

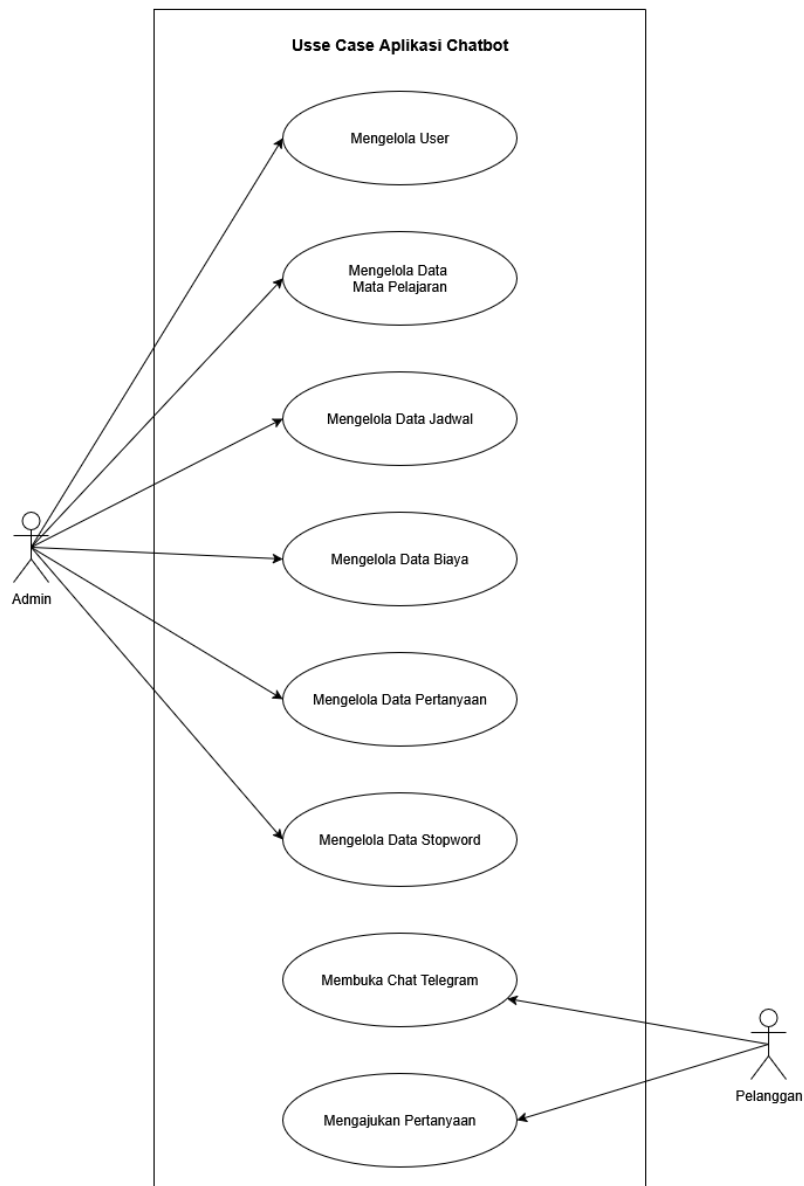
Implementasi Class Diagram

NPM-Nama lengkap : 5240411230-Muhammad Zaki As Shidiqi

Judul Tugas Akhir : Rancang Bangun Chatbot sebagai Pusat Informasi Lembaga Kursus dan Pelatihan Menggunakan Pendekatan Natural Language Processing

Nama pemilik karya : AGRA LAKSMI MAITRI

Use Case Diagram :

