

MODUL TEORI

KOMPUTER DAN PEMROGRAMAN



Disusun Oleh :

Fitri Ramadhaniati, S.ST.,M.Keb

Kurnia Dewiani, S.ST.,M.Keb

Rini Mustikasari Kurnia Pratama, S.SiT., M.Keb.

**PROGRAM STUDI PENDIDIKAN PROFESI BIDAN
FAKULTAS MATEMATIKA DAN ILMU
PENGETAHUAN ALAM
UNIVERSITAS BENGKULU**

Visi dan Misi

**PROGRAM PENDIDIKAN PROFESI BIDAN
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

Visi

Menghasilkan Lulusan Profesi Bidan yang Berbudaya, Unggul dan Profesional Dalam Menjalankan Praktik Kebidanan Holistik Berdasarkan *Evidence Based Midwifery* dengan Penerapan *Interprofessional Education*

Misi

1. *Menyelenggarakan dan mengembangkan pendidikan akademik dan profesi bidan yang berbudaya, unggul dan profesional pada pelayanan kebidanan holistik berdasarkan evidence based midwifery dengan menerapkan Interprofessional Education (IPE)*
2. Meningkatkan kualitas penelitian dan publikasi ilmiah yang berkontribusi pada IPTEK dan *evidence based midwifery* melalui pendekatan lintas profesi (*Interprofessional Collaboration/IPC*)
3. Menyebarkan ilmu pengetahuan dan teknologi melalui kegiatan pengabdian masyarakat bidang asuhan kebidanan yang berorientasi pada pengembangan kesehatan masyarakat khususnya kesehatan ibu dan anak.
4. Menerapkan sistem tata kelola yang dapat dipertanggungjawabkan;
5. Meningkatkan kerjasama bidang pendidikan dan penelitian dengan berbagai institusi tingkat nasional dan internasional

LEMBAR PENGESAHAN

Modul Komputer dan Pemrograman ini sah untuk digunakan di Program Studi
Pendidikan Profesi Bidan
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Bengkulu

Disahkan oleh :

Ketua Program Studi



Yetti Purnama, S.ST.,M.Keb
NIP: 197705302007012007

KATA PENGANTAR

Puji syukur kami ucapkan kepada Allah SWT berkat rahmat dan hidayahNya kami dapat menyelesaikan modul ini. Modul yang berjudul “Komputer dan Pemrograman” disusun dari berbagai sumber literatur yang telah disesuaikan dengan silabus mata kuliah, sehingga relevan digunakan untuk pembelajaran pada mata kuliah Komputer dan Pemrograman (*Coding*).

Modul ini dibuat dengan mengacu pada kurikulum pendidikan Sarjana Kebidanan sehingga dapat memudahkan mahasiswa dalam mempelajari mata kuliah Asuhan Kebidanan Pada Kehamilan. Akhir kata, semoga modul ini bermanfaat bagi para pembaca. Meskipun penulis telah berusaha menyusun buku ini secara sistematis dan mendalam, tetapi buku ini masih jauh dari kata sempurna. Untuk itu, penulis mengharapkan kritik dan saran dari pembaca untuk penyempurnaan tulisan di edisi berikutnya. Wassalamualaikum Wr. Wb

Bengkulu, Januari 2025

Penulis

DAFTAR ISI

DAFTAR ISI	ii
DAFTAR GAMBAR.....	vi
DAFTAR TABEL.....	vii
STANDART OPERASIONAL PROSEDUR	viii
A. Sistem Komputer dan Perkembangan Kommputer.....	1
B. Dampak dan Etika Komputer (Computer Etichs).....	2
C. Logika dan Algoritma Pemrograman.....	3
D. Dasar-Dasar Pemrograman.....	4
E. Perintah Percabangan, Perulangan, dan Larik (Array).....	5
F. Dasar-Dasar Basic Data.....	6
G. Basis Data Relasional dan Perancangan Basis Data	7
H. Bahasa Pemrograman SQL.....	8
I. Dasar-Dasar Pemrograman Web.....	9
J. Content Management System (CMS).....	10

PERCOBAAN I TIPE DATA DAN OPERATOR

1.1 Tujuan.....	11
1.2 Alat yang digunakan	11
1.3 Dasar Teori.....	11
1.4 Prosedur Percobaan	14
1.5 Listing Program Angka Penting.....	14
1.5.1 Flowchart.....	14
1.5.2 Data Hasil Percobaan.....	14
1.5.3 Analisa Data	15
1.5.4 Kesimpulan.....	15
1.6 Listing Program Perkalian	15
1.6.1 Flowchart.....	15
1.6.2 Data Hasil Percobaan.....	15
1.6.3 Analisa Data	15

