

VERSION 1.0

JULY, 2022



PEMROGRAMAN LANJUT

MODUL 6 – DOCUMENTATION AND PROGRAMMING STYLE

TIM PENYUSUN :

- WILDAN SUHARSO, S.KOM., M.KOM
- HANIA PRATIWI NINGRUM
- FILDZAH LATHIFAH

PRESENTED BY : LAB. TEKNIK INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

PEMROGRAMAN LANJUT

CAPAIAN PEMBELAJARAN MATA KULIAH

1. Mahasiswa mampu memahami tata cara mendokumentasikan sebuah program.
2. Mahasiswa mampu memahami style program dalam bahasa pemrograman Java.

KEBUTUHAN HARDWARE & SOFTWARE

- Laptop/PC
- Netbeans/IntelliJ/Eclipse

MATERI PRAKTIKUM

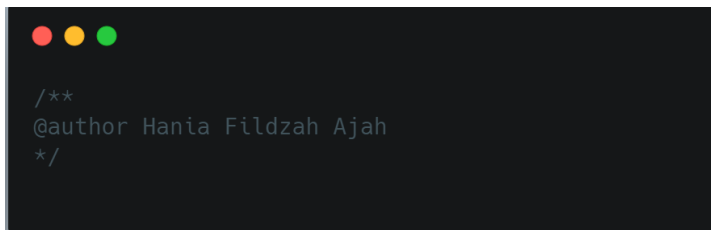
1. Documentation and Programming Style

a. Documentation

Dokumentasi program adalah catatan mengenai program yang telah dirancang, dapat meliputi catatan tentang tujuan program, data yang digunakan dalam program, ataupun logika yang digunakan pada class, method, constructor, dsb.

Documentation comment dikenal secara informal dengan sebutan “JavaDoc”. Struktur Doc comment terdiri dari karakter `/**` untuk memulai komentar dan `*/` untuk penutup. Teks yang ada di dalam komentar ini bisa berlanjut ke beberapa baris (tidak hanya sebaris saja). Berikut uraiannya adalah sejumlah tag JavaDoc :

- Author/pembuat program (`@author namaAuthor`)



```
/**
 * @author Hania Fildzah Ajah
 */
```

- Menambah informasi versi dalam class (`@param namaParameter deskripsi`)



```
/**
 * Mengembalikan hasil dari 2 int
 * @param a angka yang akan dikalikan
 * @param b angka mengalikan
 * @return an int
 */
public int multiple(int a, int b) {
    return a*b;
}
```

- Menjelaskan nilai return dari method (@return deskripsi-isi-return)

```
/**
 * Mengembalikan hasil dari 2 int
 * @return an int
 */
```

- Menjelaskan exception yang dilempar oleh sebuah class, method, constructor, atau interface (@exception namaClass deskripsi)

```
/**
 * Replace multiple().
 * @throws IniException kecuali kalau misal .....
 * @exception IyaIniException
 */
public multiple(int a, int b){
    // isinya
}
```

***Note :** Tag javaDoc di atas adalah yang paling dasar. Jika ingin mempelajari selengkapnya, kamu bisa membacanya pada link yang tertera pada referensi. (Nilai pada lembar kerja praktikum modul ini juga didasarkan pada kelengkapan tag JavaDoc yang sesuai dengan program masing-masing)

b. Programming Style

Programming style merupakan sekumpulan aturan yang digunakan dalam menulis source code/kode program. Aturan tersebut tidak berhubungan dengan logika program, tetapi lebih ke cara penulisan kode. Jadi, secara tidak langsung programming style memberikan manfaat kepada programmer untuk membaca dan memahami sebuah source code/kode program. Beberapa contoh hal yang diatur dalam programming style, antara lain:

- Komentar

Komentar bisa kamu lakukan dengan beberapa cara. Misalnya :

```
//saya suka ngoding
```

Atau bisa juga menggunakannya seperti berikut.

```
/**
 *
 *Nama          : Budi
 *Cita-cita      : Pro player
 */
```

- Aturan penamaan

Buat nama yang sesuai dengan tujuan dibuatnya. Hal ini akan membantu kamu agar lebih mudah memahami masalah seperti error pada class/method/variable/lainnya pada program yang kamu buat. Contoh :

- Nama class


```
class Calculator{
}
```
- Nama method


```
public int calculate(){
}
public int multipleResult(){
}
```
- Nama variable


```
int input1, input2, myVar;
```
- Kurung pembuka dan penutup program
 - Tempatkan posisi pembuka program di awal baris yang sama dengan kata kunci.

