PERTEMUAN 3

TAHAPAN PEMBANGUNAN PROGRAM

POKOK BAHASAN

1. Definisi Masalah
2. Outline Solusi
3. Pengembangan outline ke dalam algoritma
4. Melakukan test terhadap algoritma
5. Pemeriksaan Algoritma
6. Memindahkan Algoritma Ke Dalam Bahasa Pemrograman

DEFINISI MASALAH

- Pada tahap ini memerlukan pemahaman terhadap permasalah dengan membaca berulang kali sampai dengan mengerti apa yang dibutuhkan.

- Permasalahan dibagi kedalam tiga komponen:
  - Input / Masukan
  - Output / Keluaran
  - Proses

CONTOH KASUS

Sebuah toko peralatan mandi menjual bak mandi plastik. Banyak customer yang bertanya volume air yang dapat ditampung pada bak mandi tersebut. Oleh karena itu dibutuhkan program untuk menghitung volume air bak mandi sesuai dengan ukuran yang diinput.

Definisi Masalah :

Input : panjang, lebar dan tinggi

Output : volume bak mandi

Proses : volume bak mandi = panjang x lebar x tinggi

OUTLINE SOLUSI

- Setelah permasalahan didefinisikan, permasalahan dapat di bagi ke dalam tugas-tugas atau langkah langkah yang lebih kecil dan menghasilkan outline solusi

- Outline solusi awal dapat terdiri dari:
  - Proses utama
  - Subtask utama
  - Struktur Kontrol
  - Variabel dan struktur record
  - Logic utama (mainline)

OUTLINE SOLUSI

Input	Proses	Output
Baca p Baca l Baca t	volume_bak = p x l x t	Volume_bak

PENGEMBANGAN OUTLINE KE DALAM ALGORITMA

- Outline solusi pada langkah kedua dikembangkan menjadi algoritma yaitu sebuah set langkah yang menggambarkan tugas yang akan dikerjakan dan urutan pengerjaannya.

PENGEMBANGAN OUTLINE KE DALAM ALGORITMA (lanjutan)

Program Hitung_Volume_Bak
{menghitung volume bak mandi apabila input data tersebut diberikan}

Deklarasi
float p, l, t, volume_bak

Deskripsi
Baca p
Baca l
Baca t
Hitung volume_bak = p x l x t
Cetak volume_bak
End

MELAKUKAN TEST TERHADAP ALGORITMA

- Tujuan utama dari melakukan test terhadap algoritma adalah adalah untuk menemukan kesalahan utama logik sejak awal, sehingga akan lebih mudah diperbaiki.

- Data test diperlukan untuk melakukan test terhadap algoritma ini.

DESK CHECKS VS *TEST PLANS*

- *Test Plan* fokus pada nilai input dan output yang dibutuhkan untuk menguji program tanpa memperdulikan kinerja internal. Contoh : Apa output yang benar dari sebuah input ?

- *Desk Check* menekankan pada nilai variabel dan logika. Contoh : Berapakah nilai variabel x setelah pernyataan; Apa pernyataan berikutnya yang akan dieksekusi ?

TAHAPAN PENGECEKAN ALGORITMA

1. Pilih data sederhana yang valid. Dua atau tiga data biasanya sudah mencukupi.
2. Tentukan hasil output yang diharapkan untuk setiap set data.
3. Buatlah tabel yang nama variabel yang ada pada algoritma di sebuah kertas
4. Jalankan test satu persatu mengikuti algoritma yang ada, mulai dari perintah / statement pertama sampai dengan selesai
5. Ulangi langkah tersebut menggunakan set data yang lain.
6. Cek apakah hasil dari langkah 5, sesuai dengan hasil yang diharapkan di langkah kedua

PEMERIKSAAN ALGORITMA

Test Plan

- Input Data

	Data 1	Data 2
p	3	2
l	1	1.5
t	2	1

- Output

	Data 1	Data 2
Volume bak	6	3

TABEL DESK CHECK

	Data 1	Data 2
Baca p	3	2
Baca l	1	1.5
Baca t	2	1
Cetak volume_bak	6	3

MEMINDAHKAN ALGORITMA KE DALAM BAHASA PEMROGRAMAN

- Setelah ke-empat langkah sebelumnya dilakukan, maka pencodingan dapat dimulai dengan menggunakan bahasa pemrograman yang dipilih.

