

NAMA : Umar Afi Sauqi

NIM : 2100018137

KELAS : C

RANGKUMAN

PART 1

Algoritma, Flowchart, dan Pseucode

A. Algoritma

Algoritma adalah sekumpulan instruksi atau langkah-langkah yang dituliskan secara sistematis dan digunakan untuk menyelesaikan masalah /persoalan logika dan matematika dengan bantuan komputer.

Contoh dari sebuah algoritma :

Cara membuat kue

1. Mulai
2. Panas oven di suhu 180°C
3. Siapkan sebuah baking pan
4. Campurkan mentega dan gula
5. Campurkan dengan tepung, telur, dan vanilla essence
6. Tuangkan adonan ke baking pan
7. Masukkan baking pan kedalam oven
8. Selesai

Mengapa kita membutuhkan algoritma?, karena dengan algoritma kita dapat membuat sebuah fondasi awal dalam melakukan sesuatu dan dengan algoritma kita dapat melihat kasus kasus yang dapat terjadi Ketika sedang melakukan sebuah proses. Sedangkan dalam membuat program komputer dengan menggunakan algoritma kita dapat memahami masalah masalah yang dapat terjadi dan dapat menyelesaikan masalah tersebut dengan terstruktur, dan dapat membuat program computer tersebut dengan baik.





Dalam dunia teknologi informasi, Suatu algoritma harus cukup spesifik sehingga dapat dengan mudah diterjemahkan ke dalam program komputer (menggunakan C++, misalnya).

Sebuah algoritma dapat ditentukan:

- Secara tekstual, Misalnya, menggunakan kode semu (lihat nanti)
- Secara grafis, Misalnya, menggunakan diagram alur atau bagan aktivitas UML

B. Flowchart

Flowchart adalah representasi grafis dari urutan operasi dalam suatu program. Sebuah algoritma dapat direpresentasikan secara grafis menggunakan flowchart. Beberapa symbol dan fungsi dalam flowchart seperti berikut.

SIMBOL	FUNGSI
	Start/End
	Process
	Input/Output
	Test
	Connector
	Flow of Activities

C. Pseudocode

Garis besar program, ditulis dalam bentuk yang dapat dengan mudah diubah menjadi pernyataan pemrograman nyata. Ini menyerupai program sebenarnya yang akan dilaksanakan nanti. Namun, itu tidak dapat dikompilasi atau dieksekusi.

Pseudocode biasanya mengkodekan tindakan berikut:

- Inisialisasi variabel
- Penetapan nilai ke variabel
- Operasi aritmatika
- Operasi relasional

Contoh Pseudocode :

1. Start
2. Read *quantity*
3. Read *price_per_kg*
4. $price \leftarrow quantity * price_per_kg$
5. Print *price*
6. End

PART 2

Problem Solving

A. Problem Solving

Problem Solving adalah aktivitas mendefinisikan masalah, menentukan penyebab masalah, menentukan prioritas, menyeleksi berbagai pilihan solusi dan mengimplementasikan solusi tersebut. Pemrograman adalah proses pemecahan masalah

Teknik pemecahan masalah

- Analisis masalahnya
- Garis besar persyaratan masalah
- Langkah-langkah desain (algoritma) untuk memecahkan masalah

Algoritma:

- Proses pemecahan masalah langkah demi langkah
- Solusi dicapai dalam waktu terbatas

B. Problem Solving Process

Berikut Langkah Langkah dari problem solving :

Langkah 1 - Analisis masalahnya

- Garis besar masalah dan persyaratannya
- Langkah-langkah desain (algoritma) untuk memecahkan masalah

Langkah 2 - Terapkan algoritma

- Menerapkan algoritma dalam kode
- Verifikasi bahwa algoritme berfungsi

Langkah 3 - Pemeliharaan

- Gunakan dan modifikasi program jika domain masalah berubah

PART 3

Program Style and Form

A. Penggunaan Whitespace

Sisipkan karakter spasi putih (seperti kosong, tab, dan baris baru) jika perlu untuk meningkatkan keterbacaan sumber Anda kode. Contoh :

```
int matrix [ ][3] = {1, 0, 0, 0, 1, 0, 0, 0, 1};

int matrix [ ][3] = { 1, 0, 0,
                      0, 1, 0,
                      0, 0, 1
                    };
```

Karakter spasi putih diabaikan oleh kompiler selama kompilasi. Ingatlah untuk memisahkan kata-kata dan pengidentifikasi yang dicadangkan dari satu sama lain dan simbol lainnya.

