

---

# 1 KONSEP DASAR PEMROGRAMAN

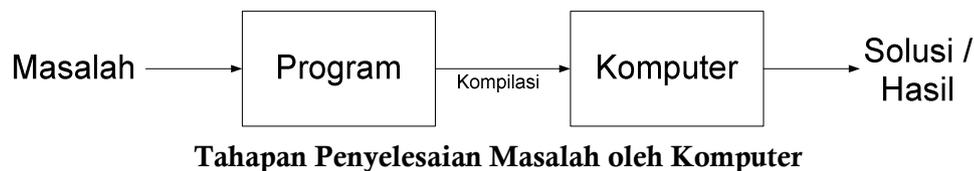
## 1.1 Pengertian Program dan Pemrograman

Komputer merupakan alat bantu penyelesaian masalah di berbagai bidang: Pendidikan, Perbankan, Industri, Penerbangan, Kedokteran, Permainan, dll

Tetapi, permasalahan tidak dapat disodorkan begitu saja ke depan komputer, karena sebenarnya mesin komputer hanyalah benda mati yang tidak bisa apa-apa. Manusia harus merumuskan langkah-langkah pemecahan masalah dalam runtunan instruksi yang dapat dilaksanakan oleh komputer yang disebut program.

**Program** merupakan runtunan atau himpunan instruksi(perintah) tertulis yang ditanamkan ke dalam komputer untuk menyelesaikan masalah.

**Pemrograman** merupakan pengerjaan penulisan instruksi oleh programmer untuk menyelesaikan suatu masalah.



## 1.2 Bahasa Pemrograman

Program harus dibuat dalam bahasa yang dimengerti oleh komputer yang disebut bahasa pemrograman supaya instruksi yang ditulis oleh *programmer* dapat dilaksanakan oleh komputer .

### 1. Bahasa Mesin (*Mnemonic Code*)

Bahasa mesin adalah bahasa yang berisi kode-kode mesin yang hanya dapat diinterpretasikan langsung oleh mesin komputer. Bahasa ini merupakan bahasa level terendah dan berupa kode numerik 0 dan 1.

Keuntungan : Eksekusi cepat

Kerugian : Sangat sulit dipelajari manusia

### 2. Bahasa Assembly

Bahasa assembly adalah bahasa simbol dari bahasa mesin. Setiap kode bahasa mesin memiliki simbol sendiri dalam bahasa assembly. Misalnya ADD untuk penjumlahan, MUL untuk perkalian, SUB untuk pengurangan, dan lain-lain.

Kelebihan : Eksekusi cepat, masih bisa dipelajari daripada bahasa mesin, file kecil

Kekurangan : Tetap sulit dipelajari, program sangat panjang

---

### 3. Bahasa Tingkat Tinggi (*High Level Language*)

Bahasa tingkat tinggi adalah bahasa pemrograman yang lebih tinggi daripada bahasa assembly. Bahasa ini lebih dekat dengan bahasa manusia dan lebih dipahami manusia.

Contoh: Pascal, Basic, Cobol, C, C++, Java, dll.

Keuntungan :

- Mudah dipelajari
- Mendekati permasalahan yang akan dipecahkan
- Kode program pendek

Kerugian : Eksekusi lambat karena ada proses pengubahan perintah dalam bahasa ini ke dalam bahasa mesin oleh **Translator(Penterjemah)**

Bahasa generasi ini disebut juga bahasa generasi ke-3 (3rd Generation Programming Language).

### 4. Bahasa yang berorientasi pada masalah spesifik

Bahasa ini adalah bahasa yang digunakan langsung untuk memecahkan suatu masalah tertentu. Misalnya SQL untuk database. Bahasa ini juga masuk ke bahasa tingkat tinggi. Bahasa ini disebut juga bahasa generasi ke-4 (4th Generation Programming Language).

#### **Translator(Penterjemah)**

Translator berfungsi untuk menterjemahkan program yang ditulis dalam bahasa pemrograman tingkat tinggi ke dalam bahasa mesin. Translator dapat dibedakan menjadi :

- **Interpreter** : menterjemahkan dan mengeksekusi baris per baris instruksi. Contoh bahasa Basic.
- **Compiler** : mengeksekusi program setelah seluruh instruksi diterjemahkan. Contoh bahasa Pascal, C, Ada ,C++, dll.

### 1.3 Belajar memprogram dan Belajar bahasa pemrograman

- Belajar memprogram adalah belajar tentang strategi, metodologi, dan sistematika pemecahan masalah, kemudian menuangkannya dalam suatu notasi tertentu yang mudah dibaca dan dipahami.

