

KELOMPOK 13



PROJECT SDA

PLAYLIST MUSIC

▶ Play

Anggota Kelompok :



Ikhrima Atusifah
23031554181



Arina Tri Yuni W. T.
23031554203



Helen Nur S.
23031554231

Struktur Data dan Algoritma 2023A



Daftar Isi



KATA PENGANTAR



LATAR BELAKANG



TUJUAN
DAN
MANFAAT



PENJELASAN
KODE



Syukur alhamdulillah senantiasa kami panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan projek ini guna memenuhi Ujian Akhir Semester untuk mata kuliah Struktur Data dan Algoritma, dengan judul projek : "Playlist Music". Projek ini dapat terselesaikan karena adanya keaktifan dari kelompok dan berbagai pihak. Untuk itu, kami mengucapkan terima kasih kepada pihak-pihak yang mendukung kami, diantaranya Dr. Atik Wintarti, M.Kom. selaku dosen pengampu dan teman-teman mata kuliah SDA kelas 2023A, serta tim yang telah bekerja keras untuk dapat menyelesaikan projek ini.

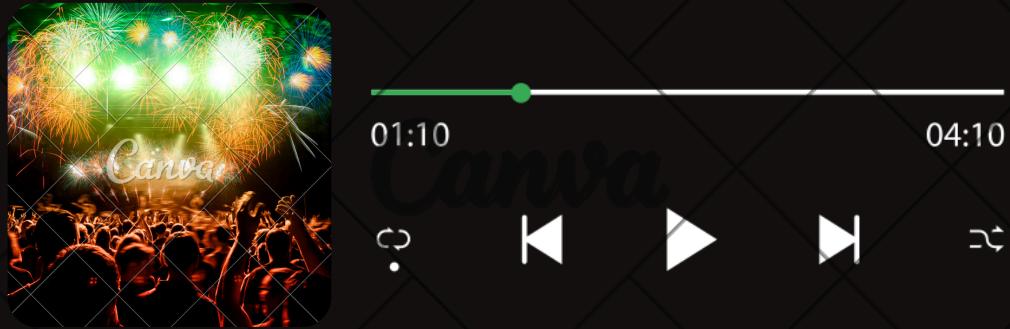
Kami menyadari sepenuhnya bahwa projek ini masih jauh dari kata sempurna dikarenakan terbatasnya pengalaman dan pengetahuan yang kami miliki. Oleh karena itu, kami mengharapkan segala bentuk saran maupun kritikan yang membangun dari berbagai pihak.

Akhir kata, kami berharap semoga Allah SWT berkenan membalaq segala kebaikan semua pihak yang telah membantu dan semoga projek ini dapat memberikan manfaat.



PLAYLIST MUSIC

LATAR BELAKANG



Dalam era digital saat ini, musik telah menjadi bagian dari kehidupan sehari-hari. Musik dapat diakses melalui berbagai macam platform yang disediakan oleh internet pada saat ini, seperti YouTube, Spotify, dan lain-lain. Para pendengar musik tentu memiliki selera musiknya masing-masing, seringkali mereka hanya ingin mendengarkan musik yang menjadi favoritnya saja dan menghindari musik-musik yang sekiranya tidak disukai.

Dari permasalahan di atas, kami memutuskan untuk membantu para pendengar musik dalam mengumpulkan lagu-lagu kesukaannya. Oleh karena itu, kami membuat program untuk menentukan lagu apa saja yang masuk ke Playlist favorit pendengar sehingga pendengar tidak kesusahan menemukan lagu kesukaannya.

PLAYLIST MUSIC



TUJUAN

1. Peningkatan kenyamanan
2. Konsumsi musik secara efisien
3. Optimalisasi waktu
4. Personalisasi selera lagu



MANFAAT

1. Memudahkan untuk mengelompokkan lagu-lagu favorit
2. Menghindari lagu yang tidak disukai
3. Menyesuaikan lagu dengan suasana dan aktivitas
4. Fleksibilitas dalam pengaturan lagu

PLAYLIST MUSIC

Playlist Musik ~

Penjelasan Kode dibagi menjadi beberapa bagian,
seperti:

Struktur Data

Algoritma

Operasi

Pengaplikasian

PENJELASAN KODE



Struktur Data yang kami gunakan adalah "**Stack**".

```
1  class Song:  
2      def __init__(self, title, artist):  
3          self.title = title  
4          self.artist = artist  
5  
6      def __str__(self):  
7          return f"{self.title} - {self.artist}"  
8  
9  class Playlist:  
10     def __init__(self):  
11         self.songs = []  
12         self.history = []  
13
```

Deklarasi Stack pada kode kami diawali dengan pembuatan class stack, atau dalam kode ini diberi nama class Playlist dan kemudian menginisialisasi list dalam class Playlist. Sedangkan class Song memiliki fungsi untuk menyimpan dan menampilkan informasi lagu dalam Playlist.



```
def bubble_sort_by_title(self):
    n = len(self.songs)
    for i in range(n):
        for j in range(0, n-i-1):
            if self.songs[j].title.lower() > self.songs[j+1].title.lower():
                self.songs[j], self.songs[j+1] = self.songs[j+1], self.songs[j]

def bubble_sort_by_artist(self):
    n = len(self.songs)
    for i in range(n):
        for j in range(0, n-i-1):
            if self.songs[j].artist.lower() > self.songs[j+1].artist.lower():
                self.songs[j], self.songs[j+1] = self.songs[j+1], self.songs[j]
```

Algoritma yang kami gunakan yakni **“Sorting”**.

