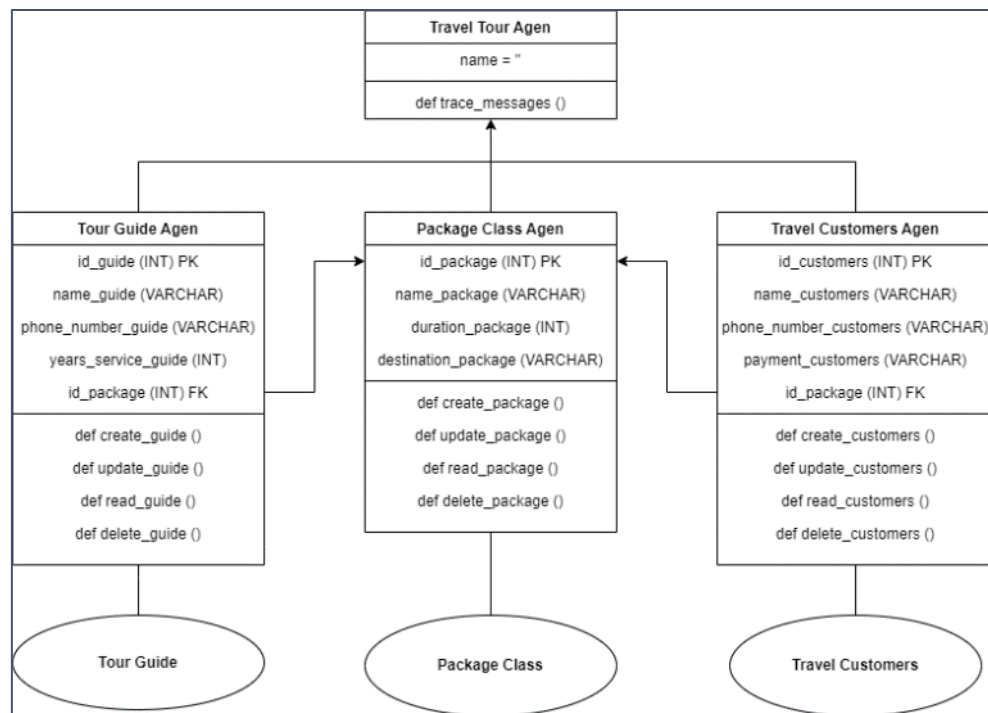


1. Class Diagram

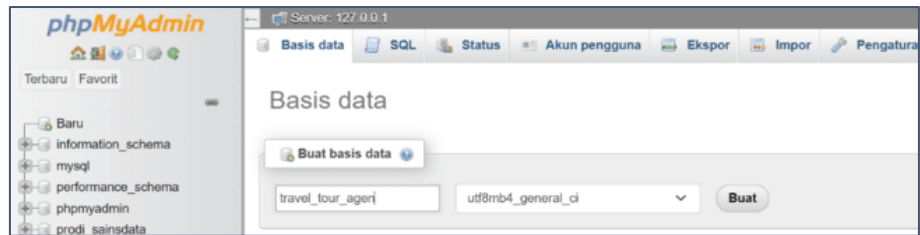


Tema class diagram untuk kebutuhan desain sistem basis data dan implementasinya ke dalam bahasa pemrograman python crud oop terintegrasi ke basis data MySQL. Di gunakan sebuah class diagram dengan judul “Travel Tour Agen” yang memiliki 3 kelas, yakni Package Class Agen, Tour Guide Agen, dan Travel Customers Agen, dengan Package Class Agen sebagai pusatnya.

Pada masing-masing kelas memiliki beberapa atribut, di antaranya yakni pada kelas “Package Class Agen” memiliki atribut `id_package`, `name_package`, `duration_package`, dan `destination_package`. Pada kelas “Tour Guide Agen” memiliki atribut `id_guide`, `name_guide`, `phone_number_guide`, `years_service_guide`, dan `id_package`. Pada kelas “Travel Customers Agen” memiliki atribut `id_customers`, `name_customers`, `phone_number_customers`, `payment_customers`, dan `id_package`.

2. Pembuatan Basis Data MySQL

2.1 Pembuatan Basis Data “Tour Travel Agen”



Terdapat basis data (database) yang di buat di MySQL dengan nama “travel_tour_agen” yang memiliki 3 kelas, yakni package_class_agen, tour_guide_agen, dan travel_customers_agen.

2.2 Pembuatan Basis Data Kelas “Package Class Agen”

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id_package	int(255)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lain
2	name_package	varchar(20)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lain
3	duration_package	int(10)			Tidak	Tidak ada			Ubah Hapus Lain
4	destination_package	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lain

Terdapat basis data kelas “package_class_agen” dengan atribut id_package yang memiliki tipe data integer, name_package yang memiliki tipe data varchar, duration_package yang memiliki tipe data integer, dan destination_package yang memiliki tipe data varchar.

