

BAB I

Pengantar Algoritma dan Bahasa C

Target :

Mengetahui Algoritma dan memahami algoritma dapat memecahkan masalah dalam kehidupan sehari-hari atau beberapa kasus sehari-hari, mampu memahami *flow chart*, *logika* dan mengetahui *variable* serta penggunaanya.

Materi :

- ✓ Sejarah Algoritma
 - ✓ Penulisan Algoritma
 - ✓ Flow Chart
 - ✓ Logika AND dan OR
 - ✓ Operator Matematika
 - ✓ Operator Pembandingan
 - ✓ Type DATA
 - ✓ VARIABEL
 - ✓ Menenal Python
 - Command Line
 - IDLE (GUI) Python
- Interpreter Python

Algoritma ?

Kata algoritma, mungkin bukan sesuatu yang asing bagi kita.

Penemunya adalah seorang ahli matematika dari uzbekistan yang bernama Abu Abdullah Muhammad Ibn Musa al-Khwarizmi (770-840).

Di literatur barat dia lebih terkenal dengan sebutan Algorizm. Panggilan inilah yang kemudian dipakai untuk menyebut konsep algorithm yang ditemukannya. Dalam bahasa Indonesia kita kemudian menyebutkannya sebagai algoritma. Algoritma adalah kunci dari bidang ilmu komputer, karena banyak bidang di bawah ilmu komputer yang lahir berdasarkan konsep algoritma ini. Pada hakekatnya algoritma juga adalah kunci dari kehidupan kita. Cara membuat masakan (resep masakan) adalah juga sebuah contoh nyata dari algoritma.

Definisi Algoritma :

Kita bisa mendefinisikan algoritma seperti dibawah:

Algoritma adalah logika, metode dan tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan.

Kamus besar bahasa Indonesia :
(Balai Pustaka 1988) secara formal

Algoritma adalah urutan logis pengambilan putusan untuk pemecahan masalah.

Menilai Sebuah Algoritma
(*Cepat Mahir Algoritma dalam Bahasa C*
Romi Satria Wahono)

Menilai Sebuah Algoritma :

Ketika manusia berusaha memecahkan masalah, metode atau teknik yang digunakan untuk memecahkan masalah itu ada kemungkinan bisa banyak (tidak hanya satu). Dan kita memilih mana yang terbaik diantara teknik-teknik itu. Hal ini sama juga dengan algoritma, yang memungkinkan suatu permasalahan dipecahkan dengan metode dan logika yang berlainan. Lalu bagaimana mengukur mana algoritma yang terbaik ?

Beberapa persyaratan untuk menjadi algoritma yang baik adalah:

- **Tingkat kepercayaannya tinggi (realibility).** Hasil yang diperoleh dari proses harus berakurasi tinggi dan benar.
- **Pemrosesan yang efisien (cost rendah).** Proses harus diselesaikan secepat mungkin dan frekuensi kalkulasi yang sependek mungkin.
- **Sifatnya general.** Bukan sesuatu yang hanya untuk menyelesaikan satu kasus saja, tapi juga untuk kasus lain yang lebih general.
- **Bisa Dikembangkan (expandable).** Haruslah sesuatu yang dapat kita kembangkan lebih jauh berdasarkan perubahan requirement yang ada.
- **Mudah dimengerti.** Siapapun yang melihat, dia akan bisa memahami algoritma anda. Susah dimengertinya suatu program akan membuat susah di maintenance (kelola).
- **Portabilitas yang tinggi (Portability).** Bisa dengan mudah diimplementasikan di berbagai platform komputer.

Dasar Pemrograman C .

Sejarah

• Bahasa C merupakan perkembangan dari bahasa BCPL yang dikembangkan oleh Martin Richards th.1967. Selanjutnya bahasa ini memberikan ide kepada Ken Thompson yang kemudian mengembangkan bahasa B tahun 70an. Selanjutnya dari bahasa B berkembang menjadi Bahasa C oleh Dennis Ritchie tahun 70 an di Bell Telephone Laboratories Inc. Pertama kali digunakan Bahasa C pada system operasi UNIX.

Kelebihan Bahasa C:

1. Bahasa C tersedia hampir di semua jenis komputer
2. Kode bahasa C sifatnya flexibel dan portable untuk semua jenis komputer
3. Bahasa C menyediakan hanya sedikit kata kunci sekitar 32 kata kunci.
4. Proses executable bahasa C lebih cepat
5. Dukungan pustaka yang banyak
6. Bahasa C bahasa yang terstruktur

Kelemahan Bahasa C:

Kekurangan antara lain :

1. Banyaknya operator serta fleksibilitas penulisan program kadang-kadang membingungkan pemakai.
2. Bagi pemula pada umumnya kesulitan menggunakan pointer.

TYPE DATA C

VARIABEL

Variabel adalah suatu pengenal (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program. Nilai suatu variabel dapat berubah-ubah sehingga berbeda dengan konstanta yang selalu tetap. Variabel merupakan penampung suatu nilai/data.

Penamaan Variable :

1. Tidak boleh menggunakan symbol-symbol khusus kecuali garis bawah.
Symbol khusus \$,?,%,#,!,&*,(,),- ,+= dsb
2. Tidak boleh diawali dengan angka.
3. ditulis dengan huruf kecil semua.

Di dalam bahasa C, semua variabel yang akan dipakai harus dideklarasikan lebih dahulu. Deklarasi menyatakan jenis data dan diikuti nama variabelnya.

Adapun jenis-jenis data yang ada :	ukuran	Range	Format
- Char	untuk type data karakter	1 byte	-128 s/d 127 %c
- Int	untuk type data bulat	2 byte	-32768 s/d 32767 %i, %d
- Float	untuk bilangan pecahan	4 byte	-3,4E-38 s/d 3.4E +38 %f
-long int	untuk bilangan integer yang lebih luas.		%ld
- double	untuk pecahan lebih luas		%lf

A. Deklarasi Konstanta (#define)

Dalam bahasa C dideklarasikan dengan #define

Contoh :

```
#define Phi 3.14
#define nim "01233"
#define nama "sri widya"
```

SYNTAX : **#define namavariabel nilai**

Contoh : #define Max 12

B. Praprosessor (#include)

instruksi praprosessor ini digunakan sebagai penyertaan file yang digunakan dalam program.

SYNTAX : **#include <namafile.h>**

Contoh :

#include <stdio.h>, pernyataan ini disertakan jika dalam program menggunakan rutin masukan dan keluaran misalkan :
fungsi printf(), scanf(), getchar(), putchar(), puts(), gets().

Suatu program C ditulis dari beberapa elemen yaitu :

- Pernyataan
- Blok pernyataan
- Blok fungsi

Suatu pernyataan adalah satu atau beberapa ekspresi yang diakhiri dengan tanda titik koma (;).

Blok pernyataan adalah terdiri satu atau beberapa pernyataan yang diawali dengan tanda { dan diakhiri dengan tanda }. Blok pernyataan selalu diawali dengan suatu instruksi.

Blok fungsi terdiri dari satu atau beberapa pernyataan yang dirancang untuk melakukan proses tertentu. Blok fungsi dimulai dengan **nama fungsinya()** dan tanda { dan diakhiri dengan tanda }

JENIS JENIS FUNGSI :

1. Fungsi Main()

Fungsi ini untuk menandai titik tempat program C mulai dilaksanakan. Fungsi ini hanya bisa digunakan satu kali dalam sebuah program.

2. Fungsi Printf

Fungsi ini digunakan untuk menampilkan argumen teks /tulisan pada layar.

SYNTAX : Printf(“.....text.....”,NamaVariabel1, NamaVariabel2, dst)

Karakter (%) untuk konversi bilangan.

Misal :

Printf(“Jumlah dari %d dan %d = %d\n”, x, y, jum);

%d adalah type format tampilan bilangan bulat desimal.

%f adalah type format tampilna bilangan real.

3. Fungsi Scanf

Fungsi ini dipakai untuk membaca karakter dari alat input. Karakter yang akan dibaca diberikan kepada variabel.

