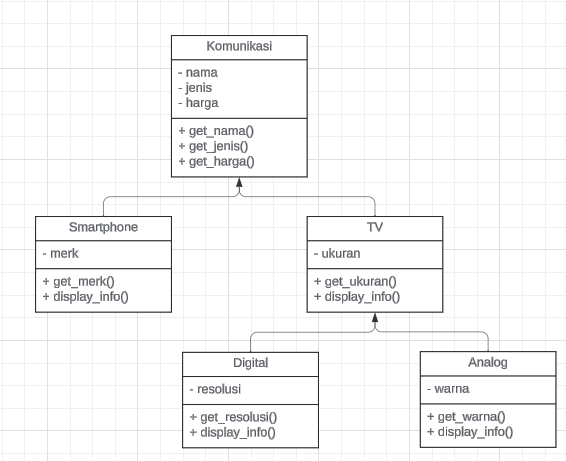
Buat class diagram dengan tema yang telah ditentukan (Teknologi).

* Class diagram memiliki induk class dan anak sejumlah 2 class. Salah satu anak class memiliki 1 class anak.
* Class induk memiliki atribut dan method yang dapat diturunkan kepada anak classnya.
* Anak class memiliki atribut dan method yang bukan hasil turunan dari induknya.
* Berikan perbedaan akses modifier pada atribut dan method yang digunakan.
* Buat kode programnya yang sesuai dengan class diagram yang anda bangun.
* Terapkan konsep inheritance, enkapsulasi, dan polimorfisme baik overloading maupun overriding.
* Gunakan operator aritmatika dan logika.
* Gunakan perulangan dan percabangan

Hasilnya dituliskan pada file Word (TAVII\_NPM\_Nama):

* Class Diagram
* Kode program >> font courier new ukuran 12 (bukan screenshot)
* Screenshot hasil running program
* Kode program up di Github yang sudah diinfokan sebelumnya

CLASS DIAGRAM:



# Kode Program

class Komunikasi:

def \_\_init\_\_(self, nama, jenis, harga):

self.\_nama = nama

self.\_jenis = jenis

self.\_harga = harga

def get\_nama(self):

return self.\_nama

def get\_jenis(self):

return self.\_jenis

def get\_harga(self):

return self.\_harga

class Smartphone(Komunikasi):

def \_\_init\_\_(self, nama, jenis, harga, merk):

super().\_\_init\_\_(nama, jenis, harga)

self.\_merk = merk

def get\_merk(self):

return self.\_merk

def display\_info(self):

print("Nama: ", self.get\_nama())

print("Jenis: ", self.get\_jenis())

print("Harga: ", self.get\_harga())

print("Merk: ", self.get\_merk())

class TV(Komunikasi):

def \_\_init\_\_(self, nama, jenis, harga, ukuran):

super().\_\_init\_\_(nama, jenis, harga)

self.\_ukuran = ukuran

def get\_ukuran(self):

return self.\_ukuran

def display\_info(self):

print("Nama: ", self.get\_nama())

print("Jenis: ", self.get\_jenis())

print("Harga: ", self.get\_harga())

print("Ukuran: ", self.get\_ukuran())

class Digital(TV):

def \_\_init\_\_(self, nama, jenis, harga, ukuran, resolusi):

super().\_\_init\_\_(nama, jenis, harga, ukuran)

self.\_resolusi = resolusi

def get\_resolusi(self):

return self.\_resolusi

def display\_info(self): # Overriding

super().display\_info()

print("Resolusi: ", self.get\_resolusi())

class Analog(TV):

def \_\_init\_\_(self, nama, jenis, harga, ukuran, warna):

super().\_\_init\_\_(nama, jenis, harga, ukuran)

self.\_warna = warna

def get\_warna(self):

return self.\_warna

def display\_info(self): # Overriding

super().display\_info()

print("Warna: ", self.get\_warna())

print(20\*"=")

smartphone1 = Smartphone("iPhone 12", "Smartphone", 15000000, "Apple")

smartphone1.display\_info()

print(20\*"=")

digital\_tv1 = Digital("Samsung OLED", "TV", 20000000, 55, "4K")

digital\_tv2 = Digital("LG OLED", "TV", 25000000, 65, "4K")

analog\_tv1 = Analog("Sony CRT", "TV", 5000000, 32, "Hitam")

daftar\_tv\_digital = [digital\_tv1, digital\_tv2]

print("Daftar TV Digital: ")

print(20\*"=")

# Perulangan

for tv in daftar\_tv\_digital:

tv.display\_info()

print(20\*"=")

print("TV Analog: ")

print(20\*"=")

analog\_tv1.display\_info()

print(20\*"=")

# Operator aritmatika

harga\_digitaltv1 = digital\_tv1.get\_harga()

harga\_digitaltv2 = digital\_tv2.get\_harga()

median\_harga\_digitaltv = (harga\_digitaltv1 + harga\_digitaltv2) / 2

# Percabangan

if harga\_digitaltv1 > median\_harga\_digitaltv:

print("Harga TV Digital 1 lebih mahal dari median harga TV Digital")

else:

print("Harga TV Digital 1 lebih murah dari median harga TV Digital")

if harga\_digitaltv2 > median\_harga\_digitaltv:

print("Harga TV Digital 2 lebih mahal dari median harga TV Digital")

else:

print("Harga TV Digital 2 lebih murah dari median harga TV Digital")

HASIL KODE PROGRAM (OUTPUT):

