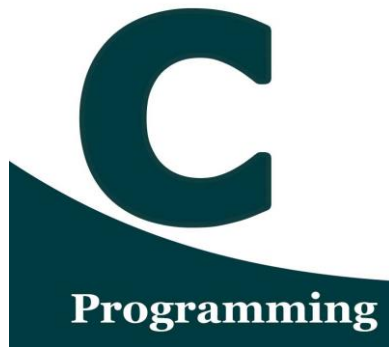


Lab Pemograman Dasar

---

**Oleh :** Tono Hartono, S.Si.,MT

(Digunakan di lingkungan sendiri, sebagai buku ajar  
mata kuliah Lab Pemograman Dasar)



**Fakultas Teknik dan Ilmu Komputer**  
**Program Studi Sistem Informasi**  
**Universitas Komputer Indonesia**

## 1. Pertemuan 1

### 1.1 Perkenalan Matakuliah

#### a. Deskripsi Matakuliah

Matakuliah ini membahas tentang bagaimana mengimplementasikan dasar-dasar algoritma dan pemrograman ke bahasa pemrograman C

Penjelasan aturan perkuliahan, komponen penilaian, silabus dan referensi buku.

### 1.2 Aturan Perkuliahan

- a. Perkuliahan diselenggarakan 14 kali pertemuan (2 SKS)
- b. Wajib kehadiran Mahasiswa 80% (-3 kali tidak masuk)
- c. Materi perkuliahan akan diberikan salinannya kepada Mahasiswa
- d. Mahasiswa dianjurkan membawa flashdisk
- e. Batas keterlambatan 15 menit setelah perkuliahan dimulai
- f. Mahasiswa diperbolehkan berkonsultasi dengan dosen; mengenai materi perkuliahan secara personal atau kelompok di luar jam perkuliahan (tatap muka; via email; kuliah online)
- g. Mengikuti tata tertib Lab
- h. Tidak diperbolehkan menggunakan perangkat komunikasi selama perkuliahan (setting silent/vibrate)
- i. Bersikap sopan dan tidak mengganggu keberlangsungan perkuliahan
- j. Tersedia waktu Shalat bagi yang beragama Islam.

### 1.3 Komponen Penilaian

- Tugas = 25%
- Absens = 10%
- UTS = 30%
- UAS = 50%

## 1.4 Silabus

Minggu ke	Kompetensi dasar	Materi dasar	Strategi perkuliahan	Sumber/referensi
1	Pendahuluan	Algoritma pemrograman, flowchart	praktik, penugasan	[1], [3]
2,3	Pengenalan struktur dasar Bahasa C	Struktur pemrograman, tipe data, syntax program	simulasi, praktik, penugasan	[1], [2], [3]
4,5	Penggunaan Operator Relasi dan Fungsi seleksi kondisi sederhana	Percabangan dalam pemrograman komputer	simulasi, praktik, penugasan	[1], [2], [3]
6,7	Fungsi Seleksi kondisi (IF dan Switch) bertingkat.	Perulangan dalam pemrograman komputer	Ceramah, simulasi, praktik, penugasan	[1], [2], [3]
8	Ujian Tengah Semester			

9	Review UTS	Fungsi dalam pemrograman	penugasan	[1], [2], [3]
10	Mampu membuat program larik dimensi 1	Larik dimensi 1	Ceramah, simulasi, praktik, penugasan	[1], [2], [3]
11,12	Mampu membuat program larik dimensi 2	Larik dimensi 1	Ceramah, simulasi, praktik, penugasan	[1], [2], [3]
13,14	Mampu membuat program pointer	Pointer dalam pemrograman komputer	Ceramah, simulasi, praktik, penugasan	[1], [2], [3]
15	Review Perkuliahan	Semua materi kuliah	Diskusi, tanya jawab, simulasi, praktik	[1], [2], [3]
16	Ujian Akhir Semester			

## 1.5 Pengertian Bahasa Pemrograman

Bahasa pemrograman adalah bahasa tertentu yang digunakan oleh para programmer untuk membuat suatu aplikasi atau software pada komputer, dasar-dasar yang dipakai adalah prinsip kerja algoritma yang sudah dipelajari sebelumnya, bahasa pemrograman itu sendiri sudah ditemukan pada tahun setelah diketemukannya mesin komputer, mulai generasi yang paling pertama kali yaitu kode mesin, sampai sampai generasi sekarang yang banyak digunakan sebagai bahasa pemrograman dalam pembuatan berbagai aplikasi, dalam konteksnya adalah setiap bahasa pemrograman mempunyai yang namanya kode, kode tersebut akan mendasari berbagai pemrosesan tiap step/langkah berdasarkan tipe data yang diinputkan oleh programmer, contohnya dalam bahasa pemrograman pascal adalah penggunaan tipe data integer untuk bilangan angka yang nantinya akan diproses dan penggunaan tipe data string untuk data-data yang terdiri dari susunan huruf dan nantinya tidak akan diproses dalam perhitungan algoritma dan akan dimunculkan (dicetak/write) pada akhir sequence.

## 2. Pertemuan 2

### 2.1 Fungsi #include

Fungsi #include adalah sebuah prosesor pengarah yang mengatakan kepada kompiler untuk meletakkan kode dari header file iostream.h kedalam program. Fungsi cout memerlukan file iostream.h.

### 2.2 Fungsi *start* dan *end* ( { } )

Fungsi start { pada fungsi main() menyatakan awal eksekusi program.

Dan fungsi end } pada fungsi main() menyatakan akhir eksekusi program.

### 2.3 Fungsi printf() dan fungsi scanf()

Fungsi printf() berfungsi untuk menampilkan keluaran data dan fungsi scanf() berguna untuk membaca masukkan data. Kedua fungsi ini terdapat pada header <stdio.h>

### 2.4 Cara penulisan komentar.

Ada 2 (dua) cara penulisan komentar dalam bahasa C/C++, anda dapat menggunakan salah satu dari kedua cara penulisan komentar tersebut.

Yang pertama garis miring dua "//", cara penulisan komentar ini disebut juga dengan komentar satu baris karena penggunaanya hanya untuk komentar pada satu baris saja. Yang kedua, diawali

dengan karakter "/\*" dan di akhiri dengan karakter "\*/", cara penulisan komentar ini disebut juga dengan blok komentar (penulisan komentar komentar bisa lebih dari satu baris).