1.6.4 Kesimpulan.....	15
1.7 Listing Program Aritmatika	16
1.7.1 Flowchart.....	16
1.7.2 Data Hasil Percobaan.....	16
1.7.3 Analisa Data	16
1.7.4 Kesimpulan.....	16

PERCOBAAN II PENGONTROLAN ALIRAN PROGRAM

2.1 Tujuan.....	17
2.2 Alat yang diperlukan.....	17
2.3 Dasar Teori.....	17
2.4 Prosedur Percobaan	19
2.5 Listing program menampilkan nilai menggunakan perintah if – else	20
2.5.1 Flowchart.....	20
2.5.2 Data Hasil Percobaan.....	20
2.5.3 Analisa Data	20
2.5.4 Kesimpulan.....	20
2.6 Listing program menampilkan nilai menggunakan perintah switch	21
2.6.1 Flowchart.....	21
2.6.2 Data Hasil Percobaan.....	21
2.6.3 Analisa Data	21
2.6.4 Kesimpulan.....	21
2.7 Listing program menjumlahkan 1+2+3... menggunakan do-while .	22
2.7.1 Flowchart.....	22
2.7.2 Data Hasil Percobaan.....	22
2.7.3 Analisa Data	22
2.7.4 Kesimpulan.....	22
2.8 Listing program menjumlahkan 1+2+3... menggunakan while	23
2.8.1 Flowchart.....	23
2.8.2 Data Hasil Percobaan.....	23
2.8.3 Analisa Data	23
2.8.4 Kesimpulan.....	23

2.9 Listing program for.....	24
2.9.1 Flowchart.....	24
2.9.2 Data Hasil Percobaan.....	24
2.9.3 Analisa Data	24
2.9.4 Kesimpulan.....	24
2.10 Listing program continue dan break.....	25
2.10.1 Flowchart continue.....	25
2.10.1 Flowchart break	25
2.10.2 Data Hasil Percobaan.....	25
2.10.3 Analisa Data	25
2.10.4 Kesimpulan.....	25

PERCOBAAN III FUNGSI

3.1 Tujuan.....	26
3.2 Alat yang digunakan	26
3.3 Dasar Teori.....	26
3.4 Prosedur Percobaan	26
3.5 Listing program pengiriman parameter dengan nilai	27
3.5.1 Data Hasil Percobaan.....	27
3.5.2 Analisa Data	27
3.5.3 Kesimpulan.....	27
3.6 Listing program pengiriman parameter dengan alamat.....	28
3.6.1 Data Hasil Percobaan.....	28
3.6.2 Analisa Data	28
3.6.3 Kesimpulan.....	28

PERCOBAAN IV ARRAY

4.1 Tujuan.....	29
4.2 Alat yang digunakan	29
4.3 Dasar Teori.....	29
4.4 Prosedur Percobaan	30
4.5 Listing program menghitung nilai rata-rata dari bilangan positif	31
4.5.1 Flowchart.....	31

4.5.2 Data Hasil Percobaan	31
4.5.3 Analisa Data	32
4.5.4 Kesimpulan.....	32
4.6 Listing program bubble sort.....	32
4.6.1 Flowchart.....	33
4.6.2 Data Hasil Percobaan	33
4.6.3 Analisa Data	33
4.6.4 Kesimpulan.....	33

PERCOBAAN V STRUKTUR

5.1 Tujuan.....	34
5.2 Alat yang digunakan	34
5.3 Dasar Teori.....	34
5.4 Prosedur Percobaan	35
5.5 Listing program mencari luas dengan operator →.....	36
5.5.1 Flowchart.....	36
5.5.2 Data Hasil Percobaan	36
5.5.3 Analisa Data	36
5.5.4 Kesimpulan.....	36
5.6 Listing program mencari luas dengan operator *.....	37
5.6.1 Flowchart.....	37
5.6.2 Data Hasil Percobaan	37
5.6.3 Analisa Data	37
5.6.4 Kesimpulan.....	37

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

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 ijin.
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 peralatan Laboratorium 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, misal 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 la

A. Dampak dan Etika Komputer (Computer Etichs)

1. Dampak Positif

- **Efisiensi dan Produktivitas:** Komputer mempercepat pekerjaan di berbagai sektor, seperti bisnis, pendidikan, dan kesehatan.
- **Akses Informasi:** Memudahkan pencarian informasi dan pengetahuan dari seluruh dunia.
- **Komunikasi Global:** Melalui internet, komunikasi menjadi lebih cepat dan murah.
- **Inovasi Teknologi:** Mendorong kemajuan di bidang AI, robotik, dan otomasi.