2.3 Pembuatan Basis Data Kelas “Tour Guide Agen”

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id_guide	int(255)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lain
2	name_guide	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lain
3	phone_number_guide	varchar(15)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lain
4	years_servive_guide	int(10)			Tidak	Tidak ada			Ubah Hapus Lain
5	id_package	int(255)			Tidak	Tidak ada			Ubah Hapus Lain

Terdapat basis data kelas “tour_guide_agen” dengan atribut id_guide yang memiliki tipe data integer, name_guide yang memiliki tipe data varchar,

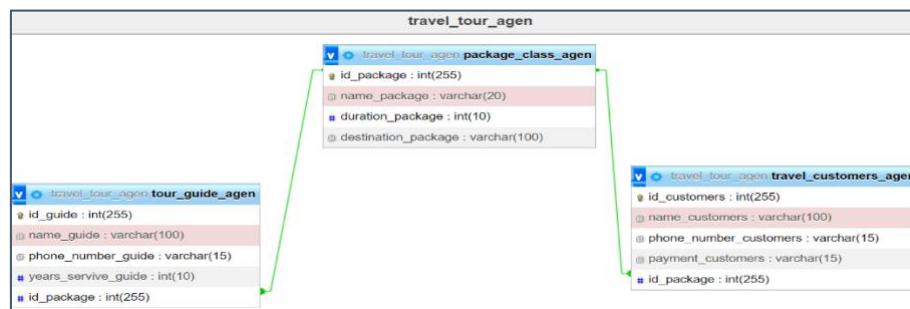
phone_number_guide yang memiliki tipe data varchar, years_service_guide yang memiliki tipe data integer, dan id_package yang memiliki tipe data integer.

2.4 Pembuatan Basis Data Kelas “Travel Customers Agen”

#	Nama	Jenis	Penyortiran	Atribut	Tak Terbilang	Bawaan	Komentar	Ekstra	Tindakan
1	id_customers	int(255)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus
2	name_customers	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus
3	phone_number_customers	varchar(15)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus
4	payment_customers	varchar(15)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus
5	id_package	int(255)			Tidak	Tidak ada			Ubah Hapus

Terdapat basis data kelas “travel_customers_agen” dengan atribut id_customers yang memiliki tipe data integer, name_customers yang memiliki tipe data varchar, phone_number_customers yang memiliki tipe data varchar, payment_customers yang memiliki tipe data varchar, dan id_package yang memiliki tipe data integer.

2.5 Pembuatan Desain Database



Terdapat desain database yang saling berelasi. Dengan id_package sebagai primary key yang berelasi dengan id_package pada tour_guide_agen dan id_package pada travel_customers_agen sebagai foreign key. Dengan kondisi cascade.

3. Implementasi Pembuatan Class CRUD (Create Read Update Delete) Pada Rancangan Class Diagram Travel Tour Agen

3.1 CRUD Pada Package Class Agen

```

In [1]: import mysql.connector

In [2]: class Database:
        my_db = my_cursor = None

        def __init__(self):
            global my_db, my_cursor
            my_db = mysql.connector.connect(
                host = "localhost",
                user = "root",
                password = "",
                database = "travel_tour_agen"
            )
            my_cursor = my_db.cursor()

        def __del__(self):
            my_db.commit()

```

Di gunakan sebuah module python mysql.connector yang di pergunakan untuk menghubungkan atau melakukan koneksi ke basis data mysql yang memiliki nama basis data “travel_tour_agen”.

```

In [3]: #package class agen

class package_class_agen(Database):

    #create
    def create_package(self, data):
        sql = "INSERT INTO package_class_agen (name_package, duration_package, destination_package) VALUES (%s, %s, %s)"

        try:
            my_cursor.execute(sql, data)
            my_db.commit()
            print(my_cursor.rowcount, "catatan di rekam.")

        except Exception as e:
            return e

        return my_cursor.lastrowid

```

```

#create many package
def create_many_package(self, data):
    sql = "INSERT INTO package_class_agen (name_package, duration_package, destination_package) VALUES (%s, %s, %s)"

    try:
        my_cursor.executemany(sql, data)
        my_db.commit()
        print(my_cursor.rowcount, "catatan di rekam.")

    except Exception as e:
        return e

#read
def read_package(self, mode = "DESC"):
    sql = "SELECT * FROM package_class_agen ORDER BY id_package {}".format(mode)

    try:
        my_cursor.execute(sql)
        result = my_cursor.fetchall()

    except Exception as e:
        return e

    return result

```

```

#update
def update_package(self, id, data):
    sql = "UPDATE package_class_agen SET name_package = %s, duration_package = %s, destination_package = %s WHERE id_package = %s"
    val = (data[0], data[1], data[2], id)

    try:
        my_cursor.execute(sql, val)
        my_db.commit()
        print(my_cursor.rowcount, "catatan di perbarui.")

    except Exception as e:
        return e

#delete
def delete_package(self, id):
    sql = "DELETE FROM package_class_agen WHERE id_package = {}".format(id)

    try:
        my_cursor.execute(sql)
        my_db.commit()
        print(my_cursor.rowcount, "catatan di hapus.")

    except Exception as e:
        return e

```

Pada kode script di atas, terdapat sebuah kelas database bernama package_class_agen yang akan di lakukan metode CRUD (create, read, update, delete) yang telah terkoneksi dengan MySQL. Create_package di gunakan untuk menambahkan satu paket, create_many_package di gunakan untuk menambahkan lebih dari satu paket, read_package di gunakan untuk menampilkan paket yang ada, update_package di gunakan untuk mengubah paket pada index/nomer tertentu, dan delete_package di gunakan untuk menghapus paket pada index/nomer tertentu.

