

Nama : I Kadek Angga Dwi Suputra

Nim : 2201010137

Kelas : C

UJIAN AKHIR SEMESTER

A. Quis 1

Sistem iphone adalah sebuah aplikasi yang dirancang untuk mengelola data terkait model, warna, RAM, dan kapasitas baterai iPhone. Aplikasi ini bertujuan untuk memudahkan pengguna dalam menyimpan, mengedit, dan menghapus informasi mengenai berbagai jenis iPhone yang dimiliki. Dengan antarmuka yang sederhana dan intuitif, pengguna dapat dengan mudah memasukkan data iPhone baru melalui bidang input yang tersedia.

- Alasan menggunakan Teori OOP adalah:

1. Encapsulation
Gunakan field private dan metode getter/setter publik di kelas Iphone.
2. Inheritance
Jika diperlukan, buat subclass dari Iphone untuk model tertentu.
3. Polymorphism
Implementasikan metode yang dapat dioverride di subclass.
4. Access Modifiers
Gunakan access modifier yang sesuai (private, protected, public) untuk mengontrol akses ke anggota kelas.

- Implementasi Project

- 1) Iphone.java

```
public class Iphone {
    private String model;
    private String color;
    private int ram;
    private int batteryCapacity;

    public Iphone(String model, String color, int ram, int
batteryCapacity) {
        this.model = model;
        this.color = color;
        this.ram = ram;
        this.batteryCapacity = batteryCapacity;
    }

    // Getter dan setter untuk setiap atribut
    public String getModel() { return model; }
    public void setModel(String model) { this.model = model;
}

    public String getColor() { return color; }
    public void setColor(String color) { this.color = color;
}
}
```

```

        public int getRam() { return ram; }
        public void setRam(int ram) { this.ram = ram; }

        public int getBatteryCapacity() { return
batteryCapacity; }
        public void setBatteryCapacity(int batteryCapacity) {
this.batteryCapacity = batteryCapacity; }
    }

```

2) IphoneManager.java

```

import java.util.ArrayList;

public class IphoneManager {
    private ArrayList<Iphone> iphones;

    public IphoneManager() {
        iphones = new ArrayList<>();
    }

    public void addIphone(Iphone iphone) {
        iphones.add(iphone);
    }

    public void editIphone(int index, Iphone iphone) {
        if (index >= 0 && index < iphones.size()) {
            iphones.set(index, iphone);
        }
    }

    public void deleteIphone(int index) {
        if (index >= 0 && index < iphones.size()) {
            iphones.remove(index);
        }
    }

    public ArrayList<Iphone> getIphones() {
        return iphones;
    }
}

```

3) IphoneGUI.java

```

import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class IphoneGUI extends JFrame {
    private JTextField modelField, colorField, ramField,
batteryField;
    private JButton addButton, editButton, deleteButton,
resetButton;
    private JTable table;
    private IphoneManager manager;
}

```

```

public IphoneGUI() {
    manager = new IphoneManager();
    initComponents();
}

private void initComponents() {
    setTitle("IPHONE");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(600, 400);
    setLayout(null);

    JLabel titleLabel = new JLabel("IPHONE",
SwingConstants.CENTER);
    titleLabel.setBounds(200, 10, 200, 30);
    add(titleLabel);

    JLabel modelLabel = new JLabel("Model Iphone:");
    modelLabel.setBounds(50, 50, 100, 25);
    add(modelLabel);

    modelField = new JTextField();
    modelField.setBounds(150, 50, 150, 25);
    add(modelField);

    JLabel colorLabel = new JLabel("Warna Iphone:");
    colorLabel.setBounds(50, 90, 100, 25);
    add(colorLabel);

    colorField = new JTextField();
    colorField.setBounds(150, 90, 150, 25);
    add(colorField);

    JLabel ramLabel = new JLabel("RAM Iphone:");
    ramLabel.setBounds(50, 130, 100, 25);
    add(ramLabel);

    ramField = new JTextField();
    ramField.setBounds(150, 130, 150, 25);
    add(ramField);

    JLabel batteryLabel = new JLabel("Kapasitas
Baterai:");
    batteryLabel.setBounds(50, 170, 100, 25);
    add(batteryLabel);

    batteryField = new JTextField();
    batteryField.setBounds(150, 170, 150, 25);
    add(batteryField);

    addButton = new JButton("Tambah");

```

```

        addButton.setBounds(50, 210, 80, 30);
        add(addButton);

        editButton = new JButton("Edit");
        editButton.setBounds(140, 210, 80, 30);
        add(editButton);

        deleteButton = new JButton("Hapus");
        deleteButton.setBounds(230, 210, 80, 30);
        add(deleteButton);

        resetButton = new JButton("Reset");
        resetButton.setBounds(320, 210, 80, 30);
        add(resetButton);

        table = new JTable();
        JScrollPane scrollPane = new JScrollPane(table);
        scrollPane.setBounds(50, 250, 500, 100);
        add(scrollPane);

        addButton.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent e) {
                // Tambahkan logika untuk menambah iPhone
di sini
            }
        });

        editButton.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent e) {
                // Tambahkan logika untuk mengedit iPhone
di sini
            }
        });

        deleteButton.addActionListener(new
        ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // Tambahkan logika untuk menghapus
iPhone di sini
            }
        });

        resetButton.addActionListener(new
        ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {

```


```

// Tambahkan logika untuk mereset bidang
teks di sini
    }
    });
}

```

B. Quis 2

1. Implementasi Desain aplikasi dari Quis 1


— □ ×

IPHONE

Model Iphone

Warna Iphone

RAM Iphone

Kapasitas Baterai

Tambah

Edit

Hapus

Reset

Model Iphoone	Warna Iphone	RAM Iphone	Kapasitas Bat...