2. Dampak Negatif

- **Kecanduan Teknologi:** Penggunaan komputer berlebihan dapat menyebabkan kecanduan, terutama game dan media sosial.
- **Pengangguran:** Otomatisasi pekerjaan bisa mengurangi kebutuhan tenaga kerja manusia.
- **Keamanan Data:** Risiko peretasan dan pencurian data semakin tinggi.
- **Privasi Terancam:** Informasi pribadi bisa tersebar tanpa izin pengguna.

Etika komputer adalah panduan moral dalam penggunaan teknologi informasi dan komunikasi.

Prinsip-Prinsip Etika Komputer

1. Menghormati Privasi Orang Lain

Jangan mengakses atau menyebarkan data pribadi orang lain tanpa izin.

2. Menghormati Hak Kekayaan Intelektual

Hindari plagiat dan pembajakan software, musik, film, atau konten digital lainnya.

3. Jujur dalam Penggunaan Komputer

Tidak membuat atau menyebarkan hoaks, virus, atau konten yang merugikan.

4. Tidak Merugikan Orang Lain

Termasuk dalam tindakan seperti hacking, phishing, atau cyberbullying.

5. Menggunakan Komputer Secara Bertanggung Jawab

Memastikan bahwa penggunaan komputer membawa manfaat, bukan kerusakan.

Contoh Pelanggaran Etika Komputer

- Menyebarkan hoaks atau ujaran kebencian di media sosial.
- Menggunakan software bajakan.

- Melakukan hacking atau pencurian identitas digital.
- Membocorkan data rahasia perusahaan atau lembaga.

B. Logika dan Algoritma Pemrograman

Logika pemrograman adalah cara berpikir sistematis dan runtut untuk memecahkan masalah menggunakan langkah-langkah yang bisa dipahami komputer.

Tujuan Logika Pemrograman:

- Menyusun solusi dari sebuah permasalahan.
- Menyusun algoritma yang efisien dan tepat.
- Menghindari kesalahan (error) dalam program.

Contoh Sederhana:

Misalnya kamu ingin membuat program menentukan apakah suatu angka ganjil atau genap.

Logika sederhananya:

- Jika angka habis dibagi 2 $\rightarrow$ genap.
- Jika tidak $\rightarrow$ ganjil.

Algoritma adalah langkah-langkah terstruktur dan berurutan yang digunakan untuk menyelesaikan masalah atau tugas tertentu.

Ciri-ciri Algoritma:

- **Pasti:** Setiap langkah harus jelas.
- **Terbatas:** Harus selesai dalam jumlah langkah terbatas.
- **Masukan (Input):** Menerima satu atau lebih input.
- **Keluaran (Output):** Menghasilkan output atau solusi.
- **Efisien:** Harus menggunakan sumber daya sehemat mungkin.

Contoh Algoritma:

Masalah: Menentukan bilangan terbesar dari tiga bilangan.

Langkah-langkah Algoritma:

1. Masukkan tiga bilangan: A, B, dan C.
2. Bandingkan A dengan B, simpan yang terbesar.
3. Bandingkan hasil dari langkah 2 dengan C.
4. Tampilkan bilangan terbesar.

Hubungan Logika dan Algoritma dalam Pemrograman

- Logika adalah dasar berpikirnya.
- Algoritma adalah implementasi dari logika tersebut.

- Bahasa pemrograman (seperti Python, Java, C++) adalah alat untuk menuliskan algoritma ke dalam kode yang dimengerti komputer.

Contoh Kode (dalam Python)

```
python
Copy code
# Menentukan bilangan terbesar dari tiga bilangan
a = int(input("Masukkan bilangan pertama: "))
b = int(input("Masukkan bilangan kedua: "))
c = int(input("Masukkan bilangan ketiga: "))

if a >= b and a >= c:
    print("Bilangan terbesar adalah:", a)
elif b >= a and b >= c:
    print("Bilangan terbesar adalah:", b)
else:
    print("Bilangan terbesar adalah:", c)
```

C. Dasar-Dasar Pemrograman

1. Pengertian Pemrograman

Pemrograman adalah proses membuat instruksi yang dimengerti oleh komputer agar bisa melakukan tugas tertentu. Instruksi ini ditulis dalam **bahasa pemrograman**.

2. Komponen Dasar Pemrograman

a. Variabel

Tempat untuk menyimpan data.

```
python
```

```
Copy code
```

```
nama = "Rina"
```

```
umur = 17
```

b. Tipe Data

Jenis data yang disimpan di dalam variabel.

