

# Taman Melodi

## Tugas Pemrograman 2 DDP1 2020/2021

Perjalanan membawa Anda ke sebuah taman. Sebuah taman yang asri dan indah. Angin berhembus lirih, perlahan menyentuh kulit. Sementara pepohonan dan rerumputan bergoyang seolah kegirangan. Anda lalu mendengar suara senandung dari tengah taman. Suara yang lembut dan penuh rasa. Ketika Anda mengikuti suara tersebut, Anda melihat sosok gadis tengah sibuk dengan dunianya sendiri. Butuh beberapa lama hingga ia menyadari kehadiran Anda.

Ia tampak sedikit tidak siap dengan kehadiran Anda, sedikit menjaga jaraknya. Sepertinya ia tidak biasa dengan kehadiran orang luar. Anda mencoba untuk membuat suasana yang lebih nyaman dengan memulai pembicaraan dengan gadis tersebut.

“Apakah kamu menyukai musik?” tanya Anda lembut.

Sang gadis menganggukkan kepala halus. Cukup memberikan pertanda persetujuan terhadap pertanyaan anda.

“Kalau begitu, apakah kamu mau memainkan sebuah *game* musik?”

Lagi-lagi, si gadis hanya memberikan persetujuan dengan anggukkan, tapi persetujuan itu yang Anda butuhkan.

“Kalau begitu, aku akan membuatkan sebuah game untukmu!”

Anda lalu mengeluarkan peralatan khusus Anda dan mulai membuatkan sebuah permainan sederhana untuk si gadis. Seperti di tempat sebelumnya, di sini ada *memory* yang bermain-main, melayang-layang di sekitar anda. Anda dapat memahami lagu-lagu apa saja yang si gadis ketahui. Dengan informasi tersebut Anda mulai membuatkan permainan yang Anda janjikan.

# Requirements

Permainan terdiri atas **5 ronde**. Sebelum mulai, pemain akan diminta untuk memasukkan namanya. Setelah pemain memasukkan **nama yang valid**, pemain diminta memilih salah satu **Game Mode**, **Normal** atau **Expert**. Permainan dimulai setelah semua persiapan selesai.

Pada setiap ronde, akan ditampilkan **4 baris lirik berurutan** dari sebuah lagu yang dipilih secara acak. Pemain akan diminta untuk **menebak kelanjutan dari lirik lagu** yang ditampilkan. Jika pemain menjawab benar, skor pemain akan bertambah. Jika salah, nyawa pemain akan berkurang sesuai Game Mode yang dipilih di awal permainan. Saat nyawa pemain habis sebelum permainan selesai, tampilkan **GAME OVER**.

Di akhir permainan, tampilkan skor yang berhasil diraih oleh pemain. Bila skor pemain melampaui skor tertinggi yang pernah ada, tampilkan **NEW HIGH SCORE** dan simpan skor tertinggi yang baru ke dalam sebuah file `highscore.txt`. Lalu, permainan selesai.

## Fitur

### 1. Nama Pemain

Tanyakan nama pemain saat `berpacu_dalam_lirik.py` pertama kali dibuka. Nama pemain ini akan kembali ditampilkan saat akhir permainan, dan disimpan dalam `highscore.txt` bila pemain berhasil meraih **new high score**.

Catatan:

- Nama pemain tidak boleh “**null**” maupun **string kosong**.
- Apabila pemain berusaha menggunakan nama “**null**” atau memasukkan **string kosong**, pemain akan diminta untuk memilih nama lain **sampai nama yang dimasukkan bukan “null” maupun string kosong**.

### 2. Game Mode

`berpacu_dalam_lirik.py` memiliki dua mode kesulitan, **Normal** dan **Expert**.

Mode Normal:

- Pemain memiliki **3 kesempatan** untuk keseluruhan permainan.
- Kesempatan **akan berkurang satu** untuk setiap jawaban pemain yang salah.
- Saat kesempatan habis, permainan akan berakhir dan tampilkan **GAME OVER**.

Mode Expert:

- Pemain hanya memiliki **1 kesempatan** untuk keseluruhan permainan.
- Sekali pemain menjawab salah, langsung tampilkan **GAME OVER**.

