



# ELEMEN DASAR BAHASA C++

Erick Paulus

Mata Kuliah : Algoritma dan Pemrograman  
Program Studi S-1 Teknik Informatika



# Pokok Bahasan

Sesi perkuliahan ini akan membahas topik-topik tentang :

- Bahasa C++
- Struktur program
- Keyword
- Tipe data
- Operator
- Alur pemrograman dengan pola Input-Proses-Output





# Kelebihan Bahasa C dan C++

- Bahasa C dan C++ tersedia hampir di semua jenis computer (PC, minikomputer dan mainframe).
- Bahasa C dan C++ hanya menyediakan sedikit kata-kata kunci
- Kode bahasa C dan C++ sifatnya adalah portable dan fleksibel untuk semua jenis komputer.
- Aplikasi yang ditulis dengan bahasa C untuk suatu komputer tertentu dapat digunakan di komputer lain hanya dengan sedikit modifikasi.
- Proses executable program bahasa C dan C++ lebih cepat
- Dukungan pustaka yang banyak.
- C dan C++ adalah bahasa yang terstruktur.
- C juga mengandung feature-feature low-level yang secara normal hanya tersedia pada Assembly atau bahasa mesin
- Program yang ditulis dengan C dikompilasi menjadi program yang kecil dan dieksekusi dengan efisien
- C++ Sudah mendukung OOP (*Object Oriented Programming*).
- C++ dapat membuat aplikasi graphic processor berkualitas tinggi.





# Kekurangan Bahasa C dan C++

- Banyaknya Operator serta fleksibilitas penulisan program kadang-kadang membingungkan pemakai (Pemula).
- Bagi pemula pada umumnya akan kesulitan menggunakan pointer dan penerapan konsep OOP.





# Susunan penulisan program dengan C++

```
/* Nama program      :  
   Nama             : Akmal  
   NPM              :  
   Tanggal buat    :  
   Deskripsi        :  
***** */  
// deklarasi header file / Preprocessor directive  
// deklarasi fungsi / void  
  
main() {  
    /* KAMUS / DEKLARASI */  
    /* ALGORITMA / DESKRIPSI */  
}
```





# Komentar

- Komentar adalah catatan yang ditulis pada kode dengan tujuan sebagai bahan dokumentasi.
- Teks tersebut bukan bagian dari program dan tidak mempengaruhi jalannya program.
- Ada 2 jenis komentar dalam C++ yaitu:
  - C++ style komentar satu baris (in line comment)  
Awalan : // komentar
  - C style beberapa baris (block comment)  
Pasangan : /\* komentar \*/





# *Preprocessor Directives*

- Preprocessor directives diawali dengan `#include` dan diikuti dengan nama header file (library file)
- Contoh :  
`#include <iostream.h>`  
`#include <conio.h>`

*using namespace std;*  
// jika menggunakan compiler MinGW





## Contoh : Pengenal identitas program dan pencetakan Hello Unpad

```
/* Nama program      : hello.cpp
   Nama          : Akmal
   NPM       : 1234
   Tanggal buat : 17 Agustus
   Deskripsi    : Pencetakan Hello Unpad 2 kali
----- */
```

```
#include <iostream.h>
```

```
main() {
    cout<<"Hello, Unpad." <<endl;
    cout<<"Hello, " << "Unpad";
}
```





## Contoh : Pengenal identitas program dan pencetakan Hello Unpad

```
/* Nama program      : hello.cpp
   Nama          : Ericj
   NPM       : 1234
   Tanggal buat : 19 Agustus
   Deskripsi    : Pencetakan Hello Unpad 2 kali
----- */  
#include <iostream>
using namespace std;
main() {
    cout<<"Hello, Unpad." <<endl;
    cout<<"Hello, " << "Unpad";
}
```





# Identifier (Pengenal)

- C++ Identifier adalah suatu tanda yang mewakili nama-nama variabel, fungsi, method, class, dsb.
- Contoh dari Identifier adalah : nilai, factorial(), Hello, dll.
- Pendeklarasian C++ adalah case-sensitive.
- Hal ini berarti bahwa Identifier : **Hello** tidak sama dengan **hello**.
- Identifier harus dimulai dengan salah satu huruf atau underscore “\_”, dan tidak diawali dengan angka.





