



DASAR PEMROGRAMAN

- **Notasi Algoritma**
- **Separator**
- **Special Character**
- **Kesalahan pada program**



Algoritma

- Program mengandung suatu algoritma (method yang digunakan untuk menyelesaikan suatu permasalahan)
- Algoritma untuk menambahkan 2 angka dapat saja berbentuk seperti ini:
 - Meminta pengguna untuk menginputkan angka pertama
 - Meminta pengguna untuk menginputkan angka kedua
 - Menambahkan kedua buah angka
 - Menampilkan hasilnya ke layar

Algoritma

- Algoritma bukanlah kode yang sebenarnya, namun algoritma menggambarkan gagasan/rencana dari program.
- Seringkali programmer mendesain programnya dalam pseude-code dan kemudian menggunakannya untuk menulis kode yang sebenarnya.

Algoritma

- **Algoritma** adalah:
 - penyusunan aspek proses logika dari suatu pemecahan masalah tanpa melihat karakteristik bahasa pemrograman yang akan digunakan
 - urutan notasi logika yang merupakan hasil analisis dan rancangan sistematis dari strategi pemecahan masalah, untuk menggambarkan urutan langkah kerja yang jika dikerjakan akan membawa ke tujuannya.
 - urutan logika langkah kerja untuk menyelesaikan suatu masalah.

Notasi Algoritma

- Notasi I : untaian kalimat deskriptif
 - Notasi II : diagram alir (flow chart)
 - Notasi III : pseudo-code
- Contoh masalah :
- Menghitung luas segiempat.

Algoritma Luas_Segiempat

Menghitung luas segiempat dengan memasukkan nilai lebar dan panjang segiempat

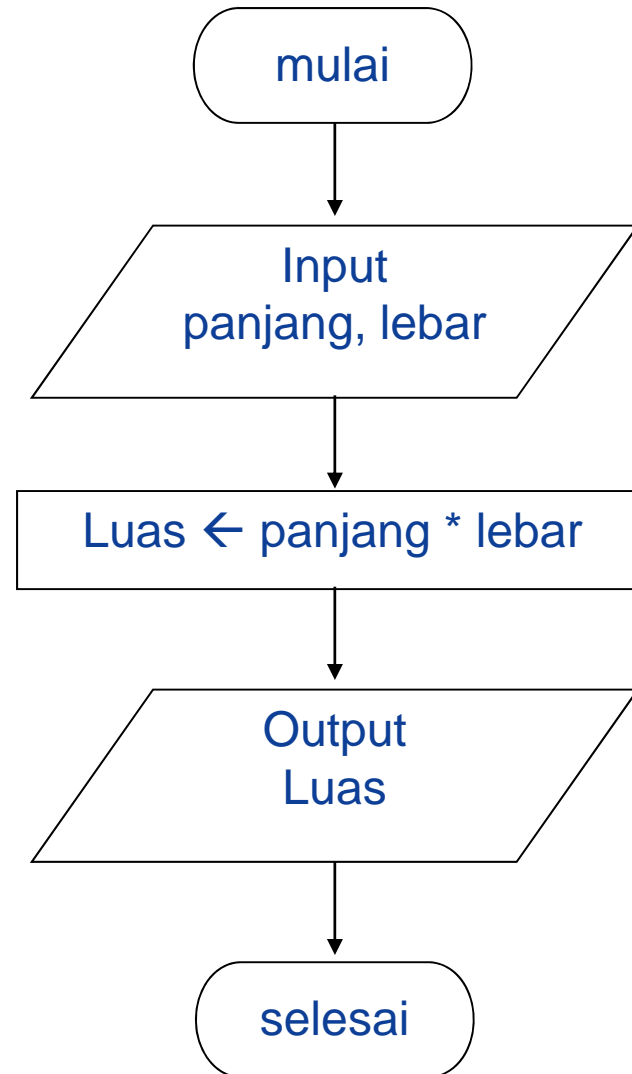
Deklarasi

luas,panjang,lebar : <bil. bulat>

Deskripsi

1. Masukkan nilai lebar dan panjang
2. Hitung luas dengan mengalikan panjang dan lebar
3. Tampilkan Luas

Notasi II : diagram alir (flow chart)



Notasi III : pseudo-code

Algoritma Luas_Segiempat

{Menghitung luas segiempat dengan memasukkan nilai lebar dan panjang segiempat}

Deklarasi

luas, panjang, lebar : integer

Deskripsi




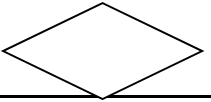


input(panjang)

input(lebar)

luas \leftarrow panjang * lebar

output(luas)

Simbol-simbol Flowchart

No.	Simbol	Gambar	Keterangan
1	Terminal		Digunakan untuk menunjukkan awal dan akhir dari program
2	Persiapan		Digunakan untuk memberikan nilai awal pada suatu variabel
3	Pengolahan/Proses		Digunakan untuk pengolahan aritmatika dan pemindahan data
4	Keputusan/Kondisi/ Percabangan		Digunakan untuk mewakili operasi perbandingan logika
5	Input/Output		Digunakan untuk menyatakan proses input/baca dan output/tulis
6	Garis		Digunakan untuk menyatakan urutan pelaksanaan, atau alur proses

