

LAPORAN UJIAN AKHIR SEMESTER

GAME DEVELOPMENT

Media Player



Dosen Pembimbing :

Yusuf Fadlila Rachman, S.Kom., M.Kom

Oleh :

Zulfi Masyita Resia Reswati

V3920063 / TI E

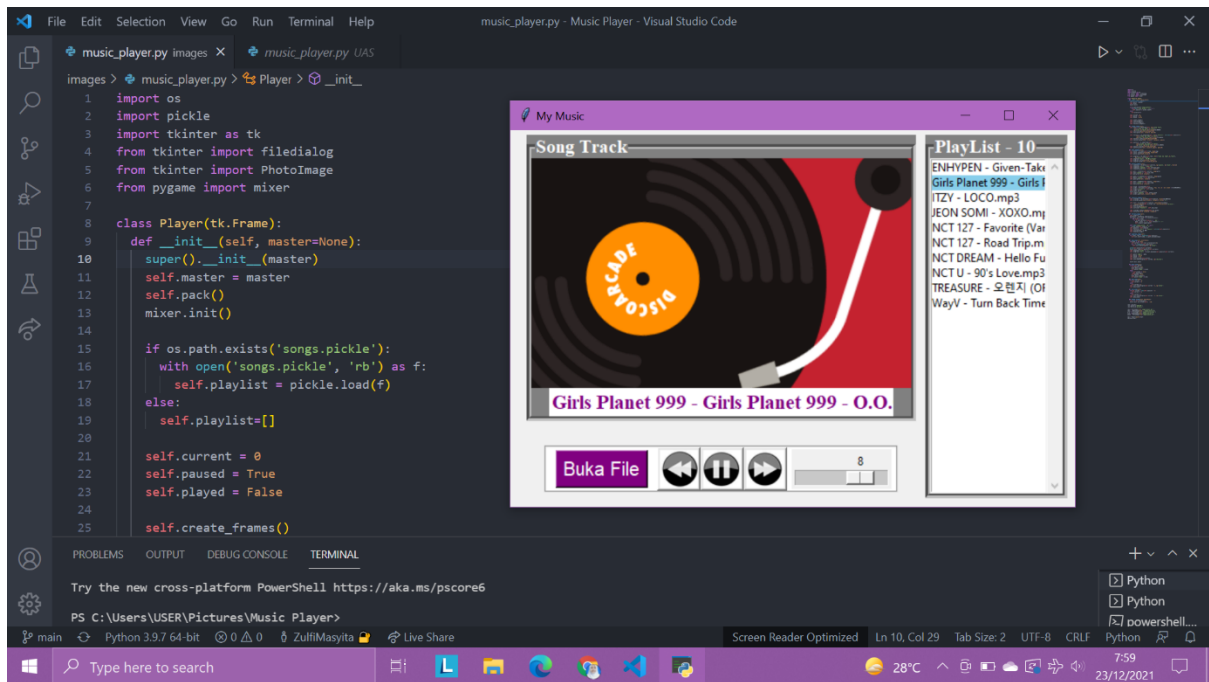
UNIVERSITAS SEBELAS MARET

D3 TEKNIK INFORMASI

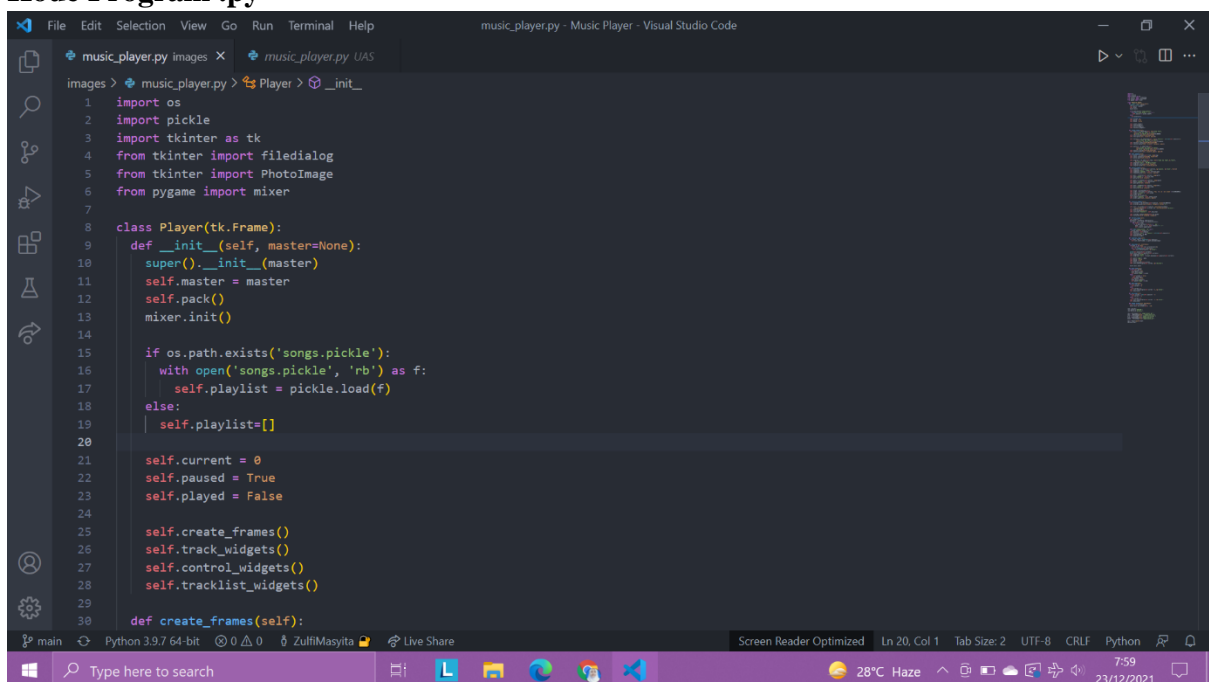
TAHUN 2020/2021

Hasil Running Python

Hasil running dibawah ini menampilkan music player yang dapat memutar beberapa lagu yang dimasukkan kedalam playlist. Untuk memasukkan lagu kedalam playlist dapat klik “buka file”, dan memilih folder yang berisikan beberapa lagu dan klik open. Jika sudah maka akan muncul pada playlist seperti dibawah ini. Music player ini juga dapat menampilkan judul file lagu sesuai dengan lagu yang diputar. Di music player ini terdapat beberapa GUI yaitu next, pause, play, previous, frame dan pengaturan volume. Yang masing masing dari GUI tersebut sudah berfungsi sesuai fungsinya. Misalnya untuk memutar lagu klik tombol play, mengatur volume lagu dapat menggeser volume yang telah tampil pada music player seperti dibawah ini.