MEMINDAHKAN ALGORITMA KE DALAM BAHASA PEMROGRAMAN (lanjutan)

```
//Program Volume Bak
include<stdio.h>
include<conio.h>
include<iostream.h>

Main() {
float p, l, t, volume_bak;
printf(”panjang :”);scanf (”%f”,p);
printf(”lebar :”);scanf (”%f”,l);
printf(”tinggi :”);scanf (”%f”,t);
Volume_bak = p* l * t;
Printf(”volume bak : %5.2f”,volume_bak); }
```


MENJALANKAN PROGRAM PADA KOMPUTER

- Setelah pengcodingan, maka program dapat dijalankan pada komputer. Jika program sudah didesain dengan baik, maka akan mengurangi tingkat kesalahan dalam melakukan testing program.

- Langkah ini perlu dilakukan beberapa kali, sehingga program yang dijalankan dapat berfungsi dengan benar

DOKUMENTASI DAN PEMELIHARAAN PROGRAM

- Dokumentasi melibatkan eksternal dokumentasi (hierarchy chart, algoritma solusi, dan hasil data test) dan internal dokumentasi (coding program).

- Pemeliharaan program meliputi perubahan yang dialami oleh program (perbaikan ataupun penambahan modul, dll)

DISKUSI

1. Mengapa dibutuhkan tahapan dalam perancangan program ?

2. Apa manfaat Desk Checking Algoritma/pemeriksaan algoritma ?

LATIHAN 1

Sebuah sebidang tanah dengan ukuran 22 m x 15 m dibangun sebidang rumah dengan ukuran 8 m x 10 m. Sisa tanah yang tidak dibangun rumah ditumbuhi rumput. Pemilik rumah berencana memanggil tukang potong rumput untuk merapikan rumput di halaman rumahnya. Tarif per jam tukang rumput sebesar 100 ribu/jam. Berapa tarif yang harus dibayar memotong rumput halaman rumah, dengan rata-rata 2 m^2 / menit.

LATIHAN 2

Seorang kontraktor sedang membangun sebuah rumah dengan ukuran 8 m x 12 m. Rumah tersebut akan dipasangkan ubin dengan ukuran 30 x 30. Setiap kardus memiliki ukuran 1 m^2. Berapa kardus ubin yang harus dibeli ? Buatlah pseudocode & flowchart serta program hitung kebutuhan ubin.

TUGAS 2

Buatlah pseudocode, flowchart dan program untuk :

1. Menentukan apakah suatu bilangan merupakan bilangan prima atau bukan ?
2. Program untuk menentukan apakah tahun yang diinputkan termasuk tahun kabisat/bukan.

Catatan Tugas :

- Tugas dibuat pada kertas folio bergaris dengan menggunakan bolpoint.
- Tugas dikumpulkan pada saat pertemuan 4. Bagi mahasiswa yang tidak mengumpulkan tugas maka tidak mendapat nilai tugas 2 (tidak ada sistem susulan).

SOAL LATIHAN

1. Fokus pengecekan yang dilakukan tanpa memperdulikan kinerja internal dan hanya pada nilai input dan output adalah jenis pengujian .....
   a. Test logic
   b. Check Algoritma
   c. Test Plan
   d. Solution Test
   e. Desk Check

2. Proses pemeliharaan yang terdiri atas inspeksi periodik, pemeriksaan sistem untuk mengungkap dan mengantisipasi permasalahan:

a. Corrective
b. Adaptive
c. Preventive
d. Perfective
e. Kurative

SOAL LATIHAN (lanjutan)

3. Berfokus pada nilai input dan output yang dibutuhkan untuk menguji program tanpa memperdulikan kinerja internal, merupakan pengertian dari?
   a. Definisi Masalah
   b. Pseducode
   c. Desk Check
   d. Test Plan
   e. Maintenance
4. Permasalahan dibagi kedalam beberapa komponen, antara lain :
   a. Input dan output
   b. Masukan dan proses
   c. Proses, keluaran dan hasil
   d. Masukan, output dan hasil
   e. Proses, input dan output

SOAL LATIHAN (lanjutan)

5. Setelah permasalahan didefinisikan, permasalahan dapat dibagi kedalam tugas-tugas atau langkah-langkah yang lebih kecil dan menghasilkan?
    a. Permasalahan
    b. Flowchart
    c. Outline Solusi
    d. Pseducode
    e. Hasil