weight then

Smaller:=Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Operasi[Kasu
s.Item[OpTotal.Item[i].Predec[l].ItemKe].JumOperasi+1].Volume
else
Smaller:=Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Operasi[Kasu
s.Item[OpTotal.Item[i].Predec[l].ItemKe].JumOperasi+1].Weight;
        Frek:=Round(Approved2/Smaller);
        if Frek<(Approved2/Smaller) then Frek:=Frek+1;

Distance:=Lokasi.JumLokasi*OpTions.FraktalMaks*Options.JumlahMaks;
        Distance:=Distance+((Genetic.Gen[m].MesinKe-
1)*Options.FraktalMaks*Options.JumlahMaks*Lokasi.JumLokasi*OpTions
.FraktalMaks*Options.JumlahMaks);
        Distance:=Distance+((Genetic.Gen[m].FraktalKe-
1)*Options.JumlahMaks*Lokasi.JumLokasi*OpTions.FraktalMaks*Options
.JumlahMaks);
        Distance:=Distance+((Genetic.Gen[m].NoMesinKe-
1)*Lokasi.JumLokasi*OpTions.FraktalMaks*Options.JumlahMaks);
        Distance:=Distance+((Genetic.Gen[k].MesinKe-
1)*OpTions.FraktalMaks*OpTions.JumlahMaks);
        Distance:=Distance+((Genetic.Gen[k].FraktalKe-
1)*OpTions.JumlahMaks);
        Distance:=Distance+Genetic.Gen[k].NoMesinKe;

System.Assign(FJarak, 'Case\' +LCase.Caption+'.jrk');
        System.Reset(FJarak);
        System.Seek(FJarak,Distance-1);
        System.Read(FJarak, Jarak);
        System.Close(FJarak);

Genetic.Gen[k].Cost:=Genetic.Gen[k].Cost+(Jarak.Jarak*Frek*Handlin
g.Handling[{Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Handling
Ke}Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Operasi[Kasus.Item
[OpTotal.Item[i].Predec[l].ItemKe].JumOperasi+1].HandlingKe].Cost)
;

        if SisaProd2<=0 then Break;
    End;
End;
```

Lampiran 5

```
Genetic.TotalCost:=Genetic.TotalCost+Genetic.Gen[k].Cost;
    if SisaProd<=0 then Break;
    End;
End
Else if (j>1) then
Begin
    JumProd:=Kasus.Item[OpTotal.Item[i].ItemKe].Produksi;
SisaProd:=JumProd;
    for k := OpTotal.Item[i].Mesin[j].Dari to
OpTotal.Item[i].Mesin[j].Ke do
    Begin

SisaKap:=Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa;
        if
SisaKap>(SisaProd*Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu) then Approved:=SisaProd
        Else
        Begin

Approved:=Round(SisaKap/Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu);
            if
Approved>SisaKap/Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu then
                Approved:=Approved-1;
            End;

Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa:=Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa-
Approved*Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu;
            SisaProd:=SisaProd-Approved;
            Genetic.Gen[k].Produksi:=Approved;
            Genetic.Gen[k].Keluar:=0;
            {Hitung Total Cost Jarak}
            SisaProd2:=Approved;
            for m := OpTotal.Item[i].Mesin[j-1].Dari to
OpTotal.Item[i].Mesin[j-1].Ke do
            if (Genetic.Gen[m].Produksi-Genetic.Gen[m].Keluar)>0
then
                Begin
                    if (Genetic.Gen[m].Produksi-
Genetic.Gen[m].Keluar)>=SisaProd2 then
                        Approved2:=SisaProd2
                    else Approved2:=Genetic.Gen[m].Produksi-
Genetic.Gen[m].Keluar;

Genetic.Gen[m].Keluar:=Genetic.Gen[m].Keluar+Approved2;
                    SisaProd2:=SisaProd2-Approved2;

                    if
Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Volume<Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Weight then
```

Lampiran 5

```
Smaller:=Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Volume
else
Smaller:=Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Weight;
Frek:=Round(Approved2/Smaller);
if Frek<(Approved2/Smaller) then Frek:=Frek+1;

Distance:=Lokasi.JumLokasi*OpTions.FraktalMaks*Options.JumlahMaks;
Distance:=Distance+((Genetic.Gen[m].MesinKe-
1)*Options.FraktalMaks*Options.JumlahMaks*Lokasi.JumLokasi*OpTions
.FraktalMaks*Options.JumlahMaks);
Distance:=Distance+((Genetic.Gen[m].FraktalKe-
1)*Options.JumlahMaks*Lokasi.JumLokasi*OpTions.FraktalMaks*Options
.JumlahMaks);
Distance:=Distance+((Genetic.Gen[m].NoMesinKe-
1)*Lokasi.JumLokasi*OpTions.FraktalMaks*Options.JumlahMaks);
Distance:=Distance+((Genetic.Gen[k].MesinKe-
1)*OpTions.FraktalMaks*OpTions.JumlahMaks);
Distance:=Distance+((Genetic.Gen[k].FraktalKe-
1)*OpTions.JumlahMaks);
Distance:=Distance+Genetic.Gen[k].NoMesinKe;
System.Assign(FJarak,'Case\''+LCase.Caption+'.jrk');
System.Reset(FJarak);
System.Seek(FJarak,Distance-1);
System.Read(FJarak,Jarak);
System.Close(FJarak);

Genetic.Gen[k].Cost:=Genetic.Gen[k].Cost+(Jarak.Jarak*Frek*Handlin
g.Handling[Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].HandlingK
e].Cost);
if SisaProd2<=0 then Break;
End;

Genetic.TotalCost:=Genetic.TotalCost+Genetic.Gen[k].Cost;
if SisaProd<=0 then Break;
End;
End;

procedure Total_Cost_Tampil;
Var JumProd,JumKeb : Real;
JumProd2,SisaProd2,Approved2 : Real;
SisaProd,SisaKeb : Real;
SisaKap, Approved : Real;
Smaller : Real;
Frek : Integer;
Distance : Integer;
Begin
Locator:=LocatorTemp;
for i := 1 to Genetic.JumGen do
Begin
Genetic.Gen[i].Produksi:=0;
Genetic.Gen[i].Keluar:=0;
Genetic.Gen[i].Cost:=0;
End;
Genetic.TotalCost:=0;
```

Lampiran 5

```
for i := 1 to OpTotal.JumItem do
  for j := 1 to OpTotal.Item[i].JumMesin do
    if (j=1) and (OpTotal.Item[i].JumPredec=0) then
      Begin
        JumProd:=Kasus.Item[OpTotal.Item[i].ItemKe].Produksi;
SisaProd:=JumProd;
        JumKeb:=Kasus.Item[OpTotal.Item[i].ItemKe].Kebutuhan;
SisaKeb:=JumKeb;
        for k := OpTotal.Item[i].Mesin[j].Dari to
OpTotal.Item[i].Mesin[j].Ke do
          Begin

SisaKap:=Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa;
            if
SisaKap>(SisaProd*Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu) then Approved:=SisaProd
              Else
                Begin

Approved:=Round(SisaKap/Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu);
                  if
Approved>SisaKap/Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu then
                    Approved:=Approved-1;
                  End;

Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa:=Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa-
Approved*Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu;
                SisaProd:=SisaProd-Approved;
                Genetic.Gen[k].Produksi:=Approved;
                Genetic.Gen[k].Keluar:=0;
                {Hitung Total Cost Jarak}
                if
Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Volume<Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Weight then
                  Smaller:=Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Volume
                else
                  Smaller:=Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Weight;
                  Frek:=Round(((Approved/JumProd)*JumKeb)/Smaller);
                  if Frek<(((Approved/JumProd)*JumKeb)/Smaller) then
                    Frek:=Frek+1;
                    Distance:=(Genetic.Gen[k].MesinKe-
1)*OpTions.FraktalMaks*OpTions.JumlahMaks;
                    Distance:=Distance+((Genetic.Gen[k].FraktalKe-
1)*OpTions.JumlahMaks);
                    Distance:=Distance+Genetic.Gen[k].NoMesinKe;
                    System.Assign(FJarak, 'Case\' +LCase.Caption+'.jrk');
                    System.Reset(FJarak);
                    System.Seek(FJarak,Distance-1);
                    System.Read(FJarak, Jarak);
```

Lampiran 5

