NAMA: Alif Arya Kusuma

NIM: 5220411242

FAKULTAS: Sains dan Teknologi

PRODI: Informatika

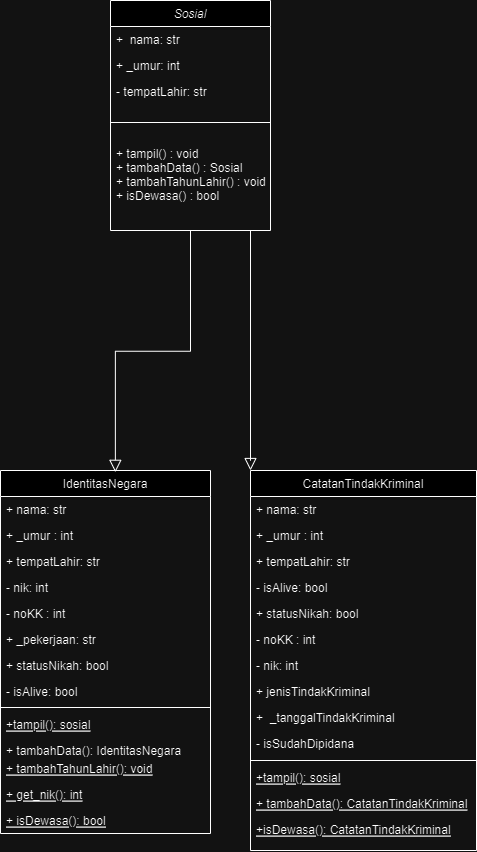
KELAS: Pemrograman Berorientasi Objek Praktik (VII)

RUANGAN: L.K 2.3

TEMA: Sosial

Sistem kependudukan dan Riwayat kriminal

1. Class Diagram



2. Kode program

data = []

class Sosial:

    def \_\_init\_\_(self, nama, umur, tempatLahir):

        self.nama = nama

        self.\_umur = umur

        self.tempatLahir = tempatLahir

    def tampil(self):

        print(

            f"Nama: {self.nama}\nUsia: {self.\_umur}\nTempat Lahir: {self.tempatLahir}"

        )

    def tambahData(self):

        print("Tambah Data")

        nama = input("Nama: ")

        umur = int(input("Usia: "))

        tempatLahir = input("Tempat Lahir: ")

        return Sosial(nama, umur, tempatLahir)  # Mengembalikan objek Sosial yang baru

    def tambahTahunLahir(self, tahun):

        self.\_umur += tahun

    def isDewasa(self):

        return self.\_umur >= 18

class IdentitasNegara(Sosial):

    def \_\_init\_\_(

        self, nama, umur, tempatLahir, nik, noKK, pekerjaan, statusNikah, isAlive

    ):

        super().\_\_init\_\_(nama, umur, tempatLahir)

        self.\_\_nik = nik

        self.\_\_noKK = noKK

        self.\_pekerjaan = pekerjaan

        self.statusNikah = statusNikah

        self.isAlive = isAlive

    def tampil(self):

        super().tampil()

        print(

            f"NIK: {self.\_\_nik}\nNomor KK: {self.\_\_noKK}\nPekerjaan: {self.\_pekerjaan}"

        )

        print(f"Status Nikah: {self.statusNikah}\nApakah Hidup: {self.isAlive}")

    def tambahData(self):

        super().tambahData()

        nik = int(input("masukan nik: "))

        noKK = int(input("masukan no kk: "))

        pekerjaan = input("masukan data pekerjaan: ")

        statusNikah = input("masukan status nikah (True or False)")

        isAlive = input("masukan status hidup (True or False)")

        return IdentitasNegara(

            nama=self.nama,

            umur=self.\_umur,

            tempatLahir=self.tempatLahir,

            nik=nik,

            noKK=noKK,

            pekerjaan=pekerjaan,

            statusNikah=statusNikah,

            isAlive=isAlive,

        )

    def tambahTahunLahir(self, tahun):

        super().tambahTahunLahir(tahun)

