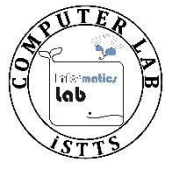




# Institut Sains dan Teknologi Terpadu Surabaya

Jl. Ngagel Jaya Tengah 73 - 77, Surabaya 60284

Telp. (031) 5027920 Fax. (031) 5041509



Laboratorium : L-304

Waktu : Rabu / 15.45-17.45

Minggu Ke : 1

Materi : Kotlin Basic

Praktikum : Mobile Device Programming

Jurusan : S1 - Informatika

Tanggal : 21 Februari 2024

Jenis Soal : Materi dan Tugas

## MATERI (TOTAL: 40)

Buatlah sebuah simulasi kucing peliharaan. Objektif dari simulasi ini adalah menjalankan program terus **hingga salah satu status dari kucing adalah 0**. Kucing memiliki 3 jenis status, yaitu **Happiness, Energy, dan Health**. Setiap status kucing di awal adalah 100, dan jika salah satu status adalah 0 maka program akan selesai.

### Main Menu

Di awal ketika program di run, program akan meminta input nama kucing. Program akan meminta input terus apabila user mengisi dengan string kosong.

```
-- My Cat --
Input nama kucing :
Mohon inputkan nama kucing
-- My Cat --
Input nama kucing : xandra
```

Sesudah menginputkan nama kucing, program akan masuk ke simulasi. Terdapat lima menu yang dapat dipilih, yaitu eat, play, sleep, status, dan next day.

```
-- My Cat --
1. Eat
2. Play
3. Sleep
4. Status
5. Next Day
Input:
```

Berikan pengecekan apabila input user di luar menu yang ditetapkan.

### Eat, Play

Apabila user memilih salah satu dari menu eat atau play, maka akan terjadi perubahan status kucing. Terdapat juga menu untuk sleep (akan dijelaskan di segmen berikutnya). Berikut adalah keterangan perubahan status kucing.

Eat -> Energy+20

Play -> Happiness+20 Energy-20 Health-20

Sleep -> Health+20 Happiness-20

You choose play! xandra's happiness : 80, energy: 40, health: 40

Input: 4

[illegible]

Health : 40

Apabila user memilih menu sleep, maka kucing akan berada dalam kondisi tidur. Saat kucing tertidur, menu “Sleep” akan berganti menjadi “Wake”, dan menu eat dan play akan ter-disable.

Input :

Status kucing belum berubah ketika tidur. Apabila user memilih input 4 (status), maka tampilkan status kucing serta ascii art kucing tertidur

```
-- My Cat --
1. Eat (x)
2. Play (x)
3. Wake
4. Status
5. Next Day
Input: 4
```

```
      | \
ZZZZz /,`.-'-'_''--''_
      |,4- ) )-,-. ,\ ( '-'-
      '----' (/_--' -'\_)
```

```
xandra's data :
Happiness : 80
Energy : 40
Health : 40
```

Apabila user memilih menu yang terdisable, maka program akan memberi pesan error "<nama kucing> is sleeping". User bisa membangunkan kucing dengan memilih input 3 (wake). Setelah bangun, berikan pesan sesuai status yang berubah, lalu kembalikan jenis menu seperti semula.

```
-- My Cat --
1. Eat (x)
2. Play (x)
3. Wake
4. Status
5. Next Day
Input: 3
You choose wake! xandra's happiness : 60, health: 60
-- My Cat --
1. Eat
2. Play
3. Sleep
4. Status
5. Next Day
Input:
```

### Next Day

Apabila user memilih menu next day, maka kurangi semua status kucing sebesar 10 poin.

```
-- My Cat --
1. Eat
2. Play
3. Sleep
4. Status
5. Next Day
Input:5
You choose next day! xandra's happiness : 50, energy: 30, health: 50
```

### Indikator

Apabila status kucing berada di antara range berikut, berikan indikator bahaya di bawah menu

1- 20 : Danger

11 – 40 : Warning

```
-- My Cat --
1. Eat
```

```

2. Play
3. Sleep
4. Status
5. Next Day
[WARNING] xandra's energy is 30
Input: 2
You choose play! xandra's happiness : 70, energy: 10, health: 30
-- My Cat --
1. Eat
2. Play
3. Sleep
4. Status
5. Next Day
[DANGER] xandra's energy is 10
[WARNING] xandra's health is 10
Input:

```

### Game Over

Apabila salah satu status kucing menyentuh 0, maka simulasi akan berakhir.

