**TUGAS AKHIR TEMA KESEHATAN**

MATA KULIAH PEMROGRAMAN BERORIENTASI OBJEK

Logo

Description automatically generated

Disusun Oleh :

1. Kholik Al Lutfi (5220411172)

Dosen Pengampu :

Rodhiyah Mardhiyyah, S.Kom., M.Kom.

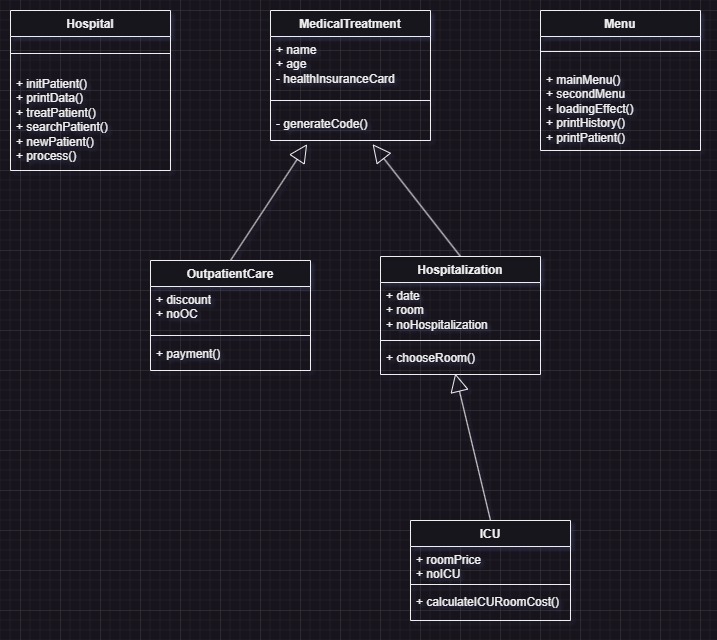
**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**FAKULTAS SAINS & TEKNOLOGI**

**PROGRAM STUDI INFORMATIKA**

**2023**

1. Class Diagram



Ada lima class yaitu class Hospital yang memiliki peran untuk menampung semua proses program dalam function proce, ini untuk mempermudah menjalankan program dalam main function. Lalu ada class Menu yang memiliki peran untuk menampung teks-teks menu pada program dan menampilkan output. Untuk class MedicalTreatment, OutpatientCare, Hospitalization, dan ICU merupakan kelas yang dibuat berdasarkan ketentuan dari studi kasus yang diberikan.

1. Kode Program

import random

import time

import os

from prettytable import PrettyTable

class Menu:

    def mainMenu(self):

        print(

        """

    Selamat Datang Di Rumah Sakit Sejahtera

    1. Daftar Pasien Baru

    2. Sudah Pernah daftar

    3. Riwayat

    4. Daftar Patient

        """

    )

    def secondMenu(self):

        print(

        """

    Jenis Pelayan apa yang akan dipilih? :

    1. Rawat Jalan

    2. Rawat Inap

    3. Unit Perawatan Intensif

        """

    )

    def loadingEffect(self, text,text2):

        print(text, end='', flush=True)

        for \_ in range(3):

            time.sleep(1)

            print('.', end='', flush=True)

        print(f"\n{text2}")

    def printHistory(self, dict):

        for treatmentType, treatments in dict.items():

            if treatments:

                table = PrettyTable()

                treatment = treatments[0]  # Ambil objek pertama untuk mendapatkan field names, cara mengambil key : treatment.\_\_dict\_\_.keys()

                table.field\_names = treatment.field\_names

                for treatment in treatments:

                    if type(treatment) is OutpatientCare:

                        table.add\_row([treatment.name,treatment.age,treatment.\_MedicalTreatment\_\_healthInsuranceCard,treatment.discount,treatment.noOC])

                    elif type(treatment) is Hospitalization :

                        table.add\_row([treatment.name,treatment.age,treatment.\_MedicalTreatment\_\_healthInsuranceCard,treatment.date,treatment.room,treatment.noHospitalization])

                    elif type(treatment) is ICU:

                        table.add\_row([treatment.name,treatment.age,treatment.\_MedicalTreatment\_\_healthInsuranceCard,treatment.date,treatment.roomPrice,treatment.noICU])

                table.title = f"Riwayat {treatmentType}"

                print("\n")

                print(table)

            else:

                print(f"\nRiwayat {treatmentType} kosong.")

