

MATA KULIAH : DASAR PEMROGRAMAN
SESI PERTEMUAN : I (SATU)
MATERI : KONSEP DASAR & ALGORITMA
DOSEN : ALUN SUJJADA, S.KOM., M.T

A. PENDAHULUAN

Pada perkembangan teknologi saat ini, komputer adalah sebuah alat (*tool*) yang memegang peranan sangat penting. Sesuai dengan fungsinya komputer berasal dari kata “compute” atau perhitungan, sehingga setiap proses yang dilakukan oleh komputer adalah sebuah perhitungan aritmatika. Teknologi komputer dibagi menjadi 3 (tiga) bagian yaitu perangkat keras (*hardware*), perangkat lunak (*software*) dan pengguna (*user/brainware*). *Hardware* adalah jenis dari perangkat komputer yang dapat disentuh secara fisik, seperti *keyboard*, *monitor*, *mouse*, *harddisk* dan lain sebagainya. Sedangkan *software* yaitu sebuah program komputer yang berfungsi sebagai sarana interaksi antara *user* dengan *hardware*.

Komputer bekerja menggunakan sistem bilangan biner yaitu 0 (nol) dan 1(satu). Hal tersebut akan menyusahkan *programmer* (seseorang yang membuat program) untuk membuat *software*. Oleh karena itu muncul sebuah bahasa pemrograman yang digunakan untuk mempermudah pemberian instruksi kepada mesin (komputer). Pemrograman adalah sebuah teknik yang digunakan untuk mendayagunakan sebuah perangkat komputer agar dapat berfungsi membantu menyelesaikan permasalahan yang dihadapi oleh manusia.

B. PROGRAM

Program adalah kumpulan instruksi komputer, sedangkan metode dan tahapan sistematis dalam program adalah algoritma. Program/pemrograman adalah kumpulan instruksi-instruksi tersendiri yang biasanya disebut source code yang dibuat oleh programmer (pembuat program). Variabel adalah suatu pengenalan (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program. Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variabel bisa diubah-ubah sesuai kebutuhan. Untuk memperoleh nilai dari suatu variabel digunakan pernyataan penugasan (*assignment statement*). Program ditulis dengan menggunakan bahasa pemrograman, dapat disimpulkan bahwa program adalah suatu implementasi dari bahasa pemrograman.

Adapun Paradigma Pemrograman yaitu:

1. Pemrograman Prosedural
 - Berdasarkan urutan-urutan, sekuensial
 - Program adalah suatu rangkaian prosedur untuk memanipulasi data. Prosedur merupakan kumpulan instruksi yang dikerjakan secara berurutan.
2. Pemrograman Fungsional
 - Berdasarkan teori fungsi matematika
 - Fungsi merupakan dasar utama program.
3. Pemrograman Terstruktur
 - Secara berurutan dan terstruktur.
 - Program dapat dibagi-bagi menjadi prosedur dan fungsi. Contoh: PASCAL dan C
4. Pemrograman Modular
 - Pemrograman ini membentuk banyak modul.
 - Modul merupakan kumpulan dari prosedur dan fungsi yang berdiri sendiri. Sebuah program dapat merupakan kumpulan modul-modul. Contoh: MODULA-2 atau ADA
5. Pemrograman Berorientasi Obyek
 - Pemrograman berdasarkan prinsip obyek, dimana obyek memiliki Data/variabel/property dan method/event/prosedur yang dapat dimanipulasi. Contoh: C++, Object Pascal, dan Java.
6. Pemrograman Berorientasi Fungsi
 - Pemrograman ini berfokus pada suatu fungsi tertentu saja. Contoh: SQL (Structured Query Language)
7. Pemrograman Deklaratif
 - Pemrograman ini mendeskripsikan suatu masalah dengan pernyataan daripada memecahkan masalah dengan implementasi algoritma. Contoh: PROLOG

Adapun tahapan dalam pembuatan program adalah:

1. Mendefinisikan masalah

Tentukan masalahnya, apa saja yang harus dipecahkan dengan menggunakan komputer, dan tentukan *input* serta *output*nya.

2. Menemukan solusi

Langkah berikutnya adalah menentukan solusi. Jika masalah terlalu kompleks, maka ada baiknya masalah dipecah menjadi modul-modul kecil agar lebih mudah diselesaikan. Contohnya masalah invers matriks, maka dapat membagi menjadi beberapa modul meminta masukkan berupa matriks bujur sangkar, mencari invers matriks, menampilkan hasil kepada pengguna.

3. Memilih algoritma

Pilihlah algoritma yang benar-benar sesuai dan efisien untuk permasalahan.

4. Menulis program

Pilihlah bahasa yang mudah dipelajari, mudah digunakan, dan lebih baik lagi jika sudah dikuasai, memiliki tingkat kompatibilitas tinggi dengan perangkat keras dan platform lainnya.