Contoh :

Menggunakan komentar  
dalam bahasa C dan C++

```
#include<iostream>
using namespace std;
main()
{
    cout <<"Komentar satu baris"; //Komentar
satu baris ya

    /*Komentar dengan blok komentar
(Multiple line comments)*/
    cout << "Blok komentar (Multiple lines
comments)";

    /*Penulisan komentar
dengan selain kedua cara yang telah
saya jelaskan, akan membuat program
tidak dapat dijalankan. Sebab, compiler
```

```
    akan menganggap itu sebagai kode
percobaan anda
    */

    system("pause");
}
```

Jika Program di atas dijalankan, maka output nya akan terlihat seperti di bawah ini:

Komentar satu baris

Blok komentar (Multiple lines comments)

### 3. Pertemuan 3

#### 3.1 Pengertian Fungsi Puts ,Puchar dan Scanf

##### a. Fungsi puts()

Fungsi ini digunakan khusus untuk menampilkan data string ke layar. Sifat fungsi ini, string yang ditampilkan secara otomatis akan diakhiri dengan `\n` (pindah baris). Dibandingkan dengan `printf()`, perintah ini mempunyai kode mesin yang lebih pendek.

##### b. fungsi putchar()

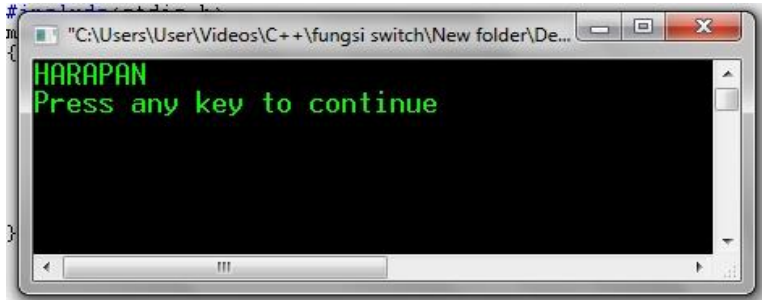


putchar berfungsi untuk menampilkan sebuah karakter di layar. Dalam menampilkan karakter tidak diakhiri dengan pindah baris.namun untuk pindah baris, dapat juga digunakan notasi "\n"

Contoh :

```
#include<stdio.h>
main()
{
    putchar('H');
    putchar('A');
    putchar('R');
    putchar('A');
    putchar('P');
    putchar('A');
    putchar('N');
        putchar('\n');
}
```

Hasil output :



### c. Fungsi scanf()

Merupakan fungsi yang dapat digunakan untuk memasukkan berbagai jenis data. Bentuk scanf ( ) sesungguhnya menyerupai fungsi printf ( ) yang melibatkan penentu format.

Bentuk penulisan : scanf(“string kontrol”, daftar argumen);

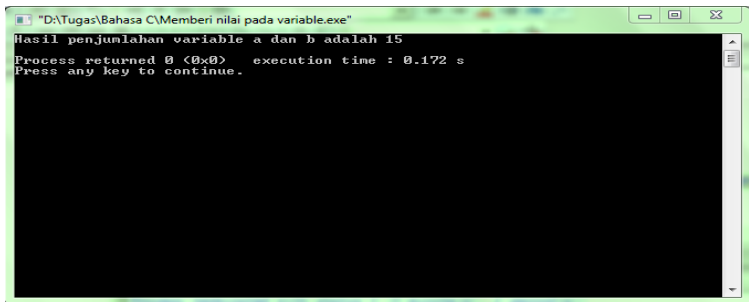
## 3.2 Penggunaan variabel, memasukan nilai ke dalam variabel

- a. cara memasukkan nilai pada sebuah variable, perlu diketahui bahwa variable adalah sebuah wadah dalam memory komputer yang digunakan untuk menampung sebuah nilai. Berikut script untuk memasukkan nilai pada sebuah variable

```
#include <stdio.h>
int main()
{
    int a=10;
```

```
int b=5;
int c;
c=a+b;
printf("Hasil penjumlahan variable a dan b
adalah %i\n",c);
return 0;
}
```

Hasil dari script berikut adalah sebagai berikut:

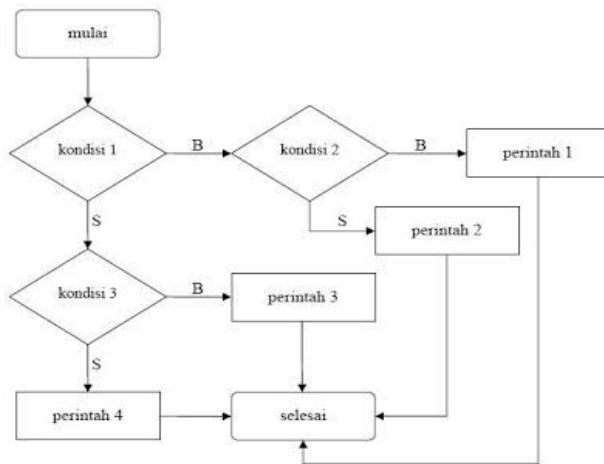


```
"D:\Tugas\Bahasa C\Memberi nilai pada variable.exe"
Hasil penjumlahan variable a dan b adalah 15
Process returned 0 (0x0) execution time : 0.172 s
Press any key to continue.
```

## 4. Pertemuan 4

### 4.1 Operator Relasi dan Fungsi seleksi kondisi sederhana

Statement percabangan dalam dunia pemrograman merupakan salah ilmu dasar pemrograman yang wajib dimiliki oleh setiap programmer. Logika dari statement percabangan adalah layaknya ketika kita mencari sebuah jalan dari beberapa jalan.



a. Operator relasi dan operator logika

Operator relasi digunakan untuk menunjukkan hubungan antara 2 operan, Operator ini sering digunakan dalam penyeleksian dengan statment if,for, while atau do-while.

Contoh :

```

#include <stdio.h>

int main()
{
    int a = 20;
    int b = 10;

    if( a==b )
  
```

```
        {
            printf("Baris 1 - a sama dengan b
\n");
        }
        else{
            printf("Baris 1 - a tidak sama dengan
b\n");
        }
        if(a < b)
        {
            printf("Baris 2 - a lebih kecil dari
b\n");
        }
        else{
            printf("Baris 2 - a tidak lebih kecil
dari b\n");
        }
        if(a > b)
        {
            printf("Baris 3 - a lebih besar dari
b\n");
        }
        else{
            printf("Baris 3 - a tidak lebih besar
dari b\n");
        }
        /* Coba Kita ubah nilai a dan b*/
        a=5;
        b=10;

        if(a <=b )
        {
```

```

        printf("baris 4 - a lebih kurang sama
dengan b\n" );
    }
    if(b >= a)
    {
        printf("baris 5 - a lebih besar sama
dengan b\n");
    }
return 0;
}

```

Jika Sudah compile dan esekusi source code diatas, maka hasilnya seperti ini:

```

Baris 1 - a tidak sama dengan b
Baris 2 - a tidak kurang dari b
Baris 3 - a lebih besar dari b
baris 4 - a lebih kurang sama dengan b
baris 5 - a lebih besar sama dengan b

```

- Operator Logika

Jika operator relasi membandingkan antara dua buah operan, maka operator logika dapat digunakan membandingkan logika hasil dari operator-operator relasi. Operator Logika membandingkan kedua nilai logika. Nilai Logika adalah benar(true) dan salah (not true).Operator logika juga banyak digunakan dalam penyeleksian kondisi yang rumit dengan statemen if , do-while atau while.

