

1. Jelaskan menurut anda apa itu inheritance/kelas turunan dan buatlah contoh codingannya serta capture hasilnya

Inheritance adalah mekanisme di dalam pemrograman berorientasi objek (OOP) dimana sebuah kelas dapat "mewarisi" properti dan metode dari kelas lain. Kelas yang "mewarisi" disebut kelas turunan atau subclass, sementara kelas yang "diedarkan" disebut kelas induk atau superclass.

Dengan inheritance, sebuah kelas dapat menggunakan properti dan metode yang sudah didefinisikan sebelumnya tanpa harus menulis ulang kode yang sama.

Coding :

```
1  class Parent:
2      def __init__(self, name):
3          self.name = name
4      def display(self):
5          print("Name:", self.name)
6
7  class Child(Parent):
8      def __init__(self, name, age):
9          super().__init__(name)
10         self.age = age
11     def display(self):
12         super().display()
13         print("Age:", self.age)
14
15     p = Parent("Agus")
16     p.display()
17
18     c = Child("Sarah", 22)
19     c.display()
```

Hasil :

```
Name: Agus
Name: Sarah
Age: 22
```

2. Jelaskan menurut anda apa itu encapsulation dan buatlah contoh codingannya serta capture hasilnya

Encapsulation adalah mekanisme di dalam pemrograman berorientasi objek (OOP) yang digunakan untuk menyembunyikan rincian implementasi suatu objek dari pemakai objek tersebut. Ini bertujuan untuk meningkatkan tingkat modularitas dan keamanan dari kode.

Encapsulation dapat diterapkan dengan menyimpan properti dan metode suatu objek dalam sebuah kelas, dan hanya menyediakan sebuah interface publik yang memungkinkan pemakai untuk mengakses dan mengubah nilai dari properti tersebut.

Secara sederhana, Encapsulation adalah proses menyembunyikan rincian implementasi dari suatu objek agar tidak dapat diakses atau diubah secara langsung dari luar objek tersebut. Ini meningkatkan tingkat keamanan dan stabilitas dari kode.

Coding:

```
1  class Car:
2      def __init__(self):
3          self.__make = "Toyota"
4          self.__year_model = 2020
5          self.__speed = 0
6
7      def set_make(self, make):
8          self.__make = make
9
10     def set_year_model(self, year):
11         self.__year_model = year
12
13     def set_speed(self, speed):
14         self.__speed = 0
15
16     def get_make(self):
17         return self.__make
18
19     def get_year_model(self):
20         return self.__year_model
21
22     def get_speed(self):
23         return self.__speed
24
25     my_car = Car()
26     print(my_car.get_make())
```

Hasil:

**Toyota**

3. Jelaskan menurut anda apa itu polymorphism dan buatlah contoh codingannya serta capture hasilnya

Polymorphism adalah mekanisme di dalam pemrograman berorientasi objek (OOP) yang memungkinkan sebuah objek untuk memiliki beberapa bentuk atau comportmen yang berbeda. Ini dicapai dengan memberikan kelas yang sama atau kelas turunan dari suatu objek untuk mengimplementasikan metode yang sama dengan cara yang berbeda.

Secara sederhana, polymorphism adalah kemampuan suatu objek untuk memiliki beberapa bentuk atau comportmen yang berbeda. Ini dicapai dengan memberikan kelas yang sama atau kelas turunan dari suatu objek untuk mengimplementasikan metode yang sama dengan cara yang berbeda.

Coding:

```
1 class Animal:
2     def speak(self):
3         raise NotImplementedError("Subclass must implement this method")
4
5 class Dog(Animal):
6     def speak(self):
7         return "Woof!"
8
9 class Cat(Animal):
10    def speak(self):
11        return "Meow!"
12
13 # Create a list of Animal objects
14 animals = [Dog(), Cat()]
15
16 # Call the speak method on each object
17 for animal in animals:
18     print(animal.speak())
```

Hasil:

```
Woof!
Meow!
```

4. Jelaskan menurut anda apa itu function exception handler dan buatlah contoh codingannya serta capture hasilnya

Function exception handler adalah sebuah mekanisme yang digunakan untuk menangani kesalahan (error) yang terjadi saat sebuah program dijalankan. Ketika sebuah kesalahan terjadi, sebuah exception akan dilemparkan (raise) dan jika tidak ditangani dengan benar, program akan berhenti di tempat kesalahan tersebut.

