

Modul Praktikum Pemrograman Dasar

Teknik Komputer

FILKOM UB

Table of Contents

Introduction	1.1
Bab 1 - Input output	1.2
Bab 2 - Seleksi kondisi	1.3
Bab 3 - Perulangan	1.4
Bab 4 - Array	1.5
Bab 5 - Method	1.6

Draft Modul Pemrograman Dasar Prodi Teknik Komputer Filkom UB 2016

Gitbook : <https://www.gitbook.com/book/99ridho/draft-modul-pd-tekkom-16/details>

Bab 1 - Program Input, Output dan Proses

Tujuan

- Praktikan mampu memahami struktur dasar bahasa pemrograman C
- Praktikan mampu mengenal penggunaan variabel dan tipe data
- Praktikan mampu mengenal dan mengimplementasikan statement input dan output dalam bahasa code C

Ringkasan Materi

Struktur Bahasa Pemrograman C

Berikut ini contoh code snippet yang menunjukkan struktur umum bahasa pemrograman C

```
#include <stdio.h> // preprocessor directive

// blok program utama (main method)
int main(int argc, char **argv) {
    // statement didalam blok main

    return 0; // return 0 jika tidak ada error
}
```

Variabel dan Tipe data

Variabel digunakan untuk menyimpan nilai data yang dapat diubah nilai datanya. Variabel memiliki tipe data dan identifier. Identifier adalah nama variabel yang digunakan sebagai pengenalan. Tipe data menandakan tipe dan jangkauan data yang dapat disimpan pada variabel tersebut.

Bahasa C menyediakan tipe data sebagai berikut

No	Tipe Data	Memori	Nilai Minimum	Nilai Maksimum
1	byte	8 bit	-128	127
2	short	16 bit	-32768	32767
3	int	32 bit	-2147483648	2147483647
4	long	64 bit	-9223372036854775808	9223372036854775807
5	float	32 bit	$\pm 1.40239846E-45$	$\pm 3.40382347E+8$
6	double	64 bit	$\pm 4.94065645841246544E-324$	$\pm 1.79769313486231570E+308$
7	char	16 bit	\u0000	\uFFFF

Cara pendeklarasian variabel seperti ini

```
tipe_data nama_variabel;

// contoh
int a;
char c;
double f;
```

Untuk menginisialisasi nilainya seperti ini

```
// contoh
char c = 'a';
int i = 89;
float f = 4.51f;
```

Format Specifier

Format specifier adalah penentu format dari suatu variabel yang biasa digunakan untuk formatting nilai variabel di standard output atau di standard input.

Secara umum, format specifier yang sering dipakai sebagai berikut, ini akan digunakan ketika kita memanfaatkan `scanf()` maupun `printf()`.

Format specifier	Tipe data
%d	Integer, basis 10
%x	Integer, basis 16
%o	Integer, basis 8
%s	String
%c	Char
%f	Float

Untuk escape character sebagai berikut

Escape character	Penjelasan
\a	audible bell
\b	backspace
\n	newline
\t	tab
\f	form feed
\r	carriage return
\v	vertical tab
\	backslash

Escape character atau escape sequence ini adalah kombinasi karakter yang diawali dengan backslash `\` dan diikuti dengan karakter huruf atau kombinasi digit. Biasa digunakan untuk menentukan carriage return atau newline di terminal atau printer, dan juga biasa digunakan untuk menyediakan representasi literal dari karakter yang mempunyai makna khusus, misalkan double quote mark `"`.

Output

Untuk mencetak suatu informasi ke layar monitor melalui stdout (standard output), kita dapat menggunakan fungsi `printf()`. Format specifier wajib dipakai.

Contoh snippet

```
#include <stdio.h>

int main() {
    int a = 45;
    int b = 30;
    printf("%d\n", a + b); // output: 75
    printf("Belajar Bahasa C\n"); // output: Belajar Bahasa C

    return 0;
}
```

Input

Untuk melakukan input kedalam variabel melalui stdin (standard input), kita dapat menggunakan fungsi `scanf()`. Format specifier juga wajib dipakai.

Contoh snippet

```
#include <stdio.h>

int main() {
    float f;
    scanf("%f", &f); // menginputkan nilai dan memasukkan nilainya ke variabel f
    printf("%.2f\n", f); // mencetak nilai f dengan 2 angka belakang koma
    return 0;
}
```

Pelaksanaan Percobaan

Contoh Code Output 1

```
#include <stdio.h>

int main() {
    printf("%s%c\n", "Pemrograman C", '|');
    printf("%30s%c\n", "Pemrograman C", '|');
    printf("%-30s%c\n", "Pemrograman C", '|');
    printf("%30.5s%c\n", "Pemrograman C", '|');

    return 0;
}
```