SYNTAX : `scanf(".....text...% .",&namavariabel);`

VARIABEL

Di dalam bahasa C, semua variabel yang akan dipakai harus dideklarasikan lebih dahulu. Deklarasi menyatakan jenis data dan diikuti nama variabelnya.

Adapun jenis-jenis data yang ada :

- Char untuk type data karakter
- Int untuk type data bulat
- Float untuk bilangan pecahan

#define

Penunjuk praprosessor `#define` mendefinisikan sebuah konstanta simbolis yang digunakan dalam program.

SYNTAX : **`#define namavariabel nilai`**

Contoh : `#define Max 12`

#include

instruksi praprosessor ini digunakan sebagai penyertaan file yang digunakan dalam program.

SYNTAX : **`#include <namafile.h>`**

Contoh :

- `#include <stdio.h>`, pernyataan ini disertakan jika dalam program menggunakan rutin masukan dan keluaran misalkan :
fungsi `printf()`, `scanf()`, `getchar()`, `putchar()`, `puts()`, `gets()`.
- `#include <ctype.h>`, pernyataan ini dipakai jika dalam program ada instruksi makro misalkan `tolower()`, `toupper()` dll.

BAB II

PERINTAH DASAR C

Target :

Mengetahui Perintah-perintah dasar pemrograman C. Memahami perintah cetak di layar, input data, proses dan output, type data dan operator matematika. Mampu mengoperasikan memecahkan masalah dengan program C melalui input, proses dan output.

Materi :

- ✓ Perintah tampilan /cetak di layar
- ✓ Perintah Masukkan
- ✓ Perintah Output terhadap input /Keluaran
- ✓ Penggunaan Komentar
- ✓ Operator matematika
- ✓ Operator Logika
- ✓ Operator Pembanding

A. PERINTAH TAMPILAN/CETAK DILAYAR

SYNTAX : printf(".....text.....");

Atau

printf(".....text...%d....",variable);



%d = type data

varibel

SYNTAX : puts(".....text.....");

printf(".....\n");

\n= bermaksan enter. Kursor berpindah ke bawah

Contoh 2.0 penggunaan fungsi printf() dan main() PUTS

```
#include <stdio.h>
main()
{
clrscr(); /* Hapus layar */
puts("Hello");
printf("Selamat datang di Program Bahasa C\n");
/* tanda : \n akan membawa kursor posisi dibawah */
getch(); /*Menahan Sampai tombol Enter ditekan*/
}
```

B. PERINTAH MASUKKAN/ INPUT

Perintah input merupakan perintah masukkan dari keyboard.

Perintah input terbagi dua yaitu :

- Input tidak langsung / pemesanan variable berupa non konstanta

SYNTAK INPUT :

`scanf("%type data", &variable)`

Type data :

`scanf("%d",&variabel);` → integer

`scanf("%c",&variabel);` → character

`scanf("%s",&variabel);` → string

`scanf("%ld",&variabel);` → long integer / double

`scanf("%f",&variabel);` → float / pecahan

`scanf("%lf",&variabel);` → long float / pecahan digit banyak

SYNTAK INPUT KHUSUS HURUF :

`gets(variable);`

contoh 2.1 penggunaan gets

```
#include <stdio.h>
main()
{
    char nama[12], alamat[20], kota[12];
    clrscr(); /** menghapus layar */
    printf("Masukkan Nama Anda = "); gets(nama); /** gets untuk char **/
    printf("Masukkan Alamat Anda = "); gets(alamat); /** gets untuk char **/
    printf("Masukkan Kota = "); gets(kota); /** gets untuk char **/
    printf("=====\n");
    printf("Hasil masukan \n");
    printf("Nama Anda adalah = %s \n",nama); /** menampilkan dengan %s **/
    /** tanda \n ke bawah **/
    printf("Alamat Anda adalah = %s \n",alamat); /** menampilkan dengan %s **/
    /** tanda \n ke bawah **/
    printf("Kota Anda adalah = %s \n",kota); /** menampilkan dengan %s **/
    /** tanda \n ke bawah **/
    getch();
}
```

C. PERINTAH OUTPUT TERHADAP INPUT

SYNTAX : `printf(".....text.....");`

Atau

`printf(".....text...%d...",variable);`



%d = type data

varibel

contoh 2.2 penggunaan Scanf untuk angka integer = bulat

```

/**** program menggunakan scanf untuk integer ****/
#include <stdio.h>
main()
{
    int no, nilai;
    clrscr(); /** menghapus layar */
    printf("Masukkan Nomor Anda  = ");scanf("%d",&no); /** scanf utk angka **/
    printf("Masukkan Nilai Anda  = ");scanf("%d",&nilai); /** scanf utk angka **/
    printf("===== \n");
    printf("Hasil masukan \n");
    printf("Nomor Anda adalah = %d \n",no); /** menampilkan dengan %d **/
    /** tanda \n ke bawah **/
    printf("Nilai Anda adalah = %d \n",nilai); /** menampilkan dengan %d **/
    /** tanda \n ke bawah **/
    getch();
}

```

Contoh : 2.3 Menggunakan Scanf pada pecahan = float

```

/**** program menggunakan scanf untuk pecahan /float ****/
#include <stdio.h>
main()
{
    int no;
    float hasilbagi, nilai;
    clrscr(); /** menghapus layar */
    printf("Masukkan Nomor Anda  = ");scanf("%d",&no); /** scanf utk angka **/
    printf("Masukkan Nilai Anda  = ");scanf("%f",&nilai); /** scanf utk angka **/
    printf("===== \n");
    hasilbagi=nilai/2;
    printf("Hasil masukan \n");
    printf("Nomor Anda adalah = %d \n",no); /** menampilkan dg %d **/
    printf("Nilai Anda adalah = %f \n",nilai); /** menampilkan dg %f **/
    printf("Nilai Hasil bagi = %8.2f \n",hasilbagi);/** menampilkan dg %f **/
}

```



```
/** %8.2f 8 di depan koma dan 2 digit di belakang koma */
getch();
}
```

D. KOMENTAR

Perintah komentar atau keterangan merupakan program yang tidak melakukan eksekusi apa-apa. Perintah tersebut boleh ditulis boleh tidak, manakala ingin ditulis hanya untuk sebagai keterangan/ dokumentasi program saja.

Memberi komentar atau keterangan tidak menyebabkan program error.

SYNTAX : /*.....komentar di sini.....*/

Contoh

```
#include <stdio.h>
```

```
/*...ini hanya tulisan keterangan program tidak dieksekusi */
```

E. OPERATOR MATEMATIKA

Operator	Makna	Contoh	Hasil
+	penjumlahan	printf("3 + 4=%d\n",3+4);	3 + 4=7
-	pengurangan	printf("10-5=%d\n",10-5);	10-5=5
/	pembagian	printf("15/3=%d\n",15/3);	15/3=5
	Pembagian pecah	printf("10/5=%2.2f\n",10.0/5.0);	10/5=2.00
*	perkalian	printf("5 x 10=%d\n",5*10);	5 x 10=50
%	Sisa bagi	printf("23 mod 10=%d\n",23%10);	23 mod 10=3
	Sisa bagi	printf("3 mod 2=%d\n",3%2);	3 mod 2 = 1

F. OPERATOR LOGIKA

&&	Logika And (DAN)
	Logika Or (ATAU)
!	Logika Not (Ingkar)

AND &&			OR		
AND		HASIL	OR		HASIL
S	S	S	S	S	S
S	B	S	S	B	B
B	S	S	B	S	B
B	B	B	B	B	B

G. OPERATOR PEMBANDING

>	Lebih Besar
<	Kurang dari
>=	Lebih Besar sama dg
<=	Kurang sama dg
= =	sama dengan
!=	tidak sama dg

Contoh 2 Latihan Menghitung Luas Persegi Panjang :

```

/**** program menghitung Luas persegi Panjang ****/
#include <stdio.h>
main()
{
    int panjang,lebar,hasil;
    clrscr();
    printf("Program Hitung Luas Persegi Panjang \n");
    printf("===== \n");
    printf("Masukkan Panjang    = ");scanf("%d",&panjang);
    printf("Masukkan Lebar      = ");scanf("%d",&lebar);
    printf("===== \n");
    hasil=panjang * lebar;
    printf("Hasil Perhitungan Luas Persegi Panjang \n");
    printf("Luas Persegi Panjang = %d %s \n",hasil, "M persegi");
    getch();
}

```

Latihan 2 : buatlah program untuk menghitung luas segi tiga (alat * tinggi) /2

BAB III

PERINTAH KONDISI (IF)

Target :

Mengetahui Perintah Kondisi dan penggunaan statemen if. Memahami ifelse, banyak kondisi1, kondisi 2 dalam if dipadukan dengan logika *and* dan *or*.