Program

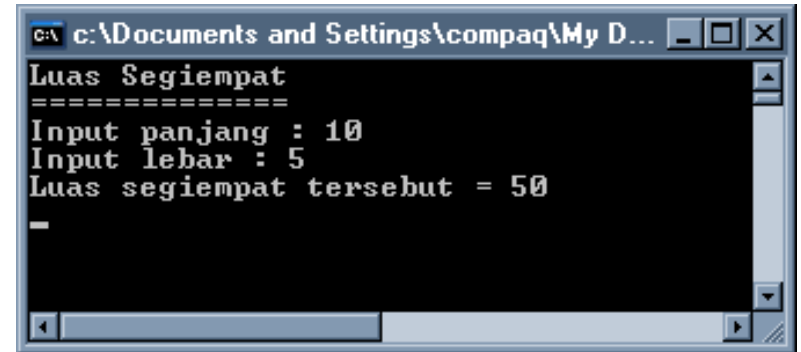
- adalah perwujudan atau implementasi teknis Algoritma yang ditulis dalam bahasa pemrograman tertentu sehingga dapat dilaksanakan oleh komputer.

Program Luas Segiempat

```
// Program Luas Segiempat
#include <iostream>
#include <conio.h>

using namespace std;

void main ()
{
    int p,l,Luas;
    cout<<"Luas Segiempat"<<endl;
    cout<<"======"<<endl;
    cout<<"Input panjang : ";
    cin>>p;
    cout<<"Input lebar : ";
    cin>>l;
    Luas = p * l;
    cout<< "Luas segiempat tersebut = "<< Luas <<endl;
    _getch();
}
```



```
c:\Documents and Settings\compaq\My D...
Luas Segiempat
====
Input panjang : 10
Input lebar : 5
Luas segiempat tersebut = 50
-
```

Special Character

Special Character	Remarks
<code>\n</code>	New line
<code>\r\n</code>	Carriage return and new line
<code>\t</code>	Tab
<code>\\</code>	Backslash
<code>\"</code>	Double quote
<code>\'</code>	Single quote
<code>\?</code>	Question mark

Comments

- C++ mendukung 2 cara untuk memasukkan baris komentar ke program :