Perlu diperhatikan bahwa masing-masing status kucing tidak bisa menembus lebih dari 100.

**DILARANG MENGGUNAKAN MATERI YANG BELUM DIAJARKAN PADA MINGGU INI  
JIKA MELANGGAR MAKA NILAI MATERI : 0**

### **PERHATIKAN KETENTUAN DIBAWAH :**

- Highlight kriteria yang dikerjakan dengan warna kuning dan kumpulkan word beserta dengan file materi, apabila tidak dikumpulkan maka materi tidak akan diperiksa.
- Akan ada pengurangan nilai sebesar -5 untuk setiap kriteria yang dihighlight namun tidak dikerjakan.
- **MENCONTEK = Nilai MOD 2**

### **MATERI : 40**

SCORE	KRITERIA
0/2/4	Program dapat meminta nama kucing dengan benar (2 jika tidak ada pengecekan input kosong)
0/2	Program dapat menerima semua input menu dengan benar dan memberikan respon yang sesuai
0/3/6	Dapat memilih menu eat dan play dengan benar (dinilai jika dapat menampilkan perubahan pada status) (3 jika tidak bisa memberikan pesan sesuai perubahan status)
0/3/6	Dapat memilih menu sleep dengan benar (dinilai jika status kucing berubah, 3 jika menu eat dan play tidak terdisable atau tidak muncul menu "Wake")
0/3/5	Dapat memilih menu wake dengan benar (dinilai jika dapat menampilkan perubahan pada status) (3 jika sesudah itu menu tidak kembali seperti semula)
0/3/6	Dapat mencetak status dengan benar (3 jika ascii art tidak bisa menyesuaikan kondisi tidur kucing)
0/3/5	Dapat melakukan next day dengan benar (3 dinilai jika dapat menampilkan perubahan pada status)
0/4	Dapat memunculkan indikator dengan benar
0/2	Simulasi dapat berakhir apabila salah satu status menyentuh 0
Total : 40	

## TUGAS (TOTAL = 30)

Buatlah game Othello sederhana dengan menu awal sebagai berikut:

```
=== OTHELLO ===  
1. New Game  
2. High Score  
0. Exit  
>>
```

### Menu 1: Game

Ketika menu 1 dipilih, mintalah input nama player pertama dan player kedua.

```
>> 1  
Player 1: Cherilyn  
Player 2: Mikhael
```

Setelah itu, tampilkan board Othello berukuran 6x6 dengan tampilan sebagai berikut:

```
Cherilyn's Turn  
  1  2  3  4  5  6  
1 [ ][ ][ ][ ][ ][ ]  
2 [ ][ ][ ][ ][ ][ ]  
3 [ ][ ][X][O][ ][ ]  
4 [ ][ ][O][X][ ][ ]  
5 [ ][ ][ ][ ][ ][ ]  
6 [ ][ ][ ][ ][ ][ ]
```

Pada game ini, player 1 dan player 2 akan bermain secara bergantian. Tampilkan nama player yang sedang bermain dengan format <Nama Player>'s Turn.

Pada awal permainan, terdapat 2 piece O (player 1) dan 2 piece X (player 2) yang terletak di tengah papan. Untuk meletakkan sebuah piece, player dapat menginputkan koordinat X dan Y. Berikan pengecekan piece hanya bisa diletakkan pada koordinat yang masih kosong dan tidak bisa diletakkan pada koordinat yang sekitarnya masih kosong (atas, bawah, kanan, dan kirinya kosong). Piece juga tidak bisa diletakkan di sebelah piece dengan simbol yang sama. Berikan pesan error jika player meletakkan piece di tempat yang tidak valid.

Setelah meletakkan piece, cetak kembali board serta ubahlah semua symbol piece yang berada di antara piece yang baru saja diletakkan dan piece terdekat player menjadi piece player (secara vertikal, horizontal, dan diagonal).