Sifat : Pemahaman persoalan, analisis, sintesis

Titik Berat : *Designing Program*

- Belajar bahasa pemrograman adalah belajar memakai suatu bahasa, aturan tata bahasanya, instruksi-instruksinya, tata cara pengoperasian compiler-nya untuk membuat program yang ditulis dalam bahasa itu saja.

Sifat : Keterampilan

Titik Berat : *Coding Program*

Produk yang dihasilkan oleh seorang pemrogram adalah program dengan rancangan yang baik (metodologis, sistematis), yang dapat dieksekusi oleh mesin, berfungsi dengan benar, sanggup melayani segala kemungkinan masukan, dan didukung dengan adanya dokumentasi.

Pengajaran pemrograman titik beratnya adalah membentuk seorang perancang "*designer*" program, sedangkan pengajaran bahasa pemrograman titik beratnya adalah membentuk seorang "*coder*"(juru kode). Pada prakteknya, suatu rancangan harus dapat dikode untuk dieksekusi dengan mesin. Oleh karena itu, belajar pemrograman dan belajar bahasa pemrograman saling komplementer, tidak mungkin dipisahkan satu sama lain

---

Tetapi, karena banyak bahasa pemrograman yang dapat digunakan dan pemilihannya akan sangat tergantung kepada masalah yang dipecahkan, maka belajar memprogram menjadi lebih penting daripada belajar bahasa pemrograman. Keduanya diperlukan tetapi dalam tingkat yang berbeda.

#### 1.4 Langkah-langkah pemrograman Komputer :

1. **Mendefinisikan masalah** : menentukan masalahnya seperti apa, apa saja yang harus dipecahkan dengan komputer, apa masukannya, dan bagaimana keluarannya
2. **Menentukan solusi** : mencari jalan bagaimana masalah tersebut diselesaikan. Jika masalah terlalu kompleks, maka ada baiknya masalah tersebut dipecah menjadi modul-modul kecil agar lebih mudah diselesaikan.
3. **Memilih algoritma** : pilihlah algoritma yang benar-benar sesuai dan efisien untuk permasalahan tersebut
4. **Menulis Program** : menuliskan program dalam salah satu bahasa pemrograman  
Pilihlah bahasa yang sesuai dengan permasalahan, mudah dipelajari, mudah digunakan, dan lebih baik lagi jika sudah dikuasai, memiliki tingkat kompatibilitas tinggi dengan perangkat keras dan platform lainnya.
5. **Menguji program** : Menentukan apakah program berhasil dikompilasi dengan baik ? apakah program dapat menghasilkan keluaran yang diinginkan?  
Langkah keempat dan kelima bisa dilakukan berulang-ulang sampai program diyakini benar-benar sesuai dengan yang diharapkan
6. **Menulis dokumentasi** : Menulis dokumentasi sangat penting agar pada suatu saat jika kita akan melakukan perubahan atau membaca source code yang sudah kita tulis dapat kita ingat-ingat lagi dan kita akan mudah membacanya. Caranya adalah dengan menuliskan komentar-komentar kecil tentang apa maksud kode tersebut, untuk apa, variabel apa saja yang digunakan, untuk apa, dan parameter-parameter yang ada pada suatu prosedur dan fungsi.
7. **Merawat program** : Program yang sudah jadi perlu dirawat untuk mencegah munculnya bug yang sebelumnya tidak terdeteksi. Atau mungkin juga pengguna membutuhkan fasilitas baru yang dulu tidak ada.

**Contoh Permasalahan** : Menghitung dan menampilkan Luas Persegi Panjang

Tahapan penyelesaian :

1. Mendefinisikan Masalah
  - Masukan / Input : panjang, lebar
  - Keluaran / Output : Luas
2. Menentukan Solusi + Memilih Algoritma  
Contoh urutan solusi :
  - Baca input panjang
  - Baca input lebar
  - Hitung Luas = panjang x lebar
  - Tampilkan Luas

---

### 3. Menulis Program

Contoh penulisan program dalam Bahasa Pascal dan Hasilnya

Program	Contoh Tampilan di layar
<pre><b>Program</b> Luas_Persegi_Panjang; <b>Var</b>     panjang, lebar, Luas : <b>integer</b>; <b>Begin</b>     <b>Write</b>('Panjang      :      ');     <b>Readln</b>(panjang);     <b>Write</b>('Lebar          :      ');     <b>Readln</b>(lebar);     Luas := panjang * lebar;     <b>Write</b>('Luas persegi Panjang = ',     Luas);     <b>Readln</b>; <b>End.</b></pre>	<pre>Panjang : 20 Lebar    : 6 Luas persegi Panjang = 120</pre>