

VERSION 1.0  
FEBRUARI, 2021



# PEMROGRAMAN LANJUT

## MODUL 1 - PROGRAM CORRECTNESS I

TIM PENYUSUN :  
- HARDIANTO WIBOWO, S.KOM., MT.  
- DINDA ARINAWATI WIYONO  
- NUR SYAHFEI

PRESENTED BY : LAB. TEKNIK INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH MALANG

## PEMROGRAMAN LANJUT

---

### CAPAIAN PEMBELAJARAN MATA KULIAH

1. Mahasiswa mampu memahami tipe-tipe error dan mampu memperbaiki error pada program.
2. Mahasiswa mampu memahami concept of spesification.
3. Mahasiswa mampu memahami excepcetion handling dan mengimplementasikannya pada program.

---

### KEBUTUHAN HARDWARE & SOFTWARE

- Laptop/PC
- Falcon/Dev++
- Netbeans/IntelliJ/Eclipse

---

### MATERI POKOK

#### 1. Types Of Errors

##### a. Syntax Error

Error tata bahasa (sintaks) merupakan jenis error yang paling banyak terjadi dalam pembuatan program. Namun error ini paling mudah terdeteksi karena umumnya compiler atau interpreter dari masing-masing bahasa program akan melakukan pengecekan sebelum program dijalankan (saat dikompilasi). Lokasi baris yang menyebabkan error juga biasanya sudah ditunjukkan. Hanya perlu kejelian untuk memperbaikinya. Contoh dalam bahasa C dan Java :

```
#include <stdio.h>

int main(){
    int a = 2, b = 0;

    printf("%i", a)
    printf("%i", b);
}
```

```
public class Modul(){
    public static void main(String[] args){
        int a = 2, b = 0;

        System.out.println(a)
        System.out.println(b);
    }
}
```

### b. Logic Error

Jenis error yang satu ini merupakan jenis error yang paling susah dideteksi karena terjadinya bukan karena kesalahan penulisan (sintaks) atau kesalahan proses runtime, namun kesalahan dari sisi programmer, dalam hal ini algoritma yang digunakan. Karena logikanya salah, tentunya output yang dihasilkan juga akan salah. Untuk mendeteksi letak kesalahannya, bukanlah hal yang mudah. Terkadang kita harus merunut algoritma yang digunakan baris per baris (line byline). Contoh dalam bahasa C dan Java :

```
#include <stdio.h>

int main() {
    int jari;
    float luas;

    printf("Masukkan jari-jari : ");
    scanf("%i", &jari);

    luas = 2 * 3.14 * jari * jari;
    printf("%f", luas);
}
```

```
import java.util.Scanner;

class Modul(){
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        int jari;
        float luas;

        System.out.println("Masukkan jari-jari : ");
        jari = input.nextInt();

        luas = 2 * 3.14 * jari * jari;
        System.out.println(luas);
    }
}
```

### c. Runtime Error

Tingkatan error selanjutnya adalah runtime error. Dimana error ini akan terdeteksi saat program dijalankan (di-*running*). Penyebabnya beragam, pada umumnya karena terjadi kesalahan dalam proses input, perhitungan dan juga dalam proses output. Sebagai contoh yang banyak terjadi adalah error runtime karena pembagian suatu bilangan dengan nol. Lihat contoh program bahasa c berikut ini! Secara sintaks tentu tidak terdapat error, namun jika dijalankan, operasi

pembagian pada baris ke-5 akan menyebabkan error “division by zero”. Contoh dalam bahasa C dan Java :

```
#include <stdio.h>

int main() {
    int a=2, b=0;

    printf("%i", a);
    printf("%i", b);
    printf("%i", a/b);
}
```

```
public class Modul(){
    public static void main(String[] args){
        int a = 2, b = 0;

        System.out.println(a);
        System.out.println(b);
        System.out.println(a/b);
    }
}
```

## 2. The Concept Of a Specification

Pada sebuah program pasti terdapat spesifikasi-spesifikasi khusus yang harus ditaati untuk menggunakan sebuah program. Seperti misalnya pada sebuah program yang menggunakan fitur login. Disana user diminta untuk memasukkan password dengan kriteria tertentu yang telah ditentukan. Seperti password harus terdiri dari minimal 8 karakter, harus terdapat kombinasi angka dan huruf dan harus terdiri dari kombinasi huruf besar dan kecil. Spesifikasi password tersebut harus dimiliki sebuah program guna mencegah terjadi error saat menggunakan program.

## 3. Defensive Programming (Exception Handling)

Pada bahasa pemrograman C/C++, terdapat suatu metode yang dinamakan Exception Handling. Exception handling ini digunakan untuk menangani error atau keadaan tertentu yang terjadi ketika program sedang berjalan. Ketika terjadi error saat program berjalan (runtime error) program akan mengarahkan alur eksekusi program ke fungsi khusus yang dinamakan handler. Namun jika tidak terjadi error yang menyebabkan exception maka semua handler akan diabaikan dan program dijalankan secara normal. Contoh :

```
#include <stdio.h>

int main(){
    int a;
    try{
        printf("Masukan angka : ");
        scanf("%d", &a);
        if(a <= 0){
            throw a;
        }
        else {
            printf("Nilai yang anda masukan adalah %d", a);
        }
    }catch(int x){
        printf("program menangkap eksepsi yang bernilai %d", x);
    }
}
```

---

## REFERENSI

<http://achmatim.net/2008/11/17/mengenal-error-dalam-program/>

<http://www.nblognlife.com/2017/04/mengenal-exception-handling-pada-c.html/>

---

## LEMBAR KERJA

### TUGAS 1

Perbaiki salah satu program berikut sehingga dapat berfungsi dengan baik.

```
int main(){
    int arr[] = {1 2 3 4 5 6 7 8 9};
    int i=0, ganjil, genap, total;

    while(i < 10){
        if(arr[i]%2 == 0) ganjil = arr[i];
        if(arr[i]/2 == 0) genap += arr[i];
        total += arr[i]
    }

    printf("Jumlah Bilangan Ganjil = %d\n", ganjil);
    printf("Jumlah Bilangan Genap = %d\n", genap);
    printf("Total = %d\n", &total);
}
```

(A)

```
int find(int x int arr){
    int i;

    while( i < 100){
        if (arr[i] == x)
        }
    return -1;
}

int main(){
    int arr[] = {1 2 3 4 5 6 7 8 9};
    if(find(9 arr) == -1) puts("Angka ada di array!");
    puts("Angka tidak ada di array!");
    return 0;
}
```

(B)

### TUGAS 2

Buatlah program bahasa c inputan nim dengan specification sebagai berikut :

1. Hanya nim mahasiswa informatika dengan kode "10370311".
2. Tidak boleh kurang atau lebih dari 15 digit.
3. Jika nim salah tampilkan pesan error, kemudian meminta inputan kembali.
4. (Opsional) Buatlah exception yang berbeda untuk setiap error. Contoh jika nim-nya kurang dari 15 digit dan nim lebih dari 15 digit.

5. (Opsional) Ubahlah program yang sudah dibuat ke dalam bahasa pemrograman java.

**Kerjakan dengan rapi ya, agar minim terjadi kesalahan syntax dan mudah dibaca.**

---

#### **RUBRIK PENILAIAN**

<b>Kriteria</b>	<b>Nilai</b>
Tugas 1 (A)    Tugas 1 (B)	15    20
Tugas 2	30
Tugas 2 (Soal Optional)	10
Pemahaman	40
<b>Total</b>	<b>100</b>

**Silahkan dikerjakan tanpa copas dan jangan lupa berdoa sebelum praktikum! Good Luck!!**