Materi :

- ✓ Statement if
- ✓ Banyak if

Pengendalian Proses IF dan else

SYNTAX :

<p>If kondisi</p> <p style="padding-left: 40px;">Pernyataan1</p> <p>Else</p> <p style="padding-left: 40px;">Pernyataan2</p>	<p>atau If Kondisi</p> <p style="padding-left: 40px;">{ }</p> <p>if Kondisi</p> <p style="padding-left: 40px;">{ }</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------

Jika kondisi benar maka kerjakan pernyataan 1 dan jika salah kerjakan pernyataan 2

“ And “ dalam Bahasa C dilambangkan dalam &&

“ Or “ dalam Bahasa C dilambangkan dalam ||

“ =” dalam perintah if akan dilambangkan ==

Contoh 1 program untuk mengetes kondisi

```
#include <stdio.h>
main()
{
char nama[12];
int nilai;
clrscr();
printf("Program Mengetes Kondisi \n");
printf("===== \n");
printf("Masukkan Nama      =");gets(nama);
printf("Masukkan Nilai      =");scanf("%d",&nilai);
if (nilai>=60)
{ printf("LULUS BAIK \n"); }
if ((nilai>=0) && (nilai<=59))
{ printf("TIDAK LULUS \n"); }
getch();
}
```

atau

contoh 2 program mengetes kondisi

```
#include <stdio.h>
main()
{
char nama[12];
int nilai;
clrscr();
printf("Program Mengetes Kondisi \n");
printf("===== \n");
printf("Masukkan Nama      =");gets(nama);
printf("Masukkan Nilai      =");scanf("%d",&nilai);
if (nilai>=60)
{ printf("LULUS BAIK \n"); }
else
{ printf("TIDAK LULUS \n"); }
getch();
}
```

Contoh 3 penggunaan case

```
#include <stdio.h>
main()
{
int pilih;
clrscr();
```

```
printf("Silahkan pilih =");scanf("%d",&pilih);
switch(pilih)
{
    case 1: puts("Anda memilih Satu");break;
    case 2: puts("Anda memilih Dua");break;
    case 3: puts("Anda memilih Tiga");break;
    case 4: puts("Anda memilih 4 Keluar");exit(0);
}
getch();
}
```

Run

Silahkan pilih =1

Anda memilih Satu

Silahkan pilih =4

Anda memilih 4 Keluar

SOAL :

Buatlah tes kondisi untuk nilai-nilai 80 – 100	Nilai huruf A Lulus Cumlaud
70 – 79	Nilai huruf B Lulus baik
60 – 69	Nilai huruf C Lulus Cukup
50 – 59	Nilai huruf D Tidak Lulus
0 – 49	Nilai huruf E Gagal

BAB IV

PERINTAH ITERASI/PERULANGAN

Target :

Mengetahui Perintah dan penggunaan statemen perulangan. Memahami kasus dalam perulangan, perulangan di dalam kondisi1, kondisi2 pada if. Mengetahui perulangan terkendali dan tak terkendali.

Materi :

- ✓ Statement for
- ✓ Perulangan terkendali
- ✓ Perulangan tak terkendali

STATEMENT FOR

Mengapa Perulangan ?

Perulangan dipakai untuk mengulangan data yang sama sebanyak beberapa kali

Jenis Perulangan :

1. Perulangan terkendali (for)
2. Perulangan tak terkendali (while)

A. PERULANGAN TERKENDALI

Contoh kasus buatlah program untuk mencetak :

Saya suka bahasa C sebanyak 3 X

Program sebagai berikut :

```
#include <stdio.h>
main()
{
    print "saya suka C \n"
    print "saya suka C \n"
    print "saya suka C \n"
}
```

program di atas betul dan tidak ada salahnya, tetapi menjadi bermasalah manakalah program tersebut tidak untuk mencetak sebanyak 3 kali tetapi sebanyak 100 X atau bahkan sebanyak 1000 X maka program tersebut menjadi bermasalah.

Syntax : For (NmVar=0; NmVar< n; NmVar++)

Contoh 1 Penggunaan Perulangan terkendali for

```
main()
{
    int i,n;
    clrscr();
    printf("Program Tes Perulangan \n");
    printf("===== \n");
    printf("Masukkan Berapa kali = ");scanf("%d",&n);
    for (i=0; i<n; i++)
    { printf("Mengulang sebanyak = %d %s \n",i," Kali"); }
    getch();
}
```

Contoh 2 Perulangan dimulaid dari 0

```
/*program perulanga*/
#include <stdio.h>
main()
{
    int i,n;
    clrscr();
    printf("Masukkan banyak n=");scanf("%d",&n);
    printf("Diulang dengan for\n");
    for (i=0; i<n; i++)
    {
        printf("Hebat.....yang ke - %d\n",i);
    }
    getch();
}
```

```
run
Masukkan banyak n=10
Diulangan dengan for
Hebat.....yang ke - 0
Hebat.....yang ke - 1
Hebat.....yang ke - 2
Hebat.....yang ke - 3
Hebat.....yang ke - 4
Hebat.....yang ke - 5
Hebat.....yang ke - 6
Hebat.....yang ke - 7
Hebat.....yang ke - 8
Hebat.....yang ke - 9
```

Contoh 2 Perulangan dimulai dari 1

```
/*program perulangan*/
#include <stdio.h>
main()
{
    int i,n;
    clrscr();
    printf("Masukkan banyak n=");scanf("%d",&n);
    printf("Diulangan dengan for dimulai 1\n");
    for (i=1; i<=n; i++)
    {
        printf("Hebat.....yang ke - %d\n",i);
    }
    getch();
}
```


Run

Masukkan banyak n=10

Diulangan dengan for dimulai 1

Hebat.....yang ke - 1

Hebat.....yang ke - 2

Hebat.....yang ke - 3

Hebat.....yang ke - 4

Hebat.....yang ke - 5

Hebat.....yang ke - 6

Hebat.....yang ke - 7

Hebat.....yang ke - 8

Hebat.....yang ke - 9

Hebat.....yang ke - 10

Contoh 3 Perulangan di dalam If mengecek bilangan genap n kali

```
main()
{
    int angka;
    int i,n;
    clrscr();
    printf("Masukkan Banyaknya Bilangan=");scanf("%d",&n);
    for (i=0; i<=n; i++)
    {
        if (i%2==0)
        {
            printf("\nAngka = %d %s",i,"Bilangan Genap");
        }
        else
            printf("\nAngka = %d %s",i,"Bilangan Ganjil");
    }
    getch();
}
```

Run

Masukkan Banyaknya Bilangan=7

Angka = 0 Bilangan Genap

Angka = 1 Bilangan Ganjil

Angka = 2 Bilangan Genap

Angka = 3 Bilangan Ganjil

Angka = 4 Bilangan Genap

Angka = 5 Bilangan Ganjil

Angka = 6 Bilangan Genap

Angka = 7 Bilangan Ganjil

B. PERULANGAN TAK TERKENDALI

Perulangan tak terkendali merupakan perulangan yang diulang tidak dapat dipastikan berapa kali mengulang datanya namun sesuai dengan kondisi memenuhi atau tidak memenuhi.