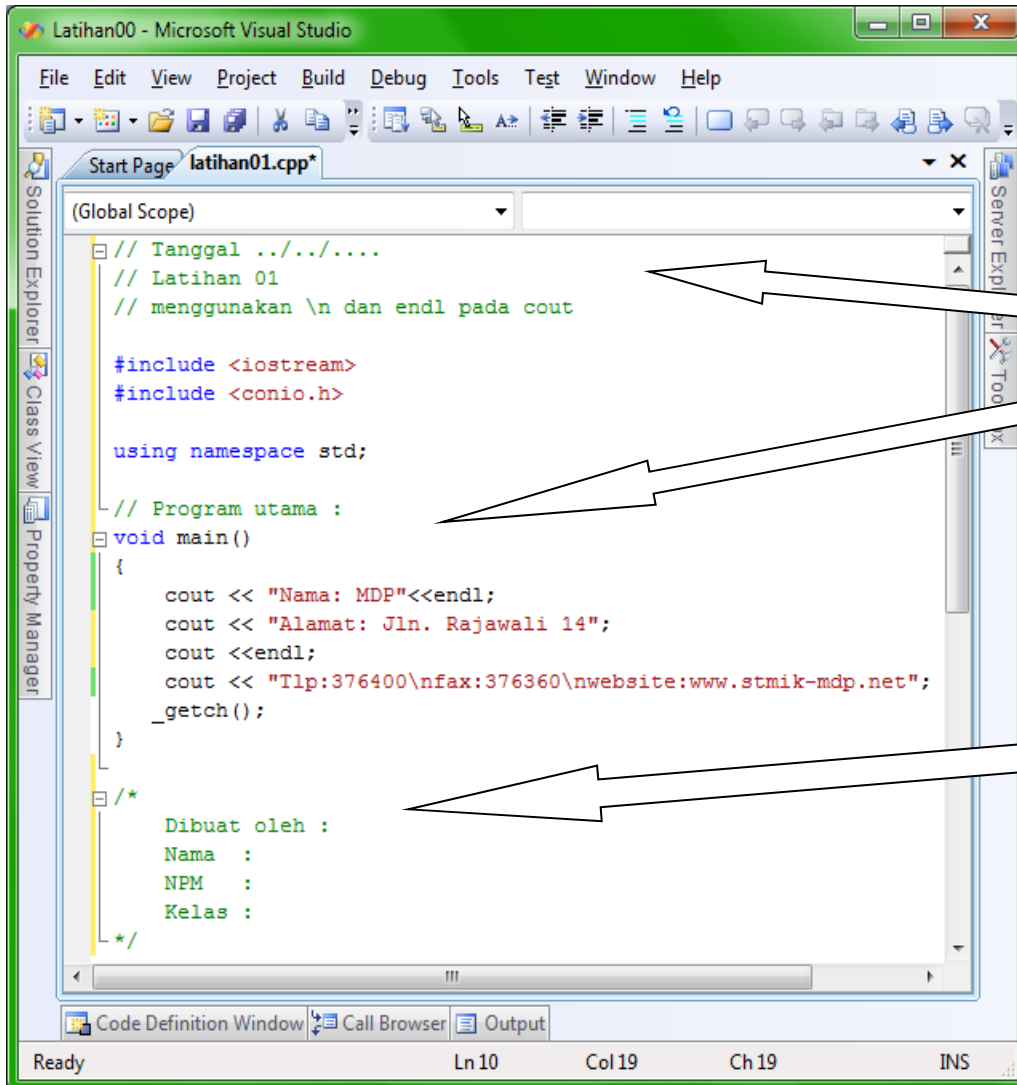
// line comment

/* block comment */

Comments merupakan bagian dari kode program yang akan diabaikan oleh compiler.

Kegunaan comments/komentar adalah mengizinkan programmer untuk memasukkan catatan-catatan atau deskripsi di sekitar kode program

Bagi kompiler hal ini tidak berguna dan akan diabaikan pada saat kompilasi.



The screenshot shows the Microsoft Visual Studio IDE with a C++ file named 'latihan01.cpp'. The code is as follows:

```
(Global Scope)
// Tanggal .././../....
// Latihan 01
// menggunakan \n dan endl pada cout

#include <iostream>
#include <conio.h>

using namespace std;

// Program utama :
void main()
{
    cout << "Nama: MDP"<<endl;
    cout << "Alamat: Jln. Rajawali 14";
    cout <<endl;
    cout << "Tlp:376400\nfax:376360\nwebsite:www.stmik-mdp.net";
    _getch();
}

/*
Dibuat oleh :
Nama :
NPM :
Kelas :
*/
```

Annotations in the image include:

- Two arrows pointing to the first three lines of code, with a callout box labeled "Contoh Line comment".
- Two arrows pointing to the multi-line comment block at the bottom, with a callout box labeled "Contoh block comment".

Contoh Line
comment

Contoh block
comment

Gaya Penulisan Program

- Contoh program sebelumnya telah terstruktur menjadi beberapa baris agar dapat lebih mudah dibaca, namun dalam C++ tidak ada aturan khusus bagaimana cara memisah baris-baris instruksi tersebut..
- Sebagai contoh, selain membuat baris program sebagai berikut :

```
int main ()  
{  
    cout <<"Hello World!!";  
    return 0;  
}
```

Kita dapat menaruh semua kode hanya dalam satu baris, dan hasil outputnya akan tetap sama seperti barisan-barisan kode sebelumnya di atas.

```
int main () { cout <<"Hello World!!"; return 0; }
```

Separator

Simbol	Nama	Penggunaan
()	Kurung	Untuk menghimpun parameter dalam definisi dan pemanggilan <i>method</i> , juga digunakan untuk menyatakan tingkat pernyataan, menghimpun pernyataan, untuk pengaturan alur program, dan untuk menyatakan tipe cast (<i>cast types</i>)
{ }	Kurung Kurawal	Untuk menghimpun nilai yang otomatis dimasukkan ke dalam <i>array</i> , digunakan juga untuk mendefinisikan blok program, untuk cakupan <i>class</i> , <i>method</i> , dan lokal.
[]	Kurung Siku	Untuk menyatakan tipe <i>array</i> dan untuk membedakan nilai <i>array</i> .
;	Titik Koma	Pemisah Pernyataan.
,	Koma	Pemisah urutan <i>identifier</i> dalam deklarasi variabel.
.	Titik	Untuk memisahkan nama paket dari sub-paket dan <i>class</i> dan untuk memisahkan variabel atau <i>method</i> dari variabel referensi.

Kesalahan pada program

- Ada dua jenis kesalahan yang mungkin terjadi pada suatu program :
 1. Kesalahan sintaks/tata bahasa (syntax error)
 2. Kesalahan logika (logical error)

- Kesalahan sintaks menyebabkan program tidak dapat dikompilasi, sedangkan apabila terdapat kesalahan logika pada program, program dapat dikompilasi tetapi jika dijalankan akan menghasilkan keluaran yang salah.

Kesalahan pada program

- Kesalahan-kesalahan pada program, baik kesalahan sintaks maupun logika, disebut dengan **bug**, yang berarti kutu. Istilah ini diambil, karena program-program yang dulu masih menggunakan kartu lubang (punched card) sering salah mengambil data karena lubang-lubang kartu tersebut tertutup kutu.
- Menemukan dan memperbaiki bug dalam program disebut dengan **debugging**.

Latihan

- Buatlah program C++ untuk menampilkan kalimat berikut:

```

*****
Hello World!
How do You Do ??
It's been nice knowing you.
"Goodbye World!"
*****
    
```

Note : gunakan Special Character

Latihan

- Jelaskan apa yang dimaksud dengan kesalahan sintaks dan kesalahan logika dalam sebuah program! Bagaimana cara mengetahui kesalahan sintaks dan kesalahan logika dalam sebuah program?