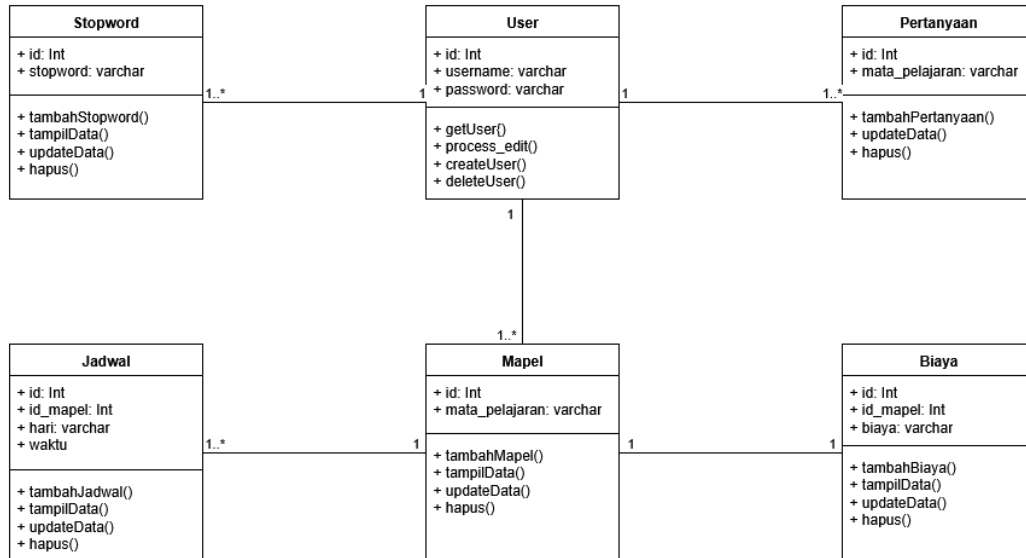


Penjelasan :

Berdasarkan usecase diagram tersebut dapat disimpulkan bahwa admin dapat melakukan 6 kegiatan, yaitu mengelola user, mengelola data mata pelajaran, mengelola data jadwal, mengelola data biaya, mengelola data pertanyaan, dan mengelola data

stopword. Pelanggan hanya dapat melakukan 2 kegiatan, yaitu membuka chat telegram dan mengajukan pertanyaan.

Class Diagram :



Penjelasan :

Berdasarkan usecase diagram tersebut terdapat 6 class yang memiliki atribut dan method sebagai berikut:

1. User

a. Atribut

- id bertipe integer.
- username bertipe varchar.
- password bertipe varchar.

b. Method

- getUser(), untuk mendapatkan / request data user.
- process_edit(), untuk melakukan edit.
- createUser(), dapat melakukan registrasi.
- deleteUser(), dapat menghapus akun.

2. Mapel

a. Atribut

- id bertipe integer.
- mata_pelajaran bertipe varchar.

b. Method

- tambahMapel(), untuk menambahkan mata pelajaran.
- tampilData(), untuk menampilkan data.
- updateData(), untuk memperbarui data.
- hapus(), untuk menghapus data.

3. Jadwal

- a. Atribut
 - id bertipe integer
 - id_mapel bertipe integer
 - hari bertipe varchar
 - waktu tidak diebutkan
 - b. Method
 - tambahJadwal(), untuk menambahkan jadwal.
 - tampilData(), untuk menampilkan data.
 - updateData(), untuk memperbarui data.
 - hapus(), untuk menghapus data.
4. Biaya
- a. Atribut
 - id bertipe integer.
 - id_mapel bertipe integer
 - biaya bertipe varchar
 - b. Method
 - tambahBiaya (), untuk menambahkan biaya.
 - tampilData(), untuk menampilkan data.
 - updateData(), untuk memperbarui data.
 - hapus(), untuk menghapus data.
5. Pertanyaan
- a. Atribut
 - id bertipe integer
 - mata_pelajaran bertipe varchar
 - b. Method
 - tambahPertanyaan (), untuk menambahkan pertanyaan.
 - updateData(), untuk memperbarui data.
 - hapus(), untuk menghapus data.
6. Stopword
- a. Atribut
 - id bertipe integer
 - stopword bertipe varchar
 - b. Method
 - tambahStopword (), untuk menambahkan stopword.
 - tampilData(), untuk menampilkan data.
 - updateData(), untuk memperbarui data.
 - hapus(), untuk menghapus data.

Implementasi Class Diagram :

```
class Stopword:
    def __init__(self, id, stopword):
        self.id = id
```

```

        self.stopword = stopword

    def tambah_stopword(self, kata):
        self.stopword = kata

    def tampil_data(self):
        return f"{self.id}. {self.stopword}"

    def update_data(self, new_word):
        self.stopword = new_word

    def hapus(self):
        del self.stopword

class User:
    def __init__(self, id, username, password):
        self.id = id
        self.username = username
        self.password = password

    def get_user(self):
        return {"id": self.id, "username": self.username}

    def process_edit(self, username=None, password=None):
        if username:
            self.username = username
        if password:
            self.password = password

    def create_user(self, id, username, password):
        return User(id, username, password)

    def delete_user(self):
        del self.username
        del self.password

class Mapel:

```

```

def __init__(self, id, mata_pelajaran):
    self.id = id
    self.mata_pelajaran = mata_pelajaran

def tambah_mapel(self, nama):
    self.mata_pelajaran = nama

def tampil_data(self):
    return f"[{self.id}] {self.mata_pelajaran}"

def update_data(self, nama_baru):
    self.mata_pelajaran = nama_baru

def hapus(self):
    del self.mata_pelajaran

class Jadwal:
    def __init__(self, id, id_mapel, waktu):
        self.id = id
        self.id_mapel = id_mapel
        self.waktu = waktu

    def tambah_jadwal(self, waktu):
        self.waktu = waktu

    def tampil_data(self):
        return f"Mapel dengan ID {self.id_mapel} dijadwalkan pada {self.waktu}"

    def update_data(self, waktu_baru):
        self.waktu = waktu_baru

    def hapus(self):
        del self.waktu

class Biaya:
    def __init__(self, id, id_mapel, biaya):