Contoh 4 Penggunaan perulangan tak terkendali dengan do while dgn I dari 1

```
main()
{
    int n,i;
    clrscr();
    printf("banyak Perulangan =");scanf("%d",&n);
    i=1;
    do
    {
        printf("ulang ke -%d\n",i);
        i=i+1;
    }
    while (i<=n);
    getch();
}
```

Run

banyak Perulangan =7

ulang ke -1

ulang ke -2

ulang ke -3

ulang ke -4

ulang ke -5

ulang ke -6

ulang ke -7

Contoh 5 Penggunaan perulangan dengan do while dengan I dimulai dari 0

```

main()
{
    int n,i;
    clrscr();
    printf("banyak Perulangan =");scanf("%d",&n);
    i=0;
    do
    {
        printf("ulang ke -%d\n",i);
        i=i+1;
    }
    while (i<n);
    getch();
}

```

Run

```

banyak Perulangan =7
ulang ke -0
ulang ke -1
ulang ke -2
ulang ke -3
ulang ke -4
ulang ke -5
ulang ke -6

```

Contoh 6 penggunaan perulangan di dalam perulangan atau for di dalam for

```

main()
{
    int n,i,j;
    clrscr();
    printf("banyak Perulangan =");scanf("%d",&n);
    for (i=1; i<=n; i++)
    {
        for (j=1;j<=n; j++)
        {
            printf("i=%d j=%d\n",i,j);
        }
    }
    getch();
}

```

Run

banyak Perulangan =4

i=1 j=1

i=1 j=2

i=1 j=3

i=1 j=4

i=2 j=1

i=2 j=2

i=2 j=3

i=2 j=4

i=3 j=1

i=3 j=2

i=3 j=3

i=3 j=4

i=4 j=1

i=4 j=2

i=4 j=3

i=4 j=4

Contoh 7 mengecek bilangan genap dan ganjil sekaligus mengcounter banyak bilangan genap dan ganjil

```
main()
{
    int n,i,jumlahgenap,jumlahganjil;
    clrscr();
    printf("banyak Perulangan =");scanf("%d",&n);
    jumlahgenap=0;
    jumlahganjil=0;
    for (i=1; i<=n; i++)
    {
        if (i % 2==0)
        {
            printf("Bilangan %d adalah genap\n",i);
            jumlahgenap=jumlahgenap+1;
        }
        else
        {
            printf("Bilangan %d adalah ganjil\n",i);
            jumlahganjil=jumlahganjil+1;
        }
    }
    printf("Jumlah bilangan Genap Ada=%d\n",jumlahgenap);
    printf("Jumlah bilangan Ganjil Ada=%d\n",jumlahganjil);
    getch();
}
```

Run

banyak Perulangan =1

Bilangan 1 adalah ganjil

Jumlah bilangan Genap Ada=0

Jumlah bilangan Ganjil Ada=1

banyak Perulangan =2

Bilangan 1 adalah ganjil

Bilangan 2 adalah genap

Jumlah bilangan Genap Ada=1

Jumlah bilangan Ganjil Ada=1

banyak Perulangan =3

Bilangan 1 adalah ganjil

Bilangan 2 adalah genap

Bilangan 3 adalah ganjil

Jumlah bilangan Genap Ada=1

Jumlah bilangan Ganjil Ada=2

banyak Perulangan =5

Bilangan 1 adalah ganjil

Bilangan 2 adalah genap

Bilangan 3 adalah ganjil

Bilangan 4 adalah genap

Bilangan 5 adalah ganjil

Jumlah bilangan Genap Ada=2

Jumlah bilangan Ganjil Ada=3

Soal

1. Buatlah program cek bilangan ganjil atau genap dengan perulangan sebanyak n kali untuk
 - a. Tak terkendali dengan while
 - b. Terkendali dengan for
 - c. Jumlahkan bilangan ganjil tersebut $1+3+5+7 = 16$
 - d. Jumlahkan bilangan genap tersebut $2+4+6+8 = 20$

2. Buatlah program untuk konversi kurs mata uang sebagai berikut :

Banyaknya Nilai Awal = 1
 Banyaknya Nilai Akhir = 5
 TENTUKAN DALAM 1 DOLAR =
 Berapa Rupiah ? = 10000
 Berapa Pound ? = 9000
 Berapa Ringgit? = 8000

	DOLAR	RUPIAH	POUND	RINGGIT
1	10000	9000	8000	
2	20000	18000	16000	
3	30000	27000	24000	
4	40000	36000	32000	
5	50000	45000	40000	

3. Buatlah konversi bilangan decimal ke dalam bilangan biner sebagai berikut :

Masukkan Angka = 100
 $100 : 2 = 50$ sisa = 0
 $50 : 2 = 25$ sisa = 0
 $25 : 2 = 12$ sisa = 1
 $12 : 2 = 6$ sisa = 0
 $6 : 2 = 3$ sisa = 0
 $3 : 2 = 1$ sisa = 1
 $1 : 2 = 0$ sisa = 1

4. Buatlah program untuk mengecek bilangan prima dengan menginputkan sebanyak n kali sebagai berikut :

Banyaknya Bilangan = 25
 2 3 5 7 11 13 17 19 23
 banyaknya Bilangan = 50
 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47

5. Buatlah program untuk membuat kotak dengan memasukkan n kali sebagai berikut :

Masukkan n=5

* *

* *

* *

Masukkan n=10

* *

* *

* *

* *

* *

* *

* *

* *

6. Buatlah program untuk konversi suhu dengan standar Celcius
Sebagai berikut :

Suhu Celcius = 10

Maka Kelvin, Reamur dan Fahrenheit tampil di layar.

BAB V

LARIK

Target :

Mengetahui dan memahami penggunaan array/larik. Memahami array di dalam program dan di dalam perulangan. Manipulasi array untuk sortir data.

Materi :

- ✓ Pengertian
- ✓ Statement array
- ✓ statement array di dalam perulangan
- ✓ Array satu dimensi
- ✓ Array dua dimensi

Pengertian

Array /Larik adalah sekumpulan type data yang sejenis pada suatu variable tertentu.

SYNTAX

Type data Variabel[nilaiangkaarray]

Contoh :

Type data integer

```
int nilai[10];
```

Mengapa harus Array ?

Contoh 1 tidak menggunakan array

```
#include<stdio.h>
main()
{
    int i,n,nilai;
    char nama;
    clrscr();
    printf("masukkan banyaknya n=");scanf("%d",&n);
    for (i=1; i<=n; i++)
    {
        fflush(stdin);
        printf("Masukkan Nama =");gets(nama);
        printf("Masukkan nilai=");scanf("%d",&nilai);
    }
}
```



```

for (i=1; i<=n; i++)
{
    printf("Nama Anda adalah =%s\n",nama);
    printf("Nilai Anda adalah=%d\n",nilai);
}
getch();
}

```

Run

masukkan banyaknya n=2

Masukkan Nama =heri

Masukkan nilai=90

Masukkan Nama =yanto

Masukkan nilai=70

Nama Anda adalah =yanto

Nilai Anda adalah=70

Nama Anda adalah =yanto

Nilai Anda adalah=70

Contoh 2 Menggunakan array satu dimensi

```

#include<stdio.h>
main()
{
    int i,n,nilai[100];
    char nama[100];
    clrscr();
    printf("masukkan banyaknya n=");scanf("%d",&n);
    for (i=1; i<=n; i++)
    {
        fflush(stdin);
        printf("Masukkan Nama =");gets(nama[i]);
        printf("Masukkan nilai=");scanf("%d",&nilai[i]);
    }
    for (i=1; i<=n; i++)
    {
        printf("Nama Anda adalah =%5s\n",nama[i]);
        printf("Nilai Anda adalah=%5d\n",nilai[i]);
    }
    getch();
}