Tepatnya, kami memilih **Bubble Sort** untuk mengurutkan lagu dalam Playlist berdasarkan judul lagu dan atau berdasarkan nama artis.



```
def sorting(self):
    print("1. Urutkan berdasarkan judul")
    print("2. Urutkan berdasarkan penyanyi")
    sub_choice = input("Masukkan pilihan Anda (1/2): ")
    if sub_choice == '1':
        self.bubble_sort_by_title()
        print("Playlist diurutkan berdasarkan Judul Lagu:")
        for song in self.songs:
            print(song)
    elif sub_choice == '2':
        self.bubble_sort_by_artist()
        print("Playlist diurutkan berdasarkan Penyanyi:")
        for song in self.songs:
            print(song)
    else:
        print("Pilihan tidak valid. Silakan coba lagi.")
```

Kode disamping menunjukkan fungsi gabungan dari kode bubble sort sebelumnya. Jadi, saat menjalankan menu 'urutkan' maka yang dipanggil adalah fungsi sorting ini dan dalam fungsi ini akan memanggil fungsi bubble sort.

OPERASI



```
def push(self, song):
    self.songs.append(song)
    self.history.append(("add", song))
```

Operasi “**pop**” berguna untuk menghapus lagu yang baru saja ditambahkan dalam Playlist. Operasi ini merupakan ciri dari Stack, yakni *Last In First Out*

```
def pop(self):
    if not self.history:
        print("Tidak ada tindakan yang dapat dibatalkan.")
        return
    action, song = self.history.pop()
    if action == "add":
        self.songs.remove(song)
        print(f'Lagu '{song.title}' dibatalkan untuk ditambahkan dari playlist.')
    elif action == "remove":
        self.songs.append(song)
        print(f'Lagu '{song.title}' dibatalkan untuk dihapus dari playlist.)
```

Operasi “**push**” merupakan salah satu operasi yang terdapat pada Stack yang berfungsi untuk menambah lagu dalam Playlist

OPERASI



```
def remove_song(self, title, artist):
    song = self.search_by_title_and_artist(title, artist)
    if song:
        self.songs.remove(song)
        self.history.append(("remove", song))
        return True
    return False
```

Kode disamping merupakan operasi untuk mencari lagu berdasarkan judul lagu dan nama artis. Sedangkan untuk fungsi paling bawah, ini akan dipanggil pada fungsi remove_song tadi.

Kode disamping merupakan operasi untuk menghapus lagu dalam Playlist.

```
def search_by_title(self, title):
    return [song for song in self.songs if song.title.lower() == title.lower()]

def search_by_artist(self, artist):
    return [song for song in self.songs if song.artist.lower() == artist.lower()]

def search_by_title_and_artist(self, title, artist):
    for song in self.songs:
        if song.title.lower() == title.lower() and song.artist.lower() == artist.lower():
            return song
    return None
```

PLAYLIST MUSIC

OPERASI



```
def display_playlist(self, sorted_by=None):
    if not self.songs:
        print("Playlist kosong.")
        return
    if sorted_by == 'title':
        self.bubble_sort_by_title()
    elif sorted_by == 'artist':
        self.bubble_sort_by_artist()
    for i, song in enumerate(self.songs, start=1):
        print(f"{i}. {song}")
```

Kode diatas merupakan fungsi untuk menampilkan lagu yang ada pada Playlist

```
def display_menu():
    print("\n===== Menu Playlist Musik =====")
    print("1. Tambah Lagu ke Playlist")
    print("2. Hapus Lagu dari Playlist")
    print("3. Hapus Lagu Terakhir di Playlist")
    print("4. Urutkan Lagu")
    print("5. Cari Lagu")
    print("6. Tampilkan Playlist")
    print("7. Keluar")
    print("=====")
```

Kode diatas merupakan menu yang akan ditampilkan saat keseluruhan kode dijalankan

PENGAPLIKASIAN



Kode disamping merupakan cara untuk mengaplikasikan setiap fungsi dan operasi yang ada di kode ini. Menu dapat dipilih hingga opsi ke-7 dengan setiap opsinya akan memanggil fungsi yang sesuai.

```
def main():
    playlist = Playlist()

    while True:
        display_menu()
        choice = input("Pilih opsi (1-7): ")

        if choice == '1':
            title = input("Masukkan judul lagu: ")
            artist = input("Masukkan nama artis: ")
            song = Song(title, artist)
            playlist.push(song)
            print("Lagu berhasil ditambahkan ke playlist.")

        elif choice == '2':
            title = input("Masukkan judul lagu yang ingin dihapus: ")
            artist = input("Masukkan nama artis: ")
            if playlist.remove_song(title, artist):
                print("Lagu berhasil dihapus dari playlist.")
            else:
                print("Lagu tidak ditemukan dalam playlist.")

        elif choice == '3':
            playlist.pop()
```

KELOMPOK 13



TERIMA KASIH

PLAYLIST MUSIC

► End