Analisis Event-Driven Programming pada Aplikasi Manajemen Keuangan C#

WAHYU SETYO BUDI – 2213020012 – 3A

Event-driven programming adalah paradigma pemrograman yang sangat penting dalam pengembangan antarmuka pengguna grafis (GUI). Dalam pendekatan ini, program tidak berjalan dalam urutan yang tetap, melainkan merespons peristiwa (event) yang terjadi, seperti interaksi pengguna berupa klik tombol, input teks, atau pemilihan item. Aplikasi berbasis Windows Forms di C# merupakan salah satu contoh nyata dari penggunaan event-driven programming.

Studi Kasus: Aplikasi Manajemen Keuangan

Aplikasi manajemen keuangan yang dibangun dengan Windows Forms C# ini memungkinkan pengguna untuk mengelola transaksi keuangan, termasuk menambahkan transaksi baru, menampilkan semua transaksi, memperbarui data transaksi berdasarkan username, dan menghapus transaksi tertentu.
Setiap fungsi utama dalam aplikasi ini dijalankan sebagai respons terhadap suatu event yang dipicu oleh pengguna, seperti menekan tombol pada form. Berikut adalah analisis mendetail dari berbagai event handler yang digunakan dalam aplikasi ini.

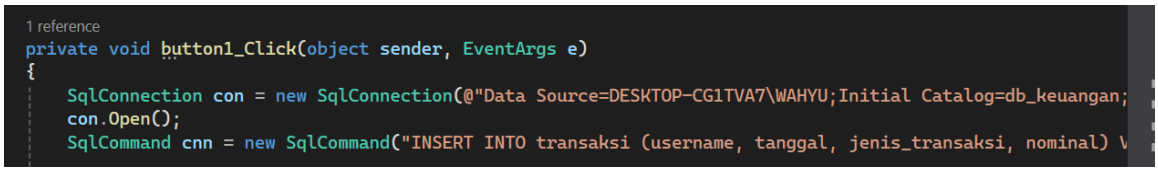
Penjelasan Event Handler

1. Form1_Load

Event ini dipicu saat form utama (Form1) dimuat. Biasanya digunakan untuk menginisialisasi komponen atau data dari database, walaupun dalam kode program ini, metode ini belum digunakan.

2. button1_Click - Tambah Transaksi

Event ini dipicu saat pengguna mengklik tombol untuk menambahkan data transaksi. Fungsi ini mengambil input dari TextBox (username dan nominal), DateTimePicker (tanggal), dan RadioButton (jenis transaksi). Kemudian, data tersebut disisipkan ke dalam tabel transaksi di database menggunakan perintah SQL INSERT.



```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-CG1TVA7\WAHYU;Initial Catalog=db_keuangan;
    con.Open();
    SqlCommand cnn = new SqlCommand("INSERT INTO transaksi (username, tanggal, jenis_transaksi, nominal) V
```

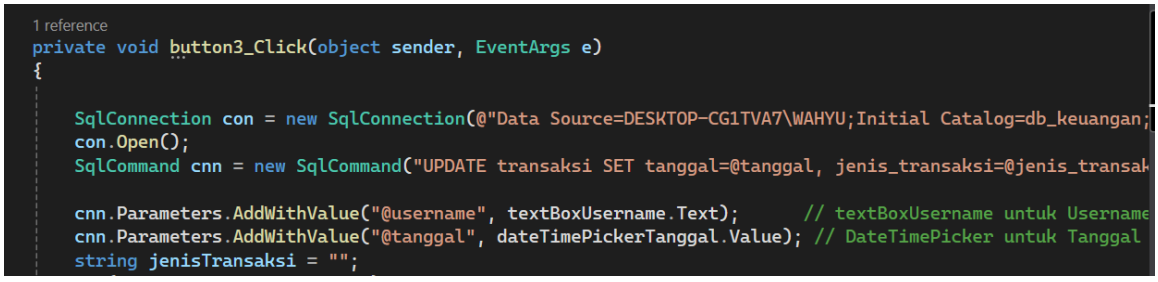
3. button2_Click - Tampilkan Data

Saat tombol ini diklik, aplikasi mengeksekusi perintah SELECT * FROM transaksi dan menggunakan SqlDataAdapter untuk mengambil semua data dari database. Data yang diperoleh ditampilkan ke dalam kontrol DataGridView.

```
1 reference
private void button2_Click(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-CG1TVA7\WAHYU;Initial Catalog=db_keuangan;
    con.Open();
    SqlCommand cnn = new SqlCommand("select * from transaksi", con);
    SqlDataAdapter da = new SqlDataAdapter(cnn);
    DataTable table = new DataTable();
    da.Fill(table);
    dataGridView1.DataSource = table;
```