➤ Menambahkan Package Class Di MySQL

The top screenshot shows a form for adding a new package class. The fields are:

- id_package: int(20)
- name_package: varchar(20)
- duration_package: int(10)
- destination_package: varchar(100)

The bottom screenshot shows the resulting data in a table view. The table has the following columns: id_package, name_package, duration_package, and destination_package. The data is as follows:

id_package	name_package	duration_package	destination_package
1	Independent Tour	1	Pantai Trikora, Pulau Penyengat, Danau Biru

Di lakukan penambahan paket pada kelas database package_class agen. Dengan name_package “Independent Tour”, duration_package 1, dan destination_package “Pantai Trikora, Pulau Penyengat, Danau Biru”.

➤ Create (Menambahkan) Package Class

The top screenshot shows a Python script for creating a package class. The code is as follows:

```
In [4]: #create single package
pc = package_class_agen()
data = ("Group Tour", 3, "Dinding Friwen, Kabui Passage, Dermaga Pulau")
pc.create_package(data)
1 catatan di rekam.
Out[4]: 2
```

The bottom screenshot shows the resulting data in a table view. The table has the following columns: id_package, name_package, duration_package, and destination_package. The data is as follows:

id_package	name_package	duration_package	destination_package
1	Independent Tour	1	Pantai Trikora, Pulau Penyengat, Danau Biru
2	Group Tour	3	Dinding Friwen, Kabui Passage, Dermaga Pulau

Pada kode script di atas, di lakukan create (penambahan) pada satu paket di package_class_agen melalui bahasa pemrograman python. Di inputkan pc sebagai package_class_agen. Di inputkan paket dengan name_package “Group Tour”, duration_package 3, dan destination_package “Dinding Friwen, Kabui Passage, Dermaga Pulau”. Maka, create (penambahan) paket tersebut akan terkoneksi dan di tampilkan pada MySQL.

➤ Create (Menambahkan) Many Package Class

```
In [5]: #create and many package

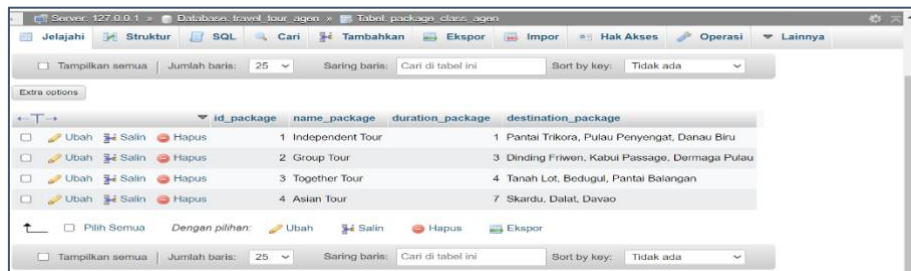
pc = package_class_agen()

data = [
    ("Together Tour", 4, "Tanah Lot, Bedugul, Pantai Balangan"),
    ("Asian Tour", 7, "Skardu, Dalat, Davao")
]

pc.create_many_package(data)

print(pc.read_package())

2 catatan di rekan.
[(4, 'Asian Tour', 7, 'Skardu, Dalat, Davao'), (3, 'Together Tour', 4, 'Tanah Lot, Bedugul, Pantai Balangan'), (2, 'Group Tour', 3, 'Dinding Friwen, Kabul Passage, Dermaga Pulau'), (1, 'Independent Tour', 1, 'Pantai Trikora, Pulau Penyengat, Danau Biru')]
```



	id_package	name_package	duration_package	destination_package
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1	Independent Tour	1	Pantai Trikora, Pulau Penyengat, Danau Biru
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	2	Group Tour	3	Dinding Friwen, Kabul Passage, Dermaga Pulau
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	3	Together Tour	4	Tanah Lot, Bedugul, Pantai Balangan
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	4	Asian Tour	7	Skardu, Dalat, Davao

Pada kode script di atas dilakukan create (penambahan) pada lebih dari satu paket di `package_class_agen` melalui bahasa pemrograman python. Di inputkan `pc` sebagai `package_class_agen`. Di inputkan 2 paket dengan paket 1 terdiri atas `name_package` “Together Tour”, `duration_package` 4, dan `destination_package` “Tanah Lot, Bedugul, Pantai Balangan” dan paket 2 terdiri atas `name_package` “Asian Tour”, `duration_package` 7, dan `destination_package` “Skardu, Dalat, Davao”. Maka, create many (penambahan lebih dari satu) paket-paket tersebut akan terkoneksi dan di tampilkan pada MySQL.