### 3. Lirik Misteri dan Prompt

`berpacu_dalam_lirik.py` akan memilih sebuah lagu secara acak dari folder lirik lagu yang disediakan. Dari lirik lagu tersebut, `berpacu_dalam_lirik.py` akan secara acak memilih sebaris Lirik Misteri, yaitu **baris lagu yang akan ditebak** oleh pemain. **Empat baris lirik sebelum Lirik Misteri** akan ditampilkan sebagai *prompt* supaya pemain dapat menebak baris Lirik Misteri nya.

Catatan:

- Prompt **akan selalu terdiri atas 4 baris**. `berpacu_dalam_lirik.py` akan mengulangi pemilihan secara acak bila ternyata Lirik Misteri termasuk dalam empat baris pertama dari lirik lagu.
- `berpacu_dalam_lirik.py` akan **mendeteksi secara otomatis** nama-nama file lirik lagu yang tersedia dalam folder, kamu tidak diminta untuk memasukkan nama file satu persatu.
- Lagu yang sudah dipilih **tidak akan terpilih kembali** untuk ronde berikutnya dalam permainan yang sama.

### 4. Score

Mekanisme *scoring* dalam `berpacu_dalam_lirik.py` adalah sebagai berikut:

- Apabila jawaban pemain benar, skor akan bertambah **sebanyak panjang karakter Lirik Misteri**.
- Bila jawaban salah, tangani sesuai Game Mode yang dipilih pemain.
- Jawaban dianggap benar bila setiap karakter dari jawaban pemain sama dengan Lirik Misteri, **tanpa mempedulikan huruf besar/huruf kecil**.

Contoh 1:

Lirik Misteri = balonku tinggal empat	
balonku tinggal empat	BENAR
BALONKU TINGGAL EMPAT	BENAR
bAlOnKU tInGgAL eMpAt	BENAR
b@lOnKu 7!n8g4L 3mp4t	SALAH
-balonku tinggal empat!	SALAH
balon ku tinggal empat	SALAH
ballonku tinggal empat	SALAH
balonku tinggal 4	SALAH
tinggal empat balonku	SALAH
satu dua tiga sayang semuanya	SALAH BANGET

Contoh 2:

Lirik Misteri = Merdeka! Merdeka!	
merdeka! merdeka!	BENAR
MerdekA! MerDekA!	BENAR
merdeka merdeka	SALAH
merdeka~ merdeka~	SALAH
merdeka merdeka!	SALAH

## 5. Winning dan Game Over

Bila pemain berhasil menyelesaikan lima ronde permainan sebelum nyawanya habis, berikan tampilan **SELAMAT** serta tunjukkan skor terakhir yang berhasil diraih.

Contoh:

```
SELAMAT!  
Anda berhasil menyelesaikan permainan  
Hasil Akhir:  
score      : 12
```

Tampilan **GAME OVER** akan muncul bila pemain kehabisan nyawa sebelum permainan berakhir. Tampilkan tulisan **GAME OVER** serta nama pemain dan skor yang berhasil diraih pada ronde-ronde sebelumnya.

Contoh:

```
GAME OVER  
Sayang sekali Lele, anda terhenti disini  
Hasil Akhir:  
score      : 12
```

Jika pemain berhasil meraih rekor baru skor tertinggi, tampilkan **NEW HIGH SCORE**, nama pemain, serta skor yang berhasil diraih.

Contoh:

```
NEW HIGH SCORE!!!  
username   : Lele  
score      : 12  
Berhasil meraih highscore di mode Expert  
~~~~~ Thanks for playing ~~~~~
```

## 6. High Score

Untuk menghargai pemain yang jago, pencapaiannya akan disimpan dalam sebuah file `highscore.txt`. File tersebut berisi **Game Mode**, **nama pemain**, dan **skor tertinggi** yang diraih. Dalam setiap permainan, apabila ada pemain baru yang berhasil meraih

skor lebih tinggi dibanding high score yang ada, file `highscore.txt` akan diperbarui. Jika skor baru tidak melampaui, `highscore.txt` tidak berubah.

File `highscore.txt` akan langsung dibuat apabila belum ada dan langsung berisi nilai sebagaimana isi file saat belum ada yang bermain.

Contoh file `highscore.txt`:

- Saat **belum ada** pemain yang bermain mode Normal maupun Expert:

```
Normal null 0
Expert null 0
```

- Saat **hanya ada salah satu** Game Mode yang pernah dimainkan:

```
Normal Foigol 39
Expert null 0
```

atau

```
Normal null 0
Expert Lele 55
```

- Saat **kedua Game Mode** sudah pernah dimainkan:

```
Normal Foigol 39
Expert Lele 55
```

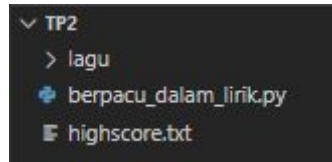
- Saat pemain Lele bermain kembali di mode Normal dan mendapatkan skor yang lebih tinggi dibanding pemain Foigol:

```
Normal Lele 42
Expert Lele 55
```

## Constraints

- Setiap file lirik lagu yang disediakan **dijamin memiliki paling sedikit 5 baris lirik**.
- Nama setiap file lirik lagu **dijamin unik**.
- Input tebakan dari user dapat berupa **string kosong**. Input tersebut akan tetap dianggap sebagai **jawaban yang salah**.
- Input tebakan dari user dapat berisi karakter alphanumeric maupun special character seperti tanda baca, simbol, dan emoji. Jawaban **hanya dianggap benar apabila memenuhi kriteria** yang sudah dijelaskan pada fitur Score.
- Beberapa file lirik lagu dalam format `.txt` **sudah disediakan**. Tetapi, kamu diperbolehkan menambah atau mengganti dengan lagu-lagu favoritmu (min. 5 file `.txt`).

Pastikan semua file lirik lagu berada dalam satu folder bernama lagu supaya bisa **terbaca**. Berikut contoh directory workspace yang benar:



# Alur Program

Format Output:

```
[ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ]  
[ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ]  
[ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ]  
DALAM  
LIRIK  
~~~~~  
masukkan username      :  
mode (normal/extreme)  :  
~~~~~  
Good Luck & Have Fun :)  
  
Round XX  
Nyawa   : XX  
Score   : XX  
~~~~~  
Judul lagu : XXXXXXXXXX  
{baris lirik 1}  
{baris lirik 2}  
{baris lirik 3}  
{baris lirik 4}  
Silakan menebak:  
{tebakan_user}  
Jawaban BENAR  
  
Round XX  
HP       : XX  
Score    : XX  
~~~~~  
Judul lagu : XXXXXXXXXX  
{baris lirik 1}  
{baris lirik 2}  
{baris lirik 3}  
{baris lirik 4}  
Silakan menebak:  
{tebakan_user}  
Jawaban SALAH
```

```

Jawaban : {lirik asli}

GAME OVER
Sayang sekali {username}, anda terhenti disini
Hasil Akhir:
score      : {score}

SELAMAT!
Anda berhasil menyelesaikan permainan
Hasil Akhir:
score      : {score}

NEW HIGH SCORE!!!
username : {username}
score    : {score}
Berhasil meraih highscore di mode {mode}
~~~~~ Thanks for playing ~~~~~

```

## To Do

1. Gunakan [template](#) untuk mempermudah pekerjaanmu.
2. Implementasikan menu awal sesuai contoh alur program.
3. Buat implementasi sesuai *requirements* di atas.
4. Berikan dokumentasi untuk menjelaskan alur program yang kamu tulis. **Kamu dianjurkan untuk menggunakan PEP-8 sebagai acuan.**
5. Kumpulkan dengan format: **[KodeAsdos]\_[Nama]\_[NPM]\_[Kelas]\_TP2.zip** yang berisi file tugas pemrograman kamu.  
Contoh: **LIT\_KinoMakoto\_1234567890\_B\_TP2.zip**
6. Kamu cukup mengumpulkan file .py dalam zip, **tidak perlu mengikutsertakan folder lagu maupun file highscore.txt.**

## Resources

File template	<a href="#">template.py</a>
Folder lagu	<a href="#">lagu.zip</a>
Contoh interaksi program	<ul style="list-style-type: none"> <li>• <a href="#">Contoh interaksi 1</a></li> <li>• <a href="#">Contoh interaksi 2</a></li> <li>• <a href="#">Contoh interaksi 3</a></li> </ul>

SEMANGAAT

(◡◡◡) ◡ ———