### Contoh 1:

```
#include <stdio.h>

int main()
{
    int a= 5;
    int b= 20;

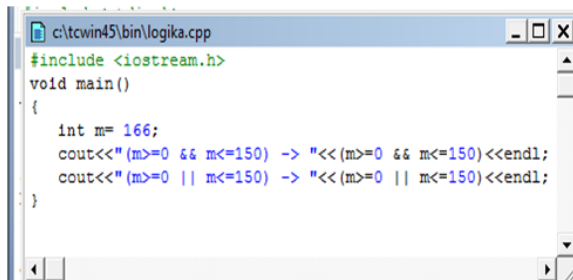
    if (a && b)
    {
        printf("Line 1 - Kondisinya adalah
true\n");
    }
    if (a || b)
    {
        printf("Line 2 - Kondisinya adalah
true\n");
    }
    /* Mari kita coba ubah nilai a dan b*/
    a=0;
    b=10;
    if(a && b)
    {
        printf("Line 3 - Kondisinya adalah
true\n");
    }
    else
    {
        printf("Line 3 - Kondisinya adalah
tidak true \n");
    }
    if (!(a && b))
    {
```

```
        printf("Line 4 - Kondisinya adalah
true\n");
    }
return 0;
}
```

Jika sudah compile dan esekusi source code diatas, maka hasilnya seperti ini:

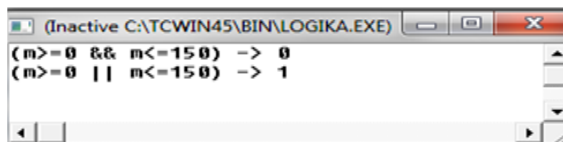
```
Line 1 - Kondisinya adalah true
Line 2 - Kondisinya adalah true
Line 3 - Kondisinya adalah tidak true
Line 4 - Kondisinya adalah true
```

Contoh 2:



```
c:\tcwin45\bin\logika.cpp
#include <iostream.h>
void main()
{
    int m= 166;
    cout<<"(m)>=0 && m<=150) -> "<<(m)>=0 && m<=150)<<endl;
    cout<<"(m)>=0 || m<=150) -> "<<(m)>=0 || m<=150)<<endl;
}
```

Hasil output :



```
(Inactive C:\TCWIN45\BIN\LOGIKA.EXE)
(m)>=0 && m<=150) -> 0
(m)>=0 || m<=150) -> 1
```



## b. Fungsi logika IF

Logika IF digunakan untuk menyeleksi sebuah kondisi bernilai benar atau salah. Jika kondisi bernilai benar maka statemen setelah IF akan dijalankan dan jika kondisi bernilai salah maka statemen setelah IF tidak dijalankan.

Cara penulisannya:

```
if (kondisi) {  
    kode yang akan dijalankan jika kondisi bernilai benar;  
}
```

Contoh :

```
#include <stdio.h>  
  
int main ()  
{  
    int panjang = 10;  
  
    if(panjang > 5)  
    {  
        printf("Panjang lebih besar dari 5");  
    }  
    return 0 ;  
}
```

## c. Fungsi logika IF .. Else

Statement ini merupakan pengembangan dari “if” statement, ketika suatu tidak terpenuhi maka akan menjalankan . Tetapi ketika menemukan “else if” statement maka dilakukan pengecekan kembali.

Cara penulisan :

```
if (kondisi) {
    statement;
}
else {
    statement;
}
else
}
```

Contoh :

```
#include
int main()
{
    int bilangan;
    printf("Masukkan sebuah bilangan : ");
    scanf("%d", &bilangan);
    if ( bilangan % 2 == 0)
        printf("%d adalah bilangan genap",
bilangan);
    else
        printf("%d adalah bilangan ganjil",
bilangan);
    return 0;
}
```

d. Fungsi logika SWITCH.

Switch digunakan ketika kita harus menyeleksi banyak kondisi. Sebetulnya bisa juga digunakan if else if namun akan banyak sekali else if nya dan berakibat pada penulisan kode program yang rumit.

Cara Penulisan Logika SWITCH pada bahasa C

```
switch (n) {
  case nilai1:
    kode yang akan dieksekusi jika n=nilai1;
    break;
  case nilai2:
    kode yang akan dieksekusi jika n=nilai2;
    break;
  case nilai3:
    kode yang akan dieksekusi jika n=nilai3;
    break;
  ...
  default:
    kode yang akan dieksekusi jika tidak ada yang
    sama dengan semua label;
}
```

Contoh penggunaan logika SWITCH pada bahasa C

```
#include <stdio.h>

int main ()
{
  int hari;
  puts ("MASUKKAN ANGKA 1 SAMPAI 3 =");
  scanf("%d",&hari);
  puts ("ANDA MEMASUKKAN ANGKA = ");
  switch (hari) {
  case 1:
    puts ("SATU");
    break;
```

```
case 2:
    puts ("DUA");
    break;
case 3:
    puts ("TIGA");
    break;
default:
    puts ("TIDAK DIANGGAP");
}
return 0;
}
```

## 5. Pertemuan 5

### 5.1 Penjelasan mengenai fungsi logika IF . . Else bertingkat.

Program Looping atau Pengulangan adalah suatu yang wajib dikuasai oleh seorang Programmer, ini karena looping adalah jiwa dari program. Banyak Algoritma yang memerlukan looping sebagai sarana terwujudnya Algoritma tersebut seperti Algoritma Array, Searching, Sorting, dll. Dari sini kebayang bagaimana pentingnya sebuah looping dalam sebuah program. Maka dari itu Aden Kejawen yang suka looping alias muter-muter kalo jelasin, akan mencoba berbagi tentang Algoritma dan Syntax Looping dalam Bahasa C.