# Keyword (Kata kunci)

- Kata yang bermakna khusus yang tidak bisa digunakan sebagai pengenal.
- Ada 48 buah keyword dalam Bahasa C++ yaitu :

asm	continue	float	new	signed	try
auto	default	for	operator	sizeof	typedef
break	delete	friend	private	static	union
case	do	goto	protected	struct	unsigned
catch	double	if	public	switch	virtual
char	else	inline	register	template	void
class	enum	int	return	this	volatile
const	extern	long	short	throw	while





# Type Data

- Data adalah fakta atau kenyataan yang tercatat mengenai suatu objek yang direpresentasikan dalam bentuk tulisan, suara, gambar, sinyal atau simbol.
- Pengertian data ini menyiratkan suatu nilai yang bisa dinyatakan dalam bentuk konstanta atau variabel sehingga dikatakan tipe data adalah macam / isi data di dalam suatu variabel dalam bahasa program





# Tipe primitive (tipe data dasar)

```
#include <limits.h>
cout << "Bilangan minimum char: "<<CHAR_MIN<<endl; //??
cout << "Bilangan minimum char: "<<UCHAR_MAX<<endl; //??
cout << "Bilangan maximum int : "<<INT_MAX; //??
```

Tipe data	Byte	Batasan
char	1	Bilangan bulat / ASCII antara -128 s.d. 127
unsigned char	1	Bilangan bulat antara 0 s.d. 255
short	2	Bilangan bulat antara -32.768 s.d. 32.767 (- $2^{15}$ s.d $2^{15}-1$ )
unsigned short	2	Bilangan bulat antara 0 s.d. 65.535 (0 s.d $2^{16}-1$ )
int	4	Bilangan bulat antara -2.147.483.648 s.d. 2.147.483.647 (- $2^{31}$ s.d $2^{31}-1$ )
unsigned int / unsigned	4	Bilangan bulat antara 0 s.d $2^{32}-1$
long int	4	Bilangan bulat antara - $2^{31}$ s.d $2^{31}-1$
unsigned long int	4	Bilangan bulat antara 0 s.d $2^{32}-1$
float	4	Bilangan real antara - 3.4 E+38 s.d. 3.4E+38 (7 digit presisi)
double	8	Bilangan real antara -1.7E+308 s.d. 1.7E+308 (15 digit presisi)
bool	1	true / false Tidak semua compiler yang support





# Variabel

- **Variabel** adalah item yang digunakan data untuk menyimpan pernyataan objek. Variabel memiliki **tipe data** dan **nama**.
- Tipe data menandakan tipe nilai yang dapat dibentuk oleh variabel itu sendiri.
- **Nama variabel** harus mengikuti aturan untuk identifier.

**Deklarasi :**

**tipe namaVar1, namaVar2 [=nilaiAwal]**

contoh :

1. Variable karakter

char var1=10, var2, var3;

2. Variable string

char kata[10]; /\* panjang karakter maksimum = 10

3. Variable integer, float

int bulat=5; /\* berisi bilangan bulat \*/  
float bil\_real=0.5;

Perhatikan :

int x=y=5 ; // ERROR

seharusnya : int x=5, y=5;

tetapi :

int x,y;

x=y=5; // DIBOLEHKAN disebut CHAINED  
**ASSIGNMENT**





# Petunjuk Membuat nama variabel

1. Sangat baik untuk menginisialisasi variabel yang dibuat
2. Gunakan nama yang bersifat menggambarkan deskriptif untuk variabel yang dibuat, Misalkan jika ingin mempunyai variabel yang terdiri atas nilai siswa, beri nama dengan nama **grade / nilai** dan jangan hanya beberapa huruf random