5. Menguji program

Menguji program dengan segala macam kemungkinan yang ada, termasuk error-handlingnya sehingga program tersebut akan benar-benar handal dan layak digunakan.

6. Menulis dokumentasi

Caranya adalah dengan menuliskan komentar-komentar kecil tentang apa maksud kode, untuk apa, variabel apa saja yang digunakan, untuk apa, dan parameter-parameter yang ada pada suatu prosedur dan fungsi.

C. JENIS BAHASA PEMROGRAMAN

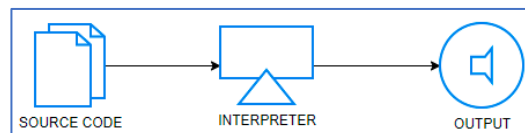
Bahasa pemrograman dikategorikan menjadi 3 jenis yaitu:

1. Bahasa tingkat rendah, seperti bahasa Assembler atau bahasa rakitan
2. Bahasa tingkat menengah, yaitu bahasa campuran antara bahasa mesin dan bahasa simbolik seperti bahasa C.
3. Bahasa tingkat tinggi, yaitu bahasa yang sudah mendekati bahasa manusia seperti C#, Visual Basic, Pascal, Python dan lain-lain.

Python termasuk dari salah satu bahasa pemrograman tingkat tinggi. Pada kenyataannya komputer hanya dapat mengeksekusi bahasa tingkat rendah, jadi bahasa pemrograman tingkat tinggi harus melewati beberapa proses untuk diubah ke bahasa pemrograman tingkat rendah, hal tersebut merupakan kelemahan yang tidak berarti bagi bahasa pemrograman tingkat tinggi. Kekurangan tersebut tidak sebanding dengan kelebihanannya. Pertama, lebih mudah memprogram sebuah aplikasi dengan bahasa tingkat tinggi. Lebih cepat, lebih mudah dimengerti

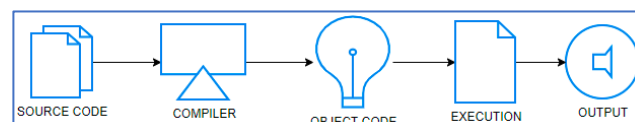
menulis program komputer dengan bahasa tingkat tinggi, dan juga kesalahan dalam penulisan program cenderung tidak mengalami kesalahan yang berarti. Kedua bahasa pemrograman tingkat tinggi lebih portable dalam arti bisa digunakan untuk menulis di berbagai jenis arsitektur komputer (seperti Intel 386, 486, 586, SPARC, RISC/6000) yang berlainan dengan sedikit modifikasi ataupun tidak memerlukan modifikasi sama sekali. Bahasa pemrograman tingkat rendah hanya dapat berjalan di satu jenis arsitektur komputer dan harus ditulis ulang untuk menjalankannya di lain mesin, hal ini dikarenakan karena perbedaan urutan register dan servisnya. Keuntungan tersebut menyebabkan aplikasi - aplikasi komputer ditulis dengan bahasa pemrograman tingkat tinggi. Penggunaan bahasa pemrograman tingkat rendah hanya digunakan pada aplikasi tertentu saja.

Terdapat 2 jenis konsep untuk memproses bahasa tingkat tinggi ke bahasa tingkat rendah, yaitu : compiler dan interpreter. Sebuah interpreter membaca sebuah program yang ditulis dengan bahasa tingkat tinggi dan langsung menjalankannya per baris, memakan waktu sedikit. Ilustrasi proses interpreter seperti pada gambar 1.1 berikut ini:



Gambar 1.1 Proses Interpreter

Sebuah kompiler menerjemahkan keseluruhan kode program sebelum menjalankan program tersebut. Dalam kasus ini kode tersebut disebut sebagai *source code* dan program yang diterjemahkan disebut dengan *object code* atau *executable*. Sekali program tersebut dikompilasi, maka proses eksekusi program dapat berulang kali tanpa menerjemahkannya lagi kedalam *object code*. Ilustrasi proses compiler seperti pada gambar 1.2 berikut ini:



Gambar 1.2 Proses Compiler

C. ALUR EKSEKUSI PROGRAM

Sebuah program adalah sejumlah instruksi yang berisi perintah - perintah dalam bahasa pemrograman komputer untuk menyelesaikan masalah dengan bantuan komputer. Masalah - masalah komputasi tersebut mungkin seperti permasalahan matematika, seperti menyelesaikan sebuah fungsi eksponen, rumus - rumus dalam matematika, tetapi dapat juga berupa mencari dan menggantikan teks, menyusun teks dalam dokumen, dan sebagainya.

Beberapa komponen pada bahasa pemrograman komputer mungkin berbeda-beda, tetapi beberapa instruksi umumnya sama di semua bahasa pemrograman komputer yaitu:

1. *Input*

Masukan dari keyboard, file, atau beberapa device.

