

**UAS OOP**



**INSTIKI**  
INSTITUT BISNIS DAN  
TEKNOLOGI INDONESIA

**Nama : Muhammad Agriyansyah**

**NIM : 2201010410**

## **Penjelasan Project Currency Converter (Konversi Mata Uang)**

### **Deskripsi Project**

Project ini adalah sebuah aplikasi desktop berbasis Java yang digunakan untuk mengkonversi berbagai mata uang dari satu mata uang ke mata uang lainnya. Aplikasi ini menggunakan antarmuka grafis pengguna (GUI) yang dibangun menggunakan Swing, sebuah toolkit GUI untuk Java.

### **Fitur Utama**

1. **Pilihan Mata Uang:** Pengguna dapat memilih mata uang asal dan mata uang tujuan dari daftar yang disediakan.
2. **Input Jumlah Mata Uang:** Pengguna dapat memasukkan jumlah mata uang yang ingin dikonversi.
3. **Konversi Mata Uang:** Aplikasi akan menghitung nilai konversi berdasarkan kurs yang telah ditentukan dan menampilkan hasilnya.
4. **Reset dan Keluar:** Pengguna dapat mereset pilihan dan input atau keluar dari aplikasi.

### **Komponen Utama**

- **JPanel:** Panel utama yang digunakan untuk menampung komponen GUI lainnya.
- **JLabel:** Label teks untuk memberikan informasi kepada pengguna.
- **JComboBox:** Kotak pilihan untuk memilih mata uang asal dan tujuan.
- **TextField:** Bidang teks untuk memasukkan jumlah mata uang yang akan dikonversi dan untuk menampilkan hasil konversi.
- **JButton:** Tombol untuk melakukan aksi tertentu seperti konversi, reset, dan keluar.

### **Struktur Kode**

Berikut adalah penjelasan rinci mengenai struktur kode yang digunakan dalam project ini:

1. Deklarasi Variabel  
“double US\_Dollar = 1.31;  
double Nigerian\_Naira = 476.57;

```
double Brazilian_Real = 5.47;
double Canadian_Dollar = 1.71;
double Kenyan_Shilling = 132.53;
double Indonesian_Rupiah = 19554.94;
double Indian_Rupee = 95.21;
double Philippine_Pisco = 71.17;
double Pakistani_Rupee = 162.74;”
```

Variabel di atas mendefinisikan kurs tetap untuk masing-masing mata uang terhadap mata uang dasar (pound sterling).

## 2. Inisialisasi Array Mata Uang

```
String[] currencyUnits = {
```

```
“ "Units",
  "US Dollar",
  "Nigerian Naira",
  "Brazilian Real",
  "Canadian Dollar",
  "Kenyan Shilling",
  "Indonesian Rupiah",
  "Indian Rupee",
  "Philippine Pisco",
  "Paskistani Rupee"
```

```
}; “ Array ini menyimpan nama-nama mata uang yang akan ditampilkan
dalam JComboBox.
```

## 3. Konstruktor

```
public CurrencyConverter() {
    initComponents();
}
```

Konstruktor ini memanggil metode initComponents() untuk menginisialisasi komponen GUI.

#### **4. Inisialisasi Komponen**

Metode initComponents() menginisialisasi dan mengatur tata letak komponen GUI, termasuk JPanel, JLabel, JComboBox, JTextField, dan JButton.

#### **5. Aksi Tombol**

- **Convert Button:** Melakukan konversi mata uang berdasarkan pilihan dan jumlah yang dimasukkan oleh pengguna.
- **Reset Button:** Mengatur ulang semua pilihan dan input.
- **Exit Button:** Keluar dari aplikasi.

#### **Pemrograman OOP yang Digunakan**

##### **1. Kelas dan Objek:**

- **CurrencyConverter** adalah kelas utama yang merepresentasikan aplikasi konversi mata uang.
- Kelas ini merupakan turunan dari javax.swing.JFrame, yang memungkinkan kita untuk membuat aplikasi GUI dengan menggunakan fitur-fitur dari JFrame.

##### **2. Encapsulation:**

- Variabel-variabel seperti US\_Dollar, Nigerian\_Naira, dll., dienkapsulasi dalam kelas CurrencyConverter.
- Penggunaan private untuk komponen GUI seperti JButton, JComboBox, dan JTextField melindungi akses langsung dari luar kelas.

##### **3. Inheritance:**

- Kelas CurrencyConverter mewarisi sifat-sifat dari javax.swing.JFrame, sehingga dapat menggunakan semua metode dan properti dari JFrame untuk membuat dan mengelola GUI.

##### **4. Polymorphism:**

- Penggunaan polymorphism dalam event handling, seperti ketika metode actionPerformed dari ActionListener dipanggil. Tombol

yang berbeda (Convert, Reset, Exit) mengimplementasikan aksi yang berbeda pada saat ditekan.

## 5. Event Handling:

- Menggunakan ActionListener untuk menangani aksi dari pengguna, seperti menekan tombol atau memilih item dari JComboBox.

## Alasan Penggunaan Teknik OOP

- **Keterbacaan Kode:** Dengan menggunakan kelas dan objek, kode menjadi lebih mudah dibaca dan dipelihara.
- **Reusability:** Komponen GUI dan logika dapat digunakan kembali dalam bagian lain dari aplikasi atau di aplikasi lain.
- **Modularity:** Memungkinkan pemisahan logika aplikasi dari antarmuka pengguna, sehingga perubahan pada satu bagian tidak mempengaruhi bagian lain.
- **Maintainability:** Penggunaan enkapsulasi membantu melindungi data dan menjaga integritas, memudahkan pemeliharaan dan debugging.

## Kesimpulan

Project Currency Converter ini menggunakan berbagai teknik pemrograman berorientasi objek (OOP) untuk membangun sebuah aplikasi konversi mata uang yang mudah digunakan dan dipelihara. Dengan memanfaatkan konsep OOP seperti kelas dan objek, enkapsulasi, inheritance, dan polymorphism, aplikasi ini menjadi modular, dapat diperluas, dan mudah untuk dipahami.