    def printPatient(self,dict):

        table = PrettyTable()

        table.field\_names = ["No","Nama","Umur","Jenis Kartu Asuransi"]

        no = 0

        for patientType, patiens in dict.items():

            no += 1

            table.add\_row([no,patiens.name,patiens.age,patiens.\_MedicalTreatment\_\_healthInsuranceCard])

        print(table)

class MedicalTreatment:

    def \_\_init\_\_(self,name,age,healthInsuranceCard):

        self.name = name

        self.age = age

        self.\_\_healthInsuranceCard = healthInsuranceCard

    def \_\_generateCode(self,jenisPerawatan):

        randomNumber = random.randint(100,999)

        treatmentNumber = f"{jenisPerawatan}{randomNumber}"

        return treatmentNumber

# kelas rawat jalan

class OutpatientCare(MedicalTreatment):

    field\_names = ["Nama", "Umur", "Jenis Kartu Asuransi", "Discount","No. Rawat Jalan"]

    def \_\_init\_\_(self, name, age, healthInsuranceCard):

        super().\_\_init\_\_(name, age, healthInsuranceCard)

        self.discount = None

        self.noOC = None

    def \_\_generateCode(self, jenisPerawatan):

        return super().\_\_generateCode(jenisPerawatan)

    def payment(self,cardType):

        if cardType == "basic":

            print("Get discount 20%")

            self.discount =  "discount 20%"

        elif cardType == "standart":

            print("Get discount 40%")

            self.discount = "discount 40%"

        elif cardType == "premium":

            print("Get discount 60%")

            self.discount = "discount 60%"

# kelas rawat inap

class Hospitalization(MedicalTreatment):

    field\_names = ["Nama","Umur","Jenis Kartu Asuransi","Tanggal Mulai Rawat","Ruangan","No. Rawat Inap"]

    def \_\_init\_\_(self, name, age, healthInsuranceCard, date):

        super().\_\_init\_\_(name, age, healthInsuranceCard)

        self.date = date

        self.room = None

        self.noHospitalization = None

    def \_\_generateCode(self, jenisPerawatan):

        return super().\_\_generateCode(jenisPerawatan)

    def chooseRoom(self):

        print(

        """

        Pilih Ruangan :

        1. Dahlia

        2. Kenanga

        3. Anggrek

        """

        )

        roomChoice = int(input("Masukkan pilihan ruangan: "))

        if roomChoice == 1:

            self.room = "Dahlia"

        elif roomChoice == 2:

            self.room = "Kenanga"

        elif roomChoice == 3:

            self.room = "Anggrek"

# kelas unit perawatan intensif (Intensif Care Unit)

class ICU(Hospitalization):

    field\_names = ["Nama","Umur","Jenis Kartu Asuransi","Tanggal Mulai Rawat","Biaya Kamar","No. ICU"]

    def \_\_init\_\_(self, name, age, healthInsuranceCard, date):

        super().\_\_init\_\_(name, age, healthInsuranceCard, date)

        self.roomPrice = None

        self.noICU = None

    def \_\_generateCode(self, jenisPerawatan):

        return super().\_\_generateCode(jenisPerawatan)

    def calculateICURoomCost(self):

        roomPrice = 2000000

        dayStayed = int(input("Perkiraan lama tinggal (hari) : "))

        totalPrice = roomPrice \* dayStayed

        print(f"Biaya kamar : {totalPrice} (belum termasuk biaya perawatan dan obat)")

        self.roomPrice = totalPrice

class Hospital:

    def initPatient(self):

        patient1 = MedicalTreatment(

            "budi",25,"premium"

        )

        patient2 = MedicalTreatment(

            "andi",30,"basic"

        )

        patient3 = MedicalTreatment(

            "Yuli",17,"standart"

        )

        dictPatient = {

            patient1.name: patient1,

            patient2.name: patient2,

            patient3.name: patient3

        }

        return dictPatient

    def printData(self,object):

        if type(object) is OutpatientCare:

            print(

            f"""

    Pasien Berhasil Diinputkan!

    Identitas Patient :

        Nama : {object.name}

        Umur : {object.age}

        Jenis Kartu Asuransi : {object.\_MedicalTreatment\_\_healthInsuranceCard}

        No. Rawat Jalan : {object.noOC}

    """

        )

        elif type(object) is Hospitalization:

            print(

            f"""

    Pasien Berhasil Diinputkan!

    Identitas Patient :

        Nama : {object.name}

        Umur : {object.age}

        Jenis Kartu Asuransi : {object.\_MedicalTreatment\_\_healthInsuranceCard}

        Tanggal Mulai Rawat : {object.date}

        Ruangan : {object.room}

        No. Rawat Inap : {object.noHospitalization}

    """

        )

        elif type(object) is ICU:

            print(

            f"""

    Pasien Berhasil Diinputkan!