- `int`: bilangan bulat → 10
- `float`: bilangan desimal → 3.14
- `str`: teks → "Halo"
- `bool`: nilai benar/salah → `True` / `False`

c. Operator

Digunakan untuk melakukan operasi pada variabel dan nilai.

- Aritmatika: `+`, `-`, `*`, `/`
- Perbandingan: `==`, `!=`, `<`, `>`
- Logika: `and`, `or`, `not`

d. Percabangan (Kondisi)

Memilih aksi berdasarkan kondisi tertentu.

```
python
Copy code
if nilai >= 75:
    print("Lulus")
else:
    print("Tidak Lulus")
```

e. Perulangan (Looping)

Menjalankan perintah berulang kali.

```
python
Copy code
for i in range(5):
    print("Halo ke-", i)
```

f. Fungsi

Kumpulan kode yang bisa digunakan berulang kali.

```
python
Copy code
def sapa(nama):
    print("Halo", nama)

sapa("Ani")
```

3. Struktur Program Sederhana

Struktur dasar dalam program biasanya terdiri dari:

1. **Input** – menerima data dari pengguna.
2. **Proses** – mengolah data sesuai logika/algoritma.
3. **Output** – menampilkan hasil.

Contoh program:

```
python
Copy code
# Program menghitung luas segitiga
alas = float(input("Masukkan alas: "))
tinggi = float(input("Masukkan tinggi: "))

luas = 0.5 * alas * tinggi
print("Luas segitiga:", luas)
```

4. Bahasa Pemrograman Populer untuk Pemula

- **Python** → mudah dibaca dan ditulis.
- **C/C++** → bagus untuk belajar logika tingkat rendah.
- **Java** → banyak dipakai di perusahaan.
- **JavaScript** → untuk pengembangan web.

5. Tips Belajar Pemrograman

- Mulai dari yang sederhana.
- Latihan setiap hari.
- Pahami logikanya, jangan hanya menghafal.

D. Perintah Percabangan, Perulangan, dan Larik (Array)

1. Perintah Percabangan (Kondisional)

Percabangan digunakan untuk membuat keputusan berdasarkan kondisi tertentu.

◆ Struktur Umum (Python):

```
python
Copy code
if kondisi:
    # kode jika kondisi benar
elif kondisi_lain:
    # kode jika kondisi lain benar
else:
    # kode jika semua kondisi salah
```

◆ Contoh:

```
python
Copy code
nilai = int(input("Masukkan nilai: "))

if nilai >= 90:
    print("Nilai A")
elif nilai >= 80:
    print("Nilai B")
else:
    print("Nilai C atau di bawahnya")
```

2. Perulangan (Looping)

Perulangan digunakan untuk menjalankan kode berulang kali.

a. for loop

Digunakan jika jumlah pengulangan sudah diketahui.

```
python
Copy code
for i in range(5): # dari 0 sampai 4
    print("Perulangan ke-", i)
```

b. while loop

Digunakan jika pengulangan bergantung pada kondisi tertentu.

```
python
Copy code
i = 1
while i <= 5:
    print("Angka:", i)
    i += 1
```

3. Larik (Array) / List

Larik (atau array) digunakan untuk menyimpan banyak data dalam satu variabel.

Contoh:

```
python
Copy code
buah = ["apel", "jeruk", "mangga"]

print(buah[0]) # output: apel
print(len(buah)) # panjang larik: 3
```

Perulangan dengan larik:

```
python
Copy code
buah = ["apel", "jeruk", "mangga"]

for b in buah:
    print("Saya suka", b)
```

Menambah & menghapus elemen:

```
python
Copy code
buah.append("pisang") # menambah
buah.remove("jeruk") # menghapus
```

Contoh Program Gabungan

```
python
Copy code
# Program menampilkan angka genap dari array
angka = [1, 4, 7, 10, 12, 15]

for a in angka:
    if a % 2 == 0:
        print(a, "adalah angka genap")
```

E. Dasar-Dasar Basic Data

Tipe data adalah klasifikasi data yang memberi tahu komputer **jenis nilai** apa yang sedang ditangani. Setiap bahasa pemrograman punya caranya sendiri, tapi konsep dasarnya mirip-mirip.