Contoh :

```
double nilai=0.0;      //jangan double a=0.0;  
int num = 10;
```

contoh : deklarasi variable dan pencetakannya

```
main()  
{  
    int num;  
    num=10;  
    cout << num << endl;  
}
```





# Konstanta

**const type nama = nilai\_konst;**  
**Atau**  
**#define nama nilai\_konst**

Contoh :  
const double phi=3.1415;

Jenis Konstanta	Contoh
Konstanta desimal	x = 10;
Konstanta hexadesimal	x = 0x1B; x = 0x10;
Konstanta Octal	x = 015;
Konstanta Float	x = 20e6; x = 6.5536E-6;
Konstanta Long	x = 30000L; x = 30000l;
Konstanta karakter	x = 'c';





# Konstanta Karakter dengan Backslash '\'

Kode	Karakter	ASCII
\a		07
\b	Bell	08
\f	Back space	12
\n	Form feed	10 dan 13
\r	Baris baru	13
\r\n	Carriage Return	00
\0	NULL	92
\\\	Backslash	11
\v	Vertical Tab	34
\"	Tanda kutip dua	44
\'	Tanda kutip satu	--
\o	Konstanta Octal	• -
\x	Konstanta Hexadesimal	





Contoh : Penulisan angka decimal, octal dan heksadesimal

```
//-----
```

```
main{
```

```
    cout << 17;           // hasil = ?  
    cout << 017;          // ??  
    cout << 0x17;          // ??
```

```
}
```

```
//-----
```

Contoh : penulisan string dan penggunaan string.h

```
// Berkas: LiteralString.Cpp
```

```
#include <string.h>
```

```
main(){
```

```
    cout << "abc\ndef";      // hasil = ??  
    cout << "abc\tdef";      // ??  
    cout << "\"Halo\"";      // ??  
    cout << "Panjang kata Unpad =" << strlen("Unpad"); // ??
```

```
}
```





# Operator Aritmatika & Increment / Decrement


```
cout<<"13 % 5 = "<<(13%5); // ??  
cout<<"13 / 5 = "<<(13/5); // ??  
cout<<"13.0/5 ="<<(13.0/5); //??
```

Simbol	Fungsi	Contoh Penggunaan	Print
++	Increment	x++ artinya $x=x+1$ ++x artinya $x=x+1$	Cetak dahulu kemudian tambah Tambah dahulu kemudian cetak
--	Decrement	x-- artinya $x=x-1$ --x artinya $x=x-1$	Cetak dahulu kemudian kurang Kurang dahulu kemudian cetak





**Contoh : Increment**

```
int m=66;  
m++;  
cout << m;      // ??
```

**Contoh : Pre Increment**

```
int m=66, n;  
n= ++m;  
cout << "m = "<<m<< " n= "<<n ; // 67, 67
```

**Perhatikan :**

```
m=m+1          //m=67  
n=m            //n=67
```

**Contoh : Post Increment**

```
int m=66, n;  
n= m++;  
cout << "m = "<<m<< " n= "<<n ; // 66, 67
```

**Perhatikan :**

```
n=m;          //n=66  
m=m+1         //m=67
```





# Operator Assignment

Simbol	Keterangan
<code>-=</code>	$x = x - y$ dapat ditulis <code>x -= y</code>
<code>+=</code>	$x = x + y$ dapat ditulis <code>x += y</code>
<code>/=</code>	$x = x / y$ dapat ditulis <code>x /= y</code>
<code>*=</code>	$x = x * y$ dapat ditulis <code>x *= y</code>
<code>%=</code>	$x = x \% y$ dapat ditulis <code>x %= y</code>

Contoh :

```
int nilai=50;  
nilai += 20;  
cout << "nilai = "<<nilai;           // hasil nilai=70
```

Contoh :

```
char c='A';      //atau c=65 dan pakai ' bukan ''  
cout << c++ << " " << int(c);    // hasil 'A' dan 65  
cout << c++ << " " << int(c);    // hasil 'B' dan 66
```