    Identitas Patient :

        Nama : {object.name}

        Umur : {object.age}

        Jenis Kartu Asuransi : {object.\_MedicalTreatment\_\_healthInsuranceCard}

        Tanggal Mulai Rawat : {object.date}

        Biaya Kamar : Rp {object.roomPrice}

        No. ICU : {object.noICU}

    """

        )

    def treatPatient(self, treatmentClass, dict, \*args):

        treatment = treatmentClass(\*args)

        treatmentCode = treatment.\_MedicalTreatment\_\_generateCode(treatmentClass.\_\_name\_\_.upper()[:3])

        # Penyesuaian atribut berdasarkan jenis perawatan

        if type(treatment) is OutpatientCare:

            treatment.noOC = treatmentCode

            treatment.payment(treatment.\_MedicalTreatment\_\_healthInsuranceCard)

            dict[treatmentClass.\_\_name\_\_].append(treatment)

            self.printData(treatment)

        elif type(treatment) is Hospitalization:

            treatment.noHospitalization = treatmentCode

            treatment.chooseRoom()

            dict[treatmentClass.\_\_name\_\_].append(treatment)

            self.printData(treatment)

        elif type(treatment) is ICU:

            treatment.noICU = treatmentCode

            treatment.calculateICURoomCost()

            dict[treatmentClass.\_\_name\_\_].append(treatment)

            self.printData(treatment)

    def searchPatient(self, dict, dictRiwayat):

        patientName = input('Nama pasien : ')

        patientFound = False  # Inisialisasi variabel untuk melacak apakah pasien ditemukan atau tidak

        Menu().loadingEffect("Mencari pasien berdasarkan nama","pasien ditemukan")

        for patient, patientObject in dict.items():

            if patientName == patientObject.name:

                patientFound = True  # Set variabel ke True ketika pasien ditemukan

                Menu().secondMenu()

                choice2 = int(input("Masukkan pilihan anda : "))

                if choice2 == 1:

                    self.treatPatient(OutpatientCare, dictRiwayat, patientObject.name, patientObject.age, patientObject.\_MedicalTreatment\_\_healthInsuranceCard)

                elif choice2 == 2:

                    date = input("Inputkan tanggal : ")

                    self.treatPatient(Hospitalization, dictRiwayat, patientObject.name, patientObject.age, patientObject.\_MedicalTreatment\_\_healthInsuranceCard, date)

                elif choice2 == 3:

                    date = input("Inputkan tanggal : ")

                    self.treatPatient(ICU, dictRiwayat, patientObject.name, patientObject.age, patientObject.\_MedicalTreatment\_\_healthInsuranceCard,date)

        if not patientFound:

            print("Pasien Tidak Ditemukan Silahkan Registrasi terlebih dahulu")

    def newPatient(self,dictRiwayat,dictPatientData):

        while True:

            patientName = input('Nama pasien baru : ')

            patientAge = int(input('Usia pasien baru : '))

            healthInsurance = input('Jenis kartu asuransi anda (basic, standart, premium) : ')

            # make object patient

            patient = MedicalTreatment(patientName,patientAge,healthInsurance)

            Menu().secondMenu()

            choice2 = int(input("Masukkan pilihan anda : "))

            if choice2 == 1:

                self.treatPatient(OutpatientCare,dictRiwayat,patient.name,patient.age,patient.\_MedicalTreatment\_\_healthInsuranceCard)

                dictPatientData[patientName] = patient

                choice3 = input("Ingin menginput lagi? (y/n) : ").upper()

                if choice3 == "Y":

                    continue

                elif choice3 == "N":

                    print("Terima Kasih")

                    return

            elif choice2 == 2:

                date = input("Inputkan tanggal : ")

                self.treatPatient(Hospitalization,dictRiwayat,patient.name,patient.age,patient.\_MedicalTreatment\_\_healthInsuranceCard,date)

                dictPatientData[patientName] = patient

                choice3 = input("Ingin menginput lagi? (y/n) : ").upper()

                if choice3 == "Y":

                    continue

                elif choice3 == "N":

                    print("Terima Kasih")

                    return

            elif choice2 == 3:

                date = input("Inputkan tanggal : ")

                self.treatPatient(ICU,dictRiwayat,patient.name,patient.age,patient.\_MedicalTreatment\_\_healthInsuranceCard,date)

                dictPatientData[patientName] = patient

                choice3 = input("Ingin menginput lagi? (y/n) : ").upper()

                if choice3 == "Y":

                    continue

                elif choice3 == "N":

                    print("Terima Kasih")

                    return

    def process(self):

        # init patient

        dataPatient = self.initPatient()

        # init riwayat

        historyMedicalTreatment = {

        "OutpatientCare" : [],

        "Hospitalization": [],

        "ICU": []

        }

        while True:

            Menu().mainMenu()

            choice = int(input("Masukkan pilihan anda : "))

            if choice == 1:

                self.newPatient(historyMedicalTreatment,dataPatient)

            elif choice == 2:

                self.searchPatient(dataPatient, historyMedicalTreatment)

            elif choice == 3:

                Menu().printHistory(historyMedicalTreatment)

            elif choice == 4:

                Menu().printPatient(dataPatient)

            else:

                print("Inputan Salah!")

                Menu().loadingEffect("Menghapus command line", "command line selesai dihapus")

                os.system("cls")

                continue

if \_\_name\_\_ == "\_\_main\_\_":

    Hospital().process()

1. Screenshot Hasil Running Program