```

Run

masukkan banyaknya n=3

Masukkan Nama =oke

Masukkan nilai=90

Masukkan Nama =and
 Masukkan nilai=80
 Masukkan Nama =moy
 Masukkan nilai=70

Nama Anda adalah = oke
 Nilai Anda adalah= 90
 Nama Anda adalah = and
 Nilai Anda adalah= 80
 Nama Anda adalah = moy
 Nilai Anda adalah= 70

Contoh 3 menggunakan array input langsung

```
#include <stdio.h>
#define banyaknya_nilai 9
main()
{
  static int nilai[]={56, 78, 43, 96, 67, 83, 51, 74, 32};
  int a,b;
  clrscr();
  printf("Nilai Ujian Array \n");
  for (a=0;a<banyaknya_nilai;a++)
  {
    printf("%d \n",nilai[a]);
  }
  getch();
}
```

Run

Nilai Ujian Array

56
 78
 43
 96
 67
 83
 51
 74
 32

Contoh 4 menggunakan array untuk nilai ranking

```

/*Program: Rangking Nilai Ujian*/
#include <stdio.h>
#define banyaknya_nilai 9
main()
{
    static int nilai[]={56, 78, 43, 96, 67, 83, 51, 74, 32};
    int rangking[banyaknya_nilai];
    int a,b;
    clrscr();
    for (a=0;a<banyaknya_nilai;a++)
    {
        rangking[a]=1;
        for (b=0;b<banyaknya_nilai;b++)
        {
            if (nilai[a]>nilai[b])
                rangking[a]++;
        }
    }

    printf("Nilai Ujian  Ranking Tekecil ke Besar\n");
    for (a=0;a<banyaknya_nilai;a++)
    {
        printf("%d  %d\n",nilai[a], rangking[a]);
    }
    getch();
}
RUN
Nilai Ujian  Ranking Tekecil ke Besar
56 4
78 7
43 2
96 9
67 5
83 8
51 3
74 6
32 1

```

Contoh 5 array untuk type data character

```

#include <stdio.h>
#define banyaknya_nilai 6
main()
{
    static char nama[]={ 'B','A','C','E','D','F' };
    int a,b;
    clrscr();

```

```

for (a=0;a<banyaknya_nilai;a++)
{
    printf("%5c \n",nama[a]);
}
getch();
}
Run
B
A
C
E
D
F

```

Contoh 6 array untuk type data long char

```

#include <stdio.h>
#define banyaknya_nilai 6
main()
{
    static char *nama[]={ "Bayu","Ahmad","Cici","Eli","Doyo","Farid" };
    int a,b;
    clrscr();
    for (a=0;a<banyaknya_nilai;a++)
    {
        printf("%10s \n",nama[a]);
    }
    getch();
}
run
    Bayu
    Ahmad
    Cici
    Eli
    Doyo
    Farid

```

Contoh 7 Penggunaan array 2 dimensi

```

/*penggunaan array 2 dimensi */
#include <stdio.h>
main()
{
    int kernel[20][10];
    float rata;
    int t,i,j,jum,n,jumlah,hasil_rata;
    clrscr();

```

```

hasil_rata=0;
printf("Banyak Data = ");scanf("%d",&n);
for (i=0;i<n;i++)
{
    clrscr();
    printf("Data Ke % d = \n",i+1);
    puts("=====");
    printf("Nomor Mhs = ");scanf("%4d",&kernel[i][0]);
    jum=0;
    for (j=1;j<4;j++)
    {
        printf("Nilai ke % d : ",j);scanf("%4d",&kernel[i][j]);
        jum=jum+kernel[i][j];
    }
    rata=jum/3;
    printf("=====+ \n");
    printf("Jumlah      = %4d\n",jum);
    kernel[i][4]=rata;
    printf("Rata-rata    = %4d\n",kernel[i][4]);
    kernel[i][j]=jum;
    getch();
}

printf("-----\n");
printf("No.Mhs   Basic   Pascal   Turbo C   Jumlah Rata2\n");
printf("      1      2      3          \n");
printf("\n-----");

for (i=0;i<n;i++)
{
    printf(" \n");
    for (j=0;j<=4;j++)
    {
        printf("%d      ",kernel[i][j]);
    }
    printf("%d ",kernel[i][4]/3);
    hasil_rata=hasil_rata+(kernel[i][4]/3);
}
printf("\n=====");
getch();
}

```

Run

Banyak Data = 1

Data Ke 1 =

=====

Nomor Mhs = 2

Nilai ke 1 : 45

Nilai ke 2 : 67

Nilai ke 3 : 44

=====+

Jumlah = 156

Rata-rata = 52

No.Mhs	Basic	Pascal	Turbo C	Jumlah	Rata2
1	2	3			
2	45	67	44	156	52

2 45 67 44 156 52

=====

Contoh 8 Manipulasi array untuk sortir data

```
#include <stdio.h>
#define banyaknya_nilai 9
main()
{
    static int nilai[]={56, 78, 43, 96, 67, 83, 51, 74, 32};
    int i,j,tukar;
    clrscr();
    printf("Nilai Ujian sebelum Sortir\n");
    for (i=0;i<banyaknya_nilai;i++)
    {
        printf("%d \n",nilai[i]);
    }
    printf("Nilai Ujian setelah Sortir\n");
    for (i=0;i<banyaknya_nilai;i++)
    {
        for (j=0;j<banyaknya_nilai;j++)
        {
            if (nilai[i] > nilai[j])
            {
                tukar=nilai[i];
                nilai[i]=nilai[j];
                nilai[j]=tukar;
            }
        }
    }
    for (i=0;i<banyaknya_nilai;i++)
```

```
{
    printf("%d \n",nilai[i]);
}
getch();
}
```

Run

Nilai Ujian sebelum Sortir

56
78
43
96
67
83
51
74
32

Nilai Ujian setelah Sortir

96
83
78
74
67
56
51
43
32

Contoh 9 sortir data dengan type data char

```
#include <stdio.h>
#define banyaknya 9
main()
{
    static char nama[]={ 'd','w','s','c','a','b','r','g','t'};
    static char tukar1;

    int i,j;
    clrscr();
    printf("Nilai Ujian sebelum Sortir\n");
    for (i=0;i<banyaknya;i++)
    {
        printf("%c \n",nama[i]);
    }
    printf("\nNilai Ujian setelah Sortir\n");
    for (i=0;i<banyaknya;i++)
    {
```

```

for (j=0;j<banyaknya;j++)
{
    if (nama[i] > nama[j])
    {
        tukar1=nama[i];
        nama[i]=nama[j];
        nama[j]=tukar1;
    }
}

```

```

}

for (i=0;i<banyaknya;i++)
{
    printf("%c \n",nama[i]);
}
getch();
}

```

Run

Nilai Ujian sebelum Sortir

d
w
s
c
a
b
r
g
t

Nilai Ujian setelah Sortir

w
t
s
r
g
d
c
b
a

Latihan

1. Buatlah program dengan array untuk sortir data sebagai berikut :

Masukkan banyak data : 3

Nomor mahasiswa : A1

Nilai Mahasiswa : 80

Nomor mahasiswa : C2

Nilai Mahasiswa : 60

Data disortir berdasarkan nilai mahasiswa

No.Mhs	Nilai Mahasiswa

C2	60
A1	80

BAB VI

PROCEDURE

Target :

Mengetahui penggunaan procedure, memahami fungsi parameter di dalam procedure, memecahkan masalah dengan *procedure*.

Materi :

- ✓ Statement procedure (void)
- ✓ Statement procedure berparameter
- ✓ Statement tidak berparameter

STATEMEN PROCEDURE

Procedure adalah suatu kumpulan instruksi yang dibuat oleh program sebagai pengikat dalam suatu blok program tersendiri/sekelompok program. Tujuan dibuat procedure untuk memudahkan dalam pemrograman yang sangat komplek. Prosedure sendiri dapat berfungsi sebagai pemanggil.

Procedure terdiri dari dua kelompok :

1. Procedure tanpa parameter

Procedure yang tidak mempunyai nilai dan sebagai pemanggil saja

Syntak Procedure tanpa parameter :

void namaprocedure():

Syntak pemanggilan procedure tanpa parameter :

namaprocedure()

2. Procedure berparameter

Procedure yang tidak hanya sebagai pamanggil saja tapi ada nilai variabel yang dibawa.