<pre>Cherilyn's Turn   1  2  3  4  5  6 1 [ ][ ][ ][ ][ ][ ] 2 [ ][ ][ ][ ][ ][ ] 3 [ ][ ][X][O][ ][ ] 4 [ ][ ][O][X][ ][ ] 5 [ ][ ][ ][ ][ ][ ] 6 [ ][ ][ ][ ][ ][ ] X: 4 Y: 5</pre>	<pre>Mikhael's Turn   1  2  3  4  5  6 1 [ ][ ][ ][ ][ ][ ] 2 [ ][ ][ ][ ][ ][ ] 3 [ ][ ][X][O][ ][ ] 4 [ ][ ][O][O][ ][ ] 5 [ ][ ][ ][O][ ][ ] 6 [ ][ ][ ][ ][ ][ ] X: 5 Y: 3</pre>	<pre>Cherilyn's Turn   1  2  3  4  5  6 1 [ ][ ][ ][ ][ ][ ] 2 [ ][ ][ ][ ][ ][ ] 3 [ ][ ][X][X][X][ ] 4 [ ][ ][O][O][ ][ ] 5 [ ][ ][ ][O][ ][ ] 6 [ ][ ][ ][ ][ ][ ] X:  </pre>
---	--	--

Game akan berakhir jika board sudah penuh atau semua piece pada board didominasi 1 player. Ketika game over, tampilkan nama player yang menang serta jumlah piece yang dimiliki oleh player tersebut dengan format <Nama Player> Wins! Score: <jumlah piece>.

## Menu 2: High Score

Ketika menu 2 dipilih, tampilkan 3 score tertinggi yang tersimpan (diurutkan secara descending dari score tertinggi) dengan tampilan sebagai berikut:

```
>> 2
High Score
1. Mikhael - 20
2. Cherilyn - 18
3. Enrico - 16
```

**Menu high score hanya akan diperiksa jika dapat bermain kembali dan data high score tersimpan.**

## Menu 3: Exit

Jika player memilih menu 0, hentikan program.

Notes:

**WAJIB** menggunakan minimal 3 syntax berbeda khusus kotlin. Berikut adalah daftar syntax yang bisa digunakan (apabila ingin menggunakan syntax di luar ini, silahkan tanya di grup):

1. For / If in range operator
2. Compact Functions
3. If sebagai expression
4. Map list
5. Lambda Function
6. Higher Order Function

Untuk mempermudah pengecekan, berikan comment pada syntax yang dimaksud contoh

```
//SYNTAX KHUSUS KOTLIN 1, Compact function
fun double(x: Int):Int = x * 2
```

Cara menggunakan syntax-syntax tersebut dapat dibaca di file tutor.

Sertakan juga file txt berisi syntax kotlin yang kalian gunakan dan line codingan untuk syntax tersebut.

**DILARANG MENGGUNAKAN KONSEP MAUPUN MATERI YANG BELUM DIAJARKAN  
JIKA MELANGGAR MAKA NILAI TUGAS: 0**

## **PERHATIKAN KETENTUAN DIBAWAH :**

- **Highlight kriteria yang dikerjakan dengan warna kuning dan kumpulkan word beserta dengan file tugas, apabila tidak dikumpulkan maka tugas tidak akan diperiksa.**
- **Akan ada pengurangan nilai sebesar -5 untuk setiap kriteria yang dihighlight namun tidak dikerjakan.**
- **MENCONTEK = Nilai MOD 2**
- **WAJIB MENGHAPUS FOLDER NODE\_MODULES SEBELUM MENGUMPULKAN**

## TUGAS : 30

SCORE	KRITERIA
0/2	Tampilan awal board Othello rapi dan sesuai
0/1	Dapat menerima inputan nama player
0/1	Dapat menampilkan giliran player yang sedang bermain
0/2	Dapat menerima input koordinat piece dan menambahkan piece ke board

0/2/4/6	Pengecekan saat menambah piece baru benar (2: jika hanya salah satu)
0/3/7	Dapat mengubah piece di antara piece baru dan piece lain terdekat setelah menambah piece (3: jika tidak bisa diagonal)
0/2	Dapat game over dengan benar
0/2	Dapat bermain kembali dengan benar
0/1/3	Dapat menampilkan 3 score tertinggi (1: jika tidakurut)
-15 / -10 / -5 / 2	Menggunakan syntax khusus kotlin (minimal 3 yang berbeda)
Total : 30	

Menyetujui

Mengetahui

Penyusun Soal

(Evan Kusuma Susanto,  
S.Kom., M.Kom.)  
Koordinator Kuliah

(Grace Levina Dewi, M.Kom.)  
Koordinator Laboratorium

(Cherilyn Eugenia)  
Asisten