1. Integer (int) – Bilangan Bulat

- Contoh: 1, -5, 1000
- Operasi: penjumlahan, pengurangan, dll.

```
python
Copy code
x = 10
y = 3
print(x + y) # output: 13
```

2. Float / Double – Bilangan Desimal

- Contoh: 3.14, -2.5, 0.001

```
python
Copy code
pi = 3.14
r = 5
luas = pi * r * r
print(luas) # output: 78.5
```

3. String (str) – Teks

Kumpulan karakter diapit tanda kutip " " atau ' '

```
python
Copy code
nama = "Budi"
print("Halo, " + nama) # output: Halo, Budi
```

4. Boolean (bool) – Nilai Benar atau Salah

- Hanya punya 2 nilai: True atau False
- Umumnya digunakan dalam logika/perbandingan

```
python
Copy code
x = 10
y = 5
print(x > y) # output: True
```

5. List / Array – Kumpulan Data

- Menyimpan banyak data dalam satu variabel

```
python
Copy code
angka = [1, 2, 3, 4]
print(angka[0]) # output: 1
```

6. Dictionary (khusus Python) – Pasangan Key:Value

```
python
Copy code
mahasiswa = {
    "nama": "Dina",
    "umur": 20,
```

```
"jurusan": "Informatika"

print(mahasiswa["nama"]) # output: Dina
```

F. Basis Data Relasional dan Perancangan Basis Data

Basis Data Relasional adalah sistem manajemen basis data (DBMS) yang menyimpan data dalam bentuk **tabel (relasi)**.

Setiap tabel berisi:

- **Baris (row)** → disebut **record**
- **Kolom (column)** → disebut **field/atribut**

Perancangan basis data adalah proses menyusun struktur database agar data **terorganisir, efisien,** dan **mudah diakses**.

Langkah-Langkah:

1. Identifikasi Kebutuhan Data

Apa saja informasi yang perlu disimpan? (misalnya: data mahasiswa, mata kuliah, dosen, dll)

2. Membuat Entitas dan Atribut

Contoh:

- **Entitas Mahasiswa** → NIM, Nama, Jurusan
- **Entitas Mata Kuliah** → KodeMK, NamaMK, SKS

3. Menentukan Relasi Antar Entitas

Contoh: Mahasiswa mengikuti banyak Mata Kuliah (relasi many-to-many)

4. Membuat Diagram ERD (Entity Relationship Diagram)

Visualisasi hubungan antar entitas

5. Normalisasi

Proses untuk menghilangkan duplikasi dan memastikan struktur data efisien

G. Bahasa Pemrograman SQL

SQL (dibaca: es-kyu-el atau sequel) adalah bahasa standar yang digunakan untuk mengelola dan mengakses data dalam basis data relasional.

Contoh-Contoh Dasar SQL

1. Membuat Tabel (DDL)

```
sql
Copy code
CREATE TABLE Mahasiswa (
    NIM VARCHAR(10) PRIMARY KEY,
    Nama VARCHAR(50),
    Jurusan VARCHAR(30),
    Umur INT
);
```

2. Menambahkan Data (DML)

```
sql
Copy code
```

```
INSERT INTO Mahasiswa (NIM, Nama, Jurusan, Umur)
VALUES ('2023001', 'Dina', 'Informatika', 20);
```

3. Mengambil Data (DQL)

```
sql
Copy code
SELECT * FROM Mahasiswa;
-- atau
SELECT Nama, Jurusan FROM Mahasiswa WHERE Umur > 18;
```

4. Memperbarui Data

```
sql
Copy code
UPDATE Mahasiswa
SET Umur = 21
WHERE NIM = '2023001';
```

5. Menghapus Data

```
sql
Copy code
DELETE FROM Mahasiswa
WHERE NIM = '2023001';
```

H. Dasar-Dasar Pemrograman Web

Pemrograman Web adalah proses membuat **aplikasi atau halaman web** yang bisa diakses melalui internet menggunakan browser seperti Chrome, Firefox, dll.

Dibagi menjadi dua bagian:

Bagian	Penjelasan
Frontend	Bagian yang terlihat oleh pengguna (tampilan/UI)
Backend	Bagian yang mengatur logika, database, dan server

Frontend (Tampilan)

Digunakan untuk membangun antarmuka pengguna.

Bahasa	Fungsi
HTML	Struktur dasar halaman web
CSS	Mengatur tampilan dan desain
JavaScript	Menambahkan interaktivitas

Backend (Server-Side)

Digunakan untuk mengelola data, proses, dan interaksi pengguna.

Bahasa	Contoh Framework/Tool
PHP	Laravel, CodeIgniter

Bahasa	Contoh Framework/Tool
Python	Django, Flask
JavaScript	Node.js, Express
Java	Spring Boot

Cara Kerja Website Sederhana

1. **Pengguna** membuka halaman web lewat browser.
2. Browser **mengirim permintaan (request)** ke server.
3. **Server memproses** permintaan dan **mengirim respon** kembali (biasanya berupa HTML).
4. Browser **menampilkan** isi web ke pengguna.