Syntak Procedure Berparameter :

void namaprocedure(variabel1):

atau

void namaprocedure(type data variabel1):

Syntak Pemanggilan Procedure Berparameter :

namaprocedure(variabel1)

contoh 1 procedure tanpa parameter

```

/* penggunaan procedure */
#include <stdio.h>
void hello()
{
    puts("STTI RESPATI");
}
main()
{
    clrscr(); hello(); hello(); hello();
    getch();
}

```

```

run
STTI RESPATI
STTI RESPATI
STTI RESPATI

```

Contoh 2 procedure Berparameter void namaprocedure(variable)

```

/* penggunaan procedure */
#include <stdio.h>
void hello(n)
{ int i;
  for(i=1;i<=n;i++)
  {
      printf("STTI RESPATI ke-%d\n",i);
  }
}
main()
{
    clrscr(); hello(5);
    getch();
}

```

```

Run
STTI RESPATI ke-1
STTI RESPATI ke-2
STTI RESPATI ke-3
STTI RESPATI ke-4
STTI RESPATI ke-5

```

Contoh 3 Procedure berparameter void namaprocedure(type data variabel)

```
/* penggunaan procedure */
#include <stdio.h>
void hello(int n)
{ int i;
  for(i=1;i<=n;i++)
  {
    printf("STTI RESPATI ke-%d\n",i);
  }
}
main()
{
  clrscr(); hello(5);
  getch();
}
```

Latihan

1. Buatlah program untuk menggunakan procedure tanpa parameter sebagai berikut :
Gunakan array dan procedure tanpa parameter.

- a. Masukkan Data
- b. Sortir Data berdasarkan nomor
- c. Sortir Data berdasarkan nilai mhs
- d. Lihat Data

Pilihan Anda = a

Masukkan Data = 3

Nomor mahasiswa = 10

Nilai mhs = 90

Nomor mahasiswa = 13

Nilai mhs = 70

Nomor mahasiswa = 3

Nilai mhs = 55

2. Buatlah program menggunakan procedure berparameter sebagai berikut :

Masukkan banyak barang : 3

Kode Barang = 201

Harga barang=600

Banyak barang=3

Kode Barang=301

Harga Barang=500

Banyak barang=2

Kode Barang=101

Harga Barang=400

Banyak Barang=4

Data tersortir berdasarkan kode

Kode Barang	Harga Barang	Banyak Barang	total
101	400	4	1600
201	600	3	1800
301	500	2	1000
	----- +		
total	XXX	XXXX	XXX

Data tersortir berdasarkan harga barang

Kode Barang	Harga Barang	Banyak Barang	total
101	400	4	1600
301	500	2	1000
201	600	3	1800
	----- +		
total	XXX	XXXX	XXX

BAB VII

RECORD

Target :

Mengetahui penggunaan record dalam data terstruktur, memahami fungsi memecahkan masalah dengan record. Dapat memakai program dengan struktru record.

Materi :

- ✓ Statement record
- ✓ Statement record dengan procedure

STATEMENT RECORD

Record merupakan kumpulan data yang tidak sejenis atau variasi. File merupakan berkas kumpulan dari beberapa record yang tersimpan dan mempunyai nama. Dalam bahasa c untuk penulisan record dikenal dengan nama struct.

Record

Penulisan Record :

<pre>struct nama_type_record { char nama[15],nomhs[5]; int nilai; } variable_record;</pre>	atau	<pre>struct nama_type_record { } variable_record[10]</pre>
----------------------------------------------------------------------------------------------	------	------------------------------------------------------------

contoh 1 penggunaan record /struktur

```
/* STRUCT */
#include <stdio.h>

struct data_mahasiswa
{ char nama[10],nomhs[10];
  int nilai;
} daftarmhs[10];

int j,i,n,menu;
char nilaihuruf[15];
void isi_data()
{
  j=0;
  clrscr();
```

```

textcolor(0);textbackground(10);
printf("Banyak data : ");scanf("%d",&n);
for(i=1;i<=n;i++)
{
j++;
printf("\nNo. [%d]\n",j);fflush(stdin);
printf("\nNomor Mahasiswa : ");gets(daftarmhs[i].nomhs);
printf("\nNama Mahasiswa : ");gets(daftarmhs[i].nama);
printf("\nNilai angka : ");scanf("%d",&daftarmhs[i].nilai);
}
}

void cetak_data()
{
clrscr();
printf("\n=====
=====\n");
printf(" No      Nomhs      N a m a      Nilai      HURUF\n");
printf("-----\n");
i=0;
i++;
for(i=1;i<=n;i++)
{
if ((daftarmhs[i].nilai >=90) && (daftarmhs[i].nilai<=100))
{
nilaihuruf[i]='A';
}
if ((daftarmhs[i].nilai >=70) && (daftarmhs[i].nilai<=89))
{
nilaihuruf[i]='B';
}
if ((daftarmhs[i].nilai >=60) && (daftarmhs[i].nilai<=70))
{
nilaihuruf[i]='C';
}
if ((daftarmhs[i].nilai >=50) && (daftarmhs[i].nilai<=59))
{
nilaihuruf[i]='D';
}
if (daftarmhs[i].nilai <49)
{
nilaihuruf[i]='E';
}
fflush(stdin);
printf("%d      %-10s      %-20s      %4d      %4c
\r\n",i,daftarmhs[i].nomhs,daftarmhs[i].nama,daftarmhs[i].nilai,nilaihuruf[i]);
printf("-----\n");
}
}

```

```

printf("Tekan sembarang tombol");getch();
clrscr();
}

main()
{
do
{clrscr();
gotoxy(30,7);printf("PROGRAM OLAH DATA MAHASISWA");
gotoxy(30,8);printf("Menu Pilihan");
gotoxy(27,10);printf("1. Memasukkan Data");
gotoxy(27,11);printf("2. Melihat Data Mahasiswa");
gotoxy(27,12);printf("3. Selesai");
gotoxy(27,14);printf("Pilihan anda : ");
gotoxy(43,14);scanf("%d",&menu);
switch (menu)
{
case 1: isi_data(daftarmhs);break;
case 2: cetak_data(daftarmhs);break;
case 3: {
clrscr();
printf("TERIMA KASIH TELAH MEMAKAI PROGRAM
INI\n");break;
}
}
}
while (menu!=3);
}

```

run

PROGRAM OLAH DATA MAHASISWA

Menu Pilihan

1. Memasukkan Data
2. Melihat Data Mahasiswa
3. Selesai

Pilihan anda :1

Banyak data : 2

No. [1]

Nomor Mahasiswa : 001

Nama Mahasiswa : agus

Nilai angka : 90

No. [2]

Nomor Mahasiswa : 002

Nama Mahasiswa : tarman

Nilai angka : 76

PROGRAM OLAH DATA MAHASISWA Menu Pilihan

1. Memasukkan Data
2. Melihat Data Mahasiswa
3. Selesai

Pilihan anda :2

=====				
No	Nomhs	N a m a	Nilai	HURUF

1	001	agus	90	A

2	002	tarman	76	B

Tekan sembarang tombol

Contoh 2 penggunaan struktur dengan menggunakan fungsi berparameter

```
/* penggunaan struct */
/* pada void berparameter */
#include <stdio.h>
#include <conio.h>
struct data_pegawai
{
    char nip[20],nama[20];
    long int gaji_pokok;
}
daftar[20];

void isi_data(struct data_pegawai daftar[20]);
void cetak_data(struct data_pegawai daftar[20]);

int i,n,jar=4,menu;
```

```

main()
{
do
{clrscr();
gotoxy(30,7);printf("PROGRAM OLAH DATA KARYAWAN");
gotoxy(30,8);printf("Menu Pilihan");
gotoxy(27,10);printf("1. Memasukkan Data");
gotoxy(27,11);printf("2. Melihat Data Karyawan");
gotoxy(27,12);printf("3. Selesai");
gotoxy(27,14);printf("Pilihan anda : ");
gotoxy(43,14);scanf("%d",&menu);
switch (menu)
{
case 1: isi_data(daftar);break;
case 2: cetak_data(daftar);break;
case 3: {
clrscr();
printf("TERIMA KASIH TELAH MEMAKAI PROGRAM
INI\n");break;
}
}
}
while (menu!=3);
}

void isi_data(struct data_pegawai daftar[20])
{ clrscr();
printf("Banyaknya data=");scanf("%d",&n);
for(i=1;i<=n;i++)
{
fflush(stdin);
printf("NIP = ");gets(daftar[i].nip);
printf("Gaji Pegawai = ");scanf("%ld",&daftar[i].gaji_pokok);
}
}

void cetak_data(struct data_pegawai daftar[20])
{
clrscr();
puts("DAFTAR GAJI PEGAWAI");
puts("=====");
puts("NIP Gaji ");
puts("=====");
for(i=1;i<=n;i++)
{
fflush(stdin);
gotoxy(2,i+jar);printf("%d",i);