```
System.Close (FJarak);

Genetic.Gen[k].Cost:=Jarak.Jarak*Frek*Handling.Handling[Kasus.Item
[OpTotal.Item[i].ItemKe].Operasi[j].HandlingKe].Cost;

Genetic.TotalCost:=Genetic.TotalCost+Genetic.Gen[k].Cost;

Str (ReplikasiJalan,DumyString);

System.Assign (Test, 'Solusi\DetailBest'+DumyString+'.txt');
System.Reset (Test);
System.Append (Test);

System.Write (Test,Kasus.Item[OpTotal.Item[i].ItemKe].Nama, ';');
System.Write (Test, 'Bahan Baku;');

System.Write (Test,Kasus.Item[OpTotal.Item[i].ItemKe].Produksi:0:2,
';');

System.Write (Test,Kasus.Item[OpTotal.Item[i].ItemKe].Kebutuhan:0:2
, ';');

System.Write (Test,Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Vo
lume:0:2, ';');

System.Write (Test,Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].We
ight:0:2, ';');
System.Write (Test, '0-0-0;');
System.Write (Test, Genetic.Gen[k].FraktalKe, '-
', Lokasi.Lokasi[Genetic.Gen[k].MesinKe].Nama, '-
', Genetic.Gen[k].NoMesinKe, ';');
System.Write (Test, SisaKap:0:2, ';');

System.Write (Test,Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Wa
ktu:0:2, ';');
System.Write (Test, Approved:0:2, ';');
System.Write (Test, Approved:0:2, ';');
System.Write (Test, Frek, ';');
System.Write (Test, Jarak.Jarak:0:2, ';');

System.Write (Test, Handling.Handling[Kasus.Item[OpTotal.Item[i].Ite
mKe].Operasi[j].HandlingKe].Nama, ';');

System.Write (Test, Handling.Handling[Kasus.Item[OpTotal.Item[i].Ite
mKe].Operasi[j].HandlingKe].Cost:0:2, ';');
System.WriteLine (Test, Genetic.Gen[k].Cost:0:2);
System.Close (Test);
if SisaProd<=0 then Break;
End;
End
Else if (j=1) and (OpTotal.Item[i].JumPredec>0) then
Begin
JumProd:=Kasus.Item[OpTotal.Item[i].ItemKe].Produksi;
SisaProd:=JumProd;
JumKeb:=Kasus.Item[OpTotal.Item[i].ItemKe].Kebutuhan;
SisaKeb:=JumKeb;
```

Lampiran 5

```
        for k := OpTotal.Item[i].Mesin[j].Dari to
OpTotal.Item[i].Mesin[j].Ke do
        Begin

SisaKap:=Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa;
        if
SisaKap>(SisaProd*Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu) then Approved:=SisaProd
        Else
        Begin

Approved:=Round(SisaKap/Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu);
        if
Approved>SisaKap/Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu then
                Approved:=Approved-1;
        End;

Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa:=Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa-
Approved*Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu;
        SisaProd:=SisaProd-Approved;
        Genetic.Gen[k].Produksi:=Approved;
        Genetic.Gen[k].Keluar:=0;
        {Hitung Total Cost Jarak}
        for l := 1 to OpTotal.Item[i].JumPredec do
        Begin

JumProd2:=Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Produksi;
        SisaProd2:=Approved/JumProd*JumProd2;
        if
OpTotal.Item[OpTotal.Item[i].Predec[l].ItemKeSort].JumMesin<=0
then
        Begin

JumProd2:=Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Kebutuhan;
        SisaProd2:=Approved/JumProd*JumProd2;
        Approved2:=SisaProd2;
        if
Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Operasi[1].Volume<Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Operasi[1].Weight then

Smaller:=Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Operasi[1].Volume
        else
Smaller:=Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Operasi[1].Weight;
                Frek:=Round(Approved2/Smaller);
                if Frek<(Approved2/Smaller) then Frek:=Frek+1;
                Distance:=(Genetic.Gen[k].MesinKe-
1)*OpTions.FraktalMaks*OpTions.JumlahMaks;
```

Lampiran 5

```
Distance:=Distance+((Genetic.Gen[k].FraktalKe-
1)*OpTions.JumlahMaks);
Distance:=Distance+Genetic.Gen[k].NoMesinKe;

System.Assign(FJarak, 'Case\' +LCCase.Caption+'.jrk');
System.Reset(FJarak);
System.Seek(FJarak, Distance-1);
System.Read(FJarak, Jarak);
System.Close(FJarak);

Genetic.Gen[k].Cost:=Genetic.Gen[k].Cost+(Jarak.Jarak*Frek*Handlin
g.Handling[{Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Handling
Ke}Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Operasi[1].Handlin
gKe].Cost);

Str(ReplikasiJalan, DummyString);

System.Assign(Test, 'Solusi\DetailBest'+DummyString+'.txt');
System.Reset(Test);
System.Append(Test);

System.Write(Test, Kasus.Item[OpTotal.Item[i].ItemKe].Nama, ';');

System.Write(Test, Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Nam
a, ';');

System.Write(Test, Kasus.Item[OpTotal.Item[i].ItemKe].Produksi:0:2,
');

System.Write(Test, Kasus.Item[OpTotal.Item[i].ItemKe].Kebutuhan:0:2
,');

System.Write(Test, Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Ope
rasi[1].Volume:0:2,');

System.Write(Test, Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Ope
rasi[1].Weight:0:2,');
System.Write(Test, '0-0-0;');
System.Write(Test, Genetic.Gen[k].FraktalKe, '-
', Lokasi.Lokasi[Genetic.Gen[k].MesinKe].Nama, '-
', Genetic.Gen[k].NoMesinKe,');
System.Write(Test, SisaKap:0:2,');

System.Write(Test, Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Wa
ktu:0:2,');
System.Write(Test, Approved:0:2,');
System.Write(Test, Approved2:0:2,');
System.Write(Test, Frek,');
System.Write(Test, Jarak.Jarak:0:2,');

System.Write(Test, Handling.Handling[{Kasus.Item[OpTotal.Item[i].It
emKe].Operasi[j].HandlingKe}Kasus.Item[OpTotal.Item[i].Predec[l].I
temKe].Operasi[1].HandlingKe].Nama,');

System.Write(Test, Handling.Handling[{Kasus.Item[OpTotal.Item[i].It
```

Lampiran 5

```
emKe].Operasi[j].HandlingKe}Kasus.Item[OpTotal.Item[i].Predec[1].ItemKe].Operasi[1].HandlingKe].Cost:0:2,'););

System.WriteLine(Test, (Jarak.Jarak*Frek*Handling.Handling[{Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].HandlingKe}Kasus.Item[OpTotal.Item[i].Predec[1].ItemKe].Operasi[1].HandlingKe].Cost):0:2);
    System.Close(Test);
End
Else for m :=
OpTotal.Item[OpTotal.Item[i].Predec[1].ItemKeSort].Mesin[OpTotal.Item[OpTotal.Item[i].Predec[1].ItemKeSort].JumMesin].Dari to
OpTotal.Item[OpTotal.Item[i].Predec[1].ItemKeSort].Mesin[OpTotal.Item[OpTotal.Item[i].Predec[1].ItemKeSort].JumMesin].Ke do
    if (Genetic.Gen[m].Produksi-
Genetic.Gen[m].Keluar)>0 then
    Begin
        if (Genetic.Gen[m].Produksi-
Genetic.Gen[m].Keluar)>=SisaProd2 then
            Approved2:=SisaProd2
        else Approved2:=Genetic.Gen[m].Produksi-
Genetic.Gen[m].Keluar;

