



### UJIAN AKHIR SEMESTER GASAL 2021-2022

Mata Kuliah	: Dasar Pemrograman (TF4536)
Hari / Tanggal	: Senin/ 13 Desember 2021
Waktu	: 180 menit (07.30 – 10.30 WIB)
Sifat Ujian	: Terbuka

**Jangan lupa menggunakan NAMA VARIABEL YANG BAIK, DAN KOMENTAR untuk proses penting/ perhitungan panjang. Kumpulkan file JAVAnyanya saja untuk setiap nomor.**

#### 1. Class & Object - 35 poin

Buatlah sebuah program permainan antara *player* dan komputer. Karena permainan ini mungkin akan digunakan lagi kedepannya, maka untuk membuat program menjadi *reusable* buatlah dalam bentuk *class*.

Berikut merupakan hal-hal yang dapat dilakukan oleh *class*.

- Menyimpan *score* permainan, nama *player*, berapa ronde permainan dilakukan.
- Mendapatkan nilai yang dipilih oleh komputer (menggunakan random class).
- Mensimulasikan permainan (menerima dua nilai dari *player* dan komputer).
- Mencetak hasil permainan yang telah dilakukan.

Beberapa properti yang harus dimiliki oleh class:

- Segala class member merupakan private member (Mahasiswa boleh menambahkan data member untuk membantu pengerjaan soal).
  - private int ronde
    - untuk menentukan berapa banyak ronde yang akan dimainkan
  - private int nilaiPemain
    - Digunakan untuk menyimpan nilai pemain, dari ronde ke ronde
  - private String namaPemain
    - Digunakan untuk menyimpan nama pemain, diisi ketika constructor dipanggil
  - private int nilaiKomputer
    - Digunakan untuk menyimpan nilai komputer, dari ronde ke ronde
- Method yang harus ada dapat dilihat dari contoh yang diberikan di bawah.
  - Constructor(String namaPemain, int ronde)
  - public void game(int pemain, int komputer)
    - penjelasan permainan di paragraf berikut (Penjelasan 1\*)



- Jika *player* menang maka cetak “{*namaPemain*} menang”, dan sebaliknya jika komputer menang maka cetak “komputer menang”, jika ronde seri maka cetak “Seri”
- public int getKomputerHand()
  - mengembalikan sebuah nilai int yang **random**. (Gunakan random class yang ada pada java.util.Random)
- public boolean isGameStillOn()
  - mengembalikan boolean yang menentukan apakah permainan masih berjalan (jumlah permainan ditentukan saat object permainan dibuat)
- public void printGameResult()
  - mencetak hasil permainan “{*nama pemenang*} menang dengan nilai {*nilai pemenang*}” atau jika terjadi seri “{*nama pemain*} seri dengan komputer dengan nilai {*nilai seri*}”
- public int numberConverter(int n)
  - fungsi yang digunakan untuk membuat mengkonvert nilai yang dipilih oleh *player* dan komputer.
  - Rumus:  $2 * (n \bmod 2) + ((n+1) \bmod 2)$

#### \*Penjelasan 1

Permainan dilakukan dengan **setiap *player* dan komputer memberikan sebuah bilangan bulat positif (int) ke fungsi *game(int pemain, int komputer)***. Lalu kedua angka tersebut masing-masing akan dimasukkan ke dalam rumus (ada pada paragraf di bawah) atau fungsi **numberConverter** dan hasilnya akan dibandingkan. Hasil *numberConverter(player)* dibandingkan dengan hasil *numberConverter(komputer)*, jika hasil dari rumus *player* lebih besar maka *player* mendapatkan poin, sebaliknya poin untuk komputer jika komputer memiliki hasil dari rumus yang lebih besar. Jika hasil seri maka tidak perlu menambahkan poin kepada masing-masing *player* dan komputer.