Exception handler menangani exception yang dilemparkan dengan mengeksekusi sebuah blok kode yang ditentukan sebagai "penanganan" (handler) untuk exception tersebut. Penanganan exception dapat digunakan untuk melakukan tugas-tugas seperti menampilkan pesan kesalahan, menyimpan informasi kesalahan ke file log, atau mengambil tindakan lain yang diperlukan.

Coding:

```
1  def divide(x, y):
2      try:
3          result = x / y
4      except ZeroDivisionError:
5          print("Cannot divide by zero!")
6      else:
7          print(f"The result is {result}")
8
9  divide(10, 2)  # The result is 5.0
10 divide(10, 0) # Cannot divide by zero!
```

Hasil:

```
The result is 5.0
Cannot divide by zero!
```

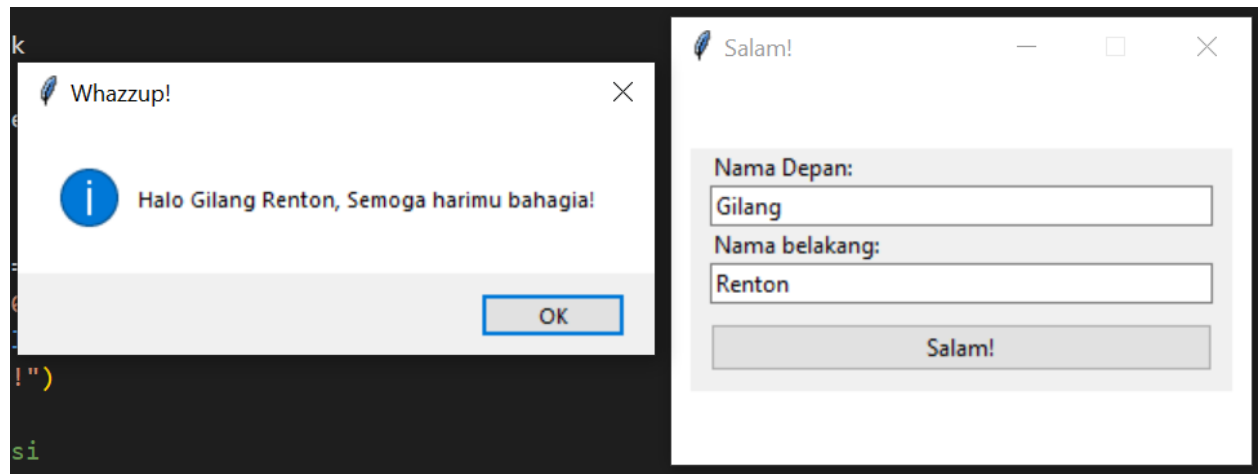
5. Jelaskan menurut anda apa itu GUI dan berikan contoh codingannya serta capture hasilnya GUI, antarmuka pengguna grafis, adalah bentuk antarmuka pengguna yang memungkinkan pengguna berinteraksi dengan perangkat elektronik melalui ikon grafis dan indikator audio seperti notasi utama, bukan UI berbasis teks, label perintah yang diketik atau navigasi teks. GUI diperkenalkan sebagai reaksi terhadap kurva pembelajaran yang dirasakan curam dari CLI (command-line interfaces), yang mengharuskan perintah diketik pada keyboard komputer.

Tindakan dalam GUI biasanya dilakukan melalui manipulasi langsung elemen grafis. Di luar komputer, GUI digunakan di banyak perangkat seluler genggam seperti MP3 player, pemutar media portabel, perangkat game, ponsel cerdas, dan kontrol rumah tangga, kantor, dan industri yang lebih kecil. Istilah GUI cenderung tidak diterapkan pada jenis antarmuka dengan resolusi layar yang lebih rendah, seperti video game (di mana HUD (head-up display) lebih disukai), atau tidak termasuk layar datar seperti tampilan volumetrik karena istilah ini dibatasi pada ruang lingkup layar tampilan 2D yang mampu menggambarkan informasi generik, dalam tradisi penelitian ilmu komputer di Xerox Palo Alto Research Center.