Genetic.Gen[m].Keluar:=Genetic.Gen[m].Keluar+Approved2;
        SisaProd2:=SisaProd2-Approved2;

        if
Kasus.Item[OpTotal.Item[i].Predec[1].ItemKe].Operasi[Kasus.Item[OpTotal.Item[i].Predec[1].ItemKe].JumOperasi+1].Volume<Kasus.Item[OpTotal.Item[i].Predec[1].ItemKe].Operasi[Kasus.Item[OpTotal.Item[i].Predec[1].ItemKe].JumOperasi+1].Weight then

Smaller:=Kasus.Item[OpTotal.Item[i].Predec[1].ItemKe].Operasi[Kasus.Item[OpTotal.Item[i].Predec[1].ItemKe].JumOperasi+1].Volume
        else
Smaller:=Kasus.Item[OpTotal.Item[i].Predec[1].ItemKe].Operasi[Kasus.Item[OpTotal.Item[i].Predec[1].ItemKe].JumOperasi+1].Weight;
        Frek:=Round(Approved2/Smaller);
        if Frek<(Approved2/Smaller) then Frek:=Frek+1;

Distance:=Lokasi.JumLokasi*OpTions.FraktalMaks*Options.JumlahMaks;
        Distance:=Distance+((Genetic.Gen[m].MesinKe-
1)*Options.FraktalMaks*Options.JumlahMaks*Lokasi.JumLokasi*OpTions.FraktalMaks*Options.JumlahMaks);
        Distance:=Distance+((Genetic.Gen[m].FraktalKe-
1)*Options.JumlahMaks*Lokasi.JumLokasi*OpTions.FraktalMaks*Options.JumlahMaks);
        Distance:=Distance+((Genetic.Gen[m].NoMesinKe-
1)*Lokasi.JumLokasi*OpTions.FraktalMaks*Options.JumlahMaks);
        Distance:=Distance+((Genetic.Gen[k].MesinKe-
1)*OpTions.FraktalMaks*OpTions.JumlahMaks);
        Distance:=Distance+((Genetic.Gen[k].FraktalKe-
1)*OpTions.JumlahMaks);
        Distance:=Distance+Genetic.Gen[k].NoMesinKe;

System.Assign(FJarak, 'Case\' +LCase.Caption+'.jrk');
System.Reset(FJarak);
```


Lampiran 5

```
System.Seek (FJarak, Distance-1);
System.Read (FJarak, Jarak);
System.Close (FJarak);

Genetic.Gen[k].Cost:=Genetic.Gen[k].Cost+(Jarak.Jarak*Frek*Handling.
Handling[{Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].HandlingKe}
Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Operasi[Kasus.Item
[OpTotal.Item[i].Predec[l].ItemKe].JumOperasi+1].HandlingKe].Cost)
;

Str (ReplikasiJalan, DummyString);

System.Assign (Test, 'Solusi\DetailBest'+DummyString+'.txt');
System.Reset (Test);
System.Append (Test);

System.Write (Test, Kasus.Item[OpTotal.Item[i].ItemKe].Nama, ';' );

System.Write (Test, Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Nam
a, ';' );

System.Write (Test, Kasus.Item[OpTotal.Item[i].ItemKe].Produksi:0:2,
'; ');

System.Write (Test, Kasus.Item[OpTotal.Item[i].ItemKe].Kebutuhan:0:2
, ';' );

System.Write (Test, Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Ope
rasi[Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].JumOperasi+1].Vo
lume:0:2, ';' );

System.Write (Test, Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].Ope
rasi[Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].JumOperasi+1].We
ight:0:2, ';' );
System.Write (Test, Genetic.Gen[m].FraktalKe, '-
', Lokasi.Lokasi [Genetic.Gen[m].MesinKe].Nama, '-
', Genetic.Gen[m].NoMesinKe, ';' );
System.Write (Test, Genetic.Gen[k].FraktalKe, '-
', Lokasi.Lokasi [Genetic.Gen[k].MesinKe].Nama, '-
', Genetic.Gen[k].NoMesinKe, ';' );
System.Write (Test, SisaKap:0:2, ';' );

System.Write (Test, Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Wa
ktu:0:2, ';' );
System.Write (Test, Approved:0:2, ';' );
System.Write (Test, Approved2:0:2, ';' );
System.Write (Test, Frek, ';' );
System.Write (Test, Jarak.Jarak:0:2, ';' );

System.Write (Test, Handling.Handling[{Kasus.Item[OpTotal.Item[i].It
emKe].Operasi[j].HandlingKe}Kasus.Item[OpTotal.Item[i].Predec[l].I
temKe].Operasi[Kasus.Item[OpTotal.Item[i].Predec[l].ItemKe].JumOpe
rasi+1].HandlingKe].Nama, ';' );

System.Write (Test, Handling.Handling[{Kasus.Item[OpTotal.Item[i].It
emKe].Operasi[j].HandlingKe}Kasus.Item[OpTotal.Item[i].Predec[l].I
```

Lampiran 5

```
temKe].Operasi[Kasus.Item[OpTotal.Item[i].Predec[1].ItemKe].JumOperasi+1].HandlingKe).Cost:0:2,');

System.WriteLine(Test, (Jarak.Jarak*Frek*Handling.Handling[{Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].HandlingKe}Kasus.Item[OpTotal.Item[i].Predec[1].ItemKe].Operasi[Kasus.Item[OpTotal.Item[i].Predec[1].ItemKe].JumOperasi+1].HandlingKe].Cost):0:2);
    System.Close(Test);
    if SisaProd2<=0 then Break;
End;
End;

Genetic.TotalCost:=Genetic.TotalCost+Genetic.Gen[k].Cost;
    if SisaProd<=0 then Break;
End;
End
Else if (j>1) then
Begin
    JumProd:=Kasus.Item[OpTotal.Item[i].ItemKe].Produksi;
SisaProd:=JumProd;
    for k := OpTotal.Item[i].Mesin[j].Dari to
OpTotal.Item[i].Mesin[j].Ke do
    Begin

SisaKap:=Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa;
        if
SisaKap>(SisaProd*Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu) then Approved:=SisaProd
        Else
        Begin

Approved:=Round(SisaKap/Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu);
            if
Approved>SisaKap/Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu then
                Approved:=Approved-1;
End;

Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa:=Locator.Lokasi[Genetic.Gen[k].MesinKe].Fraktal[Genetic.Gen[k].FraktalKe].MesinKe[Genetic.Gen[k].NoMesinKe].Sisa-
Approved*Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu;
        SisaProd:=SisaProd-Approved;
        Genetic.Gen[k].Produksi:=Approved;
        Genetic.Gen[k].Keluar:=0;
        {Hitung Total Cost Jarak}
        SisaProd2:=Approved;
        for m := OpTotal.Item[i].Mesin[j-1].Dari to
OpTotal.Item[i].Mesin[j-1].Ke do
            if (Genetic.Gen[m].Produksi-Genetic.Gen[m].Keluar)>0
then
                Begin
```

Lampiran 5

```
        if (Genetic.Gen[m].Produksi-
Genetic.Gen[m].Keluar)>=SisaProd2 then
            Approved2:=SisaProd2
        else Approved2:=Genetic.Gen[m].Produksi-
Genetic.Gen[m].Keluar;

Genetic.Gen[m].Keluar:=Genetic.Gen[m].Keluar+Approved2;
SisaProd2:=SisaProd2-Approved2;

        if
Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Volume<Kasus.Item[Op
Total.Item[i].ItemKe].Operasi[j].Weight then

Smaller:=Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Volume
        else
Smaller:=Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Weight;
        Frek:=Round(Approved2/Smaller);
        if Frek<(Approved2/Smaller) then Frek:=Frek+1;