Rumus (dalam notasi matematika) yang digunakan untuk mengkonversi nilai(*n*) (fungsi *numberConverter*) yang dipilih dari *player* dan komputer sebagai berikut:

$$2 * (n \bmod 2) + ((n+1) \bmod 2)$$



**\*\* Gunakan main class berikut untuk memastikan bahwa class sudah sesuai dengan permintaan.**

```
public class Main {  
  
    static Scanner scanner = new Scanner(System.in);  
  
    public static void main(String[] args) {  
  
        System.out.println("Selamat datang di Permainan XYZ");  
        System.out.print("Masukkan nama player: "); String nama = scanner.nextLine();  
        System.out.print("Masukkan jumlah ronde: "); int ronde = scanner.nextInt();  
  
        Permainan permainan = new Permainan(nama, ronde);  
  
        while(permainan1.isGameStillOn()) {  
  
            System.out.print("Masukkan angka yang anda pilih: ");  
            int player = scanner.nextInt();  
  
            int komputer = permainan.getKomputerHand();  
            System.out.println("Komputer memilih: " + komputer);  
  
            permainan.game(player, komputer);  
        }  
  
        permainan.printGameResult();  
    }  
}
```

### Contoh Interaksi Program

Selamat datang di Permainan XYZ

Masukkan nama player: john

Masukkan jumlah ronde: 5

Masukkan angka yang anda pilih: 1

Komputer memilih: 1681737993

john menang

Masukkan angka yang anda pilih: 2

Komputer memilih: 1776019383



komputer menang

Masukkan angka yang anda pilih: 2

Komputer memilih: 1841709471

komputer menang

Masukkan angka yang anda pilih: 2

Komputer memilih: 809855043

komputer menang

Masukkan angka yang anda pilih: 2

Komputer memilih: 82038507

komputer menang

Komputer menang dengan nilai 4

Penilaian:

1. (CPMK 2.6) Mampu membuat sebuah class dan object- (25 point)

Mampu membuat Class sesuai dengan permintaan	- 10 Point
Mampu membuat Objek dari Class yang telah dibuat	- 5 Point
Mampu membuat Class dengan data member yang sesuai	- 5 Point
Mampu membuat Class dengan method yang sesuai	- 5 Point
  
2. (CPMK 2.5) Mampu membuat function - (10 poin)

Mampu membuat function dengan parameter yang diminta	- 4 Point
Mampu membuat function dengan return type yang diminta	- 4 Point
Mampu membuat function sesuai dengan permintaan	- 2 Point



## 2. Array 1D + loop - 30 poin

Multiplikator sebuah bilangan adalah nilai perkalian dari semua faktor positif dari sebuah bilangan. Contoh:

Multiplikator dari 8 adalah  $1 * 2 * 4 * 8 = 64$

Buatlah sebuah program yang menerima input sekumpulan bilangan bulat (integer) yang lebih besar dari 0 dari user (asumsikan input selalu sesuai ketentuan, tidak perlu dicek) dan menyimpan bilangan-bilangan tersebut dalam sebuah array, kemudian mengurutkan bilangan-bilangan yang ada di array tersebut sesuai dengan nilai multiplikatornya (dari nilai multiplikator terkecil sampai terbesar). Jika nilai multiplikatornya sama, maka bilangan yang lebih kecil didahulukan

Contoh:

Banyak Bilangan: 9

Bilangan-1: 2

Bilangan-2: 8

Bilangan-3: 3

Bilangan-4: 6

Bilangan-5: 7

Bilangan-6: 5

Bilangan-7: 13

Bilangan-8: 15

Bilangan-9: 11

Array : [2, 8, 3, 6, 7, 5, 13, 15, 11]

Hasil : [2, 3, 5, 7, 11, 13, 6, 8, 15]

Penjelasan:

- Nilai Multiplikator:

2 ->  $1 * 2 = 2$

3 ->  $1 * 3 = 3$

5 ->  $1 * 5 = 5$

7 ->  $1 * 7 = 7$

11 ->  $1 * 11 = 11$

13 ->  $1 * 13 = 13$

6 ->  $1 * 2 * 3 * 6 = 36$



$$8 \rightarrow 1 * 2 * 4 * 8 = 64$$

$$15 \rightarrow 1 * 3 * 5 * 15 = 225$$

Keterangan:

- Boleh membuat fungsi-fungsi yang dapat memudahkan untuk mengerjakan soal, misal fungsi untuk menghitung nilai multiplikator sebuah bilangan (penggunaan fungsi tidak dinilai)
- Boleh menggunakan lebih dari satu array
- Hasil boleh disimpan di array yang sama dengan array yang digunakan untuk menyimpan input maupun disimpan di array yang berbeda
- Tidak boleh menggunakan fungsi sorting bawaan dari Java