```

```

gotoxy(5,i+jar);printf("%-6s",daftar[i].nip);
gotoxy(13,i+jar);printf("%ld",daftar[i].gaji_pokok);
}
printf("\n-----\n");
getch();
}

```

RUN

PROGRAM OLAH DATA KARYAWAN

Menu Pilihan

1. Memasukkan Data
2. Melihat Data Karyawan
3. Selesai

Pilihan anda : 1

Banyaknya data=3

NIP = 001

Gaji Pegawai = 1000000

NIP = 002

Gaji Pegawai = 2000000

NIP = 003

Gaji Pegawai = 3000000

PROGRAM OLAH DATA KARYAWAN

Menu Pilihan

1. Memasukkan Data
2. Melihat Data Karyawan
3. Selesai

Pilihan anda : 2

DAFTAR GAJI PEGAWAI

=====

NIP GAJI

=====

1 001 1000000

2 002 2000000

3 003 3000000

Contoh 3 penggunaan struktur dengan void sortir data record

```

/*program sortir data dengan record*/
#include<stdio.h>

struct mahasiswa
{ int nomor[100];
  int nilai[100];
} datamhs[100];

int n;

void isimhs()
{
  int i;
  for(i=1;i<=n;i++)
  {
    printf("Masukkan Nomor Mhs =");scanf("%d",&datamhs[i].nomor[i]);
    printf("Masukkan Nilai   =");scanf("%d",&datamhs[i].nilai[i]);
  }
}

void sortmhs()
{
  int i,j, tukar1, tukar2;
  for(i=1;i<=n;i++)
  {
    for(j=1;j<=n;j++)
    {
      if(datamhs[i].nilai[i] < datamhs[j].nilai[j])
      {
        tukar1=datamhs[i].nomor[i];
        tukar2=datamhs[i].nilai[i];
        datamhs[i].nomor[i]=datamhs[j].nomor[j];
        datamhs[i].nilai[i]=datamhs[j].nilai[j];
        datamhs[j].nomor[j]=tukar1;
        datamhs[j].nilai[j]=tukar2;
      }
    }
  }
  printf("Nomor      nilai\n");
  for(i=1;i<=n;i++)
  {
    printf(" %d    %d \n",datamhs[i].nomor[i],datamhs[i].nilai[i]);
  }
}

```

```

main()
{ clrscr();
  printf("Masukkan N =");scanf("%d",&n);
  isimhs(datamhs);
  sortmhs(datamhs);
  getch();
}

```

RUN (CTRL+F9)

```

Masukkan N =3
Masukkan Nomor Mhs =1
Masukkan Nilai    =78
Masukkan Nomor Mhs =2
Masukkan Nilai    =67
Masukkan Nomor Mhs =3
Masukkan Nilai    =45
Nomor    nilai
3    45
2    67
1    78

```

LATIHAN

1. Buatlah program untuk dengan record menggunakan struct dan procedure sebagai berikut : (gunakan sortir data sehingga nama urut)

Banyaknya n :

DAFTAR GAJI PEGAWAI

=====

NIP Nama GAJI

=====

1 001	Beni	1000000
2 002	Arman	2000000
3 003	Cahyo	3000000

total :

BAB VIII

FILE

Target :

Mengetahui penggunaan operasi file dan berkas. Memahami fungsi membuat file, memasukkan data tersimpan di dalam file

Materi :

- ✓ Statement membuat file baru
- ✓ Statement membaca file
- ✓ Statement mengisi data di dalam file dan melihat data di file

Fungsi Input dan Output

Perintah `#include` merupakan direktif praprosesor yang menginstruksikan kompilasi C untuk membaca isi dari sebuah file/menyertakan/including perintah di dalam file .H tersebut.

Bentuk umum :

```
#include <filename.h>
```

keterangan :

1. setiap program yang menggunakan fungsi pustaka I/O input output standart harus memiliki pernyataan `#include <stdio.h>`
2. program yang menggunakan Microsoft c compiler misalnya fungsi `isupper(C)` dan `tolower(c)` didefinisikan dulu pada `#include <ctype.h>` dan masih banyak file include yang lain berextension *.h pada compiler bahasa C. uniknya praprocessor c dapat di buka layaknya file text biasa.

OPERASI FILE BUFFERED FILE

Buffered file digunakan untuk memindahkan data dari dan ke disk. Fungsi-fungsi di bawah ini dirancang dalam file yang berupa teks dan berkaitan dengan buffer file adalah :

fopen()	getc()	fscanf()	cputs()
fclose()	ungetc()	fgets()	fread()
fflush()	putc()	fputs()	

MEMBACA FILE :

Bentuk Umum : SYNTAX : FILE *pf (dimana pf adalah pointer file)

FUNGSI BUKA FILE :

Bentuk Umum : SYNTAX : fopen(filename,mode)

Filename adalah nama file yang terdapat pada penyimpanan disk/harddisk

Mode : menentukan apa yang akan dilakukan pada file :

r adalah (read) untuk membaca file

w adalah (write) untuk menulis file ke disk/simpan

a adalah (append) untuk menambahkan data atau add.

FUNGSI fread/fwrite

Fungsi ini digunakan untuk membaca/tulis block data dari ke suatu file.

Bentuk Umum :

Nakses = fread(p, s, n, pf)

Nakses = fwrite(p, s, n, pf)

Dimana

Nakses adalah jumlah blok yang sebenarnya di akses.

p adalah pointer ke blok data pertama

s adalah ukuran tiap blok dalam byte

n adalah jumlah blok yang diakses

pf adalah pointer file

FUNGSI TUTUP FILE

Bentuk Umum : fclose(pf)

Ket: pf adalah pointer file

Contoh 1. Penggunaan File

```

/* FILE dan STRUCT */
#include <stdio.h>

FILE *mahasiswa;
struct data_mahasiswa
{ char nama[15],nomhs[5];
  char nilaih;
} daftarmhs;

int j,i,m,n,rek,menu;
int bobot[5];
void isi_data()
{
  j=0;
  switch (m)
  { case 0:
    if ((mahasiswa=fopen("c:mhs1.dat","wb"))==NULL)
    { cputs("File tak dapat dibuat..!\r\n");getch();
      exit(1);
    } break;
  case 1:
    if ((mahasiswa=fopen("c:mhs1.dat","rb"))==NULL)
    { cputs("File belum dibuat..!\r\n");getch();
      exit(1);
    }
  }
  clrscr();
  textcolor(0);textbackground(10);
  printf("Banyak data : ");scanf("%d",&n);
  for(i=1;i<=n;i++)
  {j++;clrscr();
  printf("\nNo. [%d]\n",j);fflush(stdin);
  printf("\nNomor Mahasiswa : ");gets(daftarmhs.nomhs);
  printf("\nNama Mahasiswa : ");gets(daftarmhs.nama);
  printf("\nNilai Huruf : ");scanf("%c",&daftarmhs.nilaih);
  fwrite(&daftarmhs,sizeof(daftarmhs),1,mahasiswa);
  }
  fclose(mahasiswa);
}

void cetak_data()
{ if ((mahasiswa=fopen("c:mhs1.dat","rb"))==NULL)
  { cputs("File tak dapat dibaca..!\r\n");getch();
    exit(1);
  }
}