## 6. Pertemuan 6

### 6.1 Perintah Perulangan (*Looping*) sederhana.

Program Looping atau Pengulangan adalah suatu yang wajib dikuasai oleh seorang Programmer, ini karena looping adalah jiwa dari program. Banyak Algoritma yang memerlukan looping sebagai sarana terwujudnya Algoritma tersebut seperti Algoritma Array, Searching, Sorting, dll. Dari sini kebayang bagaimana pentingnya sebuah looping dalam sebuah program. Maka dari itu Aden Kejawan yang suka looping alias muter-muter kalo jelasin, akan mencoba berbagi tentang Algoritma dan Syntax Looping dalam Bahasa C.

#### a. Penjelasan mengenai perulangan.

Looping adalah suatu fungsi pengulangan dalam sebuah script program C++. Gunanya adalah untuk mempermudah melakukan suatu proses yang berulang-ulang, seperti membuat angka dari 1 – 10. Jika memakai script biasa kita akan memakan banyak memori, tetapi dengan looping kita dapat melakukannya dengan memori yang sedikit dan waktu yang singkat.

#### b. Penjelasan mengenai perintah FOR ()

Berfungsi untuk mengulang satu pernyataan sebanyak yang kita inginkan.

```
Script for(int i=0;i<=9;i++) { statement }
```

Pengulangan akan dilakukan dari 0-9, berarti ada 10x pengulangan.

Arti variable i=0 adalah nilai awal dari i adalah 0.

Arti i<=9 adalah maksimal pengulangan yang dilakukan adalah sampai 9.

Arti i++ adalah menambahkan nilai setiap pengulangan yang sudah di eksekusi, contoh nya jika kita sudah memasukkan nilai di i [0] , maka akan langsung lanjut ke i [1] dan seterusnya hingga i [9].

c. Penjelasan mengenai perintah WHILE ()

Hampir sama dengan For, namun dalam while sistemnya hampir sama juga dengan If.

Berikut contoh scriptnya :

```
Int i=1;
While(i<=4)
{
Cout<<;
I++;
}
```

Sebelum penggunaan while, nilai integer harus di deklarasikan terlebih dahulu (int i=1).

Penggunaannya hampir sama dengan if, tetapi di akhir statement harus ada tanda variable++ (i++) untuk menyatakan pengulangan hingga 4x

d. Penjelasan mengenai perintah DO..WHILE ()

Merupakan statement pengulangan yang biasa digunakan untuk membuat menu dalam program. Secara umum proses do while sama dengan for ataupun while.

Berikut contohnya :

```
do
{
  clrscr();

  cout<<"1. Jual";

  cout<<"2. Beli ";

  int i;

  cin>>i;

}while(i!=3);
```

Dalam statement di atas, ada 2 menu yang kita buat, yaitu menu jual dan beli. Dalam hal ini jika kita tidak menginput 3 (while(i!=3)) maka pengulangan terus berlangsung

Latihan penggunaan perintah perulangan sederhana.

## 7. Pertemuan 7

### 7.1 Review Materi sebelum UTS / QUIZ

8. Pertemuan 8  
8.1 Ujian Tengah Semester

**9. Pertemuan 9**

9.1 Pembahasan soal – soal UTS

**10. Pertemuan 10**

10.1 Pembahasan soal – soal UTS.

1. Penjelasan mengenai perintah FOR() bertingkat.
2. Penjelasan mengenai perintah WHILE() bertingkat.
3. Penjelasan mengenai perintah DO..WHILE() bertingkat.

**11. Pertemuan 11**

11.1 Penggunaan *Function* (Fungsi)

Fungsi merupakan suatu bagian dari program yang dimaksudkan untuk mengerjakan suatu tugas tertentu dan letaknya terpisah dari program yang memanggilmnya. Fungsi merupakan elemen utama dalam bahasa C karena bahasa C sendiri terbentuk dari kumpulan fungsi-fungsi. Dalam setiap program bahasa C, minimal terdapat satu fungsi yaitu fungsi main(). Fungsi banyak diterapkan dalam program-program C yang terstruktur. Keuntungan penggunaan fungsi dalam program yaitu program akan memiliki struktur yang jelas (mempunyai readability yang tinggi) dan juga akan menghindari penulisan bagian program yang sama.

Dalam bahasa C fungsi dapat dibagi menjadi dua, yaitu fungsi pustaka atau fungsi yang telah tersedia dalam Turbo C dan fungsi yang didefinisikan atau dibuat oleh programmer.

Fungsi dalam bahasa C adalah sekumpulan kode yang ditaruh dalam sebuah blok dan dibuat untuk menjalankan tugas khusus. Salah satu tujuan dari penggunaan fungsi dalam pemrograman seperti pada bahasa



C adalah untuk membuat program lebih terstruktur dan efisien sehingga program tersebut mudah dipahami atau dibaca alur programnya.

### 11.2 Penjelasan mengenai penggunaan *function*.

Penggunaan fungsi membuat program jadi lebih efisien karena mengurangi pengulangan penulisan kode yang sama. Hal ini karena jika sekelompok kode sudah dibuat fungsi maka selanjutnya kita tinggal memanggil nama fungsinya saja beserta parameter yang menyertainya jika ada. Sebuah fungsi tidak dijalankan secara otomatis saat program dijalankan karena fungsi berada diluar rutin program utama yang berjalan dan hanya dijalankan jika fungsi dipanggil saja.

Pada bahasa C terdapat beberapa fungsi standar seperti misalnya `main()`, `printf()` dan `getch()` yang telah kita gunakan sebelumnya. Fungsi `main()` adalah fungsi yang mempunyai kedudukan sangat istimewa. Fungsi `main()` merupakan fungsi utama yang harus ada dalam setiap program bahasa C. Fungsi `main()` berisi rutin utama program yang akan dijalankan ketika program bahasa C kita eksekusi. Jadi semua kontrol alur proses program berada dalam fungsi `main()` ini.

Selain fungsi `main()` ada fungsi standar bahasa C yang lain misalnya `printf()` dan `getch()`. Fungsi-fungsi tersebut mengerjakan tugas khusus jika dipanggil seperti misalnya fungsi `printf()` digunakan untuk menampilkan informasi teks pada layar dan fungsi `getch()` digunakan untuk membaca masukan dari tombol keyboard.

## 11.2 Cara mendeklarasikan *function*.

### Deklarasi Fungsi dalam bahasa C

Pada umumnya sebuah fungsi dalam bahasa C tidak perlu dideklarasikan jika penulisannya ditaruh diatas fungsi main(). Deklarasi fungsi diperlukan jika penulisan fungsi berada dibawah fungsi main(). Deklarasi fungsi juga diperlukan jika kita membuat fungsi dengan tipe data keluaran bukan integer misalnya float atau string.

