

Pengenalan Pemrograman Prosedural

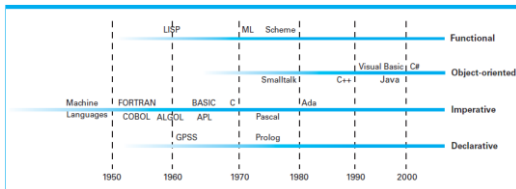
Dasar Pemrograman

Tujuan kuliah Dasar Pemrograman

- memahami cara kerja program komputer
- memecahkan masalah dengan paradigma prosedural dan menuliskan spesifikasi serta menuliskan algoritmanya tanpa tergantung bahasa pemrograman.
- Menulis algoritma dengan metodologi dan skema standar yang diajarkan.
- Menulis program dengan baik

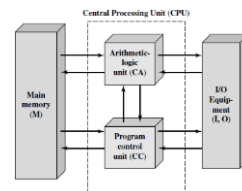
Paradigma Pemrograman

- Procedural/Imperative
- Object Oriented
- Declarative
- Functional



Konsep mesin von new mann

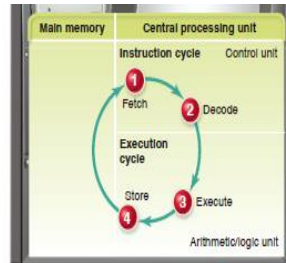
- Desain mesin komputer berdasarkan konsep mesin **John Von Neumann**
- Data dan instruksi disimpan di dalam memory
- instruksi dibaca dan dijalankan oleh Control Unit secara berurutan.



Proses pada CPU

- Pada setiap instruksi, control unit melakukan empat operasi dasar, yang dikenal dengan *machine cycle*.

1. Mengambil instruksi
2. decode instruksi
3. Eksekusi instruksi
4. Menyimpan hasil



Sistem Biner

- Merupakan *digit* yang terdiri dari 1 dan 0
- Komputer merupakan perangkat elektrik yang hanya mengenal sinyal on atau off saja.
- Ada arus on atau 1, tidak ada arus off atau 0
- Sistem biner merupakan bahasa pada komputer
- CPU melakukan eksekusi terhadap instruksi yang diberikan. Instruksi hrs dalam bentuk bits (bhs mesin) dg pola tertentu.

Paradigma procedural

- Cara menyelesaikan persoalan pemrograman dengan melakukan dekomposisi "aksional" suatu solusi menjadi aksi yang akan dijalankan secara berurutan (instruksi-instruksi).
- pemrograman *procedural* terdiri dari algoritma dan struktur data.
- Paradigma ini didasari oleh konsep mesin von newmann (stored program concept), memory utk menyimpan instruksi dan memory utk menyimpan data.
- Program dalam paradigma ini didasari pada **strukturasi informasi** di dalam memori dan **manipulasi** dari informasi yang disimpan tersebut.

Definisi program

- terdiri dari instruksi – instruksi elektronik yang memberitahu komputer untuk melakukan tugas-tugas yang diperintahkan untuknya.
- Program terdiri dari urutan instruksi yang telah di encode sebagai numeric digit.
- Encoding system dikenal dengan machine language.

pemrograman prosedural(imperatif): berbentuk "AKSI":

- Aksi memiliki IS,FS, dan dalam bekerjanya memiliki batas waktu.
- Aksi dapat dibagi menjadi sub-aksi
- Aksi: diterjemahkan menjadi sederetan instruksi yang dapat dijalankan oleh mesin.

Belajar mem-program

- Belajar mem-program adalah belajar tentang strategi pemecahan masalah, metodologi dan sistematika pemecahan masalah tersebut kemudian menuangkannya dalam suatu notasi yang disepakati bersama.
- Belajar memrogram lebih bersifat pemahaman persoalan, analisis, sintesis

Belajar bahasa pemrograman

- . Belajar bahasa pemrograman adalah belajar memakai suatu bahasa, aturan sintaks (tata bahasa), setiap instruksi yang ada, dan tata cara pengoperasian kompilator bahasa yang bersangkutan pada mesin tertentu.
- belajar bahasa pemrograman adalah belajar untuk memanfaatkan instruksi-instruksi dan kiat yang dapat dipakai secara spesifik hanya pada bahasa itu.
- Belajar bahasa pemrograman lebih bersifat keterampilan daripada analisis dan sintesa

pemrograman

- hakekatnya, penggunaan komputer untuk memecahkan persoalan adalah untuk tidak mengulang-ulang kembali hal yang sama.
- Strategi pengenalan masalah melalui dekomposisi, pemakaian kembali modul yang ada, sintesa, selalu dipakai untuk semua pendekatan, dan seharusnya mendasari semua pengajaran pemrograman.
- Karena itu,perlu diajarkan metodologi, terutama karena sebagian besar pemrogram pada akhirnya memakai rutin-rutin yang ditulis orang lain. Memang ada algoritma baru yang lahir, tapi relatif lebih sedikit dibandingkan dengan bongkar pasang program yang sudah ada.