Tools yang Digunakan

Tool	Fungsi
VS Code	Code editor
Browser (Chrome/Firefox)	Melihat hasil web
XAMPP / WAMP	Menjalankan server lokal (untuk PHP)
Git & GitHub	Mengelola versi proyek

Contoh Struktur Folder Web

```
bash
Copy code
/project-website
├── index.html      → file utama
├── style.css       → file desain
├── script.js       → file JavaScript
└── /images        → folder gambar
```

Contoh Kode HTML Dasar

```
html
Copy code
<!DOCTYPE html>
<html>
<head>
  <title>Website Pertamaku</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Halo, Dunia!</h1>
  <p>Ini adalah halaman web pertamaku.</p>
  <script src="script.js"></script>
</body>
</html>
```

I. Content Management System (CMS)

1. Apa itu CMS?

CMS (Content Management System) adalah perangkat lunak yang digunakan untuk **membuat, mengelola, dan memodifikasi** konten digital (seperti website, blog, toko online)

tanpa perlu menulis kode dari nol.

2. Fungsi Utama CMS

- Membuat dan mengedit halaman web
- Mengatur konten (teks, gambar, video)
- Mengelola pengguna dan hak akses
- Menambahkan fitur melalui plugin/ekstensi
- Mengubah tampilan web (tema/template)

3. Contoh CMS Populer

CMS	Keterangan
WordPress	Paling populer, cocok untuk blog, company profile, toko online (WooCommerce)
Joomla!	Lebih kompleks, cocok untuk portal berita, komunitas
Drupal	Sangat fleksibel, digunakan di website pemerintahan dan perusahaan besar
Magento	Khusus untuk e-commerce atau toko online yang kompleks
Shopify	CMS berbasis cloud khusus untuk membuat toko online dengan cepat

4. Komponen CMS

Komponen	Fungsi
Dashboard	Panel admin untuk mengelola semua isi dan pengaturan
Editor	Untuk menulis/mengedit konten seperti artikel atau halaman
Template/Themes	Mengatur tampilan website
Plugin/Modules	Menambahkan fitur seperti SEO, form, slider, dll
Media Manager	Kelola gambar, video, file

5. Keuntungan Menggunakan CMS

- ✓ Tidak perlu coding (user-friendly)
- ✓ Cepat membuat website
- ✓ Banyak pilihan template & plugin
- ✓ Dapat diakses dan dikelola online
- ✓ Komunitas besar (mudah cari bantuan)

6. Kekurangan CMS (Beberapa)

- ✗ Kurang fleksibel untuk sistem yang sangat kompleks
- ✗ Bisa lambat jika terlalu banyak plugin
- ✗ Perlu update rutin agar aman

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

Tipe Data	Jangkauan	Ketelitian (bit)
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

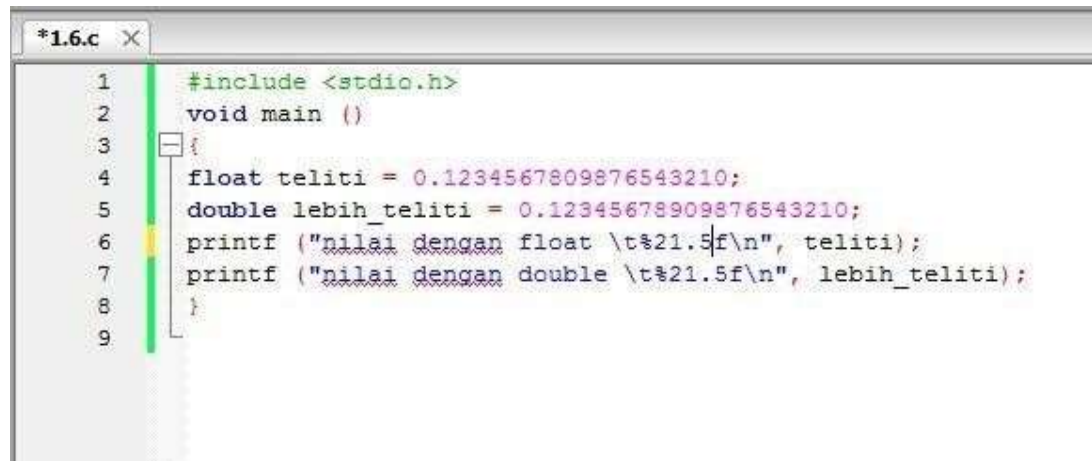
8	<=	Kurang dari atau sama dengan	Dari kiri ke kanan
8	>	Lebih dari	Dari kiri ke kanan
8	>=	Lebih dari atau sama dengan	Dari kiri ke kanan
9	==	Sama dengan	Dari kiri ke kanan
9	!=	Tidak sam dengan	Dari kiri ke kanan
10	&	AND Bitwise	Dari kiri ke kanan
11	^	XOR Bitwise	Dari kiri ke kanan
12		OR Bitwise	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  #include <stdio.h>
2  void main ()
3  {
4      float teliti = 0.1234567809876543210;
5      double lebih_teliti = 0.1234567809876543210;
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