Sebagai contoh jika program diatas kita modifikasi dengan mengubah letak penulisan definisi fungsi hitung\_luas menjadi dibawah fungsi main() tanpa deklarasi fungsi maka akan tampil warning : implicit declaration of function 'hitung\_luas' [-Wimplicit-function-declaration].

Untuk itu diperlukan deklarasi fungsi hitung\_luas seperti contoh berikut ini:

```
#include <stdio.h>
int hitung_luas(int panjang, int lebar); //deklarasi fungsi
int main(){
int panjang =5;

int lebar =7;
int luas;
luas = hitung_luas(panjang,lebar);
printf("%d \n",luas);
return 0;
}
int hitung_luas(int panjang, int lebar)
{
int luas;
luas = panjang * lebar;
return(luas);
}
```

### 11.3 Jenis-jenis fungsi pada bahasa C

#### a. Penggunaan fungsi VOID()

Fungsi yang void sering disebut juga prosedur. Disebut void karena fungsi tersebut tidak mengembalikan suatu nilai keluaran yang didapat dari hasil proses fungsi tersebut.

- Ciri-ciri dari jenis fungsi Void adalah sebagai berikut:

Tidak adanya keyword return.

Tidak adanya tipe data di dalam deklarasi fungsi.

Menggunakan keyword void.

Tidak dapat langsung ditampilkan hasilnya.

Tidak memiliki nilai kembalian fungsi

Keyword void juga digunakan jika suatu function tidak mengandung suatu parameter apapun.

Contoh fungsi void :

```
#include <iostream.h>
#include <conio.h>
void volume(int &vol, int p, int l, int t)
{ vol = p*l*t; }
main(){
int pj,lb,tg, hsl;
cout<<"Panjang = ";cin>>pj;
cout<<"Lebar = ";cin>>lb;
cout<<"Tinggi = ";cin>>tg;
volume(hsl,pj,lb,tg);
cout<<"\nVolume = "<<hsl;
getch();
}
```

b. Penggunaan fungsi REKURSIF()

Fungsi rekursif merupakan fungsi yang memanggil dirinya sendiri. Terdapat dua komponen penting dalam fungsi rekursif, yaitu kondisi kapan berhentinya fungsi dan pengurangan atau pembagian data ketika fungsi memanggil dirinya sendiri. Optimasi fungsi rekursif dapat dilakukan dengan menggunakan teknik tail call, meskipun teknik ini tidak selalu diimplementasikan oleh semua bahasa pemrograman.

```
#include <stdio.h>

int faktorial(int angka){
    if(angka<=1){
        return 1;
    }else {
        return angka*faktorial(angka-1);
    }
}

int main(){
    printf("Faktorial dari 5 = %d", faktorial(5));
    getchar();
    return 0;
}
```

### Latihan penggunaan FUNCTION

Tulislah fungsi untuk menghasilkan nilai  $F(x)=3x^2+5x-1, x \in \mathbb{R}$ . Pada soal terlihat bahwa  $x$  bertipe bilangan Real maka  $x$  harus dideklarasikan dengan tipe data float. Maka solusi program dari soal tersebut adalah

## 12 Pertemuan 12

### 12.1 Array (senarai)

Array adalah kumpulan dari nilai-nilai data bertipe sama dalam urutan tertentu yang menggunakan sebuah nama yang sama. Nilai-nilai data di suatu larik disebut dengan elemen-elemen larik.

Letak urutan dari suatu elemen larik ditunjukkan oleh suatu subscript atau suatu index.

Menurut dimensinya, array dapat dibedakan menjadi :

- a. Array berdimensi satu
  - Setiap elemen array dapat diakses melalui index
  - Index array secara default dimulai dari 0

Tipe\_array nama\_array[ukuran]

- b. Array berdimensi dua
  - Array dua dimensi merupakan array yang terdiri dari m buah baris dan n buah buah kolom. Bentuknya dapat berupa matriks atau tabel.
  - Deklarasi array :

Tipe\_array nama\_array [baris][kolom]

- c. Array multidimensi

- Array multidimensi merupakan array yang mempunyai ukuran lebih dari dua. Bentuk pendeklarasian array multidimensi sama saja dengan deklarasi array dimensi satu maupun dimensi dua.
- Deklarasi array :

Tipe\_array nama\_array [ukuran 1][ukuran 2] . . . [ukuran N]

## 12.2 Penjelasan dan contoh array satu dimensi

Suatu elemen bisa dikatakan array satu dimensi jika banyaknya index hanya ada satu. Perlu diperhatikan, sebelum variabel array digunakan maka variabel array tersebut harus terlebih dahulu dideklarasikan. Untuk pendeklarasiannya array satu dimensi ini sebenarnya hampir sama dengan pendeklarasian variabel yang lainnya, hanya saja pada pendeklarasian array satu dimensi ini variabel arraynya diikuti dengan maksimum banyaknya elemen yang dapat disimpan dalam variabel array yang dituliskan dalam pasangan tanda siku pembuka dan siku penutupnya. Didalam bahasa pemrograman C++, harga awal indeks dimulai dengan nilai 0 (nol). Maka jika dituliskan banyaknya maksimum elemen adalah  $N$ , hal ini dapat diartikan indeks yang akan digunakan adalah  $0, 1, 2, 3, \dots, N-1$

Berikut ini adalah contoh dari bentuk umum untuk pendeklarasian array satu dimensi

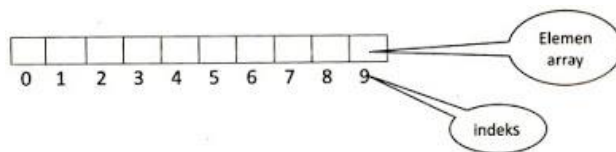
**tipe\_data nama\_var\_array [ukuran];**

Artinya :

- **tipe\_data** : Menyatakan jenis tipe data elemen array (int, char, float, dsb)
- **nama\_var\_array** : Menyatakan nama variabel yang akan digunakan
- **ukuran** : menunjukkan maksimal banyaknya elemen array
- 

Contoh : `int nilai[10];`

Dengan contoh deklarasi diatas maka akan disiapkan lokasi memori untuk variabel array sebanyak 10, lihat gambar di bawah ini untuk lebih jelasnya



#### a. Mengisi Elemen Array Satu Dimensi

Suatu variabel array satu dimensi yang telah dideklarasikan pada dasarnya sudah bisa dilakukan pengisian sebuah elemen, perlu anda ketahui bahwa untuk mengisi sebuah elemen kedalam suatu variabel satu dimensi dapat dilakukan dengan 3 (tiga) cara :