Distance:=Lokasi.JumLokasi*OpTions.FraktalMaks*Options.JumlahMaks;
        Distance:=Distance+((Genetic.Gen[m].MesinKe-
1)*Options.FraktalMaks*Options.JumlahMaks*Lokasi.JumLokasi*OpTions
.FraktalMaks*Options.JumlahMaks);
        Distance:=Distance+((Genetic.Gen[m].FraktalKe-
1)*Options.JumlahMaks*Lokasi.JumLokasi*OpTions.FraktalMaks*Options
.JumlahMaks);
        Distance:=Distance+((Genetic.Gen[m].NoMesinKe-
1)*Lokasi.JumLokasi*OpTions.FraktalMaks*Options.JumlahMaks);
        Distance:=Distance+((Genetic.Gen[k].MesinKe-
1)*OpTions.FraktalMaks*OpTions.JumlahMaks);
        Distance:=Distance+((Genetic.Gen[k].FraktalKe-
1)*OpTions.JumlahMaks);
        Distance:=Distance+Genetic.Gen[k].NoMesinKe;
        System.Assign(FJarak, 'Case\' +LCCase.Caption+'.jrk');
        System.Reset(FJarak);
        System.Seek(FJarak, Distance-1);
        System.Read(FJarak, Jarak);
        System.Close(FJarak);

Genetic.Gen[k].Cost:=Genetic.Gen[k].Cost+(Jarak.Jarak*Frek*Handlin
g.Handling[Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].HandlingK
e].Cost);

        Str(ReplikasiJalan, DummyString);

System.Assign(Test, 'Solusi\DetailBest'+DummyString+'.txt');
        System.Reset(Test);
        System.Append(Test);

System.Write(Test, Kasus.Item[OpTotal.Item[i].ItemKe].Nama, ';' );

System.Write(Test, Kasus.Item[OpTotal.Item[i].ItemKe].Nama, ';' );

System.Write(Test, Kasus.Item[OpTotal.Item[i].ItemKe].Produksi:0:2,
';');
```

Lampiran 5

```
System.Write (Test, Kasus.Item[OpTotal.Item[i].ItemKe].Kebutuhan:0:2,
','');

System.Write (Test, Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Volume:0:2,
','');

System.Write (Test, Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Weight:0:2,
','');
        System.Write (Test, Genetic.Gen[m].FraktalKe, '-
', Lokasi.Lokasi[Genetic.Gen[m].MesinKe].Nama, '-
', Genetic.Gen[m].NoMesinKe, ','');
        System.Write (Test, Genetic.Gen[k].FraktalKe, '-
', Lokasi.Lokasi[Genetic.Gen[k].MesinKe].Nama, '-
', Genetic.Gen[k].NoMesinKe, ','');
        System.Write (Test, SisaKap:0:2, ','');

System.Write (Test, Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].Waktu:0:2,
','');
        System.Write (Test, Approved:0:2, ','');
        System.Write (Test, Approved2:0:2, ','');
        System.Write (Test, Frek, ','');
        System.Write (Test, Jarak.Jarak:0:2, ','');

System.Write (Test, Handling.Handling[Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].HandlingKe].Nama,
','');

System.Write (Test, Handling.Handling[Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].HandlingKe].Cost:0:2,
','');

System.WriteLine (Test, (Jarak.Jarak*Frek*Handling.Handling[Kasus.Item[OpTotal.Item[i].ItemKe].Operasi[j].HandlingKe].Cost):0:2);
        System.Close (Test);
        if SisaProd2<=0 then Break;
    End;

Genetic.TotalCost:=Genetic.TotalCost+Genetic.Gen[k].Cost;
        if SisaProd<=0 then Break;
    End;
End;

procedure Solusi_Awal;
Var Acak : Array[1..201] of Integer;
    JumAcak, JumAcakDummy : Integer;
    BilAcak : Integer;
Begin
    Genetic:=GeneticKsg;
    Genetic.JumGen:=Kromosom.JumGen;

    for i := 1 to OpTotal.JumItem do
    Begin
        if OpTotal.Item[i].JumMesin=0 then
        Begin
            for j := 1 to 1 do
```

Lampiran 5

```
Genetic.Gen[OpTotal.Item[i].Mesin[j].Dari]:=Kromosom.Gen[OpTotal.Item[i].Mesin[j].Dari];
End
Else if (OpTotal.Item[i].Item[1]='*') then
Begin
for j := 1 to OpTotal.Item[i].JumMesin do
for k := OpTotal.Item[i].Mesin[j].Dari to
OpTotal.Item[i].Mesin[j].Ke do
Begin
Genetic.Gen[k]:=Kromosom.Gen[k];
End;
End
Else for j := 1 to OpTotal.Item[i].JumMesin do
Begin
JumAcak:=0;
for k := OpTotal.Item[i].Mesin[j].Dari to
OpTotal.Item[i].Mesin[j].Ke do
Begin
JumAcak:=JumAcak+1;
Acak[JumAcak]:=k;
End;
JumAcakDummy:=JumAcak;
Repeat
BilAcak:=Random(JumAcakDummy)+1;
Acak[201]:=Acak[BilAcak];
Acak[BilAcak]:=Acak[JumAcakDummy];
Acak[JumAcakDummy]:=Acak[201];
JumAcakDummy:=JumAcakDummy-1;
Until JumAcakDummy<=0;;
JumAcakDummy:=0;
for k := OpTotal.Item[i].Mesin[j].Dari to
OpTotal.Item[i].Mesin[j].Ke do
Begin
JumAcakDummy:=JumAcakDummy+1;
Genetic.Gen[k]:=Kromosom.Gen[Acak[JumAcakDummy]];
End;
End;
End;

Total_Cost;

str(ReplikasiJalan,DumyString);
System.Assign(Test,'Solusi\Summary'+DumyString+'.txt');
System.Reset(Test);
System.Append(Test);

System.Write(Test,IterasiJalan,';',PopulasiJalan,';',Genetic.Total
Cost:0:2,';');
for i := 1 to Genetic.JumGen do
Begin
if (Genetic.Gen[i].MesinKe<>0) and
(Kasus.Item[Genetic.Gen[i].ItemKe].Nama[1]<> '*') then
if i<Genetic.JumGen then
System.Write(Test,Kasus.Item[Genetic.Gen[i].ItemKe].Lambang,Geneti
c.Gen[i].OperasiKe,'-',Genetic.Gen[i].FraktalKe,'-
```

Lampiran 5

```
' , Lokasi.Lokasi[Genetic.Gen[i].MesinKe].Lambang, '-
', Genetic.Gen[i].NoMesinKe, ';' )
    Else
System.WriteLine (Test, Kasus.Item[Genetic.Gen[i].ItemKe].Lambang, Gene
tic.Gen[i].OperasiKe, '-', Genetic.Gen[i].FraktalKe, '-
', Lokasi.Lokasi[Genetic.Gen[i].MesinKe].Lambang, '-
', Genetic.Gen[i].NoMesinKe);
    End;
    System.Close (Test);
End;
procedure Crossover;
Var CrossOver : Array[1..5001] of Integer;
    JumlahParent, JumlahParentDumy, Terpilih : Integer;
    x, Titik1, Titik2, Dumy : Integer;
Begin
    System.Assign (FGenetic, 'Temp\Genetic.gen');
    System.Reset (FGenetic);
    Str (ReplikasiJalan, DumyString);
    System.Assign (Test, 'Solusi\Summary'+DumyString+'.txt');
    System.Reset (Test);
    System.Append (Test);
    System.WriteLine (Test);
    System.Close (Test);

    JumlahParent:=0;
    for i := 1 to System.FileSize (FGenetic) do
        if (Random (10001)/10000)<=Parameter.Pc then
            Begin
                JumlahParent:=JumlahParent+1;
                CrossOver [JumlahParent] :=i;
            End;

    JumlahParentDumy:=JumlahParent;
    while JumlahParentDumy>0 do
        Begin
            Terpilih:=Random (JumlahParentDumy)+1;
            CrossOver [5001] :=CrossOver [JumlahParentDumy];
            CrossOver [JumlahParentDumy] :=CrossOver [Terpilih];
            CrossOver [Terpilih] :=CrossOver [5001];
            JumlahParentDumy:=JumlahParentDumy-1;
        End;

    for x := 1 to (JumlahParent Div 2) do
        Begin
            System.Seek (FGenetic, CrossOver [(2*x)-1]-1);
            System.Read (FGenetic, Parent1);
            System.Seek (FGenetic, CrossOver [(2*x)]-1);
            System.Read (FGenetic, Parent2);

            Dumy:=0;
            Repeat
                Titik1:=Random (OpTotal.JumItem-2)+2;
                if (OpTotal.Item [Titik1].JumMesin>0) and
                (OpTotal.Item [Titik1].Item [1]<>'*') then
                    Dumy:=1;
            Until Dumy=1;
```

```

    Dummy:=0;
    Repeat
        Titik2:=Random(OpTotal.JumItem-2)+2;
        if (OpTotal.Item[Titik2].JumMesin>0) and
(OpTotal.Item[Titik2].Item[1]<>'*') then
            Dummy:=1;
    Until Dummy=1;

    if Titik1>Titik2 then
    Begin
        Dummy:=Titik1; Titik1:=Titik2; Titik2:=Dummy;
    End;

    Anak1.JumGen:=Parent1.JumGen;
    Anak2.JumGen:=Parent2.JumGen;
    for i := 1 to OpTotal.Item[Titik1].Mesin[1].Dari-1 do
    Begin
        Anak1.Gen[i]:=Parent1.Gen[i];
        Anak2.Gen[i]:=Parent2.Gen[i];
    End;
    for i :=
OpTotal.Item[Titik2].Mesin[OpTotal.Item[Titik2].JumMesin].Ke+1 to
Kromosom.JumGen do
    Begin
        Anak1.Gen[i]:=Parent1.Gen[i];
        Anak2.Gen[i]:=Parent2.Gen[i];
    End;
    for i := OpTotal.Item[Titik1].Mesin[1].Dari to
OpTotal.Item[Titik2].Mesin[OpTotal.Item[Titik2].JumMesin].Ke do
    Begin
        Anak1.Gen[i]:=Parent2.Gen[i];
        Anak2.Gen[i]:=Parent1.Gen[i];
    End;

    Genetic:=Anak1;
    Total_Cost;
    System.Seek(FGenetic, System.FileSize(FGenetic));
    System.Write(FGenetic, Genetic);
    str(ReplikasiJalan, DummyString);
    System.Assign(Test, 'Solusi\Summary'+DummyString+'.txt');
    System.Reset(Test);
    System.Append(Test);
    System.Write(Test, IterasiJalan, ';C', Crossover[(2*x)-
1], '&', Crossover[2*x], ';', Genetic.TotalCost:0:2, ';');
    for i := 1 to Genetic.JumGen do
    Begin
        if (Genetic.Gen[i].MesinKe<>0) and
(Kasus.Item[Genetic.Gen[i].ItemKe].Nama[1]<>'*') then
            if i<Genetic.JumGen then
System.Write(Test, Kasus.Item[Genetic.Gen[i].ItemKe].Lambang, Geneti
c.Gen[i].OperasiKe, '-', Genetic.Gen[i].FraktalKe, '-
', Lokasi.Lokasi[Genetic.Gen[i].MesinKe].Lambang, '-
', Genetic.Gen[i].NoMesinKe, ';')
            Else
System.Writeln(Test, Kasus.Item[Genetic.Gen[i].ItemKe].Lambang, Gene

```

Lampiran 5

```
tic.Gen[i].OperasiKe, '-', Genetic.Gen[i].FraktalKe, '-
', Lokasi.Lokasi[Genetic.Gen[i].MesinKe].Lambang, '-
', Genetic.Gen[i].NoMesinKe);
    End;
    System.Close (Test);

    Genetic:=Anak2;
    Total_Cost;
    System.Seek (FGenetic, System.FileSize (FGenetic));
    System.Write (FGenetic, Genetic);
    str (ReplikasiJalan, DumyString);
    System.Assign (Test, 'Solusi\Summary'+DumyString+'.txt');
    System.Reset (Test);
    System.Append (Test);
    System.Write (Test, IterasiJalan, ';C', Crossover [(2*x) -
1], '&', Crossover [2*x], ';', Genetic.TotalCost:0:2, ';');
    for i := 1 to Genetic.JumGen do
    Begin
        if (Genetic.Gen[i].MesinKe<>0) and
(Kasus.Item[Genetic.Gen[i].ItemKe].Nama[1]<>'*') then
            if i<Genetic.JumGen then
System.Write (Test, Kasus.Item[Genetic.Gen[i].ItemKe].Lambang, Gene
tic.Gen[i].OperasiKe, '-', Genetic.Gen[i].FraktalKe, '-
', Lokasi.Lokasi[Genetic.Gen[i].MesinKe].Lambang, '-
', Genetic.Gen[i].NoMesinKe, ';')
            Else
System.Writeln (Test, Kasus.Item[Genetic.Gen[i].ItemKe].Lambang, Gene
tic.Gen[i].OperasiKe, '-', Genetic.Gen[i].FraktalKe, '-
', Lokasi.Lokasi[Genetic.Gen[i].MesinKe].Lambang, '-
', Genetic.Gen[i].NoMesinKe);
                End;
                System.Close (Test);
            End;

            System.Close (FGenetic);
End;
procedure Mutasi;
Var x, dumy : Integer;
    GenDumy : Rgen;
Begin
    System.Assign (FGenetic, 'Temp\Genetic.gen');
    System.Reset (FGenetic);
    m:=System.FileSize (FGenetic);
    for x := 1 to m do
    Begin
        System.Seek (FGenetic, x-1);
        System.Read (FGenetic, Genetic);
        for i := 1 to Genetic.JumGen do
Genetic.Gen[i].Pm:=Random (10001)/10000;

        Dumy:=0;
        for i := 1 to OpTotal.JumItem do
            if (OpTotal.Item[i].JumMesin>0) and
(OpTotal.Item[i].Item[1]<>'*') then
                for j := 1 to OpTotal.Item[i].JumMesin do
```



```

        for k := OpTotal.Item[i].Mesin[j].Dari to
OpTotal.Item[i].Mesin[j].Ke do
        if Genetic.Gen[k].Pm<=Parameter.Pm then
        Begin
        Dumpy:=1;
        if k<>OpTotal.Item[i].Mesin[j].Ke then
        Begin
        GenDumy:=Genetic.Gen[k];
        Genetic.Gen[k]:=Genetic.Gen[k+1];
        Genetic.Gen[k+1]:=GenDumy;
        Genetic.Gen[k+1].Pm:=Genetic.Gen[k].Pm;
        Genetic.Gen[k].Pm:=GenDumy.Pm;
        End
        Else
        Begin
        GenDumy:=Genetic.Gen[k];
Genetic.Gen[k]:=Genetic.Gen[OpTotal.Item[i].Mesin[j].Dari];
Genetic.Gen[OpTotal.Item[i].Mesin[j].Dari]:=GenDumy;
Genetic.Gen[OpTotal.Item[i].Mesin[j].Dari].Pm:=Genetic.Gen[k].Pm;
        Genetic.Gen[k].Pm:=GenDumy.Pm;
        End;
        End;
        if Dumpy=1 then
        Begin
        Total_Cost;
        System.Seek(FGenetic, System.FileSize(FGenetic));
        System.Write(FGenetic, Genetic);
        str(ReplikasiJalan, DumyString);
        System.Assign(Test, 'Solusi\Summary'+DumyString+'.txt');
        System.Reset(Test);
        System.Append(Test);
System.Write(Test, IterasiJalan, 'M', x, ' ', Genetic.TotalCost:0:2, '
');
        for i := 1 to Genetic.JumGen do
        Begin
        if (Genetic.Gen[i].MesinKe<>0) and
(Kasus.Item[Genetic.Gen[i].ItemKe].Nama[1]<>'*') then
        if i<Genetic.JumGen then
System.Write(Test, Kasus.Item[Genetic.Gen[i].ItemKe].Lambang, Geneti
c.Gen[i].OperasiKe, '-', Genetic.Gen[i].FraktalKe, '-
', Lokasi.Lokasi[Genetic.Gen[i].MesinKe].Lambang, '-
', Genetic.Gen[i].NoMesinKe, ';')
        Else
System.Writeln(Test, Kasus.Item[Genetic.Gen[i].ItemKe].Lambang, Gene
tic.Gen[i].OperasiKe, '-', Genetic.Gen[i].FraktalKe, '-
', Lokasi.Lokasi[Genetic.Gen[i].MesinKe].Lambang, '-
', Genetic.Gen[i].NoMesinKe);
        End;
        System.Close(Test);
        End;
        End;
        System.Close(FGenetic);

```

Lampiran 5

```
End;
procedure Seleksi;
Var BestMasuk : Integer;
    EfisiensiBest, EfisiensiRata2 : Real;
    MeasureBest, MeasureRata2 : Real;
    TotalEfisiensi : Real;
    AngkaRandom : Real;
Begin
    System.Assign(FGenetic, 'Temp\Genetic.gen');
    System.Reset(FGenetic);

    for i := 1 to System.FileSize(FGenetic) do
    Begin
        System.Seek(FGenetic, i-1);
        System.Read(FGenetic, Genetic);
        GenSeleksi[i].Urutan:=i;
        GenSeleksi[i].TotalCost:=Genetic.TotalCost;
    End;

    for i := 1 to System.FileSize(FGenetic)-1 do
        for j := i+1 to System.FileSize(FGenetic) do
            if GenSeleksi[i].TotalCost>GenSeleksi[j].TotalCost then
            Begin
                GenSeleksi[5001]:=GenSeleksi[i];
                GenSeleksi[i]:=GenSeleksi[j];
                GenSeleksi[j]:=GenSeleksi[5001];
            End;

            if IterasiJalan=Parameter.Generasi then
            Begin
                System.Seek(FGenetic, GenSeleksi[1].Urutan-1);
                System.Read(FGenetic, GeneticBest);
            End;

            CostRata2:=0;
            CostGood:=GenSeleksi[1].TotalCost;

            for i := 1 to Parameter.Populasi-1 do
                for j := i+1 to Parameter.Populasi do
                    if GenSeleksi[i].Urutan>GenSeleksi[j].Urutan then
                    Begin
                        GenSeleksi[5001]:=GenSeleksi[i];
                        GenSeleksi[i]:=GenSeleksi[j];
                        GenSeleksi[j]:=GenSeleksi[5001];
                    End;

                    for i := 1 to Parameter.Populasi do
                    Begin
                        CostRata2:=CostRata2+GenSeleksi[i].TotalCost;
                        System.Seek(FGenetic, GenSeleksi[i].Urutan-1);
                        System.Read(FGenetic, Genetic);
                        System.Seek(FGenetic, i-1);
                        System.Write(FGenetic, Genetic);
                    End;

                    CostRata2:=CostRata2/Parameter.Populasi;
```

Lampiran 5

```
System.Seek (FGenetic, Parameter.Populasi);
System.Truncate (FGenetic);

str (ReplikasiJalan, DumyString);
System.Assign (Test, 'Solusi\MeanEfi'+DumyString+'.txt');
System.Reset (Test);
System.Append (Test);

System.WriteLine (Test, IterasiJalan, ';', CostGood:0:2, ';', CostRata2:0:
2);
System.Close (Test);

Str (ReplikasiJalan, DumyString);
System.Assign (Test, 'Solusi\Summary'+DumyString+'.txt');
System.Reset (Test);
System.Append (Test);
for j := 1 to System.FileSize (FGenetic) do
Begin
System.Seek (FGenetic, j-1);
System.Read (FGenetic, Genetic);

System.Write (Test, IterasiJalan, ';', j, ';', Genetic.TotalCost:0:2, ';
');
for i := 1 to Genetic.JumGen do
Begin
if (Genetic.Gen[i].MesinKe<>0) and
(Kasus.Item[Genetic.Gen[i].ItemKe].Nama[1]<>'*') then
if i<Genetic.JumGen then
System.Write (Test, Kasus.Item[Genetic.Gen[i].ItemKe].Lambang, Geneti
c.Gen[i].OperasiKe, '-', Genetic.Gen[i].FraktalKe, '-
', Lokasi.Lokasi[Genetic.Gen[i].MesinKe].Lambang, '-
', Genetic.Gen[i].NoMesinKe, ';');
Else
System.WriteLine (Test, Kasus.Item[Genetic.Gen[i].ItemKe].Lambang, Gene
tic.Gen[i].OperasiKe, '-', Genetic.Gen[i].FraktalKe, '-
', Lokasi.Lokasi[Genetic.Gen[i].MesinKe].Lambang, '-
', Genetic.Gen[i].NoMesinKe);
End;

End;
System.Close (Test);
System.Close (FGenetic);
End;
procedure Penulisan_Hasil;
Begin
WaktuSelesai:=GetTime;
WaktuSelesai1:=DateTimetoStr (now);
Str (ReplikasiJalan, DumyString);
System.Assign (Test, 'Solusi\Best'+DumyString+'.txt');
System.Reset (Test);
System.Append (Test);
System.WriteLine (Test, 'Mulai : ', WaktuMulai);
System.WriteLine (Test, 'Selesai : ', WaktuSelesai1);
DumyString:=TimetoStr (WaktuSelesai-WaktuMulai);
```

Lampiran 5

```
TIterasi.Text:=DumyString;
System.Writeln(Test, 'Waktu      : ', DumyString);
System.Writeln(Test, 'Total OMH   :
', GeneticBest.TotalCost:0:2);
Str(GeneticBest.TotalCost:0:2, DumyString);
TCost.Text:=DumyString;
System.Writeln(Test);
System.Writeln(Test, 'No;Item;Mesin;Qty Produksi;OMH');
{System.Writeln(Test, 'No;Item;Mesin;Handling;Qty
Produksi;Cost');}

Tampilan.JumGen:=0;
for i := 1 to GeneticBest.JumGen do
Begin
  if GeneticBest.Gen[i].Produksi>0 then
  Begin
    Tampilan.JumGen:=Tampilan.JumGen+1;

Tampilan.Gen[Tampilan.JumGen].Item:=Kasus.Item[GeneticBest.Gen[i].
ItemKe].Nama;

Str(GeneticBest.Gen[i].FraktalKe, Tampilan.Gen[Tampilan.JumGen].Mes
in);

Tampilan.Gen[Tampilan.JumGen].Mesin:=Tampilan.Gen[Tampilan.JumGen]
.Mesin+'-'+Lokasi.Lokasi[GeneticBest.Gen[i].MesinKe].Nama;
  Str(GeneticBest.Gen[i].NoMesinKe, DumyString);

Tampilan.Gen[Tampilan.JumGen].Mesin:=Tampilan.Gen[Tampilan.JumGen]
.Mesin+'-'+DumyString;

Str(GeneticBest.Gen[i].Produksi:0:2, Tampilan.Gen[Tampilan.JumGen].
Produksi);

Str(GeneticBest.Gen[i].Cost:0:2, Tampilan.Gen[Tampilan.JumGen].Cost
);

{Tampilan.Gen[Tampilan.JumGen].Handling:=Kasus.Item[GeneticBest.Ge
n[i].ItemKe].Operasi[GeneticBest.Gen[i].OperasiKe].Handling; }
  End;
End;

for i := 1 to Tampilan.JumGen do
Begin
  System.Write(Test, i, ';');
  System.Write(Test, Tampilan.Gen[i].Item, ';');
  System.Write(Test, Tampilan.Gen[i].Mesin, ';');
  {System.Write(Test, Tampilan.Gen[i].Handling, ';'); }
  System.Write(Test, Tampilan.Gen[i].Produksi, ';');
  System.Writeln(Test, Tampilan.Gen[i].Cost);
End;
System.Writeln(Test);
System.Close(Test);

Str(ReplikasiJalan, DumyString);
```

Lampiran 5

```
System.Assign(Test, 'Solusi\DetailBest'+DumyString+'.txt');
System.Rewrite(Test);
System.Reset(Test);
System.Append(Test);

System.Writeln(Test, 'Item;Material;Kebutuhan;Produksi;Volume;Weight;Dari;Ke;Kapasitas;Waktu;Jumlah Produksi;Jumlah Pindah;Frekuensi;Jarak;Handling;Cost;Total OMH');
System.Close(Test);
Genetic:=GeneticBest;
Total_Cost_Tampil;

End;
Procedure Tampilan_Layar;
Begin
  if Tampilan.JumGen>=1 then
  Begin
    No1.Text:='1';
    Item1.Text:=Tampilan.Gen[1].Item;
    Mesin1.Text:=Tampilan.Gen[1].Mesin;
    Qty1.Text:=Tampilan.Gen[1].Produksi;
    Cost1.Text:=Tampilan.Gen[1].Cost;
    {Handling1.Text:=Tampilan.Gen[1].Handling;}
  End;
  if Tampilan.JumGen>=2 then
  Begin
    No2.Text:='2';
    Item2.Text:=Tampilan.Gen[2].Item;
    Mesin2.Text:=Tampilan.Gen[2].Mesin;
    Qty2.Text:=Tampilan.Gen[2].Produksi;
    Cost2.Text:=Tampilan.Gen[2].Cost;
    {Handling2.Text:=Tampilan.Gen[2].Handling;}
  End;
  if Tampilan.JumGen>=3 then
  Begin
    No3.Text:='3';
    Item3.Text:=Tampilan.Gen[3].Item;
    Mesin3.Text:=Tampilan.Gen[3].Mesin;
    Qty3.Text:=Tampilan.Gen[3].Produksi;
    Cost3.Text:=Tampilan.Gen[3].Cost;
    {Handling3.Text:=Tampilan.Gen[3].Handling;}
  End;
  if Tampilan.JumGen>=4 then
  Begin
    No4.Text:='4';
    Item4.Text:=Tampilan.Gen[4].Item;
    Mesin4.Text:=Tampilan.Gen[4].Mesin;
    Qty4.Text:=Tampilan.Gen[4].Produksi;
    Cost4.Text:=Tampilan.Gen[4].Cost;
    {Handling4.Text:=Tampilan.Gen[4].Handling;}
  End;
  if Tampilan.JumGen>4 then
  Begin
    SB.Enabled:=True;
    SB.Position:=1;
    SB.Min:=1;
```

Lampiran 5

```
        SB.Max:=Tampilan.JumGen-3;
        SB.SmallChange:=1;
        SB.LargeChange:=3;
    End;
End;
begin
    Val (TReplikasi.Text,Replikasi,i);
    Cleansing_Layar;

    PBReplikasi.Min:=0;
    PBReplikasi.Max:=Replikasi;
    PBReplikasi.Position:=0;
    for ReplikasiJalan := 1 to Replikasi do
    Begin
        Create_File;
        PBGenerasi.Min:=0;
        PBGenerasi.Max:=Parameter.Generasi;
        PBGenerasi.Position:=0;

        IterasiJalan:=0;
        PopulasiJalan:=0;
        CostRata2:=0;
        for PopulasiJalan := 1 to Parameter.Populasi do
        Begin
            Solusi_Awal;
            System.Assign (FGenetic, 'Temp\Genetic.gen');
            System.Reset (FGenetic);
            System.Seek (FGenetic, PopulasiJalan-1);
            System.Write (FGenetic, Genetic);
            System.Close (FGenetic);
            if (PopulasiJalan=1) or (CostGood>Genetic.TotalCost) then
                CostGood:=Genetic.TotalCost;
                CostRata2:=CostRata2+Genetic.TotalCost;
            End;
            CostRata2:=CostRata2/Parameter.Populasi;
            str (ReplikasiJalan, DumyString);
            System.Assign (Test, 'Solusi\MeanEfi'+DumyString+'.txt');
            System.Reset (Test);
            System.Append (Test);

            System.WriteLine (Test, IterasiJalan, ' ', CostGood:0:2, ' ', CostRata2:0:
2);

            System.Close (Test);

            For IterasiJalan := 1 to Parameter.Generasi do
            Begin
                CrossOver;
                Mutasi;
                Seleksi;
                PBGenerasi.Position:=IterasiJalan;
            End;
            Penulisan_Hasil;
            PBReplikasi.Position:=ReplikasiJalan;
        End;
        Tampilan_Layar;
    end;
```

Lampiran 5

```
procedure TFormProses.CButtonClick(Sender: TObject);
Var n : Integer;
begin
    System.Assign(FKasus, 'Case\' + TCase.Text + '.kss');
    {$I-}
    System.Reset(FKasus);
    {$I+}
    n:=IORESULT;
    System.Reset(FKasus);
    System.Seek(FKasus, 0);
    Read(FKasus, Kasus);
    System.Close(FKasus);

    if n=0 then
    Begin
        Click1:=1;
        LCase.Caption:=TCase.Text;
    End;

    System.Assign(FHandling, 'Case\' + TCase.Text + '.han');
    System.Reset(FHandling);
    System.Seek(FHandling, 0);
    System.Read(FHandling, Handling);
    System.Close(FHandling);

    System.Assign(FLokasi, 'Case\' + TCase.Text + '.lok');
    System.Reset(FLokasi);
    System.Seek(FLokasi, 0);
    System.Read(FLokasi, Lokasi);
    System.Close(FLokasi);
    LocatorTemp.JumLokasi:=Lokasi.JumLokasi;
    for i := 1 to Lokasi.JumLokasi do
    Begin
        LocatorTemp.Lokasi[i].Nama:=Lokasi.Lokasi[i].Nama;
        LocatorTemp.Lokasi[i].Lambang:=Lokasi.Lokasi[i].Lambang;
        for j := 1 to 6 do
        Begin
            LocatorTemp.Lokasi[i].Fraktal[j].JumMesin:=Lokasi.Lokasi[i].Fraktal[j];
            for k := 1 to Lokasi.Lokasi[i].Fraktal[j] do
                Begin
                    LocatorTemp.Lokasi[i].Fraktal[j].MesinKe[k].Kapasitas:=Lokasi.Lokasi[i].Kapasitas;
                    LocatorTemp.Lokasi[i].Fraktal[j].MesinKe[k].Sisa:=Lokasi.Lokasi[i].Kapasitas;
                End;
            End;
        End;
    End;

    System.Assign(FOptions, 'Case\' + TCase.Text + '.opt');
    System.Reset(FOptions);
```

Lampiran 5

```
System.Seek(FOptions, 0);
System.Read(FOptions, Options);
System.Close(FOptions);

System.Assign(FOpTotal, 'Case\'+'+TCase.Text+'.ott');
System.Reset(FOpTotal);
System.Seek(FOpTotal, 0);
System.Read(FOpTotal, OpTotal);
System.Close(FOpTotal);

System.Assign(FKromosom, 'Case\'+'+TCase.Text+'.krm');
System.Reset(FKromosom);
System.Seek(FKromosom, 0);
System.Read(FKromosom, Kromosom);
System.Close(FKromosom);

Kromosom.JumGen:=OpTotal.Item[OpTotal.JumItem].Mesin[OpTotal.Item[
OpTotal.JumItem].JumMesin].Ke;

Str(Lokasi.JumLokasi, DumyString);
TLokasi.Text:=DumyString;

Str(Handling.JumHandling, DumyString);
THandling.Text:=DumyString;

Str(Kasus.JumItem, DumyString);
TItem.Text:=DumyString;

TCase.Text:='';

if (Click1+Click2)= 2 then BProses.Enabled:=True;
if (Click1+Click2)< 2 then BProses.Enabled:=False;
end;

procedure TFormProses.FormCreate(Sender: TObject);
begin
    Randomize;
    Click1:=0;
    Click2:=0;
end;

procedure TFormProses.PButtonClick(Sender: TObject);
var n : integer;
begin

System.Assign(FParameter, 'Parameter\'+'+TParameter.Text+'.prm');
    {$I-}
    Reset(FParameter);
    {$I+}
    n:=IORESULT;
    Reset(FParameter);
    System.Seek(FParameter, 0);
    Read(FParameter, Parameter);
    System.Close(FParameter);

    if n=0 then
```


Lampiran 5

```
Begin
    Click2:=1;
    Str(Parameter.Populasi,DumyString);
    TPopulasi.Text:=DumyString;
    Str(Parameter.Generasi,DumyString);
    TGenerasi.Text:=DumyString;
    Str(Parameter.Pc:0:4,DumyString);
    TPc.Text:=DumyString;
    Str(Parameter.Pm:0:4,DumyString);
    TPm.Text:=DumyString;
End;
TParameter.Text:='';

if (Click1+Click2)= 2 then BProses.Enabled:=True;
if (Click1+Click2)< 2 then BProses.Enabled:=False;
end;

procedure TFormProses.SBChange(Sender: TObject);
begin
    str(SB.Position+1-1,DumyString);
    No1.Text:=DumyString;
    str(SB.Position+2-1,DumyString);
    No2.Text:=DumyString;
    str(SB.Position+3-1,DumyString);
    No3.Text:=DumyString;
    str(SB.Position+4-1,DumyString);
    No4.Text:=DumyString;
    Item1.Text:=Tampilan.Gen[SB.Position+1-1].Item;
    Item2.Text:=Tampilan.Gen[SB.Position+2-1].Item;
    Item3.Text:=Tampilan.Gen[SB.Position+3-1].Item;
    Item4.Text:=Tampilan.Gen[SB.Position+4-1].Item;
    Mesin1.Text:=Tampilan.Gen[SB.Position+1-1].Mesin;
    Mesin2.Text:=Tampilan.Gen[SB.Position+2-1].Mesin;
    Mesin3.Text:=Tampilan.Gen[SB.Position+3-1].Mesin;
    Mesin4.Text:=Tampilan.Gen[SB.Position+4-1].Mesin;
    Qty1.Text:=Tampilan.Gen[SB.Position+1-1].Produksi;
    Qty2.Text:=Tampilan.Gen[SB.Position+2-1].Produksi;
    Qty3.Text:=Tampilan.Gen[SB.Position+3-1].Produksi;
    Qty4.Text:=Tampilan.Gen[SB.Position+4-1].Produksi;
    Cost1.Text:=Tampilan.Gen[SB.Position+1-1].Cost;
    Cost2.Text:=Tampilan.Gen[SB.Position+2-1].Cost;
    Cost3.Text:=Tampilan.Gen[SB.Position+3-1].Cost;
    Cost4.Text:=Tampilan.Gen[SB.Position+4-1].Cost;
    {Handling1.Text:=Tampilan.Gen[SB.Position+1-1].Handling;
    Handling2.Text:=Tampilan.Gen[SB.Position+2-1].Handling;
    Handling3.Text:=Tampilan.Gen[SB.Position+3-1].Handling;
    Handling4.Text:=Tampilan.Gen[SB.Position+4-1].Handling;}
end;

procedure TFormProses.TCaseChange(Sender: TObject);
begin
    if TCase.Text<>' ' then CButton.Enabled:=True;
    if TCase.Text=' ' then CButton.Enabled:=False;
end;
```

Lampiran 5

```
procedure TFormProses.TCaseKeyPress(Sender: TObject; var Key:
Char);
begin
    if Key = #13 then CButton.Click;
end;

procedure TFormProses.TGenerasiChange(Sender: TObject);
begin
    Val(TGenerasi.Text,Parameter.Generasi,i);
    if (TGenerasi.Text<>'') and (TPopulasi.Text<>'') and
(TPC.Text<>'') and (TPm.Text<>'') then CLick2:=1;
    if (TGenerasi.Text='') or (TPopulasi.Text='') or
(TPC.Text='') or (TPm.Text='') then CLick2:=0;
    if (Click1+Click2)= 2 then BProses.Enabled:=True;
    if (Click1+Click2)< 2 then BProses.Enabled:=False;
end;

procedure TFormProses.TParameterChange(Sender: TObject);
begin
    if TParameter.Text<>' ' then PButton.Enabled:=True;
    if TParameter.Text=' ' then PButton.Enabled:=False;
end;

procedure TFormProses.TParameterKeyPress(Sender: TObject; var Key:
Char);
begin
    if Key = #13 then PButton.Click;
end;

procedure TFormProses.TPcChange(Sender: TObject);
begin
    Val(TPc.Text,Parameter.Pc,i);
    if (TGenerasi.Text<>'') and (TPopulasi.Text<>'') and
(TPC.Text<>'') and (TPm.Text<>'') then CLick2:=1;
    if (TGenerasi.Text='') or (TPopulasi.Text='') or
(TPC.Text='') or (TPm.Text='') then CLick2:=0;
    if (Click1+Click2)= 2 then BProses.Enabled:=True;
    if (Click1+Click2)< 2 then BProses.Enabled:=False;
end;

procedure TFormProses.TPmChange(Sender: TObject);
begin
    Val(TPm.Text,Parameter.Pm,i);
    if (TGenerasi.Text<>'') and (TPopulasi.Text<>'') and
(TPC.Text<>'') and (TPm.Text<>'') then CLick2:=1;
    if (TGenerasi.Text='') or (TPopulasi.Text='') or
(TPC.Text='') or (TPm.Text='') then CLick2:=0;
    if (Click1+Click2)= 2 then BProses.Enabled:=True;
    if (Click1+Click2)< 2 then BProses.Enabled:=False;
end;

procedure TFormProses.TPopulasiChange(Sender: TObject);
begin
    Val(TPopulasi.Text,Parameter.Populasi,i);
    if (TGenerasi.Text<>'') and (TPopulasi.Text<>'') and
(TPC.Text<>'') and (TPm.Text<>'') then CLick2:=1;
```

Lampiran 5

```
        if (TGenerasi.Text='') or (TPopulasi.Text='') or
(TPC.Text='') or (TPm.Text='') then CLick2:=0;
        if (Click1+Click2)= 2 then BProses.Enabled:=True;
        if (Click1+Click2)< 2 then BProses.Enabled:=False;
end;

end.
```

KOMENTAR DOSEN PENGUJI

Nama mahasiswa : Angelia Sary Putry

NRP : 0723052

Judul Tugas Akhir : Usulan Penugasan Mesin pada Tata Letak Fraktal
Menggunakan Algoritma Genetika di PT. Gamatara,
Cimahi

Komentar dan Saran Dosen Penguji:

1. Heuristik 1 alternatif, metaheuristik banyak alternatif.

DATA PENULIS

Nama : Angelia Sary Putry
Alamat di Bandung : Jln. Sarimadu Blok 26 No. 100A
No. Handphone : 081947051115
Alamat email : Miel_Forever05@yahoo.com
Pendidikan : SMAK 1 BPK Penabur Cirebon
Jurusan Teknik Industri Universitas Kristen
Maranatha, Bandung
Nilai Tugas Akhir : A
Tanggal USTA : 25 Januari 2011