4. button3_Click - Update Transaksi

Event ini digunakan untuk memperbarui transaksi yang sudah ada berdasarkan username. Input pengguna digunakan untuk menentukan nilai baru dari tanggal, jenis transaksi, dan nominal. Data kemudian diupdate di database menggunakan perintah SQL UPDATE.



```
1 reference
private void button3_Click(object sender, EventArgs e)
{

    SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-CG1TVA7\WAHYU;Initial Catalog=db_keuangan;
    con.Open();
    SqlCommand cnn = new SqlCommand("UPDATE transaksi SET tanggal=@tanggal, jenis_transaksi=@jenis_transak

    cnn.Parameters.AddWithValue("@username", textBoxUsername.Text);      // textBoxUsername untuk Username
    cnn.Parameters.AddWithValue("@tanggal", dateTimePickerTanggal.Value); // DateTimePicker untuk Tanggal
    string jenisTransaksi = "";
```

5. button4_Click - Hapus Transaksi

Ketika tombol ini diklik, aplikasi menghapus data transaksi berdasarkan username yang dimasukkan pengguna. Hal ini dilakukan menggunakan perintah SQL DELETE.

```
1 reference
private void button4_Click(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-CG1TVA7\WAHYU;Initial Catalog=db_keuangan;
    con.Open();
    SqlCommand cnn = new SqlCommand("delete transaksi where username=@username", con);
    cnn.Parameters.AddWithValue("@username", textBoxUsername.Text); // Kirim sebagai string
    cnn.ExecuteNonQuery();
    con.Close();
    MessageBox.Show("Record Deleted Successfully!");
```

6. textBox2_TextChanged dan label_Click Events

Event ini menangani perubahan teks atau klik pada label. Meskipun tersedia dalam kode, fungsinya belum diimplementasikan.

```
1 reference
private void textBox2_TextChanged(object sender, EventArgs e)
{
```

Keuntungan Event-Driven Programming

Beberapa keuntungan dari penggunaan event-driven programming adalah sebagai berikut:

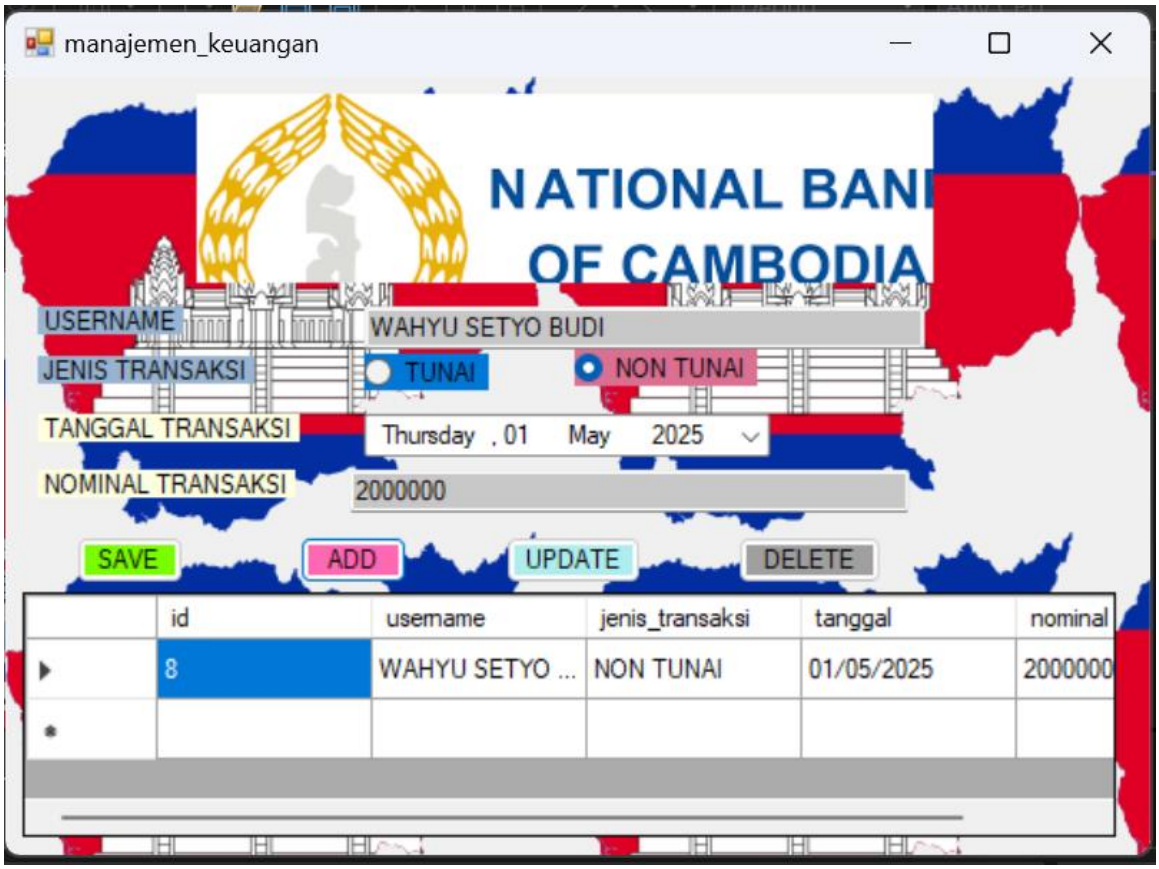
- Responsif terhadap interaksi pengguna.

- Modularitas yang baik karena setiap event memiliki handler tersendiri.

- Cocok untuk aplikasi berbasis GUI karena fleksibel dan interaktif.

Kesimpulan

Event-driven programming merupakan pendekatan yang sangat efisien untuk membangun aplikasi GUI. Melalui aplikasi manajemen keuangan ini, kita bisa melihat bagaimana event seperti klik tombol dan input pengguna dikelola melalui event handler yang mengakses dan memanipulasi data di database. Dengan memahami dan mengimplementasikan event handler dengan baik, pengembang dapat membangun aplikasi yang tidak hanya fungsional tetapi juga ramah pengguna.



Aplikasi ini merupakan sebuah sistem keuangan berbasis Windows Forms (C#) yang memungkinkan pengguna untuk memperbarui data transaksi ke dalam database SQL Server, tepatnya pada tabel bernama transaksi. Saat pengguna menekan tombol yang terhubung dengan event button3_Click, aplikasi akan membuka koneksi ke database db_keuangan, lalu mengambil data dari form input yang meliputi username, tanggal transaksi, jenis transaksi (TUNAI atau NON TUNAI), serta nominal transaksi. Setelah itu, aplikasi menjalankan perintah UPDATE untuk memperbarui data transaksi di database berdasarkan username yang dimasukkan. Kolom-kolom yang diperbarui adalah tanggal, jenis_transaksi, dan nominal. Jika proses update berhasil, aplikasi akan menampilkan pesan notifikasi kepada pengguna berupa pop-up dengan isi "Data transaksi berhasil diperbarui!". Sebagai contoh, jika sebelumnya transaksi dengan username "wahyu setyo budi" memiliki data tanggal 1 mei 2025, jenis transaksi "TUNAI", dan nominal 200.000, lalu pengguna mengubah tanggal menjadi 1 Mei 2025, jenis transaksi menjadi "NON TUNAI", dan nominal menjadi 150.000, maka data pada tabel akan diperbarui sesuai input updatet. Dengan kata lain, hasil dari aplikasi ini adalah data transaksi dalam database akan berubah sesuai input baru yang dimasukkan oleh pengguna melalui form aplikasi.