```



```

clrscr();
printf("\n=====
===\n");
printf(" No      Nomhs      NAMA  Nilai huruf      BOBOT\n");
printf("-----\n");
i=0;
while(fread(&daftarmhs,sizeof(daftarmhs),1,mahasiswa)==1)
{
i++;
if (daftarmhs.nilaih=='A')
{
bobot[i]=4;
}
if (daftarmhs.nilaih=='B')
{
bobot[i]=3;
}
if (daftarmhs.nilaih=='C')
{
bobot[i]=2;
}
if (daftarmhs.nilaih=='D')
{
bobot[i]=1;
}
cprintf("%3d %3s %10s %10c %10d
\r\n",i,daftarmhs.nomhs,daftarmhs.nama,daftarmhs.nilaih,bobot[i]);
}
printf("-----\n");
fclose(mahasiswa);
printf("Tekan sembarang tombol\n");getch();
clrscr();
}

main()
{
do
{clrscr();
gotoxy(30,7);printf("PROGRAM OLAH DATA MAHASISWA");
gotoxy(30,8);printf("Menu Pilihan");
gotoxy(27,10);printf("1. Memasukkan Data");
gotoxy(27,11);printf("2. Melihat Data Mahasiswa");
gotoxy(27,12);printf("3. Selesai");
gotoxy(27,14);printf("Pilihan anda : ");
gotoxy(43,14);scanf("%d",&menu);
switch (menu)
{

```

```
case 1: isi_data(daftarmhs);break;
case 2: cetak_data(daftarmhs);break;
case 3: {
        clrscr();
        printf("TERIMA KASIH TELAH MEMAKAI PROGRAM INI\n");break;
    }
}
while (menu!=3);
}
```

Run

PROGRAM OLAH DATA MAHASISWA

Menu Pilihan

1. Memasukkan Data
2. Melihat Data Mahasiswa
3. Selesai

Pilihan anda :1

Banyak data : 3

No. [1]

Nomor Mahasiswa : 001

Nama Mahasiswa : agus

Nilai Huruf : A

No. [2]

Nomor Mahasiswa : 002

Nama Mahasiswa : BAYU

Nilai Huruf : C

No. [3]

Nomor Mahasiswa : 003

Nama Mahasiswa : CICI

Nilai Huruf : B

PROGRAM OLAH DATA MAHASISWA

Menu Pilihan

1. Memasukkan Data
2. Melihat Data Mahasiswa
3. Selesai

Pilihan anda :2

No	Nomhs	NAMA	Nilai huruf	BOBOT
1	001	agus	A	4
2	002	BAYU	C	2
3	003	CICI	B	3

Tekan sembarang tombol

Contoh 2 penggunaa file

```

/* FILE dan STRUCT */
#include <stdio.h>

FILE *mahasiswa;
struct data_mahasiswa
{ char nama[15],nomhs[5];
  int nilai;
} daftarmhs;

int j,i,m,n,rek,menu;
char nilaihuruf[15];
void isi_data()
{
  j=0;
  switch (m)
  { case 0:
    if ((mahasiswa=fopen("c:mhs.dat","wb"))==NULL)
    { cputs("File tak dapat dibuat..!\r\n");getch();
      exit(1);
    } break;
    case 1:
    if ((mahasiswa=fopen("c:mhs.dat","rb"))==NULL)
    { cputs("File belum dibuat..!\r\n");getch();
      exit(1);
    }
  }
  clrscr();
  textcolor(0);textbackground(10);
  printf("Banyak data : ");scanf("%d",&n);
  for(i=1;i<=n;i++)
  { j++;
    printf("\nNo. [%d]\n",j);fflush(stdin);
    printf("\nNomor Mahasiswa : ");gets(daftarmhs.nomhs);
  }
}

```

```

printf("\nNama Mahasiswa : ");gets(daftarmhs.nama);
printf("\nNilai angka   : ");scanf("%d",&daftarmhs.nilai);
fwrite(&daftarmhs,sizeof(daftarmhs),1,mahasiswa);
}
fclose(mahasiswa);
}

void cetak_data()
{ if ((mahasiswa=fopen("c:mhs.dat","rb"))==NULL)
  { cputs("File tak dapat dibaca..!\r\n");getch();
    exit(1);
  }
  clrscr();
  printf("\n=====
=====\\n");
  printf(" No      Nomhs      N a m a      Nilai      HURUF\\n");
  printf("-----\\n");
  i=0;
  while(fread(&daftarmhs,sizeof(daftarmhs),1,mahasiswa)==1)
  {
    i++;
    if ((daftarmhs.nilai >=90) && (daftarmhs.nilai<=100))
    {
      nilaihuruf[i]='A';
    }
    if ((daftarmhs.nilai >=70) && (daftarmhs.nilai<=89))
    {
      nilaihuruf[i]='B';
    }
    if ((daftarmhs.nilai >=60) && (daftarmhs.nilai<=70))
    {
      nilaihuruf[i]='C';
    }
    if ((daftarmhs.nilai >=50) && (daftarmhs.nilai<=59))
    {
      nilaihuruf[i]='D';
    }
    if (daftarmhs.nilai <49)
    {
      nilaihuruf[i]='E';
    }
    cprintf("%3d   %-10s   %-20s   %4d   %4c \\r\\n",i,daftarmhs.nomhs,daftarmhs.nama,
            daftarmhs.nilai,nilaihuruf[i]);
  }
  printf("-----\\n");
  fclose(mahasiswa);
  printf("Tekan sembarang tombol");getch();
}

```

```

clrscr();
}

main()
{
do
{clrscr();
gotoxy(30,7);printf("PROGRAM OLAH DATA MAHASISWA");
gotoxy(30,8);printf("Menu Pilihan");
gotoxy(27,10);printf("1. Memasukkan Data");
gotoxy(27,11);printf("2. Melihat Data Mahasiswa");
gotoxy(27,12);printf("3. Selesai");
gotoxy(27,14);printf("Pilihan anda : ");
gotoxy(43,14);scanf("%d",&menu);
switch (menu)
{
case 1: isi_data(daftarmhs);break;
case 2: cetak_data(daftarmhs);break;
case 3: {
clrscr();
printf("TERIMA KASIH TELAH MEMAKAI PROGRAM INI\n");break;
}
}
}
while (menu!=3);
}

```

Run

PROGRAM OLAH DATA MAHASISWA
Menu Pilihan

1. Memasukkan Data
2. Melihat Data Mahasiswa
3. Selesai

Pilihan anda :1

Banyak data : 3

No. [1]

Nomor Mahasiswa : 001

Nama Mahasiswa : adi

Nilai angka : 60

No. [2]

Nomor Mahasiswa : 022

Nama Mahasiswa : saputra
 Nilai angka : 70

No. [3]
 Nomor Mahasiswa : 023
 Nama Mahasiswa : herman
 Nilai angka : 90

PROGRAM OLAH DATA MAHASISWA

Menu Pilihan

1. Memasukkan Data
2. Melihat Data Mahasiswa
3. Selesai

Pilihan anda :2

No	Nomhs	N a m a	Nilai	HURUF
1	001	adi	60	C
2	022	saputra	70	C
3	023	herman	90	A

Tekan sembarang tombol

Latihan :

1. Buatlah program mensortir data yang tersimpan di file sebagai berikut :
 Data sebelum disortir

No	Nomhs	N a m a	Nilai	HURUF
1	001	adi	60	C
2	022	saputra	70	C
3	023	herman	90	A

Data setelah disortir

No	Nomhs	N a m a	Nilai	HURUF
1	001	adi	60	C
2	023	herman	90	A
3	022	saputra	70	C