LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK

PERTEMUAN 13

*QUIZ*



Disusun oleh:

Nama : Wahhab Awaludin

NIM : 23/522216/SV/23632

Kelas : A1

Dosen Pengampu : Margareta Hardiyanti, S.Kom., M.Eng.

PROGRAM STUDI D-IV TEKNOLOGI REKAYASA PERANGKAT LUNAK

DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA

SEKOLAH VOKASI

UNIVERSITAS GADJAH MADA

YOGYAKARTA

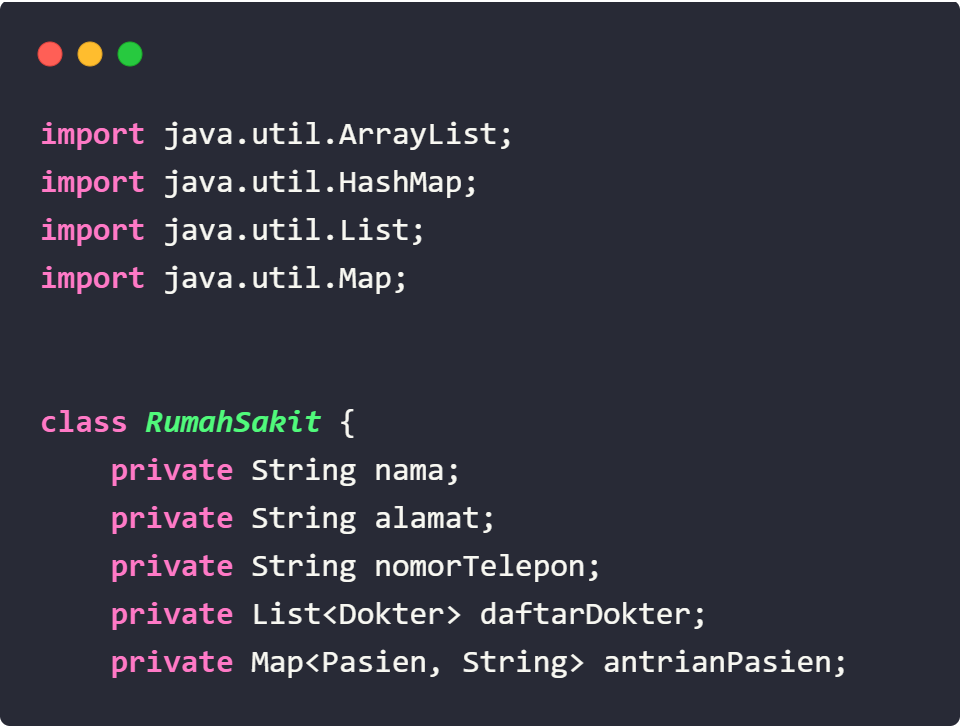
2024

**HASIL DAN PEMBAHASAN**

**SOAL 1**

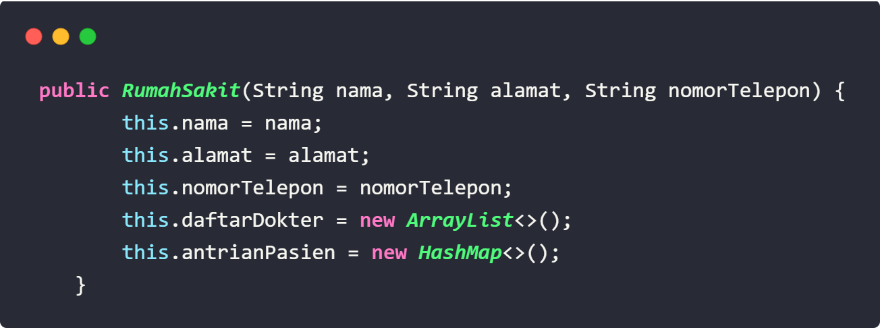
# *Class* RumahSakit

*Class* rumah sakit merupakat pusat perawatan, pemeriksaan para pasien. Rumah sakit ini dapat melakukan pemeriksaan pasien dengan dokter tentetu. Sebelum diperiksa, pasien harus melakukan pendaftaran terlebih dahulu. Selain itu, dokter juga harus tersedia dan terdaftar di rumah sakit tersebut. *Class* ini menampung 5 atribut, di antaranya : nama, alamat, nomor telpon, daftar dokter, dan antrian pasien.