Penilaian:

1. (CPMK 2.4) Mampu menggunakan Array 1D - (20 point)
  - Mampu membuat array sesuai input user - 4 Point
  - Mampu menampilkan isi array - 4 Point
  - Mampu memproses array sesuai permintaan soal - 12 Point
2. (CPMK 2.3) Mampu menggunakan perintah loop - (10 poin)
  - Mampu menghitung multiplikator sebuah bilangan - 3 point
  - Mampu memproses array sesuai permintaan soal - 7 Point

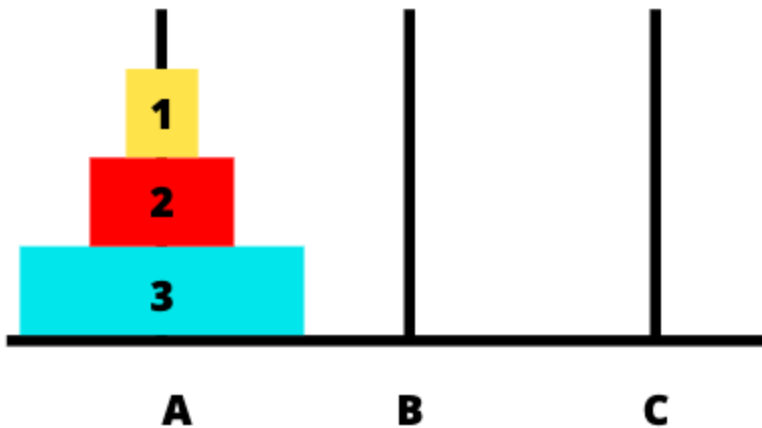


### 3. ArrayList + Exception Handling - 35 poin

Buatlah permainan sederhana bernama "Hanoi Tower" menggunakan ArrayList.

Sediakan 3 Variable ArrayList sebagai Tower dari Hanoi Tower.

Saat program dijalankan, Variable ArrayList pertama ( Tower pertama ) berisi 3 angka yang digunakan sebagai item pada Tower tersebut.



Seperti pada gambar di atas, **Sediakan 3 Angka pada program Anda.**

Variable ArrayList pertama (A) berisi angka 1, 2, dan 3 ( **urut dari kecil ke besar : 1 di index 0, 2 di index 1, 3 di index 2** )

Hasil akhir yang diharapkan dari permainan ini: variable ArrayList ketiga (C) berisi angka 3, 2, dan 1 ( **urut dari besar ke kecil : 3 di index 0, 2 di index 1, 1 di index 2** )

Tujuan dari game ini adalah memindahkan seluruh isi dari Tower pertama ke tower ketiga, dengan bantuan tower kedua dan memiliki aturan sebagai berikut:

- Program hanya dapat menerima input user dengan format :

**Input asal dan input Tujuan**

**Input Asal : 1**

**Input Tujuan : 2**

→ artinya memindahkan balok dari Tower pertama ke Tower kedua (memindahkan balok angka pada **index pertama** dari ArrayList pertama ke **index terakhir** ArrayList kedua)

atau

**Input asal dan input Tujuan**

**Input Asal : 3**

**Input Tujuan : 1**



→ artinya memindahkan balok dari Tower ketiga ke Tower pertama (memindahkan balok angka pada **index pertama** dari Arraylist ketiga ke **index terakhir** Arraylist pertama)

- Sediakan **Exception Handling** agar setiap input dari User tidak membuat program Crash. **Exception handling digunakan untuk input yang bukan angka.** Gunakan `Integer.parseInt(scan.nextLine());` Untuk menerima Input awal dan tujuan.
- **Jika input tidak sesuai format**, maka perintah tersebut tidak dilakukan, tampilkan kesalahannya kenapa dan meminta inputan ulang.
- (Pakai IF untuk aturan A,B,D dan E) hanya aturan C yang menggunakan Exception handling.  
List aturan yang ada:
  - a. Perintah Input bisa dilakukan jika balok yang hendak dipindahkan bernilai lebih kecil dari balok **index terakhir** yang berada pada Tower tujuan atau pada Tower Tujuan tidak memiliki balok.
  - b. Tidak bisa menerima inputan Awal dan Tujuan yang sama.
  - c. Input tidak bisa menerima Huruf.
  - d. Asal Tower tidak boleh Tower yang kosong.
  - e. Input Awal dan Tujuan hanya boleh antara 1, 2 dan 3
- **Jika input sesuai format**, maka perintah akan diproses :
  - Jika balok bisa dipindahkan, maka dipindahkan
  - Jika balok tidak bisa dipindahkan, maka perintah tidak dilakukan, tampilkan info kalau input tersebut melanggar aturan dan meminta input ulang.
- **Bantuan aturan A : untuk pengecekan nilai tower asal dan tujuan, bisa diambil dulu nilai dari masing-masing tower, di cek, kalau memenuhi aturan, baru di proses pindahkan.**