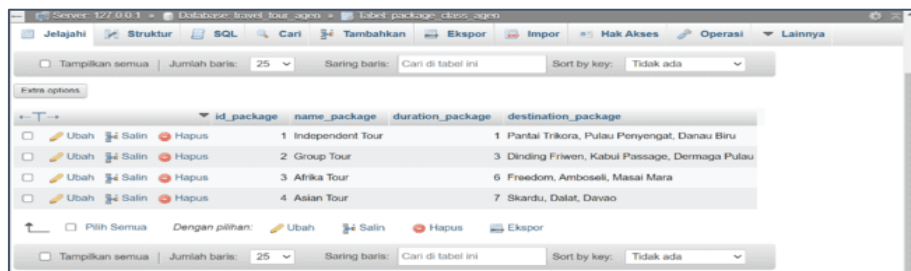
➤ Update (Memperbarui) Package Class

```
In [11]: #update package

pc = package_class_agen()

pc.update_package(3, ("Afrika Tour", 6, "Freedom, Amboseli, Masai Mara"))

1 catatan di perbarui.
```



	id_package	name_package	duration_package	destination_package
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1	Independent Tour	1	Pantai Trikora, Pulau Penyengat, Danau Biru
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	2	Group Tour	3	Dinding Friwen, Kabul Passage, Dermaga Pulau
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	3	Afrika Tour	6	Freedom, Amboseli, Masai Mara
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	4	Asian Tour	7	Skardu, Dalat, Davao

Pada kode script di atas, dilakukan update (pembaruan) pada paket di package_class_agen melalui bahasa pemrograman python. Di inputkan pc sebagai package_class_agen. Di inputkan paket yang akan di update pada data ke 3 dengan name_package “Afrika Tour”, duration_package 6, dan destination_package “Freedom, Amboseli, Masai Mara”. Maka, update (pembaruan) pada paket data ke 3 akan otomatis di update yang terkoneksi dan di tampilkan pada MySQL.

➤ Read (Membaca/Menampilkan) Package Class

```
In [13]: #read_package

pc = package_class_agen()

pc.read_package()

Out[13]: [(4, 'Asian Tour', 7, 'Skardu, Dalat, Davao'),
(3, 'Afrika Tour', 6, 'Freedom, Amboseli, Masai Mara'),
(2, 'Group Tour', 3, 'Dinding Friwen, Kabui Passage, Dermaga Pulau'),
(1, 'Independent Tour', 1, 'Pantai Trikora, Pulau Penyengat, Danau Biru')]
```

Pada kode script di atas dilakukan read (pembacaan/menampilkan) pada paket di package_class_agen melalui bahasa pemrograman python. Di inputkan pc sebagai package_class_agen dan pc.read_package. Maka, akan di tampilkan data-data dari paket di package_class_agen yang terkoneksi pada MySQL.

➤ Delete (Menghapus) Package Class

```
In [14]: #delete_package

pc = package_class_agen()

pc.delete_package(4)

1 catatan di hapus.
```

id_package	name_package	duration_package	destination_package
1	Independent Tour	1	Pantai Trikora, Pulau Penyengat, Danau Biru
2	Group Tour	3	Dinding Friwen, Kabui Passage, Dermaga Pulau
3	Afrika Tour	6	Freedom, Amboseli, Masai Mara

Pada kode script di atas dilakukan delete (penghapusan) pada paket di package_class_agen melalui bahasa pemrograman python. Di inputkan pc

sebagai package_class_agen. Di inputkan akan di lakukan penghapusan pada paket ke 4 di package_class_agen. Maka, akan di tampilkan paket 1-3 yang terkoneksi dan di tampilkan di MySQL.