B. Koma dan Semicolons

Penggunaan koma dan titik koma (semicolons) sangat berpengaruh dalam sebuah program. oleh karena itu, fungsi dari koma dan titik koma (semicolons) harus di ketahui.

- Koma memisahkan item dalam daftar.
Contoh: `int a, b, c;`
- Semua pernyataan C++ diakhiri dengan titik koma.
Contoh: `luas = panjang * lebar;`

Titik koma juga disebut terminator pernyataan.

C. Dokumentasi

Program lebih mudah dibaca dan dipelihara jika didokumentasikan dengan baik. Komentar dapat digunakan untuk mendokumentasikan kode

- Komentar satu baris dimulai dengan `//` di mana saja di baris
- Komentar beberapa baris diapit di antara `/*` dan `*/`
- Hindari memasukkan komentar yang tidak berguna seperti yang ditunjukkan di bawah:

```
int main() {
    min = elapsed_time / 60;           // assign elapsed_time / 60 to min
    sec = elapsed_time % 60;           // assign elapsed_time % 60 to sec
    hr = min / 60;                     // assign min / 60 to hr
    min = min % 60;                    // assign min % 60 to min
}
```

- Komentar program di bawah ini lebih berguna, contoh:

```
int main() {
    ...

    // Convert elapsed_time to min:sec

    min = elapsed_time / 60;
    sec = elapsed_time % 60;

    // Convert min:sec to hr:min:sec

    hr = min / 60;
    min = min % 60;
    ... }
```

- Pengidentifikasi nama dengan nama yang bermakna.

D. Form dan Style

Pertimbangkan dua cara untuk mendeklarasikan variabel. Berikut contoh dari penulisan dari sebuah variabel, kedua contoh benar namun metode ke 2 susah untuk dibaca.

- Method 1

```
int feet, inch;

double x, y;
```

- Method 2

```
int a,b;double x,y;
```

PART 4

Syntax dan Logical Error

A. Syntax Error

Kesalahan sintaks adalah kesalahan dalam kode sumber yang terkait dengan sintaks bahasa. Kesalahan sintaks terdeteksi oleh kompiler. File yang dapat dieksekusi akan dihasilkan oleh kompiler hanya jika kode sumber yang dikompilasi tidak memiliki kesalahan sintaksis. Kesalahan sintaks dilaporkan oleh kompiler dalam bentuk pesan kesalahan.

```
sufian@fujitsu3 /cygdrive/d/Pa1mOScygwin/sufian
$ g++ prog.cpp
prog.cpp:4:17: missing terminating " character
prog.cpp: In function `int main()':
prog.cpp:5: error: syntax error before `return'
prog.cpp:6:2: warning: no newline at end of file

sufian@fujitsu3 /cygdrive/d/Pa1mOScygwin/sufian
$
```

Contoh error yang
di tampilkan oleh
compiler

B. Logical Error

Kesalahan logika adalah kesalahan yang berhubungan dengan logika program. Biasanya, kesalahan logis tidak terdeteksi oleh kompiler. Kesalahan logis biasanya terdeteksi selama runtime program. Misalnya, sebuah program yang menghasilkan hasil yang tidak diharapkan merupakan indikasi bahwa ia memiliki kesalahan logis. Penting untuk diingat bahwa jika kompilator tidak menghasilkan pesan kesalahan, itu tidak berarti bahwa program Anda bebas dari kesalahan logis.

Kemungkinan untuk menghapus semua kesalahan sintaks dalam suatu program dan masih belum menjalankannya. Bahkan jika itu berjalan, itu mungkin masih tidak melakukan apa yang Anda maksudkan. Sebagai contoh,

$$2 + 3 * 5 \text{ dan } (2 + 3) * 5$$

keduanya adalah ekspresi yang benar secara sintaksis, tetapi memiliki arti yang berbeda.