Coding:

```
1  import tkinter as tk
2  from tkinter import ttk
3  from tkinter.messagebox import showinfo
4
5  # Init
6  window = tk.Tk()
7  window.configure(bg="white")
8  window.geometry("300x200")
9  window.resizable(False,False)
10 window.title("Salam!")
11
12 # Variabel dan Fungsi
13 NAMA_DEPAN = tk.StringVar()
14 NAMA_BELAKANG = tk.StringVar()
15
16 def tombol_click():
17     '''fungsi ini akan dipanggil oleh tombol'''
18     pesan = f"Halo {NAMA_DEPAN.get()} {NAMA_BELAKANG.get()}, Semoga harimu bahagia!"
19     showinfo(title="Whazzup!",message=pesan)
20
21 # frame input
22 input_frame = ttk.Frame(window)
23 # penempatan Grid, Pack, Place
24 input_frame.pack(padx=10,pady=10,fill="x",expand=True)
25
26 # komponen-komponen
27 # 1. Label nama depan
28 nama_depan_label = ttk.Label(input_frame,text="Nama Depan:")
29 nama_depan_label.pack(padx=10,fill="x",expand=True)
30
31 # 2. entry nama depan
32 nama_depan_entry = ttk.Entry(input_frame,textvariable=NAMA_DEPAN)
33 nama_depan_entry.pack(padx=10,fill="x",expand=True)
34
35 # 3. Label nama belakang
36 nama_belakang_label = ttk.Label(input_frame,text="Nama belakang:")
37 nama_belakang_label.pack(padx=10,fill="x",expand=True)
38
39 # 4. entry nama belakang
40 nama_belakang_entry = ttk.Entry(input_frame,textvariable=NAMA_BELAKANG)
41 nama_belakang_entry.pack(padx=10,fill="x",expand=True)
42
43 # 5. Tombol
44 tombol_sapa = ttk.Button(input_frame,text="Salam!",command=tombol_click)
45 tombol_sapa.pack(fill='x',expand=True,padx=10,pady=10)
46
47 # Main Loop window
48 window.mainloop()
```

Hasil:

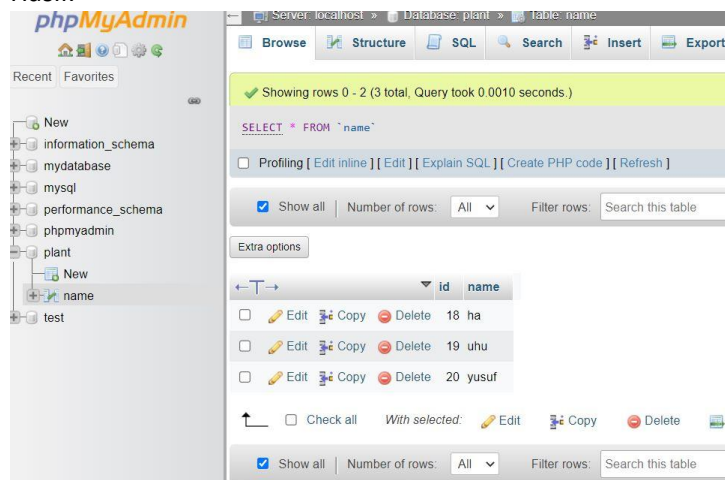


6. Jelaskan menurut anda bagaimana App bisa berkomunikasi dengan database dan buatlah contoh codingnya serta capture hasilnya
- Aplikasi dapat berkomunikasi dengan database dengan menggunakan protokol standar seperti SQL (Structured Query Language) melalui koneksi API (Application Programming Interface). API akan mengizinkan aplikasi untuk mengirimkan perintah SQL ke database, seperti untuk mengambil atau menambahkan data, dan menerima hasilnya dalam bentuk data yang dapat diolah oleh aplikasi. Beberapa contoh teknologi yang digunakan untuk koneksi API antara aplikasi dan database adalah ORM (Object-Relational Mapping), JDBC (Java Database Connectivity) dan ODBC (Open Database Connectivity).

Coding:

```
1 self.db = mysql.connector.connect(  
2     host="localhost",  
3     user="root",  
4     password="",  
5     database="plant"  
6 )
```

Hasil:



7. Buatlah Aplikasi sederhana yang terkoneksi dengan database push ke github source codenya dan capture hasilnya

Hasil:

