

# **MODUL PRAKTIKUM DASAR PEMROGRAMAN KOMPUTER**

**NAMA MAHASISWA :**

**NIM MAHASISWA :**

**LABORATORIUM TEKNIK ELEKTRO  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS MUHAMMADIYAH MALANG  
2019**

## DAFTAR ISI

<b>DAFTAR ISI</b> .....	<b>ii</b>
<b>DAFTAR GAMBAR</b> .....	<b>vi</b>
<b>DAFTAR TABEL</b> .....	<b>vii</b>
<b>STANDART OPERASIONAL PROSEDUR</b> .....	<b>viii</b>
<b>A. PRA PRAKTIKUM</b> .....	<b>viii</b>
<b>B. PRA PELAKSANAAN PERCOBAAN PRAKTIKUM</b> .....	<b>viii</b>
<b>C. PRAKTIKUM BERLANGSUNG</b> .....	<b>ix</b>
<b>D. PRAKTIKUM BERAKHIR</b> .....	<b>ix</b>
<b>E. PASCA PRAKTIKUM</b> .....	<b>x</b>
<b>F. SANKSI</b> .....	<b>x</b>
<b>G. KESELAMATAN DAN KESEHATAN KERJA (K3)</b> .....	<b>xi</b>

### PERCOBAAN I TIPE DATA DAN OPERATOR

1.1 Tujuan .....	1
1.2 Alat yang digunakan .....	1
1.3 Dasar Teori .....	1
1.4 Prosedur Percobaan.....	4
1.5 Listing Program Angka Penting .....	4
1.5.1 Flowchart .....	4
1.5.2 Data Hasil Percobaan .....	4
1.5.3 Analisa Data .....	5
1.5.4 Kesimpulan .....	5
1.6 Listing Program Perkalian .....	5
1.6.1 Flowchart .....	5
1.6.2 Data Hasil Percobaan .....	5
1.6.3 Analisa Data .....	5
1.6.4 Kesimpulan .....	5
1.7 Listing Program Aritmatika .....	6

1.7.1 Flowchart .....	6
1.7.2 Data Hasil Percobaan.....	6
1.7.3 Analisa Data .....	6
1.7.4 Kesimpulan .....	6
<b>PERCOBAAN II PENGONTROLAN ALIRAN PROGRAM</b>	
2.1 Tujuan .....	7
2.2 Alat yang diperlukan.....	7
2.3 Dasar Teori .....	7
2.4 Prosedur Percobaan.....	9
2.5 Listing program menampilkan nilai menggunakan perintah if – else	10
2.5.1 Flowchart .....	10
2.5.2 Data Hasil Percobaan.....	10
2.5.3 Analisa Data .....	10
2.5.4 Kesimpulan .....	10
2.6 Listing program menampilkan nilai menggunakan perintah switch	11
2.6.1 Flowchart .....	11
2.6.2 Data Hasil Percobaan.....	11
2.6.3 Analisa Data .....	11
2.6.4 Kesimpulan .....	11
2.7 Listing program menjumlahkan 1+2+3... menggunakan do-while .	12
2.7.1 Flowchart .....	12
2.7.2 Data Hasil Percobaan.....	12
2.7.3 Analisa Data .....	12
2.7.4 Kesimpulan .....	12
2.8 Listing program menjumlahkan 1+2+3... menggunakan while .....	13
2.8.1 Flowchart .....	13
2.8.2 Data Hasil Percobaan.....	13
2.8.3 Analisa Data .....	13
2.8.4 Kesimpulan .....	13
2.9 Listing program for .....	14
2.9.1 Flowchart .....	14

2.9.2 Data Hasil Percobaan.....	14
2.9.3 Analisa Data .....	14
2.9.4 Kesimpulan .....	14
2.10 Listing program continue dan break .....	15
2.10.1 Flowchart continue.....	15
2.10.1 Flowchart break .....	15
2.10.2 Data Hasil Percobaan.....	15
2.10.3 Analisa Data .....	15
2.10.4 Kesimpulan .....	15

### **PERCOBAAN III FUNGSI**

3.1 Tujuan .....	16
3.2 Alat yang digunakan .....	16
3.3 Dasar Teori .....	16
3.4 Prosedur Percobaan.....	16
3.5 Listing program pengiriman parameter dengan nilai .....	17
3.5.1 Data Hasil Percobaan.....	17
3.5.2 Analisa Data .....	17
3.5.3 Kesimpulan .....	17
3.6 Listing program pengiriman parameter dengan alamat .....	18
3.6.1 Data Hasil Percobaan.....	18
3.6.2 Analisa Data .....	18
3.6.3 Kesimpulan .....	18

### **PERCOBAAN IV ARRAY**

4.1 Tujuan .....	19
4.2 Alat yang digunakan .....	19
4.3 Dasar Teori .....	19
4.4 Prosedur Percobaan.....	20
4.5 Listing program menghitung nilai rata-rata dari bilangan positif ...	21
4.5.1 Flowchart .....	21
4.5.2 Data Hasil Percobaan.....	21
4.5.3 Analisa Data .....	22

4.5.4 Kesimpulan .....	22
4.6 Listing program bubble sort .....	22
4.6.1 Flowchart .....	23
4.6.2 Data Hasil Percobaan.....	23
4.6.3 Analisa Data .....	23
4.6.4 Kesimpulan .....	23

## **PERCOBAAN V STRUKTUR**

5.1 Tujuan .....	24
5.2 Alat yang digunakan .....	24
5.3 Dasar Teori .....	24
5.4 Prosedur Percobaan.....	25
5.5 Listing program mencari luas dengan operator $\rightarrow$ .....	26
5.5.1 Flowchart .....	26
5.5.2 Data Hasil Percobaan .....	26
5.5.3 Analisa Data .....	26
5.5.4 Kesimpulan .....	26
5.6 Listing program mencari luas dengan operator $*$ .....	27
5.6.1 Flowchart .....	27
5.6.2 Data Hasil Percobaan .....	27
5.6.3 Analisa Data.....	27
5.6.4 Kesimpulan .....	27

## DAFTAR FLOWCHART

Flowchart 1.5.1 Program Angka Penting .....	4
Flowchart 1.6.1 Program Perkalian.....	5
Flowchart 1.7.1 Program Aritmatika .....	6
Flowchart 2.5.1 Program if – else .....	10
Flowchart 2.6.1 Program switch .....	11
Flowchart 2.7.1 Program do-while.....	12
Flowchart 2.8.1 Program while.....	13
Flowchart 2.9.1 Program for .....	14
Flowchart 2.10.1 Program continue .....	15
Flowchart 2.10.1 Program break .....	15
Flowchart 4.5.1 Program menghitung nilai rata-rata dari bilangan positif ...	21
Flowchart 4.6.1 Program bubble sort.....	23
Flowchart 5.5.1 Program mencari luas dengan operator → .....	26
Flowchart 5.6.1 Program mencari luas dengan operator *.....	27

## DAFTAR TABEL

Tabel 1.1 Jangkauan dan Ketelitian Tipe Data .....	1
Tabel 1.2 Hirarki Operator .....	2
Tabel 1.3 Hasil percobaan program ketelitian tipe data.....	4
Tabel 1.4 Hasil percobaan program perkalian .....	5
Tabel 1.5 Hasil percobaan program aritmatika .....	6
Tabel 2.1 Hasil percobaan program if-else .....	10
Tabel 2.2 Hasil percobaan program switch.....	11
Tabel 2.3 Hasil percobaan program do – while .....	12
Tabel 2.4 Hasil percobaan program while .....	13
Tabel 2.5 Hasil percobaan program for.....	14
Tabel 2.6 Hasil percobaan program continue dan break.....	15
Tabel 3.1 Program pengirim parameter dengan nilai .....	17
Tabel 3.2 Program pengirim parameter dengan alamat .....	18
Tabel 4.1 Hasil percobaan program menghitung rata-rata.....	21
Tabel 4.2 Hasil percobaan bubble sort.....	23
Tabel 5.1 Hasil percobaan program mencari luas dengan operartor ->.....	26
Tabel 5.2 Hasil percobaan program mencari luas dengan operartor *.....	27

**STANDART OPERASIONAL PROSEDUR  
LABORATORIUM TEKNIK ELEKTRO  
UNIVERSITAS MUHAMMADIYAH MALANG**

**A. PRA PRAKTIKUM**

1. Ka Laboratorium bersama Ketua Prodi menetapkan daftar Mata Praktikum yang akan dilaksanakan pada semester berjalan
2. Laboran atau Staf mengumumkan daftar Mata Praktikum dan pengumuman lainnya via web lab-elektro.umm.ac.id
3. Staf / Laboran menerima pendaftaran calon praktikan yang mengulang
4. Staf / Laboran mengumumkan daftar peserta Mata Praktikum berdasarkan data peserta mata kuliah dan peserta mengulang di web lab-elektro.umm.ac.id
5. Kepala lab dan wakil kepala lab menetapkan daftar Instruktur dan Asisten Mata Praktikum dan diusulkan untuk ditetapkan SK Dekan
6. Ka. Lab mengundang Peserta Mata Praktikum untuk mengikuti pertemuan persiapan dan pembagian jadwal peserta mengikuti praktikum dan peraturan serta prosedur praktikum dan K3
7. Instruktur dan Asisten mengundang peserta Mata Praktikum untuk mengikuti Ujian Pra Praktikum (Memberikan Tugas Pra Praktikum)

**B. PRA PELAKSANAAN PERCOBAAN PRAKTIKUM**

1. Asisten dan Praktikan hadir 15 menit sebelum dimulai jam praktikum
2. Asisten mempersiapkan instrumen ukur serta modul praktikum dan peralatan pendukung seperti kabel, jumper dan lain lain
3. Praktikan membaca petunjuk praktikum dan mempersiapkan kebutuhan peralatan sebelum masuk ruang/lab
4. Asisten memberikan salam dan ucapan selamat datang dengan senyum serta memberikan arahan kepada kelompok Praktikan tentang prosedur pelaksanaan praktikum dan penjelasan daftar peralatan dan modul



5. Asisten menunjuk peserta yang menjadi petugas pencatat, melakukan pengukuran dan pembantu pelaksanaan
6. Asisten meminta kelompok Praktikum untuk membaca doa/Basmalah sebelum dimulai pemasangan dan instalasi praktikum dan dipandu oleh Asisten

### **C. PRAKTIKUM BERLANGSUNG**

1. Asisten memberikan instruksi kepada kelompok praktikan pemasangan atau instalasi modul dan mengawasi dan mengevaluasi serta memeriksa hasil pemasangan dan memastikan kebenaran instalasi
2. Praktikan dan asisten saling menjaga kenyamanan dan ketertiban praktikum sesuai tata tertib yang berlaku serta menjaga keamanan perangkat lab selama pelaksanaan praktikum dari satu percobaan ke percobaan berikutnya.
3. Asisten berhak menegur dan menindak praktikan apabila ketahuan merusak, mengubah atau memindahkan perlengkapan lab tanpa izin.
4. Asisten melakukan penilaian dan pengawasan tiap praktikan melakukan pengukuran selama percobaan.
5. Asisten dan kelompok praktikan mengakhiri praktikum dengan membaca hamdallah dan mengucapkan salam serta meminta praktikan untuk merapikan peralatan dan modul serta kursi dan membuang sampah di sekitarnya.

### **D. PRAKTIKUM BERAKHIR**

1. Praktikan meninggalkan ruangan dengan rapi dan teratur.
2. Asisten Mengkondisikan ruangan kembali,
  - a. Mengembalikan/mengatur kursi kembali.
  - b. Merapikan sampah yang ditemukan berserakan dalam ruangan.
  - c. Mengembalikan peralatan dan modul ke Lemari Alat dan Modul sesuai nama jenis Mata Praktikum
  - d. Mengunci pintu
  - e. Mematikan lampu apabila tidak ada praktikum berikutnya.

3. Asisten menandatangani presensi kelompok dan memberikan daftar penilaian kerja percobaan kelompok ke ruang administrasi (Laboran).
4. Instruktur dan atau asisten melakukan evaluasi reguler praktikum jika diperlukan.

#### **E. PASCA PRAKTIKUM**

1. Praktikan menyusun laporan semua percobaan
2. Praktikan melakukan asistensi laporan ke Asisten Praktikum min 4 kali
3. Setelah laporan praktikum ditandatangani oleh Asisten, Tiap Praktikum menghadap Instruktur sesuai jadwal yang ditetapkan Instruktur
4. Instruktur menguji praktikum mengenai proses pelaksanaan praktikum
5. Instruktur memberikan nilai akhir praktikan
6. Nilai akhir pratikum diserahkan ke Lab untuk proses administrasi

#### **F. SANKSI**

1. Keterlambatan asistensi pertama kali sanksi point 1
2. Tidak memenuhi minimal 4 kali asistensi sanksi point 2
3. Datang terlambat 15 menit dari waktu yang telah ditentukan sanksi point 3
4. \* Tidak mengikuti proses praktikum tanpa adanya konfirmasi sanksi point 4
5. \* Tidak mengikuti ujian koordinator tanpa adanya konfirmasi sanksi point 5
6. Keterlambatan pengumpulan laporan resmi sanki point 6
7. \* Tidak mengikuti ujian instruktur sesuai dengan jadwal yang ditentukan instruktur sanksi point 7
8. Pemalsuan tanda tangan selama proses praktikum berlangsung sanksi point 8
9. Merusakkan perlatan Lab. Teknik Elektro sanksi point 9

\* Maksimal konfirmasi 2 x 24 jam sejak jadwal resmi diumumkan untuk penggantian jadwal ujian

Point 1	Menulis materi modul bab 1
---------	----------------------------

Point 2	Menulis materi modul bab 1-3 & Pengurangan Nilai
Point 3	Menulis materi 1 bab & Pengurangan Nilai
Point 4	Mengulang (tidak konfirmasi sesuai waktu yang telah ditentukan) atau Pengurangan Nilai
Point 5	Mengulang (tidak konfirmasi sesuai waktu yang telah ditentukan) atau Pengurangan Nilai
Point 6	Membeli buku berkaitan dengan bidang Teknik elektro
Point 7	Pengurangan Nilai Instruktur
Point 8	Mengulang Praktikum atau mendapat Nilai E
Point 9	Mengganti peralatan tersebut sesuai dengan spesifikasi atau mirip dan memiliki fungsi yang sama

### **G. KESELAMATAN DAN KESEHATAN KERJA (K3)**

1. Sebelum memulai praktikum, praktikan memahami tata tertib dan keselamatan di Laboratorium
2. Mengetahui tempat dan cara penggunaan peralatan Laboratorium
3. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik ( stop kontak dan circuit breaker)
4. Praktikan harus memperhatikan dan menaati peringatan (warning) yang biasa tertera pada badan peralatan praktikum maupun rambu peringatan yang terdapat di ruangan Laboratorium
5. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera laporkan ke asisten terkait atau dapat langsung melapor ke laboran.
6. Hindari daerah atau benda yang berpotensi menimbulkan bahaya listrik ( sengatan listrik) secara tidak sengaja, missal seperti jala-jala kabel yang terkelupas
7. Keringkan bagian tubuh yang basah, seperti keringat atau sisa air wudhu
8. Selalu waspada terhadap bahaya listrik pada setiap aktifitas praktikum.
9. Jika terjadi kecelakaan akibat bahaya listrik, berikut ini adalah hal-hal yang harus diikuti praktikan:
  - a) Jangan panik
  - b) Matikan semua peralatan elektronik dan sumber listrik di meja masing-masing dan di meja praktikum yang tersengat arus listrik.
  - c) Bantu praktikan yang tersengat arus listrik untuk melepaskan diri dari sumber listrik

- d) Beritahukan dan minta bantuan kepada laboran, praktikan lain dan orang di sekitar anda tentang terjadinya kecelakaan akibat bahaya listrik.
10. Jangan membawa benda-benda mudah terbakar (korek api, gas, dll) ke dalam ruangan laboratorium bila tidak disyaratkan dalam modul praktikum.
  11. Jangan melakukan sesuatu yang menimbulkan api, percikan api, atau panas yang berlebihan.
  12. Jangan melakukan sesuatu yang menimbulkan bahaya api atau panas berlebih pada diri sendiri atau orang lain.
  13. Selalu waspada terhadap bahaya api atau panas berlebih pada setiap aktivitas di laboratorium.
  14. Jika terjadi kecelakaan akibat bahaya listrik, berikut ini adalah hal-hal yang harus diikuti praktikan:
    - a) Jangan panik
    - b) Matikan semua peralatan elektronik dan sumber listrik di meja masing-masing.
    - c) Beritahukan dan minta bantuan laboran, praktikan lain dan orang di sekitar anda tentang terjadinya bahaya api atau panas berlebih
    - d) Menjauh dari ruang praktikum
  15. Dilarang membawa benda tajam (pisau, gunting dan sejenisnya) ke ruang praktikum bila tidak diperlukan untuk pelaksanaan percobaan
  16. Dilarang memakai perhiasan dari logam misalnya cincin, kalung, gelang, dll
  17. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai.
- Tidak melakukan sesuatu yang dapat menimbulkan luka pada diri sendiri atau orang lain

# PERCOBAAN I

## *TIPE DATA DAN OPERATOR*

### 1.1. Tujuan

Mempelajari jangkauan, ketelitian dan penggabungan data serta hirarki operator.

### 1.2. Alat yang digunakan

- 1 Set Komputer
- Sistem Operasi Windows 10
- Kompiler Code Blocks

### 1.3. Dasar Teori

Tipe data dan operator menentukan ketepatan suatu nilai yang disimpan di komputer. Suatu nilai disimpan di komputer dengan menggunakan tipe data berupa variable, literal dan konstanta. Tipe data tersebut mempunyai jangkauan serta ketelitian seperti yang diperlihatkan pada table berikut.

Tabel 1.1 Jangkauan dan Ketelitian Tipe Data

<b>Tipe Data</b>	<b>Jangkauan</b>	<b>Ketelitian (bit)</b>
Char	-128 – 127	8
unsigned char	0 – 255	8
int.	-32.768 – 32.767	16
unsigned int.	0 – 65.535	16
Short	-32.768 – 32.767	16
unsigned short	0 – 65.535	16
Long	-2.147.483.648 – 2.147.483.647	32
unsigned long	0 – 4.294.967.295	32
Float	3.4E-38 - 3,4E38	32

Double	1,7E-308 - 1,7E308	64
--------	--------------------	----

Operator merupakan simbol untuk mengubah nilai dari tipe data. Dalam suatu pernyataan matematis, hirarki. Operator mempengaruhi hasil perhitungan. Hirarki Operator tersebut berupa tingkatan pengerjaan seperti yang diperlihatkan pada tabel berikut

Tabel 1.2 Hirarki Operator

Tingkat	Simbol	Penjelasan	Keterkaitan
1	()	Pemanggilan Fungsi	Dari kiri ke kanan
2	[]	Subskrip Array	Dari kiri ke kanan
2	!	Negasi Logika	Dari kiri ke kanan
3	~	Komplemen Satu/bitwise	Dari kanan ke kiri
3	+	Tanda Plus Unary	Dari kanan ke kiri
3	-	Tanda Minus Unary	Dari kanan ke kiri
3	&	Alamat	Dari kanan ke kiri
3	*	Tak Langsung	Dari kanan ke kiri
3	sizeof	Ukuran Tipe Data	Dari kanan ke kiri
3	Type	Typecast (pengonversian)	Dari kanan ke kiri
4	()	Kurung Ekspresi	Dari kiri ke kanan
5	*	Perkalian	Dari kiri ke kanan
5	/	Pembagian	Dari kiri ke kanan
5	%	Sisa (modulus)	Dari kiri ke kanan
6	+	Tanda Plus Unary	Dari kiri ke kanan
6	-	Tanda Minus Unary	Dari kiri ke kanan
7	<<	Penggeseran ke kiri	Dari kiri ke kanan
7	>>	Penggeseran ke kanan	Dari kiri ke kanan
8	<	Kurang dari	Dari kiri ke kanan

8	<=	Kurang dari atau sama dengan	Dari kiri ke kanan
8	>	Lebih dari	Dari kiri ke kanan
8	>=	Lebih dari atau sama dengan	Dari kiri ke kanan
9	==	Sama dengan	Dari kiri ke kanan
9	!=	Tidak sam dengan	Dari kiri ke kanan
10	&	AND Bitwise	Dari kiri ke kanan
11	^	XOR Bitwise	Dari kiri ke kanan
12		OR Bitwise	Dari kiri ke kanan
13	&&	AND Logika	Dari kiri ke kanan
14		OR Logika	Dari kiri ke kanan
15	?:	Operator Kondisional	Dari kiri ke kanan
16	=	Penugasan Sederhana	Dari kanan ke kiri
16	*=	Penugasan Majemuk Perkalian	Dari kanan ke kiri
16	/=	Penugasan Pembagian	Dari kanan ke kiri
16	%=	Penugasan Majemuk Sisa Pembagian	Dari kanan ke kiri
16	+=	Penugasan Majemk Penjumlahan	Dari kanan ke kiri
16	-=	Penugasan Majemuk Pengurangan	Dari kanan ke kiri
16	&=	Penugasan Majemuk AND Bitwise	Dari kanan ke kiri
16	^=	Penugasan Majemuk XOR Bitwise	Dari kanan ke kiri
16	=	Penugasan Majemuk OR Bitwise	Dari kanan ke kiri
16	<<=	Pergeseran Majemuk geser kiri	Dari kanan ke kiri
16	>>=	Penugasan Majemuk geser kanan	Dari kanan ke kiri
17	,	Petunjuk Urutan	Dari kanan ke kiri

#### 1.4. Prosedur Percobaan

- Bukalah Window baru dari program Code Blocks(New Project).
- Ketiklah listing program yang di berikan berikut ini.
- Compile listing program yang telah dibuat (F9). Apabila masih ada yang error, tanyakan lah pada asisten pendamping.
- Simpanlah listing program yang telah berhasil di DRIVE D,dalam Folder yang telah anda buat.

#### 1.5 Listing program angka penting

##### 1.5.1 Flowchart



```
*1.6.c X
1  #include <stdio.h>
2  void main ()
3  {
4  float teliti = 0.1234567809876543210;
5  double lebih_teliti = 0.12345678909876543210;
6  printf ("nilai dengan float \t%21.5f\n", teliti);
7  printf ("nilai dengan double \t%21.5f\n", lebih_teliti);
8  }
9
```

##### 1.5.2 Data Hasil Percobaan

Tabel 1.4 Hasil percobaan program ketelitian tipe data

Tipe Data	Operand	Hasil
Float	5 angka penting	
Float	10 angka penting	
Float	20 angka penting	
Double	5 angka penting	
Double	10 angka penting	
Double	20 angka penting	



1.5.3 Analisa Data

1.5.4 Kesimpulan

## 1.6 Listing program perkalian

```
Start here x 1.7.c x
1  #include <stdio.h>
2  void main()
3  {
4  int i=8, hasil1;
5  float hasil2, f=6.9;
6  hasil1=f*i;
7  hasil2=(int) f*i;
8  printf("hasil1:%d\n", hasil1);
9  printf("hasil2:%f\n", hasil2);
10 }
11
```

1.6.1 Flowchart

1.6.2 Data Hasil Percobaan

Tabel 1.5 Hasil percobaan program perkalian

Operand	Operand II	Operator	Hasil
(float) 6.9	(int) 8	Perkalian	
(int) 6.9	(int) 8	Perkalian	

1.6.3 Analisa Data

1.6.4 Kesimpulan

## 1.7 Listing program aritmatika

```
Start here x 1.8.c x
1  #include <stdio.h>
2  void main()
3  {
4  int x,y,z,hasil1,hasil2,hasil3,hasil4;
5  printf ("masukkan nilai x :");
6  scanf ("%d",&x);
7  printf ("masukkan nilai y :");
8  scanf ("%d",&y);
9  printf ("masukkan nilai z :");
10 scanf ("%d",&z);
11 hasil1=x+y*z;
12 hasil2=(x+y)*z;
13 hasil3=x*y+z;
14 hasil4=(x*y)+z;
15 printf ("\n hasil1 = %d",hasil1);
16 printf ("\n hasil2 = %d",hasil2);
17 printf ("\n hasil3 = %d",hasil3);
18 printf ("\n hasil4 = %d",hasil4);
19 }
20
```

1.7.1 Flowchart

1.7.2 Data Hasil Percobaan

Tabel 1.5 Hasil percobaan program aritmatika

Pernyataan Matematika	Hasil
$x+y*z$	
$(x+y)*z$	
$x*y+z$	
$(x*y)+z$	

1.7.3 Analisa Data

1.7.4 Kesimpulan

## PERCOBAAN II

### *PENGONTROLAN ALIRAN PROGRAM*

#### 2.1. Tujuan

Mempelajari pengontrolan aliran program, meliputi aliran program berurutan, pencabangan dan perulangan.

#### 2.2. Alat yang diperlukan

- 1 Set Komputer
- Sistem Operasi Windows 10
  - Kompiler Code Blocks

#### 2.3. Dasar Teori

Pengontrolan aliran program adalah upaya untuk mengatur aliran eksekusi supaya berkembang dan bercabang sesuai dengan perubahan keadaan data. Pengontrolan aliran program membentuk pencabangan dan perulangan. Pengontrolan aliran program mengubah aliran eksekusi yang biasa dilakukan secara berurutan.

Bahasa C menyediakan cara untuk menghasilkan pengontrolan aliran program yang akan dijalankan berupa pernyataan-pernyataan pencabangan, yaitu *If – else* dan *switch*. Bentuk *If – else* menyebabkan aliran eksekusi dijalankan melalui pernyataan-boolean sehingga hanya bagian tertentu dari program saja yang dijalankan.

```
If (Pernyataan-boolean) Pernyataan1;  
[else Pernyataan2;]
```

Pernyataan-boolean berupa sembarang pernyataan yang menghasilkan besaran boolean, misalkan pernyataan yang menggunakan operator relasi untuk membentuk pernyataan perbandingan. Besaran boolean adalah suatu besaran dengan nilai dua saja yaitu „0“ dan „1“. *If-else* ini akan mengerjakan *pernyataan1*

jika hanya pernyataan boolean bernilai „0“. Bentuk *switch* adalah cara lain untuk menyatakan *if-else* dengan banyak pernyataan-boolean.

```
switch( pernyataan){  
    case (boolean1):prnyataan1;break;  
    case(boolean2):pernyataan2;break;  
    default:pernyataan3;break;  
}
```

Cara lain pengontrolan aliran program adalah perulangan yaitu mengulangi eksekusi blok program tertentu sampai tercapainya kondisi untuk menghentikannya (terminasi). Setiap perulangan memiliki empat bagian yaitu: inisiali, badan program, interasi dan terminasi. Inisialii adalah program yang menyiapkan keadaan awal awal perulangan. Badan program adalah pernyataan yang diulang. Iterasi adalah program yang dijalankan setelah badan program tetapi sebelum bagian tersebut dijalankan lagi. Interasi digunakan untuk menambah atau mengurangi pencacah. Sedangkan terminasi adalah boolean yang diperiksa setiap kali selama perulangan untuk melihat apakah sudah waktunya untuk menghentikan eksekusi. Bahasa C menyediakan 3 macam perulangan yaitu: *while*, *do-while*, dan *for*, dalam bentuk;

```
[inisialisasi;]  
While(terminasi) {  
    badan_program;  
    [iterasi;]  
}
```

Bagian inisialisasi dan iterasi bersifat pilihan, yaitu selama pernyataan terminasi menghasilkan keadaan “benar”, maka pernyataan dibadan program akan terus dieksekusi.

Kadang – kadang diijinkan untuk mengeksekusi badan program sekurang-kurangnya sekali sehingga pemeriksaan pernyataan terminasi di akhir perulangan seperti dalam bentuk berikut :

```
[inisialisasi;]
```

```
do {  
    badan_program;  
    [iterasi;]  
} while(terminasi);
```

Cara lain untuk menyatakan perulangan *while* adalah dengan pernyataan *for*. *for(inisialisasi;terminasi;iterasi)badan\_program;*

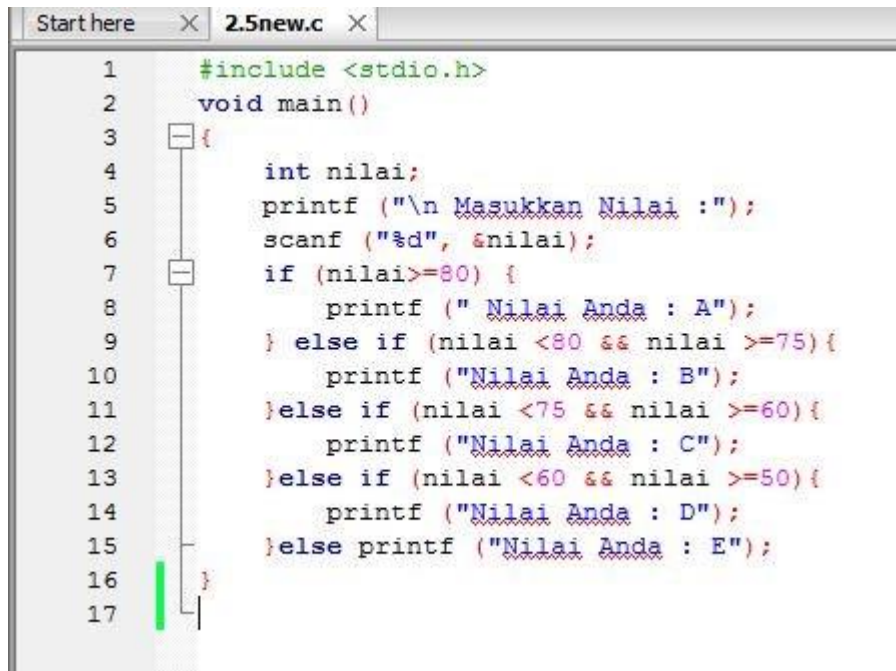
Dalam hal ini, jika inisialisasi tidak menyebabkan terminasi menghasilkan keadaan “benar” ketika pertama berjalan, maka pernyataan iterasi dan badan\_program tidak akan dijalankan. Sebaliknya akan dicacah dari suatu nilai minimum sampai maksimumnya.

Pernyataan aliran program yang lebih rumit dilakukan dengan memberikan pernyataan break yaitu pernyataan untuk memutus perulangan dan pernyataan continue yaitu pernyataan untuk meneruskan ke iterasi berikutnya tanpa mengeksekusi pernyataan-pernyataan dibawahnya.

#### 2.4 Prosedur Percobaan

- Bukalah Window baru dari program Code Blocks(New Project).
- Ketiklah listing program yang di berikan berikut ini.
- Compile listing program yang telah dibuat (F9). Apabila masih ada yang error, tanyakan lah pada asisten pendamping.
- Simpanlah listing program yang telah berhasil di DRIVE D,dalam Folder yang telah anda buat.

## 2.5 Listing program menampilkan nilai menggunakan perintah if – else



```
1  #include <stdio.h>
2  void main()
3  {
4      int nilai;
5      printf ("\n Masukkan Nilai :");
6      scanf ("%d", &nilai);
7      if (nilai>=80) {
8          printf (" Nilai Anda : A");
9      } else if (nilai <80 && nilai >=75){
10         printf ("Nilai Anda : B");
11     }else if (nilai <75 && nilai >=60){
12         printf ("Nilai Anda : C");
13     }else if (nilai <60 && nilai >=50){
14         printf ("Nilai Anda : D");
15     }else printf ("Nilai Anda : E");
16 }
17
```

### 2.5.1 Flowchart

### 2.5.2 Data Hasil Percobaan

Tabel 2.1 Hasil percobaan program if-else

Nilai	Hasil
48	
77	
85	
63	
55	

### 2.5.3 Analisa Data

### 2.5.4 Kesimpulan

## 2.6 Listing program menampilkan nilai menggunakan perintah switch

```
Start here x 2.6.c x
1  #include <stdio.h>
2  void main()
3  {
4      char nilai;
5      printf ("\n nilai ?");
6      scanf ("%c", &nilai);
7      switch (nilai)
8      {
9          case ('a'):printf("baik sekali");break;
10         case ('b'):printf("baik");break;
11         case ('c'):printf("cukup");break;
12         case ('d'):printf("kurang");break;
13         default:printf("kurang sekali");break;
14     }
15 }
16
```

### 2.6.1 Flowchart

### 2.6.2 Data Hasil Percobaan

Tabel 2.2 Hasil percobaan program switch

Nilai	Hasil
A	
B	
C	
D	
G	

### 2.6.3 Analisa Data

### 2.6.4 Kesimpulan

## 2.7 Listing program menjumlahkan 1+2+3+... menggunakan perintah do-while

```
Start here x 2.7.c x
1  #include <stdio.h>
2  void main()
3  {
4  int i, batas, jumlah;
5  printf("\n menjumlah 1+2+3+...");
6  printf("\n sampai berapa ?");
7  scanf("%d",&batas);
8  jumlah=0;
9  i=1;
10 do
11 {
12     jumlah=jumlah+i;
13     i=i+1;
14 }
15 while (i<(batas+1));
16 printf ("\n 1+2+3+...+%d=%d",batas,jumlah);
17 }
18
```

### 2.7.1 Flowchart

### 2.7.2 Data Hasil Percobaan

Tabel 2.3 Hasil percobaan program do-while

I	Jumlah
0	
2	
4	
6	

### 2.7.3 Analisa Data

### 2.7.4 Kesimpulan



## 2.8 Listing program menjumlahkan 1+2+3+... menggunakan perintah while

```
Starthere x 2.8.c x
1  #include <stdio.h>
2  void main()
3  {
4  int i, batas, jumlah;
5  printf("\n menjumlah : 1+2+3+...");
6  printf("\n sampai berapa ?");
7  scanf("%d", &batas);
8  jumlah=0;
9  i=1;
10 while (i<(batas+1)) {
11 jumlah=jumlah+i;
12 i=i+1;
13 }
14 printf ("\n 1+2+3+...+%d=%d", batas, jumlah);
15 }
16
```

### 2.8.1 Flowchart

### 2.8.2 Data Hasil Percobaan

Tabel 2.4 Hasil percobaan program while

<b>I</b>	<b>Jumlah</b>
<b>0</b>	
<b>2</b>	
<b>4</b>	
<b>6</b>	

### 2.8.3 Analisa Data

### 2.8.4 Kesimpulan

## 2.9 Listing program for

```
Start here  x  2.9.c  x
1  #include <stdio.h>
2  void main()
3  {
4      int i, batas, jumlah;
5      printf("\n menjumlah : 1+2+3+...");
6      printf("\n sampai berapa ?");
7      scanf("%d", &batas);
8      jumlah=0;
9      for (i=1; i<(batas+1); i++)
10     {
11         jumlah=jumlah+i;
12     }
13     printf ("\n 1+2+3+...+%d=%d", batas, jumlah);
14 }
15
```

### 2.9.1 Flowchart

### 2.9.2 Data Hasil Percobaan

Tabel 2.5 Hasil percobaan program for

<b>I</b>	<b>Jumlah</b>
<b>0</b>	
<b>2</b>	
<b>4</b>	
<b>6</b>	

### 2.9.3 Analisa Data

### 2.9.4 Kesimpulan

## 2.10 Listing program continue dan break

```
Start here x 2.10.1.c x
1  #include <stdio.h>
2  void main()
3  {
4  int i;
5  for (i=0;i<10;i++)
6  {
7  printf(" %d" , i);
8  if(i%2==0)break;
9  }
10 }
11
```

### 2.10.1 Flowchart Continue Flowchart Break

### 2.10.2 Data Hasil Percobaan

Tabel 2.6 Hasil percobaan program continue dan break

Continue	
Break	

### 2.10.3 Analisa Data

### 2.10.4 Kesimpulan

## PERCOBAAN III

### *FUNGSI*

#### 3.1. Tujuan

Mempelajari parameter fungsi, mengirimkan parameter ke fungsi dengan nilai dan alamat.

#### 3.2. Alat yang digunakan

- 1 Set Komputer
- Sistem Operasi Windows 10
- Kompiler Code Blocks

#### 3.3. Dasar Teori

Parameter fungsi adalah bagaian dari fungsi yang digunakan untuk menghubungkan ke fungsi lain. Parameter fungsi terdiri *Parameter Formal* yang berada di fungsi yang dipanggil dan *Parameter Aktual* berada di fungsi pemanggil.

Mengirimkan parameter ke fungsi dengan nilai berarti mengirimkan parameter aktual ke suatu fungsi dengan suatu fungsi dengan suatu nilai dari variabel bukan dengan variabel itu sendiri sehingga perubahan pada parameter aktual. Fungsi penerima tidak dapat mengubah nilai variabel dari fungsi pemanggil.

Mengirimkan parameter ke fungsi dengan alamat berarti mengirimkan parameter aktual ke suatu fungsi dengan alamat yaitu variabelnya itu sendiri, hal ini dapat menyebabkan ketergantungan antara parameter formal dengan parameter aktual yaitu jika nilai dari parameter formal berubah maka nilai dari parameter aktual juga berubah. Fungsi penerima dapat mengubah nilai variabel fungsi pemanggil.

#### 3.4 *Prosedur Percobaan*

- a. Bukalah Window baru dari program Code Blocks(New Project).

- Ketiklah listing program yang di berikan berikut ini.
- Compile listing program yang telah dibuat (F9). Apabila masih ada yang error, tanyakan lah pada asisten pendamping.
- Simpanlah listing program yang telah berhasil di DRIVE D,dalam Folder yang telah anda buat.

### 3.5 Listing program pengiriman parameter dengan nilai

```

Start here x 2.10.1.c x 3.5.c x
1      #include<stdio.h>
2      void proses(int i);
3      void main ()
4      {
5          int i;
6          printf("\n i ?");
7          scanf("%d",&i);
8          proses(i);
9          printf("\n i=%d", i);
10     }
11     void proses(j)
12     int j;
13     {
14         printf("j=%d", j);
15         j*=2;
16         printf("\n j=%d", j);
17         return;
18     }
19

```

#### 3.5.1 Data Hasil Percobaan

Tabel 3.1 Program pengiriman parameter dengan nilai

Masukan Nilai i	Nilai j	Nilai j	Nilai i
1			
2			
3			

#### 3.5.2 Analisa Data

#### 3.5.3 Kesimpulan

### 3.6 Program pengiriman parameter dengan alamat

```
Start here X 3.6.c X
1  #include<stdio.h>
2  void proses(int *i);
3  void main ()
4  {
5      int i;
6      printf("\n i ?");
7      scanf("%d",&i);
8      proses(&i);
9      printf("\n i=%d", i);
10     return;
11 }
12 void proses(j)
13 int *j;
14 {
15     printf("j=%d",*j);
16     *j*=2;
17     printf("\n j=%d",*j);
18     return;
19 }
20
```

#### 3.6.1 Data Hasil Percobaan

Tabel 3.2 Program pengiriman parameter dengan alamat

Masukan Nilai i	Nilai j	Nilai j	Nilai i
1			
2			
3			

#### 3.6.2 Analisa Data

#### 3.6.3 Kesimpulan

## PERCOBAAN IV

### *ARRAY*

#### 4.1. Tujuan

Mempelajari cara menyimpan dan mengakses data bertipe sama.

#### 4.2. Alat yang digunakan

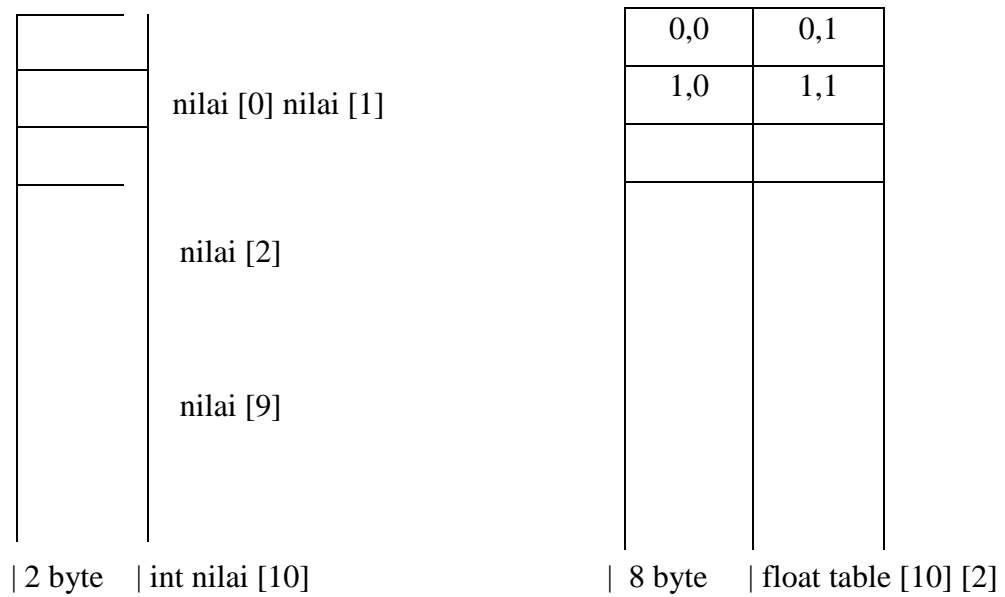
- 1 Set Komputer
- Kompiler Code Blocks
- Sistem Operasi Windows 10

#### 4.3. Dasar Teori

*Variabel Skalar* hanya dapat digunakan untuk menyimpan sebuah nilai saja sehingga untuk menyimpan beberapa nilai sekaligus dalam satu variabel diperlukan suatu variabel khusus yang disebut dengan *Variabel Array*. Variabel Array adalah sebuah variabel yang mampu menyimpan data bertipe sama. Variabel Array mempunyai bagian tipe data, pengenal, dan ukuran. Ukuran menunjukkan banyaknya data yang dapat disimpan. Data tersimpan dapat diakses dengan menunjukkan indeks yang berharga diantara 0 sampai dengan satu lokasi sebelum ukuran array.

Kompiler C mengalokasikan memori yang cukup untuk menampung data sesuai dengan tipenya dan mendukung array berdimensi tunggal, ganda maupun banyak. Sebuah berdimensi tunggal dengan elemen bertipe int memerlukan sebanyak  $10 \times 2$  atau 20 byte, sedangkan  $10 \times 2 \times 4$  atau 80 byte seperti terlihat pada gambar berikut:

:



#### 4.4 Prosedur Percobaan

- Bukalah Window baru dari program Code Blocks(New Project).
- Ketiklah listing program yang di berikan berikut ini.
- Compile listing program yang telah dibuat (F9). Apabila masih ada yang error, tanyakan lah pada asisten pendamping.
- Simpanlah listing program yang telah berhasil di DRIVE D,dalam Folder yang telah anda buat.



#### 4.5 Listing program menghitung nilai rata – rata dari bilangan positif

```

1  #include <stdio.h>
2  main ()
3  {
4  int cacah,i,n ;
5  float jumlah, rata;
6  float x[100];
7
8  jumlah = 0;
9  cacah = 0;
10 printf ("MENGHITUNG NILAI RATA-RATA DARI BILANGAN YANG BERNILAI POSITIF\n");
11 printf ("Banyak Data :");
12 scanf ("%d",&n);
13 for (i=0;i<n;i++)
14 {
15 printf("Data ke-%d:", i+1);
16 scanf ("%f",&x[i]);
17 if(x[i]>0)
18 {
19 jumlah = jumlah + x[i];
20 cacah = cacah +1;
21 }
22 }
23 rata = jumlah/cacah;
24 printf ( "hasil penjumlahan : %.2f\n", jumlah);
25 printf ( "Rata-rata : %.2f\n", rata);
26 }
27

```

##### 4.5.1 Flowchart

##### 4.5.2 Data Hasil Percobaan

Tabel 4.1 Hasil percobaan program menghitung rata-rata

Banyak Data		Nilai Data	Nilai Rata Rata
<b>7</b>	Data Ke -1		
	Data Ke -2		
	Data Ke -3		
	Data Ke -4		
	Data Ke -5		

	Data Ke -6		
	Data Ke -7		

4.5.3 Analisa Data

4.5.4 Kesimpulan

4.6 Listing program bubble sort

```
Start here x 4.6.c x
1  #include <stdio.h>
2  main ()
3  {
4      int i, n, j, temp, k;
5      int A[100];
6      printf("Bubble sort menaik \n");
7      printf("Banyak data : ");
8      scanf ("%d", &n);
9      for(i=0;i<n;i++)
10     {
11         printf("Data Ke-%d:", i+1);
12         scanf ("%d", &A[i]);
13     }
14     printf("Sebelum terurut : \n");
15     for(i=0;i<n;i++)
16     {
17         printf("%d", A[i]);
18         printf("\n");
19     }
20     printf("Sebelum terurut : \n");
21     for(i=0;i<n-1;i++)
22     {
23         for(j=0;j<(n-1);j++)
24         {
25             if(A[j] > A[j+1])
26             {
27                 temp=A[j];
28                 A[j]=A[j+1];
29                 A[j+1]=temp;
30             }
31     }
```

```

32     printf("Proses ke-%d : ",i+1);
33     for(k=0;k<n;k++)
34         printf( " %d " ,A[k]);
35         printf ( " \n");
36     }
37     printf("Setelah terurut : \n");
38     for(i=0;i<n;i++)
39         printf( "%d \n" ,A[i]);
40     }

```

4.6.1 Flowchart

4.6.2 Data Hasil Percobaan

Tabel 4.2 Hasil Percobaan bubble sort

Banyak Data	Data Ke	Nilai Data	Elemen Ke- Sebelum Terurut	1	2	3	4	5
5	1		Proses ke-1					
	2		Proses ke-2					
	3		Proses ke-3					
	4		Proses ke-4					
	5		Proses ke-5					

4.6.3 Analisa Data

4.6.4 Kesimpulan

## PERCOBAAN V

### *STRUKTUR*

#### 5.1. Tujuan

Mempelajari cara menyimpan dan mengakses data dengan tipe berbeda sebagai satu kesatuan.

#### 5.2. Alat yang digunakan :

- 1 Set Komputer
- Sistem Operasi Windows 10
- Kompiler Code Blocks

#### 5.3. Dasar Teori

Struktur adalah kumpulan variabel dimana tipe datanya berbeda. Kumpulan variabel ini dinyatakan dengan nama tunggal sehingga data dapat ditangani sebagai kesatuan. Dengan demikian diharapkan penanganan data dapat lebih mudah dilakukan.

Struktur mempunyai beberapa anggota. Sebuah anggota adalah nilai individu.

Kompiler C menampung setiap anggota seperti terlihat pada gambar berikut;

```
struct Segitiga {
    int      alas
    }      2 byte int alas;
    tinggi }      2 byte int tinggi;
    luas   }      4 byte
    float luas;
                                sg      8 byte
};
struct Segitiga sg;
```

Segitiga adalah tipe data baru yang merupakan komposisi tipe data integer dan float. Segitiga memerlukan tempat di memori sebesar  $2+2+4 = 8$  byte.

Sg mempunyai anggota alas, tinggi dan luas yang merupakan variable struktur dengan tipe data segitiga. Anggota Sg dapat diakses dengan dua cara:

1. Dengan oprator \*

Contoh =(\*Sg). Alas = 1 ;

2. Dengan oprator

Contoh = Sg alas = 2 ;

Variable struktur dapat juga berupa Array, yaitu kumpulan nilai yang bertipe Struktur seperti terlihat pada gambar brikut :

```

struct          Segitiga          {
    alas      }          2 byte int alas ;
    sgt[0]    tinggi }          2 byte
int tinggi ;                                luas }          4 byte
float luas ;                                alas }          2 byte
} ; sgt [1]    tinggi } 2 byte struct Segitiga sgt [2] luas }
4 byte
sgt          18 byte

```

Sgt merupakan array variable struktur dengan 2 elemen, masing-masing elemen dengan tipe data segitiga sehingga sgt menempati  $2*(2+2+4)= 16$  byte.

#### 5.4 Prosedur Percobaan

- Bukalah Window baru dari program Code Blocks(New Project).
- Ketiklah listing program yang di berikan berikut ini.
- Compile listing program yang telah dibuat (F9). Apabila masih ada yang error, tanyakan lah pada asisten pendamping.

- Simpanlah listing program yang telah berhasil di DRIVE D,dalam Folder yang telah anda buat.

### 5.5 Listing program mencari luas dengan operator ->

```

Start here x 4.6.c x 5.5 new.c x
1  #include<stdio.h>
2  struct segitiga
3  {
4  int alas;
5  int tinggi;
6  float luas;
7  };
8  segitiga;
9  void nilai_segitiga (sgt)
10 struct segitiga *sgt;
11 {
12 sgt->alas=10;
13 sgt->tinggi=3;
14 sgt->luas=0.5*sgt->alas*sgt->tinggi;
15 }
16 void main()
17 {
18 struct segitiga sg;
19 nilai_segitiga (&sg);
20 printf ("\nAlas =%d\n", sg.alas);
21 printf ("Tinggi=%d\n", sg.tinggi);
22 printf ("Luas=%f\n", sg.luas);
23 }

```

#### 5.5.1 Flowchart

#### 5.5.2 Data Hasil Percobaan

Tabel 5.1 Hasil percobaan program mencari luas dengan operator ->

Alas	Tinggi	Luas
1	2	
1.5	3.5	
1.3	4	
0.2	0.7	

#### 5.5.3 Analisa Data

#### 5.5.4 Kesimpulan

## 5.6 Listing program mencari luas dengan operator \*

```
Start here x 4.6.c x 5.6new.c x
1  #include <stdio.h>
2  struct segitiga
3  {
4  int alas;
5  int tinggi;
6  float luas;
7  };
8  segitiga;
9  void nilai_segitiga(segitiga *sgt)
10 struct segitiga *sgt;
11 {
12 (*sgt).alas=10;
13 (*sgt).tinggi=3;
14 (*sgt).luas=0.5*(*sgt).alas*(*sgt).tinggi;
15 }
16 void main()
17 {
18 struct segitiga sg;
19 nilai_segitiga(&sg);
20 printf ("\n Alas=%d\n", sg.alas);
21 printf ("Tinggi=%d\n", sg.tinggi);
22 printf ("Luas=%f\n", sg.luas);
23 }
24
```

### 5.6.1 Flowcharts

### 5.6.2 Data Hasil Percobaan

Tabel 5.2 Hasil percobaan program mencari luas dengan operator \*

Alas	Tinggi	Luas
1	2	
1.5	3.5	
1.3	4	
0.2	0.7	

### 5.6.3 Analisa Data

### 5.6.4 Kesimpulan