3.2 CRUD Pada Tour Guide Agen

```
In [5]: #tour guide agen

class tour_guide_agen(Database):

    #create
    def create_guide(self, data):
        sql = "INSERT INTO tour_guide_agen (name_guide, phone_number_guide, years_service_guide, id_package) VALUES (%s, %s, %s, %s)"

        try:
            my_cursor.execute(sql, data)
            my_db.commit()
            print(my_cursor.rowcount, "catatan di rekam.")

        except Exception as e:
            return e

        return my_cursor.lastrowid
```

```
#create many guide
def create_many_guide(self, data):
    sql = "INSERT INTO tour_guide_agen (name_guide, phone_number_guide, years_service_guide, id_package) VALUES (%s, %s, %s, %s)"

    try:
        my_cursor.executemany(sql, data)
        my_db.commit()
        print(my_cursor.rowcount, "catatan di rekam.")

    except Exception as e:
        return e

#read
def read_guide(self, mode = "DESC"):
    sql = "SELECT * FROM tour_guide_agen ORDER BY id_package {}".format(mode)

    try:
        my_cursor.execute(sql)
        result = my_cursor.fetchall()

    except Exception as e:
        return e

    return result
```

```
#update
def update_guide(self, id, data):
    sql = "UPDATE tour_guide_agen SET name_guide = %s, phone_number_guide = %s, years_service_guide = %s, id_package = %s WHERE id = %s"
    val = (data[0], data[1], data[2], data[3], id)

    try:
        my_cursor.execute(sql, val)
        my_db.commit()
        print(my_cursor.rowcount, "catatan di perbarui.")

    except Exception as e:
        return e

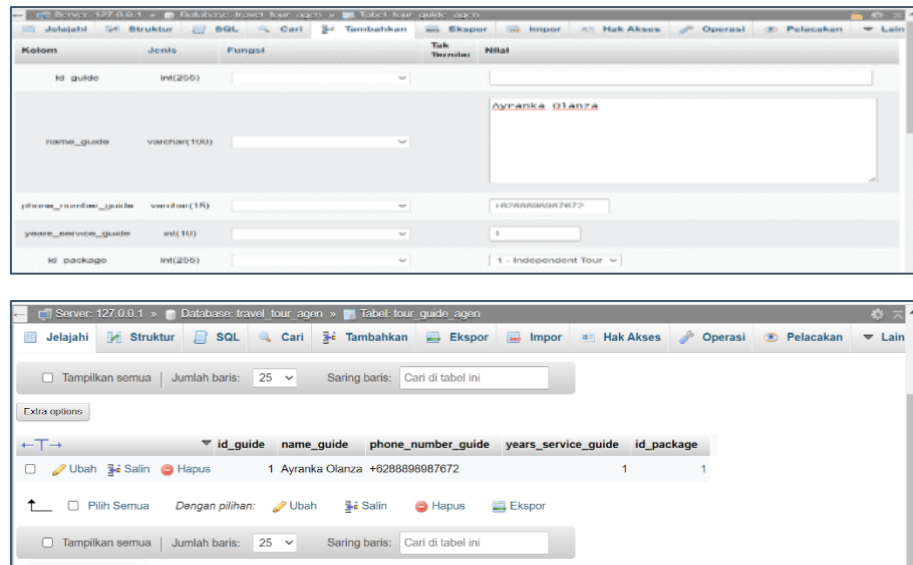
#delete
def delete_guide(self, id):
    sql = "DELETE FROM tour_guide_agen WHERE id_package = {}".format(id)

    try:
        my_cursor.execute(sql)
        my_db.commit()
        print(my_cursor.rowcount, "catatan di hapus.")

    except Exception as e:
        return e
```

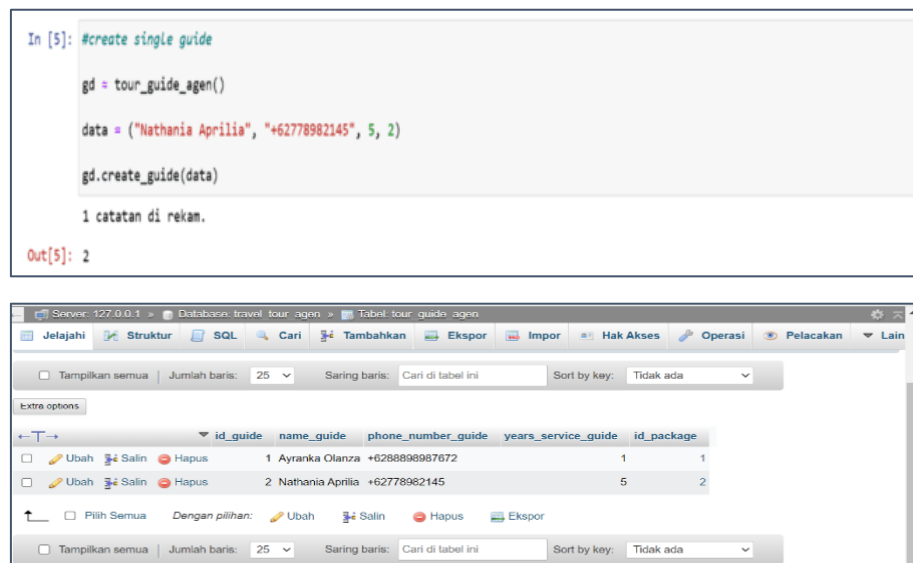
Pada kode script di atas, terdapat sebuah kelas database bernama tour_guide_agen yang akan di lakukan metode CRUD (create, read, update, delete) yang telah terkoneksi dengan MySQL. Create_guide di gunakan untuk menambahkan satu pemandu, create_many_guide di gunakan untuk menambahkan lebih dari satu pemandu, read_guide di gunakan untuk menampilkan pemandu yang ada, update_guide di gunakan untuk mengubah pemandu pada index/nomer tertentu, dan delete_guide di gunakan untuk menghapus pemandu pada index/nomer tertentu.

➤ Menambahkan Package Class Di MySQL



Di lakukan penambahan pemandu pada kelas database tour_guide_agan. Dengan name_guide “Ayranka Olanza”, phone_number_guide “+6288898987672”, years_service_guide 1, dan id_package 1 yang berarti bahwa pemandu ini akan memandu paket “Independent Tour”.

➤ Create (Menambahkan) Tour Guide



Pada kode script di atas, di lakukan create (penambahan) pada satu paket pemandu di tour_guide_agan melalui bahasa pemrograman python. Di inputkan gd sebagai tour_guide_agan. Di inputkan name_guide “Nathania Aprilia”, phone_number_guide “+62778982145”, years_service_guide 5, dan id_package 2. Maka, create (penambahan) pemandu tersebut akan terkoneksi dan di tampilkan pada mysql.