1. Deklarasi variabel (inisialisasi)
2. Menggunakan penugasan (=)
3. Dilakukan dengan menggunakan / dibaca dari media masukan

#### b. Inisialisasi Array Satu Dimensi (Deklarasi Variabel)

Untuk melakukan inisialisasi pada array satu dimensi dapat dilakukan dengan memberikan nilai awal dari array pada saat dideklarasikan, berikut ini adalah bentuk umum dari inisialisasi array satu dimensi.

```
tipe_data nama_var_array [ukuran] = {elemen-0, elemen-1, .....,  
elemen-n-1};
```

Artinya :

- elemen-0, elemen-1, ....., elemen-n-1 merupakan elemen - elemen dari array dan dari banyaknya ukuran.
- Jika banyaknya elemen kurang dari ukuran, maka sisanya akan diberi harga atau nilai nol (0).
- Jika banyaknya elemen lebih dari ukuran maka akan mengalami kesalahan " *Too many initializers* ".

```
int nilai[10] = {8, 12, 20, 15, 7, 5, 6, 4, 32, 3};
```

Contoh dari data diatas, dapat kita artikan bahwa kita telah memesan tempat pada memori komputer sebanyak 10 dengan indeks dari 0 sampai dengan 9, dimana elemen - elemen akan dimasukan ke lokasi nilai secara berturut - turut, mulai dari indeks 0 sampai indeks 9, seperti pada contoh dibawah ini.

8	12	20	15	7	5	6	4	32	3
0	1	2	3	4	5	6	7	8	9



Perhatikan contoh dibawah ini untuk melakukan penginisialisasian 10 bilangan bulat yang kemudian akan dilakukan penghitungan rata - rata (Lat\_I\_01)

```

/*Array Satu Dimensi
   Dengan Inisialisasi
   Nama File : Lat_I_01 */

#include<iostream.h>
#include<conio.h>
main()
{
    int Nilai[10]={8, 12, 20, 15, 7, 5, 6, 4, 32, 3};
    int i, Jumlah=0;
    float Rata_Rata;
    //Menghitung jumlah
    for(i=0;i<10;i++)
        Jumlah+=Nilai[i];

    Rata_Rata=(float) Jumlah / 10;
    //Mencetak Elemen Array
    cout<<"\n\nDeretan Bilangan    = ";
    for(i=0;i<10;i++)
        cout<<Nilai[i]<<" ";
    //Mencetak Harga Jumlah
    cout<<"\nJumlah Bilangan      = "<<Jumlah;
    cout<<"\nRata-Rata Bilangan = "<<Rata_Rata;
    getch();
}

```

### c. Menggunakan Operator Penugasan (=)

Mengisi sebuah elemen array satu dimensi dengan menggunakan operator penugasan dapat dilakukan dengan bentuk seperti dibawah ini

**nama\_var\_array [indeks/posisi] = elemen;**

Dimana elemen adalah harga yang akan disimpan di bagian **nama\_var\_array** pada sebuah indeks atau poisisi

Contoh :

**Nilai [0] = 8;**

**Nilai [1] = 12;**

**Nilai [2] = 20;**

...

**Nilai [9] = 3;**

### d. Dibaca melalui Media Masukan

pada cara yang ketiga ini adalah dengan menggunakan media masukan atau lebih kita kenal dengan istilah media input berupa keyboard, perhatikan format dibawah ini untuk melakukan pembacaan pada elemen array satu dimensi

**cin>>nama\_var\_array [indeks];**

Contoh 1 :

- **cin>>nilai [0]** → Membaca elemen array posisi ke-0
- **cin>>nilai [1]** → Membaca elemen array posisi ke-1
- **cin>>nilai [2]** → Membaca elemen array poisis ke-2
- **cin>>nilai [i]** → Membaca elemen array posisi ke-i dimana harga "i" ini harus diketahui sebelumnya, Misalnya harga i=7 berarti "i" disini akan membaca elemen array pada posisi ke-7

Contoh 2 :

Membuat sebuah program untuk membaca 10 bilangan bulat yang kemudian akan dihitung jumlah dan rata - rata (**Lat\_I\_02**).

## 12.2 Penjelasan dan contoh array dua dimensi

Array dua dimensi adalah array yang terdiri dari n buah baris dan m buah kolom, atau array dua dimensi juga biasa disebut sebagai array yang mempunyai dua subskrip, yaitu baris dan kolom. Bentuknya dapat sobat bayangkan seperti matriks atau tabel. dimana indeks pertama menunjukkan baris dan indeks kedua menunjukkan kolom.

Berikut gambar yang dapat mengilustrasikan sebuah array dua dimensi.

Int A [3][4];

Materi   Dosen	Kolom 0	Kolom 1	Kolom 2	Kolom 3
Baris 0	A[0][0]	A[0][1]	A[0][2]	A[0][3]
Baris 1	A[1][0]	A[1][1]	A[1][2]	A[1][3]
Baris 2	A[2][0]	A[2][1]	A[2][2]	A[2][3]

ilustrasi array dua dimensi

Pada ilustrasi array dua dimensi diatas, dimisalkan terdapat sebuah array dua dimensi dengan nama "A", array A memiliki jumlah elemen baris sebanyak 3 dan jumlah elemen kolom sebanyak 4.

#### a. Bentuk / Cara Pendeklarasian Array Dua Dimensi

Untuk mendeklarasikan sebuah **array dua dimensi dalam C++**, sobat harus menggunakan tanda [ ](bracket). Adapun bentuk umum dari pendeklarasian array dua dimensi adalah sebagai berikut:

```
tipe_data          nama_array          [jumlah_element_baris]
[jumlah_element_kolom];
```

Sebagai contoh misal kita ingin mendeklarasikan sebuah array dua dimensi (misalnya dengan nama LARIK) yang mempunyai jumlah elemen baris sebanyak 3, jumlah elemen kolom sebanyak 5 dan array LARIK memiliki tipe data int, maka bentuk array nya adalah sebagai berikut:

```
int LARIK [3][5];
```

Seandainya array LARIK akan kita beri nilai saat pendeklarasian (inisialisasi), maka contoh penulisannya dapat ditulis sebagai berikut:

```
int LARIK [3][5] = {{1,2,3,4,5},{6,7,8,9,10},{11,12,13,14,15}};
```