```

```

        self.id = id
        self.id_mapel = id_mapel
        self.biaya = biaya

    def tampil_data(self):
        return f"Biaya untuk mapel dengan ID {self.id_mapel}:  
Rp {self.biaya}"

    def update_data(self, biaya_baru):
        self.biaya = biaya_baru

    def delete(self):
        del self.biaya

    def tambah_biaya(self, biaya):
        self.biaya = biaya

class Pertanyaan:
    def __init__(self, id, mata_pelajaran, pertanyaan):
        self.id = id
        self.mata_pelajaran = mata_pelajaran
        self.pertanyaan = pertanyaan

    def tambah_pertanyaan(self, pertanyaan):
        self.pertanyaan = pertanyaan

    def tampil_data(self):
        return f"[{self.mata_pelajaran}] {self.pertanyaan}"

    def hapus(self):
        del self.pertanyaan

# Implementasi
if __name__ == "__main__":
    user1 = User(1, "admin", "1234")
    mapel1 = Mapel(1, "Bahasa Inggris")
    jadwal1 = Jadwal(1, mapel1.id, "Senin 09:00")

```

```
biaya1 = Biaya(1, mapel1.id, "500000")

print(user1.get_user())
print(mapel1.tampil_data())
print(jadwal1.tampil_data())
print(biaya1.tampil_data())
```

Penjelasan :

Program ini menggunakan konsep OOP, artinya terdapat class, atribut, dan method. Program ini dibuat berdasarkan class diagram sebelumnya telah dibahas. Setelah sintaks class terdapat nama class yang sudah sesuai dengan diagram. Pada awal baris setiap class terdapat `__init__`, ini akan menginisialisasi class tersebut akan memiliki atribut yang letaknya ada di dalam `__init__`. Setelah `__init__` akan berisi method-method, artinya setiap class dapat memiliki method lebih dari satu.

1. User

a. Atribut

- id bertipe integer.
- username bertipe varchar.
- password bertipe varchar.

b. Method

- `getUser()`, untuk mendapatkan / request data user.
- `process_edit()`, untuk melakukan edit.
- `createUser()`, dapat melakukan registrasi.
- `deleteUser()`, dapat menghapus akun.

2. Mapel

a. Atribut

- id bertipe integer.
- mata_pelajaran bertipe varchar.

b. Method

- `tambahMapel()`, untuk menambahkan mata pelajaran.
- `tampilData()`, untuk menampilkan data.
- `updateData()`, untuk memperbarui data.
- `hapus()`, untuk menghapus data.

3. Jadwal

a. Atribut

- id bertipe integer
- id_mapel bertipe integer

- hari bertipe varchar
- waktu tidak diebutkan
- b. Method
 - tambahJadwal(), untuk menambahkan jadwal.
 - tampilData(), untuk menampilkan data.
 - updateData(), untuk memperbaiki data.
 - hapus(), untuk menghapus data.
- 4. Biaya
 - a. Atribut
 - id bertipe integer.
 - id_mapel bertipe integer
 - biaya bertipe varchar
 - b. Method
 - tambahBiaya (), untuk menambahkan biaya.
 - tampilData(), untuk menampilkan data.
 - updateData(), untuk memperbaiki data.
 - hapus(), untuk menghapus data.
- 5. Pertanyaan
 - a. Atribut
 - id bertipe integer
 - mata_pelajaran bertipe varchar
 - b. Method
 - tambahPertanyaan (), untuk menambahkan pertanyaan.
 - updateData(), untuk memperbaiki data.
 - hapus(), untuk menghapus data.
- 6. Stopword
 - a. Atribut
 - id bertipe integer
 - stopword bertipe varchar
 - b. Method
 - tambahStopword (), untuk menambahkan stopword.
 - tampilData(), untuk menampilkan data.
 - updateData(), untuk memperbaiki data.
 - hapus(), untuk menghapus data.

Pada akhir program ini terdapat contoh implementasi untuk menjalankan program, berikut adalah output dari kode tersebut.


```
[ >_pwsh ] [ G:\..\..\..\..\meet-3 ] on [ P main ≡ ?1 -1 ] - [ base 3.12.8 ] 109ms
> py main.py
{'id': 1, 'username': 'admin'}
[1] Bahasa Inggris
Mapel dengan ID 1 dijadwalkan pada Senin 09:00
Biaya untuk mapel dengan ID 1: Rp 500000
```