Contoh Code Output 2

```
#include <stdio.h>

int main() {
    float x = 7654.123456789f;
    printf("%d %3d %8d\n", 1234, -567, 8910);
    printf("%d %3d %+8d\n", 1234, -567, 8910);
    printf("%f %15f %15.3f\n", x, x, x);

    return 0;
}
```

Contoh Code Variabel

```
#include <stdio.h>

int main() {
    int nilai=10;
    double nilai_2=5.3;
    int hasil;
    char s[] = "Belajar Java";
    printf("%d\n", nilai+nilai_2);
    printf("Kita sedang %s\n", s);

    return 0;
}
```

Contoh Code Input

```
#include <stdio.h>

int main() {
    int nilai1, nilai2, hasil;
    printf("Masukkan nilai 1 : ");
    scanf("%d", &nilai1);
    printf("Masukkan nilai 2 : ");
    scanf("%d", &nilai2);

    hasil = nilai1 + nilai2;
    printf("maka hasil = %d\n", hasil);

    return 0;
}
```

Data dan Analisis Hasil Percobaan

A. Output

1. Jalankan program Output1 – Output2 dan benahi jika menemukan kesalahan!
2. Pada program Output1 baris 3 – 6, jelaskan arti dari kode tersebut dan temukan perbedaannya!
3. Pada program output1 baris 3 ganti karakter 'j' menjadi 'ab' amati dan jelaskan!
4. Pada program Output1 setelah baris 6 tambahkan kode program dengan syntax `printf("%-30.8s%c\n", "Pemrograman C", '|');` kemudian amati perbedaannya dengan baris 6 dan simpulkan!
5. Pada program Output2 tambahkan kode program yang mencetak nilai x dengan format 2 angka dibelakang koma!
6. Pada program Output2 tambahkan kode berikut `printf("%g %15e %15.3f\n", x, x, x);` amati apa yang terjadi dan jelaskan!
7. Dari kedua jenis output tersebut simpulkan dan temukan perbedaannya!
8. Carilah format specifier lain yang tersedia di bahasa C selain yang ada di contoh!

B. Variabel

1. Jalankan program Variabel dan benahi jika menemukan kesalahan!
2. Setelah baris 5 tambahkan syntax untuk mengurangi nilai dari variabel nilai dan nilai_2, amati yang terjadi dan jelaskan! (benahi jika menemukan kesalahan)
3. Setelah baris 8 tambahkan variabel dengan type data char dengan nama variabel: ch dan mempunyai nilai sama seperti pada variabel s dan amati apa yang terjadi, jelaskan!
4. Tambahkan variabel dengan nama nilai3 yang bertipe data integer dan diberi nilai awal 100 kemudian lakukan pembagian yang disimpan pada variabel hasil kemudian cetak menggunakan kelas `printf()` !
5. Tambahkan variabel dengan nama nilai3 yang bertipe data integer dan diberi nilai awal 100 kemudian lakukan pembagian tanpa harus disimpan disebuah variabel melainkan cetak pada layar menggunakan `printf()` !
6. Jelaskan perbedaan antara no 4 dan no 5!

C. Input

1. Jalankan program Input dan benahi jika menemukan kesalahan!

2. Tambahkan perintah untuk memasukkan nama dan NIM user dan ditampilkan pada akhir program!

Tugas Praktikum

- 1) Buatlah program dengan tampilan sebagai berikut :

```
Masukkan operator pertama : 3
Masukkan operator kedua : 2
Hasil penjumlahan : 5
Hasil pengurangan : 1
Hasil perkalian : 6
Hasil pembagian : 1.5
```

- 2) Buatlah program untuk menghitung pemakaian daya listrik dirumah tangga secara sederhana. Tampilan program sebagai berikut :

```
Program penghitung pemakaian listrik sederhana
Masukkan Nama : Bpk Asisten
Keluarahan : Java
Masukkan posisi awal Kwh Meter : 8000
Masukkan posisi akhir Kwh Meter : 9000
Masukkan biaya beban saat ini : 140
Masukkan PPJ (dalam persen) : 10
=====PLN Java=====
Nama : Bpk Asisten
Kelurahan : Java
Pemakaian bulan ini : 1000 Kwh Meter
Tarif Listrik : Rp 140000,-
PPJ 10% : Rp 14000,-
Total Bayar : Rp 154000,-
=====
```

Bab 2 - Seleksi Kondisi

Tujuan

- Praktikan mampu memahami konsep penggunaan conditional assignment
- Praktikan mampu memahami konsep percabangan menggunakan if-else, if-else if-else
- Praktikan mampu memahami konsep percabangan menggunakan

Ringkasan Materi

Conditional Assignment

Sebelum mengenal menggunakan if kita dikenalkan bagaimana cara menggunakan dan kode untuk conditional assignment. Berikut kode dari conditional assignment :

```
type_data variabel = kondisi ? pernyataan_benar : pernyataan_salah
```

Dari kode diatas dapat dijelaskan bahwa pertama harus dilakukan pendeklarasian variabel dan type data dari variabel yang kita buat, kemudian kita beri suatu kondisi setelah itu jika kondisi benar maka program akan berjalan ke pernyataan benar namun jika salah maka akan melakukan pernyataan salah.