2. *Output*

Hasil / keluaran program ke monitor display, file, atau beberapa device.

3. *Math*

Perhitungan matematika atau kalkulasi matematika seperti pengurangan, penjumlahan, perkalian, pembagian dan sebagainya.

4. *Conditional*

Memeriksa beberapa kondisi dan mengeksekusi beberapa perintah tertentu, sesuai dengan kondisi yang telah diperiksa.

5. *Looping*

Menjalankan beberapa perintah secara berulang - ulang kali, biasanya dengan beberapa variasi.

Semua program yang ada, betapapun rumitnya program tersebut dibuat dengan beberapa instruksi yang telah disebutkan di atas, walaupun kelihatannya sebuah program dibagi menjadi modul yang lebih kecil dan dari modul-modul tersebut terbagi lagi menjadi sub modul untuk mengerjakan fungsi-fungsi dasar program tersebut. Hal inilah yang kemudian dikenal dengan istilah Algoritma.

D. DEFINISI ALGORITMA

Beberapa definisi algoritma adalah:

- Urutan aksi-aksi yang dinyatakan dengan jelas dan tidak rancu untuk memecahkan suatu masalah dalam rentang waktu tertentu. Setiap aksi harus dapat dikerjakan dan mempunyai efek tertentu.
- Algoritma adalah barisan langkah-langkah perhitungan dasar yang mengubah masukan (dari beberapa fungsi matematika) menjadi keluaran.
- Algoritma adalah urutan aksi-aksi yang dinyatakan dengan jelas dan tidak rancu untuk memecahkan suatu masalah dalam rentang waktu tertentu.
- Algoritma merupakan logika, metode dan tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan.

Algoritma secara umum direpresentasikan kedalam 3 bentuk yaitu:

1. Uraian Deskripsi yaitu menceritakan sesuai urutan logika menggunakan bahasa sehari-hari.
2. Flowchart yaitu menyusun algoritma menggunakan bahasa simbolis yang sudah menjadi kesepakatan bersama
3. Pseudocode yaitu menyusun algoritma dengan bahasa yang menyerupai bahasa kode program yang bersifat general.

Kriteria Algoritma Menurut Donald E. Knuth

1. Input: algoritma dapat memiliki nol atau lebih inputan dari luar.
2. Output: algoritma harus memiliki minimal satu buah output keluaran.
3. Definiteness (pasti): algoritma memiliki instruksi yang jelas dan tidak ambigu.
4. Finiteness (ada batas): algoritma harus memiliki titik berhenti (stopping role).
5. Effectiveness (tepat dan efisien): algoritma sebisa mungkin harus dapat dilaksanakan dan efektif. Contoh instruksi yang tidak efektif adalah: $A = A + 0$ atau $A = A * 1$

Jenis proses algoritma, yaitu:

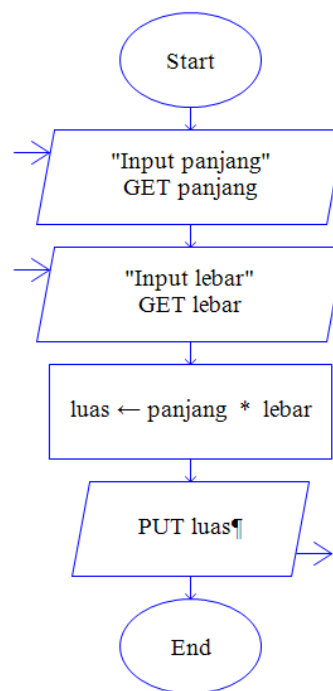
1. Sequence Process: instruksi dikerjakan secara sekuensial, berurutan.
2. Selection Process: instruksi dikerjakan jika memenuhi kriteria tertentu
3. Iteration Process: instruksi dikerjakan selama memenuhi suatu kondisi tertentu.
4. Concurrent Process: beberapa instruksi dikerjakan secara bersama

Berikut ini adalah contoh algoritma untuk menghitung luas persegi panjang dengan menggunakan :

A. Uraian Deskripsi

- Deklarasikan variabel *panjang*, *lebar* dan *luas*
- Isikan nilai variabel *panjang* dan *lebar*
- Lakukan perkalian antara *panjang* dan *lebar*, kemudian tampung nilainya pada variabel *luas*
- Tampilkan nilai dari variabel *luas*

B. Flowchart



Gambar 1.3 Flowchart menghitung luas persegi panjang

C. Pseudocode



```
program hitung_luas_persegi

deklarasi
var panjang,lebar : integer;
var luas:integer;

algoritma:
read(panjang,lebar); {input user}
luas < panjang * lebar;

write(luas);
```

Gambar 1.4 Pseudocode menghitung luas persegi panjang