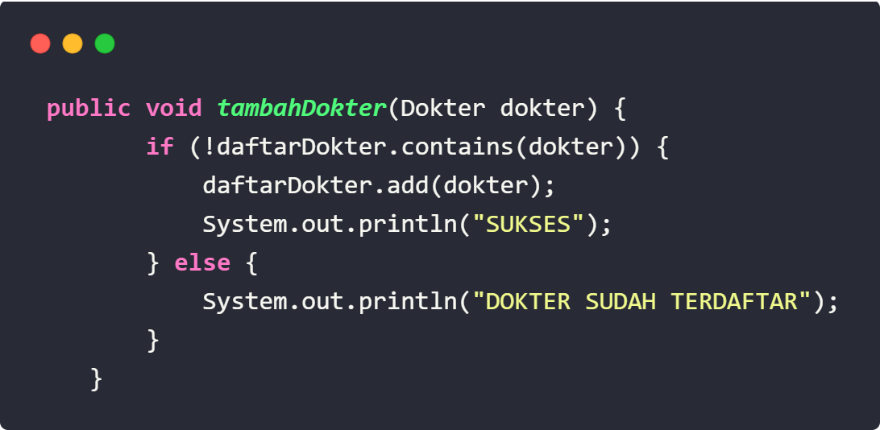
1. Konstruktor

Konstruktor RumahSakit menerima tiga parameter (nama, alamat, dan nomorTelepon) dan menginisialisasi properti-properti tersebut. Selain itu, konstruktor juga menginisialisasi daftarDokter sebagai ArrayList kosong dan antrianPasien sebagai HashMap kosong.



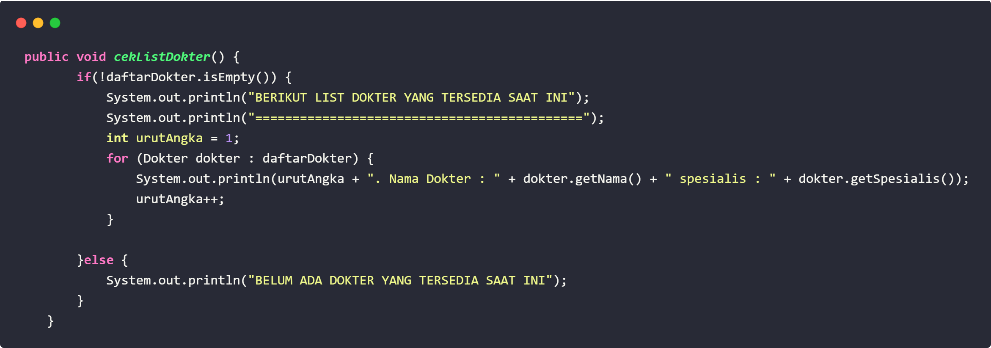
1. Metode Tambah Dokter

Metode tambahDokter menambahkan objek Dokter ke dalam daftarDokter jika dokter tersebut belum terdaftar. Jika dokter sudah terdaftar, akan menampilkan pesan "DOKTER SUDAH TERDAFTAR"; jika belum, akan menambahkannya ke dalam daftar dan menampilkan pesan "SUKSES".



1. Metode Cek Daftar Dokter

Metode cekListDokter menampilkan daftar dokter yang tersedia saat ini. Jika daftarDokter tidak kosong, metode ini akan mencetak daftar nama dan spesialisasi setiap dokter. Jika kosong, akan menampilkan pesan bahwa belum ada dokter yang tersedia.



1. Metode Tambah Pasien

Metode tambahPasien menambahkan pasien ke dalam antrianPasien jika pasien tersebut belum terdaftar dan dokter yang diminta tersedia. Metode ini memeriksa keberadaan dokter berdasarkan nama, kemudian menambahkan pasien dengan informasi keluhan, nama dokter, dan tipe pasien. Jika berhasil, akan menampilkan pesan "SUKSES"; jika dokter tidak tersedia atau pasien sudah terdaftar, akan menampilkan pesan yang sesuai.

1. Metode Cek Antrian Pasien

Metode cekAntrianPasien menampilkan antrian pasien yang ada di rumah sakit, dikelompokkan berdasarkan dokter. Metode ini membuat map baru untuk mengelompokkan pasien sesuai dengan dokter yang menangani, kemudian mencetak antrian untuk setiap dokter. Jika tidak ada pasien dalam antrian, akan menampilkan pesan "ANTRIAN KOSONG".

1. Metode Periksa Pasien

Metode periksaPasien menghapus pasien dari antrianPasien dan mencetak pesan bahwa pasien sedang diperiksa jika pasien tersebut ada dalam antrian. Jika pasien tidak ditemukan dalam antrian, akan menampilkan pesan bahwa pasien tidak ada dalam antrian.