Contoh :

```
int m = (5%2==0) ? 1 : 0;
```

Jika program tersebut dijalankan maka akan mencetak "Berhasil" kondisi pada conditional assignment tersebut benar.

Seleksi Kondisi menggunakan if-else

Untuk melakukan percabangan tunggal kita dapat menggunakan if saja namun untuk percabangan yang lebih dari satu (percabangan majemuk) maka kita dapat menggunakan ifelse.

Bentuk dasar dari statemen ini adalah :

```
if (kondisi){
    Blok pernyataan;
}
else{
    Blok pernyataan
}
```

Namun untuk percabangan yang lebih dari 2, bentuk dasar yang digunakan adalah :

```
if (kondisi){
    Blok pernyataan 1;
}
else if (kondisi){
    Blok pernyataan 2;
}
else if (kondisi){
    Blok pernyataan 3;
}
else{
    Blok pernyataan 4;
}
```

Nested If

Suatu if memungkinkan untuk terapat if didalam if inilah yang disebut sebagai nested if. Alur programnya adalah jika kondisi if pertama benar maka program akan mengecek if kedua jika benar maka mengecek if ketiga begitu seterusnya.

Bentuk dasar dari nested if adalah sebagai berikut :

```
if (kondisi){
    if(kondisi){
        if(kondisi){
            blok pernyataan;
        }
        Else{
            Blok pernyataan;
        }
    }
    Else{
        Blok pernyataan;
    }
}
Else{
    Blok pernyataan;
}
```

Switch case

Selain menggunakan if untuk seleksi kondisi terdapat sintaks lain yaitu menggunakan Switch case. Program akan menampilkan output sesuai dengan inputan yang diberikan dengan batasan input berupa nilai awal sampai nilai akhir tertentu.

Bentuk dasar dari switch case adalah sebagai berikut :

```
switch (variabel) {  
    case nilai1 : statement1; break;  
    case nilai2 : statement2; break;  
    case nilai3 : statement3; break;  
    default : statement-def;  
}
```

Pelaksanaan Percobaan

Contoh Code Conditional Assignment

```
#include <stdio.h>  
  
int main() {  
  
    int x = 5;  
    ((x==5)) ? printf("Benar\n") : printf("Salah\n");  
  
    return 0;  
}
```

Contoh Code If-else

```
#include <stdio.h>

int main() {
    float nilai;

    printf("masukkan nilai : ");
    scanf("%f",&nilai);

    if (nilai > 60){
        printf("Anda Lulus\n");
    } else if (nilai >= 40){
        printf("Anda harus mengulang\n");
    } else {
        printf("Anda gagal\n");
    }

    return 0;
}
```

Contoh Code Nested If

```
#include <stdio.h>

int main() {

    int nilai;

    printf("masukkan nilai : ");
    scanf("%d",&nilai);

    if(nilai * 2 < 50){
        nilai += 10;
    }

    if (nilai <= 20){
        printf("Filkom\n");
        if (nilai % 2 == 1){
            printf("UB\n");
        } else{
            printf("Brawijaya\n");
        }
    } else {
        printf("PTIIK\n");
        if (nilai % 2 == 1){
            printf("UB\n");
        } else {
            printf("Brawijaya\n");
        }
    }

    return 0;
}
```

Contoh Code Switch case

```

#include <stdio.h>

int main() {

    int uang;

    printf("Tipe Mata uang yang tersedia\n");
    printf("1. Dolar - USA (kurs USD 1 = Rp 13000,-)\n");
    printf("2. Yen - Jepang( kurs JPY 1 = Rp. 4000,- )\n");
    printf("3. Poundsterling - Inggris( kurs 1 Poundsterling = Rp. 10.500, -)\n");
    printf("4. Euro - MEE( kurs EUR 1 = Rp. 8900,- )\n");
    printf("5. Riyal - Arab Saudi( kurs 1 Riyal = Rp. 1100,-)\n");
    printf("Masukkan jenis mata uang anda : ");
    int pilihan;
    scanf("%d", &pilihan);

    switch(pilihan){
        case 1 :
            printf("Data diterima, jenis valuta Anda : Dolar Amerika Serikat\n");
            printf("Masukkan banyak uang nda (dalam dolar) : ");
            scanf("%d", &uang);
            printf("Uang yang diterima : Rp. %d ,-",uang*13000); break;
        case 2 :
            printf("Data diterima, jenis valuta Anda : Yen Jepang\n");
            printf("Masukkan banyak uang nda (Yen Jepang) : ");
            scanf("%d", &uang);
            printf("Uang yang diterima : Rp. %d ,-",uang*4000); break;
        case 3 :
            printf("Data diterima, jenis valuta Anda : Poundsterling Inggris\n");
            printf("Masukkan banyak uang nda (Poundsterling Inggris) : ");
            scanf("%d", &uang);
            printf("Uang yang diterima : Rp. %d ,-",uang*15000); break;
        case 4 :
            printf("Data diterima, jenis valuta Anda : Euro MEE\n");
            printf("Masukkan banyak uang nda (Euro MEE) : ");
            scanf("%d", &uang);
            printf("Uang yang diterima : Rp. %d ,-",uang*14000); break;
        case 5 :
            printf("Data diterima, jenis valuta Anda : Riyal Arab Saudi\n");
            printf("Masukkan banyak uang nda (Riyal Arab Saudi) : ");
            scanf("%d", &uang);
            printf("Uang yang diterima : Rp. %d ,-",uang*5000); break;
        default: printf("system not found\n");

    }

    return 0;
}