**Perhatikan setelah dicetak lalu increment**





# Operator Logika

Simbol	Keterangan
&&	AND
	OR
!	Not

```
Contoh :    int nilai = 70;  
             if (nilai>=60 && nilai <=100)  
                 cout<<"Lulus ";
```





# Operator Bitwise

Simbol	Keterangan
	OR
^	XOR
~	Complement
>>	Shift Right (geser kanan)

Contoh :

```
int a = 13;           // biner : 1101
int b = 6;           // biner : 0110
cout<<"a & b =" << (a&b); // biner : 0100 = 4
```

Tabel Kebenaran

X	Y	AND (&)	OR ( )	XOR (^)
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0





# Operator Shift

Bentuk umum dari operator shift :

variabel >> nomor posisi bit  
variable << nomor posisi bit

Contoh :

```
int a = 13;           // biner : 1101
cout<<"a << 1 = "<<(a<<1); // biner : 11010 = 26
```





# Type casting (Pengarah type)

## Deklarasi

```
typeData ( var);
```

- Perubahan tipe ke yang lebih besar ukurannya, jika ke yang lebih kecil akan terjadi pemotongan nilai.

Contoh :

```
int jumlah=10, nData=3;  
float rata;  
cout<<(jumlah / nData); // hasil = 3  
rata=float (jumlah) / nData; // hasil = 3.333  
cout<<rata;
```





# I/O stream manipulator

Manipulator	Keterangan
	Mengkonversi menjadi bilangan desimal Mengkonversi menjadi bilangan hexadesimal Mengkonversi menjadi bilangan octal Baris baru atau sama saja dengan \n dan flush stream Menyisipkan null zero kedalam string flush output stream Mengkonversi ke base n (0 untuk desimal default, 8 untuk octal , 16 untuk hexadesimal) Menghilangkan format dengan format flag yang terdapat pada tabel format flag) setting format sesuai dengan tabel format flag. setting isi dengan karakter setting floating point dengan n presisi setting lebar field





# Format flag untuk resetiosflags dan setiosflags

Nama format flag	Keterangan
ios::left	Output rata kiri dengan setw()
ios::right	Output rata kanan dengan setw()
ios::scientific	Format output dengan notasi scientific
ios::fixed	Format angka dengan desimal format
ios::dec	Format menjadi bilangan desimal
ios::hex	Format menjadi bilangan hexadesimal
ios::oct	Format menjadi bilangan oktal
ios::uppercase	Merubah semua hexadesimal dan notasi scientific menjadi huruf besar (contoh 1.22E+03)
ios::showbase	Mencetak base prefix (0x untuk hexadesimal atau 0 untuk octal)
ios::showpoint	Mencetak titik untuk menunjukkan presisi





# Alur pemrograman (Input-Proses-Output)

panjang = 5  
Lebar = 3

**Input**

luas = panjang \* Lebar

**Proses**

luas = 15

**Output**

```
cin >> panjang;  
cin >> lebar;
```

```
luas = panjang * lebar;
```

```
cout << "luas :" << luas << endl;
```

```
#include <iostream>  
#include <stdlib.h>  
  
using namespace std;  
int main() {  
    int panjang, lebar, luas ;  
  
    cin >> panjang;  
    cin >> lebar;  
    luas = panjang * lebar;  
    cout << "luas :" << luas << endl;  
    system("pause");  
    return 0;  
}
```





# Input (Masukan) dan Output (Keluaran)

- Contoh : Menggunakan style bahasa C (**dengan scanf dan printf**)

```
/* Nama program      : tulisInt.c
   Nama              : Akmal
   NPM               : 007
   Tanggal buat     : 15/8/2015
   Deskripsi         : Contoh membaca dan menulis nilai
                       integer dgn bahasa C
 ****
# include <stdio.h>
main() {
    /* KAMUS */
    int a;
    /* ALGORITMA */
    printf("Contoh baca dan tulis, ketik integer :");
    scanf("%d", &a);
    printf("Nilai yang dibaca : %d \n", a);
    return 0;
}
```