    # Menambahkan method untuk mengakses variabel private \_\_nik

    def get\_nik(self):

        return self.\_\_nik

class CatatanTindakKriminal(IdentitasNegara):

    def \_\_init\_\_(

        self,

        nama,

        umur,

        tempatLahir,

        nik,

        noKK,

        pekerjaan,

        statusNikah,

        isAlive,

        jenisTindakKriminal,

        tanggalTindakKriminal,

        isSudahDipidana,

    ):

        super().\_\_init\_\_(

            nama, umur, tempatLahir, nik, noKK, pekerjaan, statusNikah, isAlive

        )

        self.jenisTindakKriminal = jenisTindakKriminal

        self.\_tanggalTindakKriminal = tanggalTindakKriminal

        self.\_\_isSudahDipidana = isSudahDipidana

    def tampil(self):

        super().tampil()

        print(f"Jenis Tindak Kriminal: {self.jenisTindakKriminal}")

        print(f"Tanggal Tindak Kriminal: {self.\_tanggalTindakKriminal}")

        print(f"Sudah Dipidana: {self.\_\_isSudahDipidana}")

    def tambahData(self):

        super().tambahData()

        jenisTindakKriminal = input("masukan jenis tindakan: ")

        tanggalTindakKriminal = input("masukan tanggal tindakan: ")

        isSudahDipidana = input("apakah sudah dipidana? (y/n): ")

        return CatatanTindakKriminal(

            nama=self.nama,

            umur=self.\_umur,

            tempatLahir=self.tempatLahir,

            nik=self.get\_nik(),

            noKK=self.\_\_noKK,

            pekerjaan=self.\_pekerjaan,

            statusNikah=self.statusNikah,

            isAlive=self.isAlive,

            jenisTindakKriminal=jenisTindakKriminal,

            tanggalTindakKriminal=tanggalTindakKriminal,

            isSudahDipidana=isSudahDipidana,

        )

    def isDewasa(self):

        return super().isDewasa() and self.\_pekerjaan != "Pelajar"

# Contoh operator logika:

sosial = Sosial("albert", 20, "Jogja")

ck = CatatanTindakKriminal("meguhunter", 17, "wonosobo", 123456789, 987654321, "mahasiswa", False, True, "kasus pelecehan terhadap anak ayam", "20", False)

print(sosial.isDewasa())  # Memeriksa apakah orang dewasa menggunakan metode dari kelas Sosial

print()

print(ck.isDewasa())  # Memeriksa apakah orang dewasa menggunakan metode dari kelas IdentitasNegara (override)

ck.tampil()

print("\n")

print("\*" \* 40)

print("\n")

# Contoh operator aritmatika:

sosial = Sosial("albert", 20, "Jogja")

identitas\_negara = IdentitasNegara(

    "John", 25, "Jakarta", 123456789, 987654321, "PNS", True, True

)

sosial.tampil()

print()

print("\n")

print("\*" \* 40)

print("\n")

sosial.tambahTahunLahir(5)  # Menambah umur menggunakan metode dari kelas Sosial

sosial.tampil()

print()

identitas\_negara.tambahTahunLahir(

    3

)  # Menambah umur menggunakan metode dari kelas IdentitasNegara (override)

identitas\_negara.tampil()

print("\n")

print("\*" \* 40)

print("\n")

# Contoh penggunaan inheritance:

sosial = Sosial("albert", 20, "Jogja")

identitas\_negara = IdentitasNegara(

    "John", 25, "Jakarta", 123456789, 987654321, "PNS", True, True

)

catatan\_kriminal = CatatanTindakKriminal(

    "Alice",

    30,

    "Surabaya",

    987654321,

    123456789,

    "Swindler",

    False,

    True,

    "Penipuan",

    "2023-01-02",

    False,

)

sosial.tampil()

print()

identitas\_negara.tampil()

print()

catatan\_kriminal.tampil()

print("\n")

print("\*" \* 40)

print("\n")

# Contoh penggunaan encapsulation:

identitas\_negara = IdentitasNegara(

    "John", 25, "Jakarta", 123456789, 987654321, "PNS", True, True

)

print(identitas\_negara.\_umur)  # Variabel dengan underscore masih bisa diakses dari luar

# Mengakses variabel private \_\_nik melalui method get\_nik

print(identitas\_negara.get\_nik())

print("\n")

print("\*" \* 40)

print("\n")

# Contoh overriding (polimorfisme pada metode):

sosial = Sosial("albert", 20, "Jogja")

identitas\_negara = IdentitasNegara(

    "John", 25, "Jakarta", 123456789, 987654321, "PNS", True, True

)

sosial.tampil()  # Memanggil metode dari kelas Sosial

print()

identitas\_negara.tampil()  # Memanggil metode dari kelas IdentitasNegara

print("\n")

print("\*" \* 40)

print("\n")