```

Data dan Analisis Percobaan

A. Conditional Assignment

1. Jalankan Code Conditional Assignment dan benahi jika menemukan kesalahan!
2. Tambahkan kode dibawah program diatas yang fungsinya meminta inputan user dengan memasukkan nama dan nim masing-masing mahasiswa dan jika benar maka akan mencetak nama dan nim mahasiswa, jika salah maka mencetak "input nama salah" jika memasukkan nama yang salah, "input nim salah" jika memasukka nim yang salah
3. Buat program yang meminta untuk memasukkan nama dan password kemudian program akan meminta user untuk memasukkan nama dan password sesuai inputan sebelumnya. Jika benar maka program akan mencetak informasi biodata mahasiswa dan jika salah maka program akan mencetak "data tak ditemukan".

B. If - Else

1. Jalankan Code If - Else dan benahi jika menemukan kesalahan!
2. Masukkan nilai 30, 60 dan 80 saat program dijalankan, dan jawablah dengan screenshot hasil keluaran dari program!

C. Nested If

1. Jalankan Code Nested If dan benahi jika menemukan kesalahan!
2. Masukkan nilai 5, 20, 30 saat program dijalankan, jelaskan alur jalan program dan beri screenshot keluaran dari program!
3. Ubah kode diatas dengan memanfaatkan operasi and!

D. Switch Case

1. Jalankan Code Switch Case dan benahi jika menemukan kesalahan!
2. hapus kode `break` yang ada di program, pengaruh apa yang terjadi setelah pengubahan kode tersebut!
3. Apa perbedaan seleksi kondisi dengan menggunakan switch case dan if-else, dan kapan kita harus menggunakan if-else dan kapan menggunakan switch case?

Tugas Praktikum

1) Buatlah program sebagai berikut dengan menggunakan metode switch case

```

Menu :
1. menghitung luas dan keliling persegi panjang
2. menghitung luas dan keliling lingkaran
3. menghitung luas dan keliling segitiga
Pilihan anda : 3
Masukkan a : 3
Masukkan b : 4
Masukkan r : 5

Keliling segitiga   : 12 cm
Luas segitiga       : 6 cm2

Pilihan anda : 10
Data tak ditemukan, program dihentikan ...

```

2) Untuk menentukan kriteria kegemukan, digunakan IMT (Indeks Massa Tubuh), yang bisa dihitung menggunakan rumus :

$$IMT = b / (t * t)$$

b = berat badan (kg) t = tinggi badan (m)

Kriteria untuk nilai IMT ditabelkan sebagai berikut :

Nilai IMT	Kriteria
$IMT \leq 18,5$	Kurus
$18,5 < IMT \leq 25$	Normal
$25 < IMT \leq 30$	Gemuk
$IMT > 30$	Kegemukan

Susun program dengan tampilan sebagai berikut dengan menggunakan metode if-else!

```

Berat badan (kg)   : 45
Tinggi badan (m)   : 1.72
IMT      = 15,21           Termasuk kurus

Berat badan (kg)   : 85
Tinggi badan (m)   : 1.71
IMT      = 27,76           Termasuk gemuk

```

3) Susun program untuk masalah pengajian sebagai berikut : Masukan yang dibutuhkan oleh program adalah : jumlah jam kerja tiap minggu. Keluaran program adalah : total upah dari pegawai tertentu. Aturan yang diterapkan adalah :

- Batas kerja maksimal adalah 60 jam / minggu, dengan upah Rp. 5000,- / jam. Kelebihan jam kerja dari batas maksimum akan dianggap sebagai lembur dengan upah Rp. 6000,- /jam.
- Batas kerja minimal adalah 50 jam / minggu. Apabila pegawai mempunyai jam kerja di bawah batas kerja minimal ini, maka akan dikenakan denda sebesar Rp. 1000, - / jam.

```
Jam kerja   : 55
Upah        = Rp. 275000
Lembur      = Rp.      0
Denda       = Rp.      0
-----
Total       = Rp. 275000
```