## Contoh : Menggunakan style C++ (**dengan cin dan cout**)

```
//Program Baca.cpp
// Deskripsi  : Contoh membaca dan menulis nilai integer
//              dengan bahasa C++
//*****  

# include <iostream.h>
main() {
    // KAMUS
    int a;
    // ALGORITMA
    cout << "Contoh baca dan tulis, ketik integer :";
    cin >>a;
    cout << "Nilai yang dibaca : " << a;
}
```





## Contoh : Input string dengan getline() □ type char dan string

```
#include <iostream>
#include <string>
using namespace std;

main() {
    // KAMUS
    string nama1;
    char nama2[50], alamat[50];

    // ALGORITMA
    cout << "Contoh baca dan tulis" << endl;
    cout << "Ketik Nama anda: "; getline (cin, nama1);
    cout << "Nama anda adalah : " << nama1 << endl;

    cout << "Ketik Alamat anda: "; cin.getline (alamat, 50);
    cout << "Nama anda adalah : " << alamat << endl

    cout << "Ketik Nama anda lagi : "; cin >> nama2;
    cout << "Nama anda adalah : " << nama2 << endl;
}
```





## Contoh : Penggunaan I/O Manipulator untuk format output

```
#include <iostream.h>
int main() {
    double pi = 3.141592654;
    // Tampilan default: left justified, presisi 6.
    cout << pi << endl;
    // Ubah dg precision 4, lebar field 12, isi dg #
    cout.precision(4);
    cout.width(12);
    cout.fill('#');
    cout << pi << endl;
    // Ubah presisi ke 10
    cout.precision(10);
    cout << pi << endl;
    return 0;
}
```





# Contoh : iomanip

```
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
main() {
    int num = 37;
    double pi = 3.141592654;
    cout << "hex: " << hex << num << endl;
    cout << "oct: " << oct << num << endl;
    cout << "dec: " << dec << num << endl;
    cout << setw(8) << num << endl;
    cout << setw(8) << setfill(' ') << num << endl;
    cout << "Pi: " << setprecision(10) << pi << endl;
    getch();
}
```





# LATIHAN DAN TUGAS

1. Tuliskan 4 perintah berbeda yang mengurangi nilai n sebanyak 1.

2. Evaluasi setiap ekspresi berikut, misalkan nilai m=24 dan nilai n=7

- a.  $m - 8 - n$
- b.  $m = n = 3$
- c.  $m \% n$
- d.  $m \% n ++$
- e.  $m \% ++ n$
- f.  $.++m - n -$
- g.  $m += n -= 2$ .

3. Tuliskan jadi satu perintah dari dua operasi berikut (*lihat Operator assignment*):

$$z = z - (x+y)$$

$$y = y+1$$

4. Tentukan hasilnya dan uraikan perintah berikut dengan nilai m=5 dan n=2

- a.  $m *= n ++;$
- b.  $m += --n;$

5. Tentukan hasil dari perintah-perintah berikut:

```
int x, y, z;
```

```
x = y = z = 5;
```

```
x *= y += z -= 1;
```

6. Seorang programmer menuliskan pernyataan sebagai berikut, setelah mendefinisikan semua nama yang dipakai sebagai tipe real/float

```
dua = tujuh + lima
```

```
Cinta = Toto + Tita
```

Apa komentar anda ?





# LATIHAN DAN TUGAS

Buatlah spesifikasi dan algoritma untuk persoalan-persoalan berikut. Gunakan notasi pseudocode kemudian implementasikan menggunakan Bahasa C++;

1. Dibaca nama dan jam kerja pegawai, harus dihitung honor pegawai tersebut jika upahnya perjam adalah Rp. 5.000,- Perhatikan bahwa upah perjam setiap pegawai tidak sama, dan perubahan upah tidak sesering perubahan kurs.
2. Dibaca 3 buah bilangan bulat yang mewakili tiga buah tahanan dalam Ohm : R1, R2 dan R3, harus dihitung dan dituliskan tahanan total yang dihasilkan jika ketiganya dipasang seri dan parallel.