# Contoh overloading (polimorfisme pada metode):

sosial = Sosial("ali",22,'bogor')

identitas\_negara = IdentitasNegara("John", 25, "Jakarta", 123456789, 987654321, "PNS", True, True)

sosial.tampil()  # Memanggil metode tanpa parameter dari kelas Sosial

print()

identitas\_negara.tampil()  # Memanggil metode tanpa parameter dari kelas IdentitasNegara

print()

sosial.tampil(

)  # Memanggil metode dengan parameter dari kelas Sosial

print()

identitas\_negara.tampil(

)  # Memanggil metode dengan parameter dari kelas IdentitasNegara

print("\n")

print("\*" \* 40)

print("\n")

def looping():

    # Looping untuk menambah data

    while True:

        print("\n1. Tambah Data Sosial")

        print("2. Tambah Data Identitas Negara")

        print("3. Tambah Data Catatan Tindak Kriminal")

        print("4. Tampilkan Semua Data")

        print("5. Keluar")

        pilihan = input("Pilih menu (1-5): ")

        if pilihan == "1":

            data.append(Sosial("", 0, "").tambahData())

        elif pilihan == "2":

            data.append(IdentitasNegara("", 0, "", 0, 0, "", False, False).tambahData())

        elif pilihan == "3":

            data.append(

                CatatanTindakKriminal("", 0, "", 0, 0, "", False, False, "", "", False).tambahData()

            )

        elif pilihan == "4":

            for item in data:

                item.tampil()

                print("\n" + "\*" \* 40 + "\n")

        elif pilihan == "5":

            print("Terima kasih! Keluar dari program.")

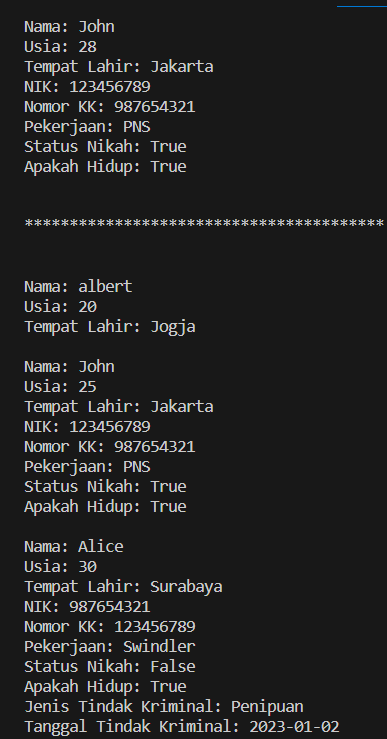
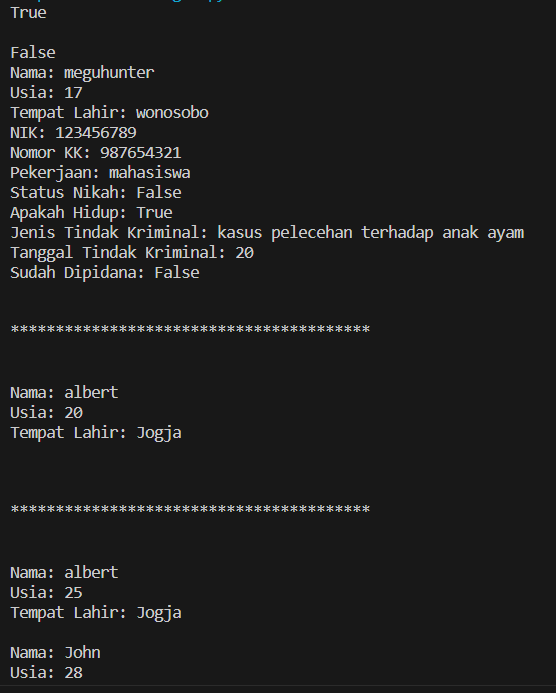
            break

        else:

            print("Pilihan tidak valid. Silakan pilih kembali.")

            #opsional sajah saya sudah berada diambang batas kewarasan sayah

3. Screenshot Hasil Running Program





A computer screen shot of a black screen

Description automatically generated