Bab 3 - Perulangan

Tujuan

- Praktikan mampu memahami konsep penggunaan perulangan dengan menggunakan `for`
- Praktikan mampu memahami konsep penggunaan perulangan dengan menggunakan `while`
- Praktikan mampu memahami konsep penggunaan perulangan dengan menggunakan `do-while`
- Praktikan mampu memahami penggunaan pernyataan `break` dan `continue`

Ringkasan Materi

Perulangan dengan menggunakan `for`

Kata kunci `for` digunakan untuk mengulang pengeksekusian satu atau sejumlah pernyataan. Perulangan menggunakan `for` mempunyai batas awal, batas akhir dan kenaikan yang telah ditentukan terlebih dahulu. Perulangan akan dilakukan dengan membandingkan pencacah dengan batas akhir hingga ditemukan kondisi benar pada batas akhir, Bentuk umum penulisan perulangan menggunakan `for` adalah :

```
for (initialization; loop condition; step) {  
    // statement  
}
```

initialization = nilai awal dari sebuah looping

loop condition = kondisi dimana looping harus dijalankan jika condition bernilai true

step = update variabel yang digunakan dalam looping

Perulangan dengan menggunakan `while`

Kata kunci `while` digunakan untuk melakukan suatu proses perulangan yang memerlukan suatu kondisi tertentu untuk menghentikan perulangan. Perulangan akan dilakukan dengan membandingkan syarat perulangan dengan kondisi saat itu hingga ditemukan kondisi salah satu pada syarat perulangan. Bentuk umum penulisannya adalah :

```
while (kondisi_perulangan) {  
    // statement  
}
```

Perulangan dengan menggunakan do while

Hampir sama dengan perulangan menggunakan while, perulangan dengan do-while juga digunakan untuk melakukan perulangan yang memerlukan suatu kondisi tertentu untuk menghentikan perulangan. Perbedaan mendasar dengan perulangan menggunakan while adalah, dengan do-while, pengecekan kondisi dilakukan di belakang setelah baris statemen dalam blok do-while dijalankan (minimal 1 kali). Bentuk umum penulisan dengan do-while sebagai berikut :

```
do {  
    // statement  
} while (kondisi_perulangan);
```

Pernyataan break dan continue

Pernyataan break adalah pernyataan untuk mengentikan perulangan, sehingga akan keluar dari perulangan tersebut walaupun proses perulangan belum berakhir.

Bentuk pernyataan continue akan melewati bagian pernyataan setelah pernyataan ini dituliskan dan memeriksa ekspresi logika (boolean) yang mengontrol pengulangan. Jika operasi logika bernilai true, maka pengulangan tetap dilanjutkan. Pada dasarnya pernyataan ini akan melanjutkan bagian pengulangan pada pernyataan loop.

Pelaksanaan Percobaan

Loop for

```
#include <stdio.h>  
  
int main() {  
    int nilai;  
    for (nilai = 1 ; nilai <= 10 ; nilai++) {  
        printf("%d\n", nilai);  
    }  
    return 0;  
}
```

Loop while

```
#include <stdio.h>

int main() {
    int nilai = 1;
    while (nilai <= 10) {
        printf("%d\n", nilai);
        nilai++;
    }
    return 0;
}
```

Loop do while

```
#include <stdio.h>

int main() {
    int nilai = 1;
    do {
        printf("%d\n", nilai);
        nilai++;
    } while (nilai <= 10);
    return 0;
}
```

Break dan continue

```
#include <stdio.h>

int main() {
    int nilai;

    printf("pernyataan break batas 10\n");
    for (nilai = 1 ; nilai <= 10 ; nilai++) {
        if (nilai == 5) break;
        else printf("%d\n", nilai);
    }

    printf("\npernyataan continue batas 10\n");
    for (nilai = 1 ; nilai <= 10 ; nilai++) {
        if (nilai == 5) continue;
        else printf("%d\n", nilai);
    }

    return 0;
}
```

Data dan Analisis Hasil Percobaan

A. Looping For

1. Jelaskan dan perbaiki jika menemui kesalahan!
2. Apa fungsi dan variabel nilai dalam statemen for?
3. Dalam statemen for hapus bagian step, kemudian apa yang terjadi, jelaskan!
4. Dalam statement for hapus satu persatu secara bergantian mulai dari initialization, loop condition, dan step , amati yang terjadi dan jelaskan!

B. Looping while

1. Jelaskan dan perbaiki jika menemui kesalahan!
2. Setelah mengamati hasil keluaran, sebutkan perbedaan looping dengan menggunakan for dan while!
3. Hapus statemen i++ pada baris 6 kemudian amati yang terjadi dan jelaskan!
4. Ubah syntaks di atas untuk membuat deret angka kelipatan 2!