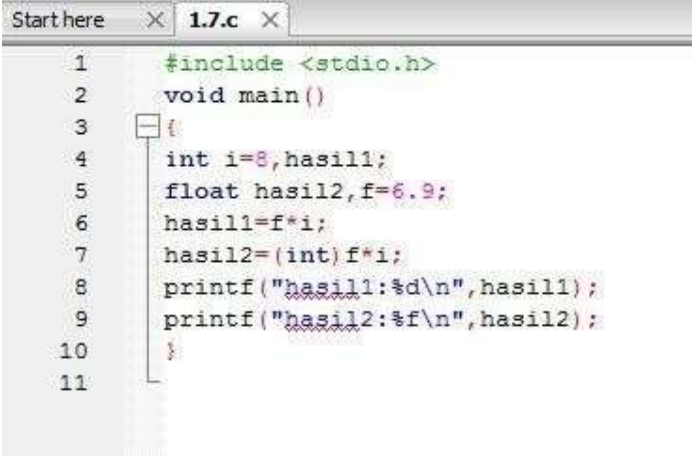
Tabel 1.4 Hasil percobaan program ketelitian tipe data

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



```
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

```

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, iterasi dan terminasi. Inisial 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. Iterasi 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 di badan 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;

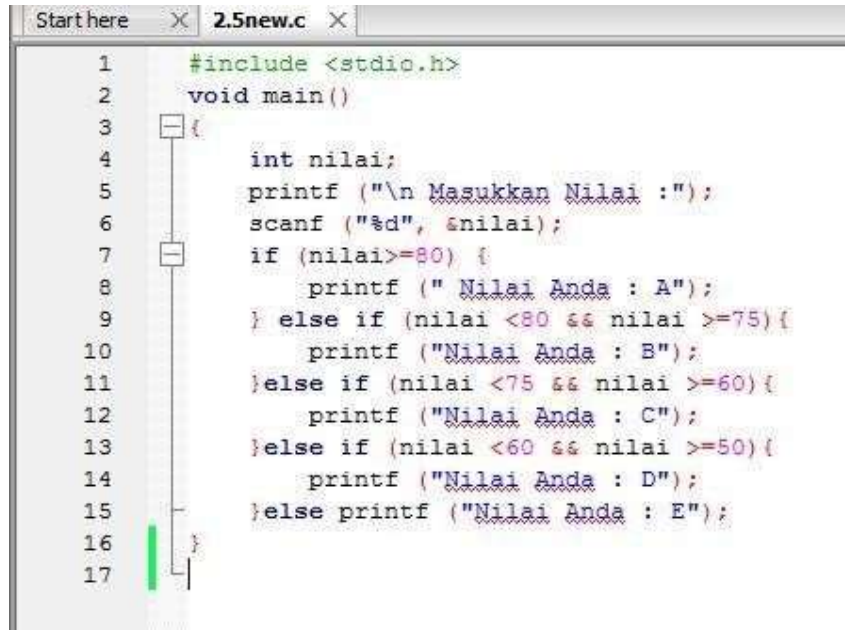
Dalam hal ini, jika inisialisasi tidak menyebabkan terminasi menghasilkan keadaan “benar” ketika pertama berjalan, maka pernyataan iterasi dan badan_program tidak akan dijalankan. Sebaliknya akan dicacah dari suatu nilai minimum sampai maksimumnya.

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

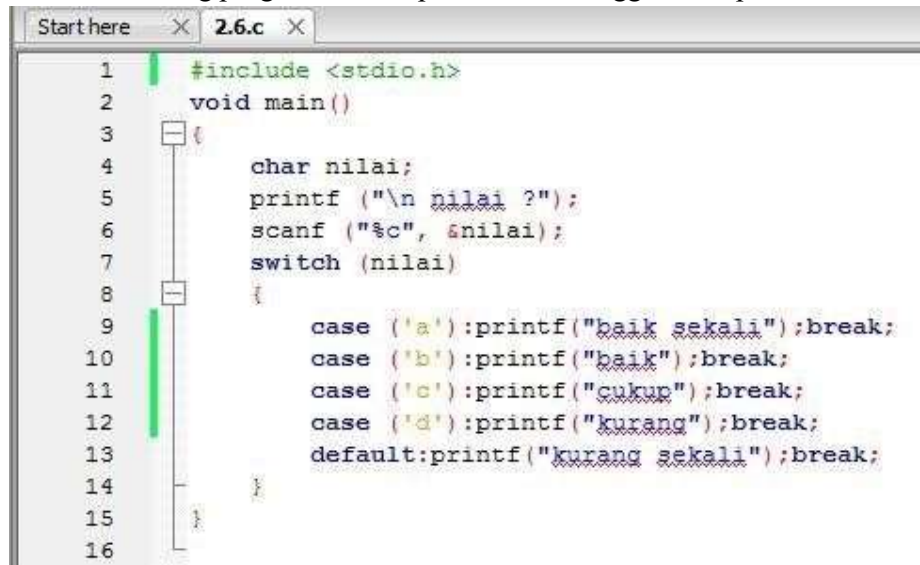
Tabel 2.1 Hasil percobaan program if-else