Kode Program .py



```
File Edit Selection View Go Run Terminal Help music_player.py - Music Player - Visual Studio Code
music_player.py images X music_player.py UAS
images > music_player.py > Player > _init_
30 def create_frames(self):
31     self.track = tk.LabelFrame(self, text='Song Track',
32                               font=("times new roman",15,"bold"),
33                               bg="grey",fg="white",bd=5,relief=tk.GROOVE)
34     self.track.config(width=410,height=300)
35     self.track.grid(row=0, column=0, padx=10)
36
37     self.tracklist = tk.LabelFrame(self, text=f'PlayList - {str(len(self.playlist))}',
38                                   font=("times new roman",15,"bold"),
39                                   bg="grey",fg="white",bd=5,relief=tk.GROOVE)
40     self.tracklist.config(width=190,height=400)
41     self.tracklist.grid(row=0, column=1, rowspan=3, pady=5)
42
43     self.controls = tk.LabelFrame(self,
44                                  font=("times new roman",15,"bold"),
45                                  bg="white",fg="white",bd=2,relief=tk.GROOVE)
46     self.controls.config(width=410,height=80)
47     self.controls.grid(row=2, column=0, pady=5, padx=10)
48
49 def track_widgets(self):
50     self.canvas = tk.Label(self.track, image=img)
51     self.canvas.config(width=400, height=240)
52     self.canvas.grid(row=0,column=0)
53
54     self.songtrack = tk.Label(self.track, font=("times new roman",16,"bold"),
55                               bg="white",fg="purple")
56     self.songtrack['text'] = 'My Music Player'
57     self.songtrack.config(width=30, height=1)
58     self.songtrack.grid(row=1,column=0,padx=10)
59
60 def control_widgets(self):
61     self.loadSongs = tk.Button(self.controls, bg='purple', fg='white', font=10)
62     self.loadSongs['text'] = 'Buka File'
63     self.loadSongs['command'] = self.retrieve_songs
64     self.loadSongs.grid(row=0, column=0, padx=10)
65
66     self.prev = tk.Button(self.controls, image=prev)
67     self.prev['command'] = self.prev_song
68     self.prev.grid(row=0, column=1)
69
70     self.pause = tk.Button(self.controls, image=pause)
71     self.pause['command'] = self.pause_song
72     self.pause.grid(row=0, column=2)
73
74     self.next = tk.Button(self.controls, image=next_)
75     self.next['command'] = self.next_song
76     self.next.grid(row=0, column=3)
77
78     self.volume = tk.DoubleVar(self)
79     self.slider = tk.Scale(self.controls, from_ = 0, to = 10, orient = tk.HORIZONTAL)
80     self.slider['variable'] = self.volume
81     self.slider.set(8)
82     mixer.music.set_volume(0.8)
83     self.slider['command'] = self.change_volume
84     self.slider.grid(row=0, column=4, padx=5)
85
86
87 def tracklist_widgets(self):
88     self.scrollbar = tk.Scrollbar(self.tracklist, orient=tk.VERTICAL)
89     self.scrollbar.grid(row=0, column=0, rowspan=3, sticky='ns')
```

```
music_player.py - Music Player - Visual Studio Code

def tracklist_widgets(self):
    self.scrollbar = tk.Scrollbar(self.tracklist, orient=tk.VERTICAL)
    self.scrollbar.grid(row=0, column=1, rowspan=5, sticky='ns')

    self.list = tk.Listbox(self.tracklist, selectmode=tk.SINGLE,
                           yscrollcommand=self.scrollbar.set, selectbackground='sky blue')
    self.enumerate_songs()
    self.list.config(height=22)
    self.list.bind('<Double-1>', self.play_song)

    self.scrollbar.config(command=self.list.yview)
    self.list.grid(row=0, column=0, rowspan=5)

def retrieve_songs(self):
    self.songlist = []
    directory = filedialog.askdirectory()
    for root_, dirs, files in os.walk(directory):
        for file in files:
            if os.path.splitext(file)[1] == '.mp3':
                path = (root_ + '/' + file).replace('\\', '/')
                self.songlist.append(path)

    with open('songs.pickle', 'wb') as f:
        pickle.dump(self.songlist, f)
    self.playlist = self.songlist
    self.tracklist['text'] = f'Playlist - {str(len(self.playlist))}'
    self.list.delete(0, tk.END)
    self.enumerate_songs()

def enumerate_songs(self):
    for index, song in enumerate(self.playlist):
        self.list.insert(index, os.path.basename(song))

def play_song(self, event=None):
    if event is not None:
        self.current = self.list.curselection()[0]
        for i in range(len(self.playlist)):
            self.list.itemconfigure(i, bg="white")

    print(self.playlist[self.current])
    mixer.music.load(self.playlist[self.current])
    self.songtrack['anchor'] = 'w'
    self.songtrack['text'] = os.path.basename(self.playlist[self.current])

    self.pause['image'] = play
    self.paused = False
    self.played = True
    self.list.activate(self.current)
    self.list.itemconfigure(self.current, bg='sky blue')

    mixer.music.play()

def pause_song(self):
    if not self.paused:
        self.paused = True
        mixer.music.pause()
        self.pause['image'] = pause
    else:
```