Contoh :

```
if(aku == kamu){
    System.out.print("Alhamdulillah");
}
```
 - Tempatkan posisi pembuka dibawah dan sejajar dengan kata kunci.


```
if(kamu == dia)
{
    System.out.print("Aku Gapapa");
}
```
- Whitespace

Spasi memang terlihat hal yang sepele, akan tetapi juga digunakan sebagai bagian dari program. Misalnya setelah atau sebelum tanda operator harus menggunakan spasi.

Masih banyak hal lain yang bisa dipelajari dari programming style terutama dalam bahasa pemrograman java ini. Biasanya programming style hanya berlaku dalam lingkup bahasa pemrograman tertentu (contohnya java yang sudah dijelaskan di atas) karena setiap bahasa pemrograman memiliki style yang berbeda pula. Jadi, sesuaikan dengan bahasa pemrograman yang kamu pakai.

2. Exception Handling

Exception handler menggunakan kombinasi try dan catch keywords. Dimana source code/kode program yang sekiranya dapat menimbulkan exception dimasukkan ke dalam try.

Sedangkan statement catch disertai dengan tipe exception yang ingin ditangani (misalnya `ArrayIndexOutOfBoundsException`). Jika exception terjadi di bagian protected code, maka blok catch akan melakukan pemeriksaan.

Berikut adalah syntax dari penggunaan try-catch :

```
try{
//Protected code
}catch(ExceptionName e){
//blok catch
}
```

REFERENSI

<https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html>

<https://docs.oracle.com/javase/7/docs/technotes/tools/solaris/javadoc.html#sourcefiles>

<https://ichi.pro/id/penjelasan-singkat-tentang-javadoc-229034430139462>

<https://husnilk.wordpress.com/2011/11/08/programming-style-what-is-your-style/>

<https://www.slideshare.net/RasyedaAufa/pengenalan-javadoc>

<https://javanewbie.wordpress.com/2013/07/26/dokumentasi-aplikasi-merupakan-hal-yangpenting/#:~:text=Dokumentasi%20Program%20adalah%20catatan%20mengenai,dalam%20program%2C%20logika%20yang%20dipergunakan.>

LEMBAR KERJA

TUGAS 1

Kamu memiliki dua array ***inNumber*** dan ***inString***, gunakan kedua array tersebut untuk menampung inputan dari user secara acak (index-nya). Ketentuannya :

- Jika inputan user adalah String maka value-nya akan masuk ke ***inString***
- Jika inputan adalah angka maka value-nya masuk ke ***inNumber***.
- Buat stopper untuk menghentikan inputan dan kemudian tampilkan semua data yg sudah diinput.
- Antisipasi setiap kemungkinan error yang terjadi pada program yang sudah kamu buat dengan ***exception*** dan menjaga agar program mu tetap berjalan sampai user ingin menghentikan program.
- Simpan hasil inputan kedalam database sehingga jika program dihentikan file tidak hilang

****Note*** : program yang dibuat bebas, asalkan memenuhi ketentuan yang sudah disebutkan di atas.

TUGAS 2

Dari program yang sudah kamu buat. Buatlah dokumentasi pada program tersebut dengan mengikuti aturan Javadoc dengan selengkap mungkin. Jangan lupa juga mencantumkan data diri sebagai author pada dokumentasi.

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

RUBRIK PENILAIAN

Kriteria	Nilai
Tugas 1	40
Tugas 2	20
Pemahaman	40
Total	100

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