➤ Create (Menambahkan) Many Tour Guide

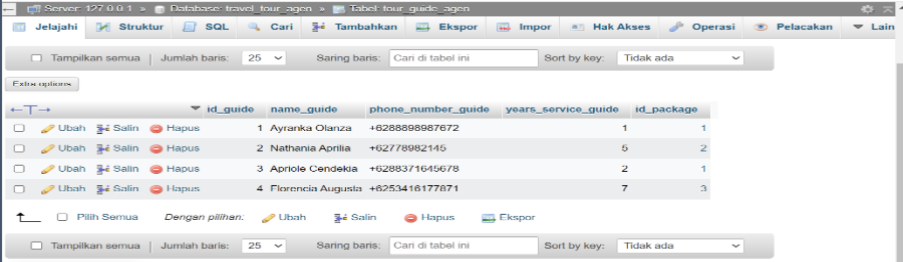
```
In [6]: #create many guide

gd = tour_guide_agen()

data = [
    ("Aprile Cendekia", "+6288371645678", 2, 1),
    ("Florenzia Augusta", "+6253416177871", 7, 3)
]

gd.create_many_guide(data)
print(gd.read_guide())

2 catatan di rekam.
[(4, 'Florenzia Augusta', '+6253416177871', 7, 3), (2, 'Nathania Aprilia', '+62778982145', 5, 2), (1, 'Ayranka Olanza', '+6288898987672', 1, 1), (3, 'Aprile Cendekia', '+6288371645678', 2, 1)]
```



	id_guide	name_guide	phone_number_guide	years_service_guide	id_package
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1	Ayranka Olanza	+6288898987672	1	1
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	2	Nathania Aprilia	+62778982145	5	2
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	3	Aprile Cendekia	+6288371645678	2	1
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	4	Florenzia Augusta	+6253416177871	7	3

Pada kode script di atas, di lakukan create (penambahan) pada lebih dari satu paket pemandu di tour_guide_agen melalui bahasa pemrograman python. Di inputkan 2 pake dengan paket 1 yang terdiri atas name_guide “Aprile Cendekia”, phone_number_guide “+6288371645678”, years_service_guide 2, dan id_package 1 dan paket ke 2 terdiri atas name_guide “Florenzia Augusta”, phone_number_guide “+6253416177871”, dan id_package 3. Maka create many (penambahan lebih dari satu) pemandu tersebut akan terkoneksi dan di tampilkan pada MySQL.

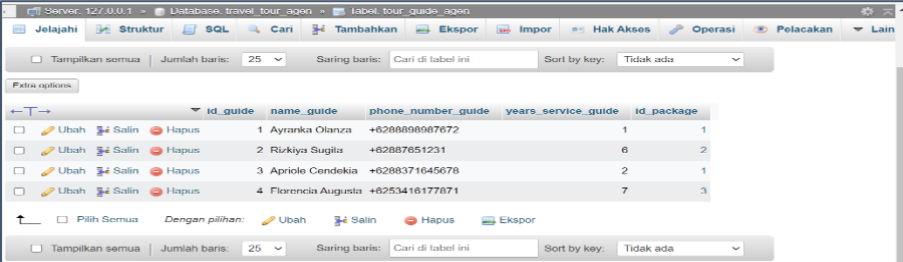
➤ Update (Memperbarui) Tour Guide

```
In [7]: #update guide

gd = tour_guide_agen()

gd.update_guide(2, ("Rizkiya Sugita", "+62887651231", 6, 2))

1 catatan di perbarui.
```



	id_guide	name_guide	phone_number_guide	years_service_guide	id_package
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1	Ayranka Olanza	+6288898987672	1	1
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	2	Rizkiya Sugita	+62887651231	6	2
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	3	Aprile Cendekia	+6288371645678	2	1
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	4	Florenzia Augusta	+6253416177871	7	3

Pada kode script di atas, di lakukan update (pembaruan) pada pemandu di tour_guide_agen melalui bahasa pemrograman python. Di inputkan gd sebagai tour_guide_agen. Di inputkan pemandu yang akan di update dengan ciri pemandu tersebut memiliki id_package 2. Pemandu tersebut akan di perbarui menjadi, name_guide “Rizkiya Sugita”, phone_number_guide “+62887651231”, years_service_guide 6, dan id_package 2. Maka, update (pembaruan) akan terkoneksi dan di tampilkan di MySQL.

➤ Read (Membaca/Menampilkan) Tour Guide

```
In [8]: #read guide

gd = tour_guide_agen()

gd.read_guide()

Out[8]: [(4, 'Florencia Augusta', '+6253416177871', 7, 3),
(2, 'Rizkiya Sugita', '+62887651231', 6, 2),
(1, 'Ayranka Olanza', '+6288898987672', 1, 1),
(3, 'Apriole Cendekia', '+6288371645678', 2, 1)]
```

Pada kode script di atas, di lakukan read (pembacaan/menampilkan) pada pemandu di di tour_guide_agen melalui bahasa pemrograman python. Di inputkan gd sebagai tour_guide_agen dan gd.read_package. Maka, akan di tampilkan data-data dari pemandu yang terkoneksi pada MySQL.