# Panduan Instalasi C++ di VSCode

- <https://code.visualstudio.com/docs/languages/cpp>





# Jawaban

1. Tuliskan 4 perintah berbeda yang mengurangi nilai n sebanyak 1.

- angka = angka -1 ;
- angka-- ;
- --angka ;
- angka -=1 ;

2. Evaluasi setiap ekspresi berikut, misalkan nilai m=24 dan nilai n=7

- a.  $m - 8 - n \Rightarrow 24 - 8 - 7 = 9$
- b.  $m = n = 3 \Rightarrow m=3 \text{ dan } n=3$
- c.  $m \% n \Rightarrow 3$
- d.  $m \% n++ \Rightarrow 3$
- e.  $m \% ++n \Rightarrow 0$
- f.  $++m - n-- \Rightarrow 18$
- g.  $m += n -= 2 \Rightarrow m=29 \text{ dan } n=5$





# Jawaban

1. Tuliskan jadi satu perintah dari dua operasi berikut (*lihat Operator assignment*):

- $z = z - (x+y) \Rightarrow z = (x+y)$
- $y = y + 1 \Rightarrow y = 1$

4. Tentukan hasilnya dan uraikan perintah berikut dengan nilai  $m=5$  dan  $n=2$

- a.  $m *= n++; \Rightarrow 10$
- b.  $m += --n; \Rightarrow 6$

**Perintah  $m *= n++;$**

**$n++$  adalah operator post-increment. Ini berarti nilai  $n$  digunakan dalam operasi terlebih dahulu, dan kemudian  $n$  di-increment.**

Langkah-langkahnya:

1. Gunakan nilai  $n$  yang saat ini adalah 2 dalam operasi  $m *= n++$ .
2.  $m *= n$  berarti  $m = m * n$ . Dengan nilai awal  $m = 5$  dan  $n = 2$ , perintah ini akan menghitung  $m = 5 * 2 = 10$ .
3. Setelah operasi,  $n$  di-increment menjadi 3.
4. Jadi, setelah perintah ini:
  - $m = 10$
  - $n = 3$

**Perintah  $m += --n;$**

**$--n$  adalah operator pre-decrement. Ini berarti nilai  $n$  dikurangi terlebih dahulu, dan kemudian nilai baru dari  $n$  digunakan dalam operasi.**

Langkah-langkahnya:

1.  $--n$  mengurangi nilai  $n$  dari 2 menjadi 1, dan kemudian nilai 1 digunakan dalam operasi.
2.  $m += --n$  berarti  $m = m + --n$ . Dengan nilai  $m = 5$  dan  $--n = 1$ , perintah ini akan menghitung  $m = 5 + 1 = 6$ .
3. Jadi, setelah perintah ini:
  - $m = 6$
  - $n = 1$





# Jawaban

5. Tentukan hasil dari perintah-perintah berikut:

- `int x, y, z;`
- `x = y = z = 5; =>x=5, y=5, z=5`
- `x *= y += z -= 1; => 45`

Perhatikan urutan penggeraan:

- **`z -= 1`**: Mengurangi nilai z dengan 1.
  - z sebelumnya adalah 5, jadi `z -= 1` menjadi 4.
  - Sekarang  $z = 4$ .
- **`y += z`**: Menambahkan nilai z yang baru ke y.
  - y sebelumnya adalah 5, dan z yang baru adalah 4.
  - Jadi `y += 4` menjadi 9.
  - Sekarang  $y = 9$ .
- **`x *= y`**: Mengalikan nilai x dengan y yang baru.
  - x sebelumnya adalah 5, dan y yang baru adalah 9.
  - Jadi `x *= 9` menjadi 45.
  - Sekarang  $x = 45$ .





1. Seorang programmer menuliskan pernyataan sebagai berikut, setelah mendefinisikan semua nama yang dipakai sebagai tipe real/float
  - dua = tujuh + lima
  - Cinta = Toto + Tita
  - Apa komentar anda ?
  - Jawaban : Kedua peryataan tersebut valid, namun nama variabel seperti dua, tujuh, lima, Cinta, Toto, dan Tita tidak sejalan dengan konvensi penamaan yang umum. Dalam pemrograman praktis terbaik, nama variabel sebaiknya deskriptif dan mencerminkan tujuan atau tipe data yang diwakili.