contoh:

aksi "mengupas kentang"

- is: T_0 , kentang berada dalam kantong di rak dapur.
- fs: T_1 , kentang terkupas di panci dan siap dimasak, kantong dikembalikan ke rak di dapur.
- sub-aksi:
 1. ambil kantong kentang dari rak
 2. ambil panci dari lemari
 3. kupas kentang
 4. letakkan kentang di panci
 5. kembalikan kantong kentang ke rak dapur.
- aksi 1 dan 2 bebas urutan, bisa digabungkan
- aksi 3, 4 bisa digabungkan

aksi "mengupas kentang"

- is: T_0 , kentang berada dalam kantong di rak dapur.
- fs: T_1 , kentang terkupas di panci dan siap dimasak, kantong dikembalikan ke rak di dapur sub-aksi:
- sub-aksi:
 1. ambil kantong kentang dari rak
 2. ambil panci dari lemari
 3. kupas kentang
 4. letakkan kentang di panci
 5. kembalikan kantong kentang ke rak dapur.
- harus dikerjakan dengan urutan yang diinginkan [aksi sekuensial]
- saat tertentu (jika baju yg dikenakan berwarna muda) diperlukan celemek [aksi kondisional/analisis kasus]
- mengupas kentang dilakukan hingga pada kriteria tertentu (misl: jumlah) [pengulangan]
- mengupas kentang merupakan bagian dari aksi menyiapkan makan siang [subprogram]

Notasi algoritmik

Berupa teks algoritma yang terdiri dari:

judul
kamus
algoritma } harus didefinisikan nama sbg identifikasi

nama digunakan sebagai identifikasi konstanta, variabel, tipe, prosedur, fungsi, modul program, algoritma harus unik.

Teks algoritma

Judul namaProgram

Spesifikasi: {spesifikasi teks algoritmik secara umum }

Kamus: { definisi konstanta, tipe, deklarasi variabel, spesifikasi prosedur, fungsi }

Algoritma:

teks algoritma yang berisi instruksi-instruksi dan bisa juga pemanggilan sub-aksi.

Instruksi dasar: i/o, assignment

Sequential Statement (aksi)

Kondisi/analisa kasus

Perulangan

Tipe data

- Data → diproses menggunakan operasi-operasi tertentu tergantung jenis dan tipe datanya.
- data bisa berupa angka bulat, pecahan, character/huruf, string.
- Tipe Data:
 - Tipe Primitive
 - Tipe Composite

Tipe primitive

- tipe yang disediakan oleh bahasa pemrograman
- misl: boolean, integer, real, char, string
bhs-C : int, float, double, char

C Data Types	
char	usually 8-bits (1 byte)
int	usually the natural word size for a machine or OS (e.g., 16, 32, 64 bits)
short int	at least 16-bits
long int	at least 32-bits
float	usually 32-bits
double	usually 64-bits
long double	usually at least 64-bits

Table 2.1: C data types and their usual sizes.

Variabel

- variabel: nama unik/identifier yang mewakili suatu tempat/address di dalam memory.
- variabel menyimpan suatu nilai yang memiliki tipe data sesuai dengan deklarasi variabel tsb.
- variabel dapat dikenakan operasi sesuai tipe datanya.
- variabel hrs dideklarasikan dan di inialisasi dengan suatu nilai.
- contoh:


```
angka_bulat : integer
angka_bulat=0
```

Konstanta

- Literal/Nilai yang ditulis langsung pada code program
misl: 100; 3.14; 'X'; "HelloWorld"
- Konstanta bisa memiliki identifier/nama dimana nilai yang ada didalamnya tdk boleh diubah.
misl: constant PI : real = 3.14159

Operator Dasar:

- Assignment
- Aritmatika
- Relasional/logical

Latihan soal:

buatlah notasi algoritmik dan translasikan ke program C untuk persoalan berikut:

1. menghitung volume gas ideal (V) dalam satuan liter, dengan memasukkan tekanan(P) dalam satuan kiloPascal, banyaknya mol gas (n) dalam satuan mol, dan temperatur (T) dalam derajat kelvin, serta diketahui konstanta gas ideal (R) yaitu : $8.314 \text{ JK}^{-1}\text{mol}^{-1}$ dengan rumus $PV=nRT$

Latihan

buatlah notasi algoritmik dan translasikan ke program C untuk persoalan berikut:

2. menghitung tegangan listrik (V) dalam volt dari masukan besarnya arus listrik (I) dalam ampere dan hambatan(R) dalam ohm, dengan rumus hukum Ohm: $V=IR$
3. menghitung luas sebuah trapesium (L) berdasarkan masukan a , dan b yang merupakan panjang dua sisi sejajar trapesium dan h yang merupakan tinggi trapesium dengan rumus $L=0.5h(a+b)$