➤ Delete (Menghapus) Tour Guide

```
In [9]: #delete guide

gd = tour_guide_agen()

gd.delete_guide(3)

1 catatan di hapus.
```

id_guide	name_guide	phone_number_guide	years_service_guide	id_package
1	Ayranka Olanza	+6288898987672	1	1
2	Rizkiya Sugita	+62887651231	6	2
3	Apriole Cendekia	+6288371645678	2	1

Pada kode script di atas, di lakukan delete (penghapusan) pada pemandu di tour_guide_agen melalui bahasa pemrograman python. Di inputkan gd sebagai

tour_guide_agen. Di inputkan akan di lakukan penghapusan pemandu yang memiliki id_package 3. Maka, delete (penghapusan) akan terkoneksi dan di tampilkan pada MySQL.

3.3 CRUD Pada Travel Customers Agen

```
In [6]: #travel customers agen

class travel_customers_agen(Database):

    #create
    def create_customers(self, data):
        sql = "INSERT INTO travel_customers_agen (name_customers, phone_number_customers, payment_customers, id_package) VALUES (%s, %s, %s, %s)"
        try:
            my_cursor.execute(sql, data)
            my_db.commit()
            print(my_cursor.rowcount, "catatan di rekam.")
        except Exception as e:
            return e
        return my_cursor.lastrowid
```

```
#create many customers
def create_many_customers(self, data):
    sql = "INSERT INTO travel_customers_agen (name_customers, phone_number_customers, payment_customers, id_package) VALUES (%s, %s, %s, %s)"
    try:
        my_cursor.executemany(sql, data)
        my_db.commit()
        print(my_cursor.rowcount, "catatan di rekam.")
    except Exception as e:
        return e

#read
def read_customers(self, mode = "DESC"):
    sql = "SELECT * FROM travel_customers_agen ORDER BY id_package {}".format(mode)
    try:
        my_cursor.execute(sql)
        result = my_cursor.fetchall()
    except Exception as e:
        return e
    return result
```

```
#update
def update_customers(self, id, data):
    sql = "UPDATE travel_customers_agen SET name_customers = %s, phone_number_customers = %s, payment_customers = %s, id_package = %s"
    val = (data[0], data[1], data[2], data[3])
    try:
        my_cursor.execute(sql, val)
        my_db.commit()
        print(my_cursor.rowcount, "catatan di perbarui.")
    except Exception as e:
        return e

#delete
def delete_customers(self, id):
    sql = "DELETE FROM travel_customers_agen WHERE id_package = {}".format(id)
    try:
        my_cursor.execute(sql)
        my_db.commit()
        print(my_cursor.rowcount, "catatan di hapus.")
    except Exception as e:
        return e
```

Pada kode script ini, terdapat sebuah kelas database bernama travel_customers_agen yang akan di lakukan metode CRUD (create, read, update, delete) yang telah terkoneksi dengan mysql. Create_customers di gunakan untuk menambahkan satu pelanggan, create_many_customers di gunakan untuk menambahkan lebih dari satu pelanggan, read_customers di gunakan untuk menampilkan pelanggan yang ada, update_customers di gunakan untuk mengubah pelanggan pada index/nomer tertentu, dan delete_customers di gunakan untuk menghapus pelanggan pada index/nomer tertentu.

➤ Menambahkan Package Class Di MySQL

id_customers	name_customers	phone_number_customers	payment_customers	id_package
1	Kienan Wardani	+6288098271613	Cash	1

Di lakukan penambahan pelanggan pada kelas database travel_customers_agan. Dengan name_customers “Kienan Wardani”, phone_number_customers “+6288098271613”, payment_customers cash, dan id_package 1 yang berarti pelanggan ini mengikuti perjalanan wisata “Independent Tour”

➤ Create (Menambahkan) Travel Customers

```
In [11]: #create single customers
sc = travel_customers_agan()
data = ("Andini Cantika", "+628890291829", "Cashless", 2)
sc.create_customers(data)
1 catatan di rekam.
Out[11]: 2
```


id_customers	name_customers	phone_number_customers	payment_customers	id_package
1	Kienan Wardani	+6288098271613	Cash	1
2	Andini Cantika	+628890291829	Cashless	2

Pada kode script di atas, di lakukan create (penambahan) pada satu pelanggan di travel_customers_agan melalui bahasa pemrograman python. Di inputkan sc sebagai travel_customers_agan. Di inputkan name_customers “Andini Cantika”, phone_number_customers “+628890291829”, payment_customers

“Cashless”, dan id_package 2. Maka, create (penambahan) pelanggan tersebut akan terkoneksi dan di tampilkan pada MySQL.