# Soal 7

Dibaca nama dan jam kerja pegawai, harus dihitung honor pegawai tersebut jika upahnya perjam adalah Rp. 5.000,- Perhatikan bahwa upah perjam setiap pegawai tidak sama, dan perubahan upah tidak sesering perubahan kurs.

**Algoritma Hitung\_Honor\_Pegawai**

{Menghitung honor pegawai berdasarkan upah per jam}

**Deklarasi**

```
string nama
int jamKerja
int upahPerJam , honor
```

**Deskripsi**

INPUT nama

INPUT jamKerja

upahPerJam = 5000

honor = jamKerja \* upahPerJam

OUTPUT "Nama pegawai: " + nama

OUTPUT "Honor pegawai: " + honor

```
/* Nama program      : Hitung_Honor_Pegawai
   Nama              : Erick
   NPM               :
   Tanggal buat    : 13 Sep 2024
   Deskripsi        : Menghitung honor pegawai berdasarkan
                      upah per jam
 *****/
#include <iostream>
#include <string>
using namespace std;

int main() {
    string nama;
    int jamKerja;
    const int upahPerJam = 5000;

    cout << "Masukkan nama pegawai: ";
    getline(cin, nama);
    cout << "Masukkan jam kerja: ";
    cin >> jamKerja;

    int honor = jamKerja * upahPerJam;

    cout << "Nama pegawai: " << nama << endl;
    cout << "Honor pegawai: Rp " << honor << endl;

    return 0;
}
```



# Soal 8

Dibaca 3 buah bilangan bulat yang mewakili tiga buah tahanan dalam Ohm : R1, R2 dan R3, harus dihitung dan dituliskan tahanan total yang dihasilkan jika ketiganya dipasang seri dan parallel

**Algoritma Hitung\_Tahanan\_Total**

{Menghitung Tahanan Total berdasarkan tiga buah tahanan}

**Deklarasi**

int R1, R2, R3

**Deskripsi**

INPUT R1

INPUT R2

INPUT R3

tahananSeri = R1 + R2 + R3

tahananParallel =  $1 / ((1 / R1) + (1 / R2) + (1 / R3))$ 

OUTPUT "Tahanan total seri: " + tahananSeri

OUTPUT "Tahanan total paralel: " + tahananParallel

```
/* Nama program      : Hitung_Tahanan_Total
Nama                 : Erick
NPM                  :
Tanggal buat : 13 Sep 2024
Deskripsi  : Menghitung Tahanan Total berdasarkan tiga buah tahanan
***** */
#include <iostream>
using namespace std;

int main() {
    int R1, R2, R3;

    cout << "Masukkan tahanan R1 (Ohm): ";
    cin >> R1;
    cout << "Masukkan tahanan R2 (Ohm): ";
    cin >> R2;
    cout << "Masukkan tahanan R3 (Ohm): ";
    cin >> R3;

    int tahananSeri = R1 + R2 + R3;
    double tahananParallel = 1.0 / ((1.0 / R1) + (1.0 / R2) + (1.0 / R3));

    cout << "Tahanan total seri: " << tahananSeri << " Ohm" << endl;
    cout << "Tahanan total paralel: " << tahananParallel << " Ohm" << endl;

    return 0;
}
```



UNPAD  
LUHUNG **LIVE**

**UNPAD  
LUHUNG**

The logo consists of the words "UNPAD" and "LUHUNG" in a dark blue sans-serif font. The letters are stylized with yellow diagonal bars and arrows pointing upwards and to the right.

**LIVE**

The word "LIVE" is written in a large, bold, sans-serif font. Each letter is a different color: L is orange, i is green, V is light blue, and E is pink. A small green circle is positioned above the letter "i".

Terima kasih

Tenaga Pengajar :

- Erick Paulus, S.Si.,
- Dr. Akmal, S.Si, MT