# *Class* Orang

# Konstruktor

Konstruktor Orang menerima tiga parameter (nama, tanggalLahir, dan jenisKelamin) untuk menginisialisasi atribut-atribut kelas ini. Konstruktor ini digunakan untuk membuat objek Orang dengan atribut yang diberikan.

# *Enum Class*

JENIS\_KELAMIN adalah enum yang mendefinisikan dua nilai konstan, yaitu LAKI\_LAKI dan PEREMPUAN. Enum ini digunakan untuk merepresentasikan jenis kelamin seseorang.

# *Class* Pasien

# Konstruktor

# Metode Daftar

# Metode Periksa

# *Class* Dokter

# Konstruktor

# *Class* Main

**SOAL 2**

## Class HRISGUI

HRISGUI adalah kelas utama yang mengatur antarmuka pengguna (GUI) untuk aplikasi HRIS. Kelas ini memperluas JFrame dari Swing untuk membuat jendela aplikasi. Ada beberapa komponen utama dalam kelas ini:

## Komponen GUI

* JFrame frame: Jendela utama aplikasi.
* JComboBox<String> departementComboBox: Pilihan dropdown untuk memilih departemen pegawai.
* JTextField nameField, tglField, posisiField: Bidang teks untuk memasukkan nama, tanggal lahir, dan posisi pegawai.
* JButton addEmployeeButton, addButton, showAllButton: Tombol untuk menambahkan pegawai dan menampilkan semua data pegawai.
* JTextArea displayArea: Area teks untuk menampilkan data pegawai.
* JTextField findIdField: Bidang teks untuk mencari pegawai berdasarkan ID (fitur ini belum sepenuhnya diimplementasikan).

## Panel dan Layout

* showMainPanel(): Metode ini menampilkan panel utama dengan tombol untuk menambah data dan menampilkan data pegawai.
* showInputPanel(): Metode ini menampilkan panel untuk memasukkan data pegawai baru.
* showAllDataPanel(): Metode ini menampilkan panel untuk menampilkan semua data pegawai yang sudah ada.

## Fungsi Tambahan

* repaintPanel(JPanel newPanel): Metode untuk menyegarkan tampilan panel dengan panel baru.

## Abstract Class Employee

Employee adalah kelas abstrak yang menjadi superclass untuk semua tipe pegawai. Kelas ini menyimpan informasi dasar seperti:

* String name, birthDate, position, department, employeeID: Properti untuk menyimpan informasi pegawai.
* double salary: Properti untuk menyimpan gaji pegawai.
* Konstruktor Employee(String name, String birthDate, String position, String department): Menginisialisasi objek pegawai dan menghasilkan ID unik.
* generateEmployeeID(String department, String birthDate): Metode untuk menghasilkan ID unik dengan format "000 + kode departemen + tanggal lahir + angka random".
* getDetails(): Metode untuk mendapatkan detail lengkap pegawai dalam bentuk string.
* setSalary(String position): Metode abstrak yang diimplementasikan di subkelas untuk menetapkan gaji berdasarkan posisi.

## SubClass Manager, Developer, HR Staff

Tiga subkelas dari Employee yang spesifik untuk masing-masing posisi:

1. *Manager*:

- Mengimplementasikan setSalary untuk mengembalikan gaji sebesar 20.000.000.

2. *Developer*:

- Mengimplementasikan setSalary untuk mengembalikan gaji sebesar 10.000.000.

3. *HRStaff*:

- Mengimplementasikan setSalary untuk mengembalikan gaji sebesar 5.000.000.

## Fungsi Tambah dan Tampilkan Pegawai

Menambah Pegawai:

* Di showInputPanel(), pengguna memasukkan detail pegawai seperti nama, tanggal lahir, posisi, dan departemen.
* Ketika tombol "Add Data" ditekan, data pegawai baru dibuat berdasarkan posisi dan disimpan dalam daftar ArrayList<Employee>.
* Pesan konfirmasi ditampilkan dengan detail pegawai baru.

Menampilkan Semua Pegawai:

* Di showAllDataPanel(), semua data pegawai yang disimpan dalam ArrayList<Employee> ditampilkan di displayArea.

Link gitHub: <https://github.com/Wahhab2004/Pemrograman-Berorientasi-Objek_A1_Wahhab-Awaludin/tree/2f6b60f8ebc03037679f6aa75011394ec67f9c97/Pertemuan%2010%20-%20Abstract%20%26%20Interface/task01>