C. Looping do-while

1. Jelaskan dan perbaiki jika menemui kesalahan!
2. Setelah mengamati hasil keluaran, sebutkan perbedaan looping dengan menggunakan for, while dan do while!
3. Hapus statement i++ pada baris 6, amati yang terjadi dan jelaskan!
4. Ubah nilai dari variabel nilai baris ke 3 menjadi 11, amati yang terjadi dan jelaskan!

D. Break dan continue

1. Jelaskan dan perbaiki jika menemui kesalahan!
2. Jelaskan alur logika untuk pernyataan break dan continue pada program diatas!
3. Hapus pernyataan break pada baris 8 dan tuliskan kembali pernyataan break setelah else baris ke 8, amati yang terjadi dan jelaskan!
4. Pada if pernyataan continue baris ke 14 ubah pernyataan samadengan (==) menjadi pernyataan kurang dari samadengan (<=)

Tugas Pratikum

1) Buatlah program dengan tampilan sebagai berikut :

```
Masukkan nilai n = 4
*
* *
* * *
* * * *
```

2) Buatlah looping yang menuliskan nama anda secara vertikal, dengan huruf yang sesuai dengan huruf yang di tulisan. Contoh menuliskan A :

```
      A
     A  A
    A    A
   A A A A
  A A A A A
 A      A
A        A
```

3) Buatlah program sederhana untuk menghitung beberapa volume bidang dengan tampilan awal sebagai berikut :

```
MENU
0. KELUAR
1. HITUNG VOLUME BALOK
2. HITUNG VOLUME BOLA
3. HITUNG VOLUME KERUCUT
4. HITUNG VOLUME SILINDER
5. HITUNG VOLUME LIMAS SEGITIGA
MASUKKAN PILIHAN ANDA :
```


Bab 4 - Array

Tujuan

- Praktikan mampu mengenal array pada pemrograman C
- Praktikan mampu mengenal penggunaan array pada pemrograman C
- Praktikan mampu mengimplementasikan macam-macam array pada pemrograman C

Ringkasan Materi

A. Array

Array adalah variabel yang dapat menyimpan beberapa nilai dengan tipe data yang sama dalam ukuran tertentu. Index array secara default dimulai dari 0, sehingga apabila ada array dengan panjang “n” maka index array dimulai dari 0 hingga n-1. Contohnya seperti dibawah ini:

Array	:	[]	[]	[]	[]	.	.	.	[]
Index	:		0		1		2		3								n-1		

Untuk mendeklarasikan array pada C dapat dilakukan seperti di bawah ini.

Syntax :

```
Tipe_data Nama_array [ Ukuran_array ];
```

Contoh :

```
int dataArr [10];  
double data [5];
```

B. Array 1 Dimensi

Array 1 dimensi adalah array yang menggunakan satu set kurung siku.

Contoh deklarasi array 1 dimensi:

```
int dataArr [8];  
char kata [20];
```

Pemberian nilai pada array bisa dilakukan dengan memberi nilai langsung ataupun satu-persatu sesuai dengan indexnya. Apabila ingin memberi nilai langsung pada array kita bisa menggunakan satu set kurung kurawal, contohnya seperti di bawah ini:

```
int data [5] = {10, 11, 7, 5, 8};  
atau  
int nilai [] = {10, 11, 7, 5, 8};
```

Jika ingin memberi nilai langsung sesuai indexnya bisa seperti contoh di bawah ini:

```
data[3] = 10;
```

Untuk lebih jelasnya tentang array 1 dimensi bisa melihat contoh dibawah ini (Array1D.c)

C. Array Multi-Dimensi

Array Multi-Dimensi adalah array yang menggunakan lebih dari satu set kurung siku. Salah satu contoh array multi-dimensi adalah array dua dimensi.

Contoh deklarasi array 2 dimensi :

```
int arr2Dimensi [5][2] ;  
double arr2 [3][7];
```

Dalam array 2 dimensi kita dapat menganggap 1 set kotak pertama adalah baris dan 1 set kotak kedua adalah kolom. Untuk memberi nilai pada array 2 dimensi kita bisa lakukan seperti contoh di bawah ini:

```
int arrData [3][2] = {{1, 2}, {3, 4}, {5, 6}};
```

atau

```
arrData[1][1] = 10;
```

Untuk lebih jelasnya tentang array 2 dimensi bisa melihat contoh dibawah ini (Array2D.c)

D. Pointer

Pointer adalah variabel yang menyimpan suatu alamat dari variabel lainnya. Untuk mendeklarasikan pointer kita menggunakan tanda asterisk (*) di depan nama variabel dan menggunakan tanda & untuk mendapatkan alamat dari suatu variabel.