---

Tower:

Tower 1: 1 2 3

Tower 2: -

Tower 3: -

*Contoh yang berhasil :*

Input Asal: 1





Input Tujuan: **3**

Output: **berhasil memindahkan Balok 1 dari Tower 1 ke Tower 3.**

---

Tower:

Tower 1: 2 3

Tower 2: -

Tower 3: 1

*Contoh Aturan (A) tidak dipenuhi :*

Input Asal: **1**

Input Tujuan: **3**

Output: **Input tidak dapat diproses.**

**Balok pada Tower 1 bernilai 2 yang lebih besar dari Tower 3 yang bernilai 1.**

**Silahkan Input lagi.**

---

Tower:

Tower 1: 2 3

Tower 2: -

Tower 3: 1

*Contoh Aturan (B) tidak dipenuhi:*

Input asal: **1**

Input Tujuan: **1**

Output: **Input tidak dapat diproses.**

**Tower Asal dan Tower tujuan tidak boleh sama.**

**Silahkan input lagi.**

---

Tower:

Tower 1: 2 3

Tower 2: -

Tower 3: 1



*Contoh Aturan (C) tidak dipenuhi:*

Input asal: **a**

Output: **java.lang.NumberFormatException: For input string: "a"**

**Silahkan Input lagi.**

---

Tower:

Tower 1: 2 3

Tower 2: -

Tower 3: 1

*Contoh Aturan (D) tidak dipenuhi:*

Input asal: **2**

Input Tujuan: **3**

Output: **Input tidak dapat diproses.**

**Tidak ada balok pada Tower 2.**

**Silahkan input lagi.**

---

Tower:

Tower 1: 2 3

Tower 2: -

Tower 3: 1

*Contoh Aturan (E) tidak dipenuhi:*

Input asal: **0**

Input Tujuan: **5**

Output: **Format Input tidak Valid.**

**Asal tower hanya boleh antara 1, 2 dan 3.**

**Tujuan tower hanya boleh antara 1, 2 dan 3.**

**Silahkan Input lagi.**

---

- NOTE: Program akan **selalu meminta Input** sampai seluruh Balok sudah pindah dari Tower pertama ke Tower ketiga (Seluruh balok berada pada Tower ketiga dalam posisi urut dari 3, 2 dan diakhiri 1, **3 di index 0, 2 di index 1, 1 di index 2**)

Tower:



Tower 1: -

Tower 2: 1

Tower 3: 3 2

Input asal: 2

Input Tujuan: 3

Output: **berhasil memindahkan Balok 1 dari Tower 2 ke Tower 3.**

**Seluruh balok sudah berada pada posisi yang benar.**

**Anda menang!!**

---

Penilaian:

1. (CPMK 2.4) Mampu menggunakan arraylist - (25 point)

- |  |            |
|--|------------|
| Mampu menerapkan arraylist                               | - 5 Point  |
| Mampu menambahkan, memindahkan, mengurangi isi arraylist | - 15 Point |
| Mampu mengecek arraylist untuk kondisi menang            | - 5 Point  |

2. (CPMK 1.5) Mampu menggunakan Exception Handling - (10 poin)

- |  |           |
|--|-----------|
| Mampu mengecek aturan a dan menampilkan kesalahannya | - 2 Point |
| Mampu mengecek aturan b dan menampilkan kesalahannya | - 2 Point |
| Mampu mengecek aturan c dan menampilkan kesalahannya | - 2 Point |
| Mampu mengecek aturan d dan menampilkan kesalahannya | - 2 Point |
| Mampu mengecek aturan e dan menampilkan kesalahannya | - 2 Point |