Dari pendeklarasian sekaligus inisialisasi array LARIK diatas, dapat diartikan bahwa kita telah memesan tempat pada memori komputer sebanyak 15 tempat, dengan indeks dari LARIK[0][0] sampai LARIK[2][4], dimana nilai - nilai akan dimasukkan ke elemen array secara berturut - turut, mulai dari LARIK[0][0] akan di isi dengan nilai '1' sampai LARIK[2][4] yang di isi dengan nilai '15'. untuk lebih jelasnya berikut kami sajikan ilustrasi dari array LARIK.

```
int LARIK [3][5] = {{1,2,3,4,5},{6,7,8,9,10},{11,12,13,14,15}};
```

Materi Dosen	Kolom 0	Kolom 1	Kolom 2	Kolom 3	Kolom 4
Baris 0	1	2	3	4	5
Baris 1	6	7	8	9	10
Baris 2	11	12	13	14	15

### Ilustrasi Array 'LARIK' yang merupakan contoh Array Dua Dimensi

#### b. Cara Akses Array Dua Dimensi

Cara yang digunakan untuk mengakses elemen array dua dimensi adalah dengan menuliskan indeks baris dan kolom nya. Misalnya pada array LARIK diatas kita ingin mengakses nilai '8' yang terdapat pada indeks baris ke 1 dan indeks kolom ke 2, kemudian menampung nilai tersebut kedalam sebuah variabel yang bertipe int (misal int tampung), maka kita perlu menuliskan kode programnya seperti berikut ini:

```
tampung = LARIK[1][2];
```

\* Perlu sobat ingat meskipun array LARIK memiliki jumlah elemen baris sebanyak 3 dan jumlah elemen kolom sebanyak 5 atau di deklarasikan sebagai berikut : "LARIK [3][5];". namun karena indeks array selalu dimulai dari 0 maka untuk mengakses elemen terakhir dari array LARIK, kita dapat mengaksesnya dengan menuliskan LARIK[2][4] bukan LARIK[3][5].

#### Contoh Program Dengan Array Dua Dimensi Dalam C++

Berikut ini adalah contoh program sederhana menampilkan matrik ordo 3x2, dimana program ini menggunakan Array Dua Dimensi.

```
D:\Blog\MateriDosen.Com\code\Contoh Program Sederhana Array Dua Dimensi.exe
=====
=== Tampil Matrik Ordo 3x2 ===
=====

Menampilkan matrikcontoh
1 2
3 4
5 6

-----
Process exited after 8.544 seconds with return value 0
Press any key to continue . . .

Materi Dosen
```

```
#include <iostream>

using namespace std;

int main(){

    // Deklarasi dan inisialisasi array dua dimensi
    // Dengan jumlah elemen baris = 3
    // dan jumlah elemen kolom = 2
    int matrikcontoh [3][2] = {{1,2},{3,4},{5,6}};

    // Mendeklarasi variabel untuk
    // Indeks perulangan
    int i,j;

    cout<<"\t=====";
    cout<<"\n\t=== Tampil Matrik Ordo 3x2 ===\n";
    cout<<"\t=====\\n\\n";
```

```

// Menampilkan matrikcontoh
cout<<"\nMenampilkan matrikcontoh\n";
for(i=0;i<3;i++){
    for(j=0;j<2;j++){
        cout<<matrikcontoh[i][j]<<" ";
    }
    cout<<endl;
}

return 0;
}

```

Berikut ini adalah contoh program penjumlahan 2 buah matrik ordo 3x2, dimana program ini menggunakan Array Dua Dimensi.

```

D:\Blog\MateriDosen.Com\code\Contoh Program dengan Array Dua Dimensi.exe
=====
= Penjumlahan Matrik Ordo 3x2 =
=====
matrik a [0] [0] = 2
matrik a [0] [1] = 3
matrik a [1] [0] = 1
matrik a [1] [1] = 4
matrik a [2] [0] = 3
matrik a [2] [1] = 1

Matrik a
2 3
1 4
3 1

Matrik b
1 2
3 4
5 6

Matrik c atau hasil
3 5
4 8
8 7

-----
Process exited after 10.66 seconds with return value 0
Press any key to continue . . .

```

```

#include <iostream>

using namespace std;

```



```

int main(){

    // Deklarasi array dua dimensi
    // Dengan jumlah elemen baris = 3
    // dan jumlah elemen kolom = 2
    int matrika [3][2];
    int matrikb [3][2] = {{1,2},{3,4},{5,6}};
    int matrikc [3][2];

    // Mendeklarasi variabel untuk
    // Indeks perulangan
    int i,j;

    cout<<"\t===== ";
    cout<<"\n\t= Penjumlahan Matrik Ordo 3x2 =\n";
    cout<<"\t===== \n\n";

    // Mengisi nilai kedalam
    // Elemen-elemen array matrika
    // Sedangkan elemen-elemen array matrikb
    // Sudah diisi saat deklarasi (inisialisasi)
    for(i=0;i<3;i++){
        for(j=0;j<2;j++){
            cout<<"matrik a ["<<i<<"] ["<<j<<"] =
";
            cin>>matrika[i][j];
        }
    }

    cout<<endl;

    // Melakukan penjumlahan array matrika dan
    matrikb
    // Dan menyimpan hasilnya ke array matrikc;
    for(i=0;i<3;i++){
        for(j=0;j<2;j++){

```

```

        matrikc[i][j]=matrika[i][j] +
matrikb[i][j];
    }
}

// Menampilkan matrika
cout<<"\nMatrik a\n";
for(i=0;i<3;i++){
    for(j=0;j<2;j++){
        cout<<matrika[i][j]<<" ";
    }
    cout<<endl;
}

// Menampilkan matrikb
cout<<"\nMatrik b\n";
for(i=0;i<3;i++){
    for(j=0;j<2;j++){
        cout<<matrikb[i][j]<<" ";
    }
    cout<<endl;
}

// Menampilkan hasil penjumlahan / matrikc
// (dalam bentuk matrik dengan ordo 3x2)
cout<<"\nMatrik c atau hasil\n";
for(i=0;i<3;i++){
    for(j=0;j<2;j++){
        cout<<matrikc[i][j]<<" ";
    }
    cout<<endl;
}

return 0;
}

```

### 12.3 Penjelasan dan contoh array tiga dimensi (dimensi banyak).

Array 3 dimensi adalah array yang tidak jauh berbeda dari array dimensi satu dan dua yang telah dijelaskan sebelumnya, kecuali pada indeks dari array. Pada tipe ruang misalnya type ruang = array [1..8,1..5,1..3] of integer; menunjukkan bahwa ruang adalah nama-pengenal/variabel yang berupa array yang komponennya bertipe integer dan terdiri atas 8 baris, mempunyai 5 kolom dan 3 halaman. Bedanya array 3 dimensi dengan 1 dan 2 dimensi yaitu menggunakan 3 index misalnya deklarasi array yang sebagai berikut :

```
Int banyakData [5][2][3]
```

- index pertama menunjukkan banyak baris
- index kedua menunjukkan banyak isi baris
- index ketiga menunjukkan banyak kolom

### Contoh Program Array Multi Dimensi pada Bahasa C

```
#include<stdio.h>
Main()
{
Char data [2][3][6] = {"1", "soto", "5000", "2", "bakso", "3500"};
Int i,j;
For(i=0; i<2; i++)
{
For(j=0; j<3; j++)
{
Printf("\t%s", data [i][j]);
}
Printf("\n");
}
```

```
}  
Return 0;  
}
```

Jika program dijalankan akan menghasilkan output sebagai berikut :

Latihan penggunaan array.