Contoh:

```
Tipe_data *nama_variabel;  
int *data;
```

untuk mendapatkan alamat dari suatu variabel bisa menggunakan tanda dan (&).

```
int nilai = 10;  
int *data = &nilai;
```

Pelaksanaan Percobaan

```
//Array1D.c  
#include<stdio.h>  
  
int main(){  
    int i ;  
    int data[]={11,3,4,7,24};  
  
    //menghitung jumlah byte yang dibutuhkan, karena int butuh 4byte maka (jumlah memo  
ry / 4)  
    int panjang = sizeof(data)/4;  
  
    printf("sebelum\n");  
    //mencetak semua nilai pada array sebelum diubah  
    for(i = 0; i < panjang; i++){  
        printf("index ke-%d adalah %d\n",i,data[i]);  
    }  
  
    for(i = 0; i < panjang; i++){  
        // nilai data sebelumnya di kali 2  
        data[i] = data[i]*2;  
    }  
    printf("\n\nSesudah\n");  
    //mencetak semua nilai pada array sesudah diubah  
    for(i = 0; i < panjang; i++){  
        printf("index ke-%d adalah %d\n",i,data[i]);  
    }  
}
```

```
//array2D
#include<stdio.h>
//mendefinisikan fungsi length
#define length(arr) ((int) (sizeof (arr) / sizeof (arr)[0]))

int main(){

    /* membuat array dengan baris 3 dan kolom 3
        0  1  2
    0  [] [] []
    1  [] [] []
    2  [] [] []
    */
    int data[3][3];
    int i,j;
    //mengetahui panjang kolom
    int kolom = length(data);
    //mengetahui panjang baris
    int baris = length(data[0]);

    // mengisi data array
    for(i=0; i < kolom; i++){
        for(j=0; j < baris; j++){
            data[i][j] = i * j;
        }
    }
    //mencetak nilai dalam array
    for(i=0; i < kolom; i++){
        for(j=0; j < baris; j++){
            printf("%d ",data[i][j]);
        }
        printf("\n");
    }

}
```

Data dan Analisis Percobaan

Array 1 Dimensi

1. Jalankan program Array1D diatas , benahi jika ada kesalahan!
2. Tambahkan statement berikut (data[5] = 10) sebelum for pertama , apa yang terjadi ? jelaskan?
3. Ubahlah nilai panjang menjadi 10, apa yang terjadi ? jelaskan?

Array 2 Dimensi

1. Jalankan program Array2D diatas, benahi jika ada kesalahan!
2. Ubahlah (`baris = length(data[0])`) menjadi (`baris = length(data[0][0])`), apa yang terjadi? Jelaskan?

Tugas Praktikum

1. Buatlah program untuk melakukan penjumlahan dan pengurangan array 2 dimensi (Matrik) ?
2. Buatlah program untuk menggabungkan 2 buah array menjadi 1?
3. Buatlah program untuk melakukan pengurutan data dari nilai terkecil hingga terbesar?

Bab 5 - Program Dengan Menggunakan Method

Tujuan

- Praktikan mampu mengenal method pada pemrograman C
- Praktikan mampu mengenal penggunaan method pada pemrograman C
- Praktikan mampu mengimplmentasikan macam-macam method pada pemrograman C

Ringkasan Materi

Method Void pada C

Method merupakan suatu program yang terletak terpisah dari program utama, tetapi tetap merupakan bagian dari program yang dibuat. Dengan menggunakan method dapat membuat program menjadi lebih mudah dimengerti dan mudah dalam pendokumentasian. Method void merupakan sebuah method yang tidak mengembalikan nilai yang ada pada method tersebut. Berikut merupakan contoh dari penggunaan method void pada C (prak_void.c).

Method Return Value pada C

Method return value merupakan method yang mengembalikan suatu nilai ketika kembali ke program utamanya disertai dengan membawa suatu nilai. Berikut merupakan contoh dari penggunaan method return value pada C (prak_retval.c)

Method rekursif pada C

Method rekursif adalah method yang dimana ia memanggil dirinya sendiri ketika kondisi tertentu. Berikut merupakan contoh dari penggunaan method rekursif pada C (prak_rekursif.c)

Pass by value dan pass by reference

Perbedaan pass by value dengan pass by reference adalah sebagai berikut..

Pass by value

- Parameter dalam method atau function merupakan copy dari variabel asli yang dilewatkan di parameter method
- Perubahan yang terjadi pada variabel yang dilewatkan tersebut tidak berpengaruh pada nilai asli variabelnya

Contoh code

```
// Pass by value
#include <stdio.h>

void test(float f) {
    f = -1.5f;
}

int main() {
    float f = 3.14f;
    test(f);
    printf("%.2f\n", f); // output : 3.14

    return 0;
}
```

Pass by reference

- Parameter dalam method atau function merupakan reference menuju lokasi memori dimana variabel tersebut menyimpan nilainya
- Perubahan yang terjadi pada variabel yang dilewatkan tersebut berpengaruh pada nilai asli variabelnya
- Menggunakan pointer untuk menunjuk ke reference variabel sebenarnya