➤ Create (Menambahkan) Many Travel Customers

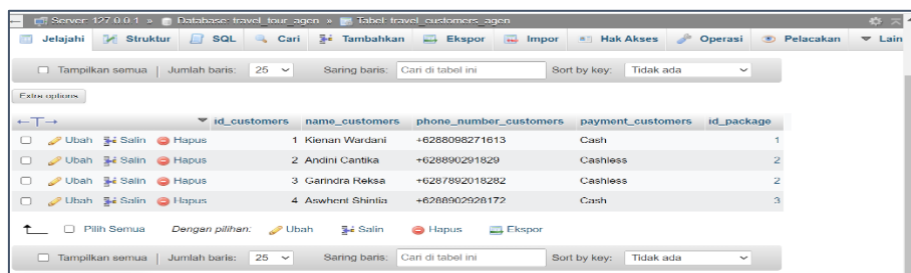
```
In [12]: #create and read many customers
sc = travel_customers_agen()

data = [
    ("Garindra Reksa", "+6287892018282", "Cashless", 2),
    ("Aswhent Shintia", "+6288902928172", "Cash", 3)
]

sc.create_many_customers(data)

print(sc.read_customers())

2 catatan di rekam.
[(4, 'Aswhent Shintia', '+6288902928172', 'Cash', 3), (2, 'Andini Cantika', '+628890291829', 'Cashless', 2), (3, 'Garindra Reksa', '+6287892018282', 'Cashless', 2), (1, 'Kienan Wardani', '+628890271613', 'Cash', 1)]
```



	id_customers	name_customers	phone_number_customers	payment_customers	id_package
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1	Kienan Wardani	+6288068271613	Cash	1
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	2	Andini Cantika	+628890291829	Cashless	2
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	3	Garindra Reksa	+6287892018282	Cashless	2
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	4	Aswhent Shintia	+6288902928172	Cash	3

Pada kode script di atas, di lakukan create (penambahan) pada lebih dari satu pelanggan di travel_customers_agen melalui bahasa pemrograman python. Di inputkan sc sebagai travel_customers_agen. Di inputkan 2 pelanggan dengan pelanggan 1 yang terdiri atas, name_customers “Garindra Reksa”, phone_number_customers “+6287892018282”, payment_customers “Cashless”, dan id_package 2, dan pelanggan 2 yang terdiri atas, name_customers “Aswhent Shintia”, phone_number_customers “+6288901928172”, payment_customers “Cash”, dan id_package 3. Maka, create many (penambahan lebih dari satu) pelanggan tersebut akan terkoneksi dan di tampilkan pada MySQL.

➤ Update (Memperbarui) Travel Customers

```
In [13]: #update customers
sc = travel_customers_agen()

sc.update_customers(1, ("Hanna Laila", "+6288902928172", "Cashless", 2))

1 catatan di perbarui.
```

	id_customers	name_customers	phone_number_customers	payment_customers	id_package
	1	Hanna Laila	+6288902928172	Cashless	2
	2	Andini Cantika	+628890291829	Cashless	2
	3	Garindra Reksa	+6287892018282	Cashless	2
	4	Aswhent Shintia	+6288902028172	Cash	3

Pada kode script di atas, di lakukan update (pembaruan) pada pelanggan di travel_customers_agen melalui bahasa pemrograman python. Di inputkn sc sebagai travel_customers_agen. Di inputkan pelanggan yang akan di perbarui dengan ciri pelanggan tersebut memiliki id_package 1. Pelanggan tersebut akan di perbarui menjadi name_customers “Hanna Laila”, phone_number_customers “+6288902028172”, payment_customers “Cashless”, dan id_package 2.

➤ Read (Membaca/Menampilkan) Travel Customers

```
In [14]: #read customers

sc = travel_customers_agen()

sc.read_customers()

Out[14]: [(4, 'Aswhent Shintia', '+6288902928172', 'Cash', 3),
(1, 'Hanna Laila', '+6288902928172', 'Cashless', 2),
(2, 'Andini Cantika', '+628890291829', 'Cashless', 2),
(3, 'Garindra Reksa', '+6287892018282', 'Cashless', 2)]
```

Pada kode script di atas, di lakukan read (pembacaan/menampilkan) pada pelanggan di travel_customers_agen melalui bahasa pemrograman python. Di inputkan sc sebagai travel_customers_agen dan sc.read_customers(). Maka, akan di tampilkan data-data dari pelanggan yang terkoneksi pada MySQL.

➤ Delete (Menghapus) Travel Customers

```
In [15]: #delete customers

sc = travel_customers_agen()

sc.delete_customers(3)

1 catatan di hapus.
```

Server: 127.0.0.1 - Database: travel_tour_agensi - Tabel: travel_customers_agensi

☐ Tampilkan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini | Sort by key: Tidak ada

Extra options:

	id_customers	name_customers	phone_number_customers	payment_customers	id_package
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1	Hanna Laila	+6288902928172	Cashless	2
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	2	Andini Cantika	+628890291829	Cashless	2
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	3	Ganindra Reksa	+6287892018282	Cashless	2

☐ Pilih Semua | Dengan pilihan: ☐ Ubah ☐ Salin ☐ Hapus ☐ Ekspor

☐ Tampilkan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini | Sort by key: Tidak ada

Pada kode script di atas, di lakukan delete (penghapusan) pada pelanggan di travel_customers_agensi melalui bahasa pemrograman python. Di inputkan sc sebagai travel_customers_agensi. Di inputkan akan di lakukan penghapusan pelanggan yang memiliki id_package 3. Maka, delete (penghapusan) akan terkoneksi dan di tampilkan pada MySQL.