Buatlah dan jelaskan langkah-langkahnya untuk membuat sebuah array/larikiga dimensi berukuran 3x3x3 menggunakan python.

### **13. Pertemuan 13**

#### 13.1 Fungsi dan tujuan penggunaan struct.

Pada tahap pembelajaran bahasa C telah dipelajari mengenai tipe data dan variabe. Pada tahap awa setiap variable hanya mewakili 1 tipe data. Penggunaan struct memungkinkan 1 variabel memiliki anggota yang terdiri dari beberapa variabel dan tipe data. Struct membebaskan pemrogram untuk menyimpan data yang kompleks, data yang disimpan tidak harus bertipe data yang sama dengan data lainnya.

#### 13.2 Cara mendeklarasikan struct.

Untuk mendeklarasikan sebuah struct, dapat dipilih salah satu dari dua metode berikut ini. Pendeklarasian struct yang pertama (bentuk umum).

```
struct namaStruct { tipeData namaVariabel; };
```

Contoh kode pendeklarasian struct:

```

struct mahasiswa {
    char nim[25];
    char nama[25];
    int usia;
};

```

## Latihan penggunaan Struct.

Buat tampilan berikut:

**Input**

PENITIPAN HEWAN PELIHARAAN  
 "MEAOW PET CARE"  
 =\*\*\*\*\*=

Nama Petugas :  
 Nama Pemilik :  
 Jumlah Data Peliharaan :

Hewan Ke- <counter>  
 KodeJenis Hewan [K/H/B] :  
 Kode Perawatan [1/2] :  
 Jumlah hewan :

**Output**

PENITIPAN HEWAN PELIHARAAN  
 "MEAOW PET CARE"  
 =\*\*\*\*\*=

Nama Petugas :  
 Nama Pemilik :

=====

No Hewan	Jenis Hewan	Jenis Perawatan	Biaya perawatan	Jumlah hewan	Subtotal
<<<<<<OUTPUT DATA>>>>>>>>					
					Total Bayar :
					Uang Bayar : <input>
					Uang Kembali :

=====

INPUTDATA LAGI [Y/T]:<input>

## 14 Pertemuan 14

### 14.1 Latihan penggabungan fungsi – fungsi dalam bahasa C

### 14.2 Penggabungan fungsi- fungsi bahasa C.

### 14.3 Array dan looping

### 14.4 Array dan struct

Array of struct adalah jenis data hasil gabungan kelebihan array dan struct,,

kelebihan array adalah dapat di implementasikan kedalam looping for, dan kelebihan struct adalah dapat menyimpan data dengan tipe yang berbeda.

- Deklarasi array of struct

```
typedef struct mahasiswa{char nama[100]; char pekerjaan[50];  
int umur}mahasiswa a[5];
```

Cara menginput data array of struct :

```
a[0].nama="Ardhy";  
a[0].pekerjaan="mahasiswa";  
a[0].umur=20;  
  
jika pakai looping for  
  
for(i=0;i<=5;i++  
{  
printf("%s%s %d",a[i].nama,a[i].pekerjaan,a[i].umur)  
}
```

#### 14.5 Latihan studi kasus

Buatlah program dengan output dibawah ini :

```

F:\KULIAH\Program Pendidikan\SEMESTER 2\Prakt. Pemrograman Sistem\Training\responsi babak ...
SELAMAT DATANG DI PROGRAM PENILAIAN MAHASISWA - UERSI ULTIMATE
=====
Masukkan Standard Nilai Anda ??? (0-100%)
Masukkan Persentase Kehadiran : 10
Masukkan Persentase TUGAS : 30
Masukkan Persentase UTS : 30
Masukkan Persentase UAS : 30
Standard Nilai Anda Sudah Mencapai 100 %
=====
Masukkan Total Pertemuan Dalam Satu Semester : 20
=====
Masukkan Jumlah Mahasiswa yang Ingin Di Input : 10
=====
:=====|=====|=====|=====|=====|=====|=====|
: No. | Nama Mahasiswa | Absen | Tugas | UTS | UAS | Na | Grade |
:=====|=====|=====|=====|=====|=====|=====|
:1 | gatewan | 20 | 100 | 100 | 100 | 100 | A |
:2 | wawanBeneran | 18 | 90 | 90 | 90 | 81 | A |
:3 | santaiarea | 17 | 70 | 78 | 70 | 65 | C |
:4 | totaltren | 19 | 80 | 80 | 75 | 70 | B |
:5 | mr. bin | 18 | 0 | 60 | 60 | 36 | E |
:6 | mr. ban | 20 | 90 | 90 | 95 | 92 | A |
:7 | mr. bon | 3 | 0 | 0 | 0 | 0 | E |
:8 | wiroSableng | 17 | 60 | 90 | 90 | 72 | B |
:9 | s intoGendeng | 19 | 90 | 98 | 89 | 82 | A |
:10 | HlPakki | 0 | 0 | 0 | 0 | 0 | E |
:=====|=====|=====|=====|=====|=====|=====|

```

## 15. Pertemuan 15

15.1 Review materi keseluruhan sebelum UAS.

15.2 Pelaksanaan quiz sebelum UAS.

## DAFTAR PUSTAKA

Jogiyanto, (2010), Konsep Dasar Pemrograman Bahasa C. Yogyakarta : Andi

Abdul Kadir, (2010), Pemrograman C++ Membahas Pemrograman Berorientasi Objek Menggunakan Turbo C++ dan Borland C++. Yogyakarta : Andi

Rinaldi Munir, Algoritma dan Pemrograman dalam Bahasa Pascal dan C (edisi revisi). Jakarta : Informatika