Contoh code

```
// Pass by reference
#include <stdio.h>

// menunjuk ke alamat memori sebenarnya dengan pointer
void test(float *f) {
    *f = -1.5f;
}

void multiplyArray(int *arr, int length, int multiplier) {
    for (int i = 0 ; i < length ; i++) {
        arr[i] *= multiplier;
    }
}

int main() {
    float f = 3.14f;
    int array[] = {1, 2, 3, 4, 5};

    test(&f);
    multiplyArray(&array, sizeof(array) / sizeof(*array), 2); // array = [2, 4, 6, 8,
10]

    printf("%.2f\n", f); // output : -1.50
    return 0;
}
```

Pelaksanaan Percobaan

Contoh code method void


```
// prak_void.c
#include <stdio.h>

void greeting(char nama[]) {
    printf("Hello, %s\n", nama);
}

void morningGreeting() {
    printf("Good morning\n");
}

int main() {
    char nama[50]; // 50 karakter nama
    printf("Masukkan nama : ");
    scanf("%s", &nama);
    morningGreeting();
    greeting(nama);

    return 0;
}
```

Contoh code method return value

```
// prak_retval.c
#include<stdio.h>

int faktorial(int n) {
    int hasil = 1;
    for (int i = 1 ; i <= n ; i++) {
        hasil *= i;
    }
    return hasil;
}

int main() {
    printf("Nilai dari 5! = %d\n", faktorial(5));

    return 0;
}
```

Contoh code method rekursif

```
// prak_rekursif.c
#include<stdio.h>

int faktorial(int n) {
    return (n == 1) ? 1 : n * faktorial(n - 1);
}

int main() {
    printf("Nilai dari 5! = %d\n", faktorial(5));

    return 0;
}
```

Contoh code pass by reference

```
// prak_reference.c
#include <stdio.h>

// menunjuk ke alamat memori sebenarnya dengan pointer
void multiplyArray(int *arr, int length, int multiplier) {
    for (int i = 0 ; i < length ; i++) {
        arr[i] *= multiplier;
    }
}

void print(int arr[], int length) {
    for (int i = 0 ; i < length ; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main() {
    int array[] = {1, 2, 3, 4, 5};
    multiplyArray(&array, sizeof(array) / sizeof(*array), 7);
    print(array, sizeof(array) / sizeof(*array));
    return 0;
}
```

Data dan Analisis Percobaan

Method Void

1. Jalankan program prak_void.c dan benahi jika ada kesalahan!

2. Ubahlah deklarasi method `void greeting(char nama[])` menjadi `void greeting(char* nama)` , apa yang terjadi? Jelaskan!
3. Ubahlah `scanf("%s", &nama);` menjadi `scanf("%c", nama);` , apa yang terjadi? Jelaskan!

Method Return Value

1. Jalankan program `prak_retval.c` dan benahi jika ada kesalahan!
2. Mengapa di method `main()` harus mengembalikan nilai 0? Jelaskan!
3. Ubahlah method signature `int faktorial(int n)` menjadi `float faktorial(int n)` , apa yang terjadi? Jelaskan!

Method rekursif

1. Jalankan program `prak_rekursif.c` dan benahi jika ada kesalahan!
2. Ubahlah `return (n == 1) ? 1 : n * faktorial(n - 1);` menjadi `return faktorial(n-1)` , apa yang terjadi? Jelaskan!
3. Gambarkan proses rekursif yang terjadi dalam program tersebut!

Pass by reference

1. Jalankan program dan benahi jika ada kesalahan!
2. Ubahlah `void multiplyArray(int *arr, int length, int multiplier)` menjadi `void multiplyArray(int arr[], int length, int multiplier)` , apa yang terjadi? Jelaskan!
3. Jelaskan mengapa `sizeof(array) / sizeof(*array)` akan mengembalikan panjang dari array!

Tugas Praktikum

- 1) Buatlah program untuk menentukan apakah bilangan tersebut bilangan prima atau bukan dengan menginputkan sebuah bilangan yang di inginkan. Setelah itu program akan mengoutputkan bilangan prima 1-100!
- 2) Buatlah program yang akan menghasilkan segitiga pascal seperti dibawah ini

n = 10

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
```

3) Buatlah sebuah program yang terdiri dari beberapa operasi terhadap matriks. Berikut ini daftar method yang harus diimplementasikan

```
void addition(int dest[][dest_cols], int a[][a_cols], int b[][b_cols])
void subtraction(int dest[][dest_cols], int a[][a_cols], int b[][b_cols])
void multiplication(int dest[][dest_cols], int a[][a_cols], int b[][b_cols])
int isDiagonalMatrix(int source[][source_cols]) // return 0 if not, else return 1
int isIdentityMatrix(int source[][source_cols]) // return 0 if not, else return 1
```