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



```
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

```

Starthere  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

```
Start here 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

I	Jumlah
0	
2	
4	
6	

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

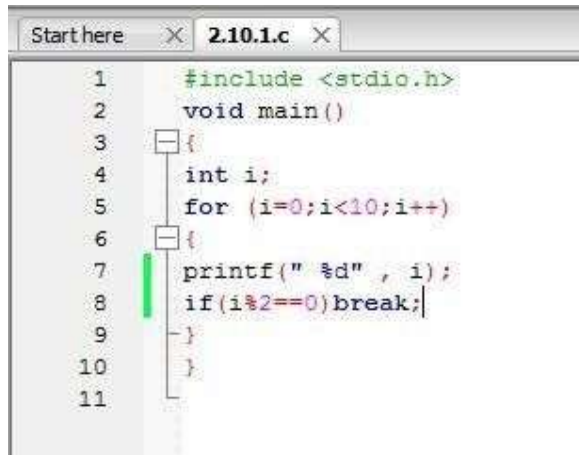
Tabel 2.5 Hasil percobaan program for

I	Jumlah
0	
2	
4	
6	

2.9.3 Analisa Data

2.9.4 Kesimpulan

2.10 Listing program continue dan break



```
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

```

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

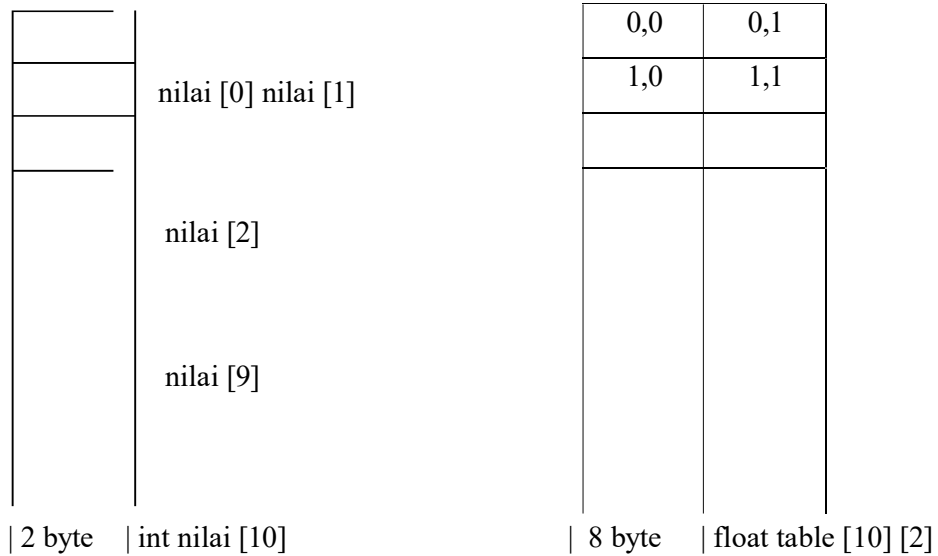
- 1 Set Komputer
- Kompiler Code Blocks
- Sistem Operasi Windows 10

4.3. Dasar Teori

Variabel Skaler 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 bernomor 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 10×2 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
7	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 operator *

Contoh =(*Sg). Alas = 1 ;

2. Dengan operator

Contoh = Sg.alas = 2 ;

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

```

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 $2 \times (2+2+4) = 16$ 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 {
11     (*sgt).alas=10;
12     (*sgt).tinggi=3;
13     (*sgt).luas=0.5*(*sgt).alas*(*sgt).tinggi;
14 }
15 void main()
16 {
17     struct segitiga sg;
18     nilai_segitiga(&sg);
19     printf ("\n Alas=%d\n", sg.alas);
20     printf ("Tinggi=%d\n", sg.tinggi);
21     printf ("Luas=%f\n", sg.luas);
22 }
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