

Nama : Ciko Nanda P.W.

Npm : 2213020263

Nama : Mochammad Malikal Mulki

Npm : 2213020265

Penerapan Konsep Event-Driven Programming dalam Pembuatan Aplikasi Kuis Pembelajaran Umum Berbasis C# Windows Forms

Pendahuluan

Dalam era digital saat ini, pemanfaatan teknologi informasi dalam dunia pendidikan menjadi suatu kebutuhan yang tidak dapat dihindari. Salah satu bentuk pemanfaatan teknologi dalam pembelajaran adalah pembuatan aplikasi kuis digital, yang dapat digunakan untuk membantu guru dalam mengevaluasi pemahaman siswa secara interaktif dan efisien. Pada artikel ini, saya akan menjelaskan secara mendetail bagaimana konsep *event-driven programming* diterapkan dalam aplikasi kuis yang telah saya buat menggunakan bahasa pemrograman C# dan platform Windows Forms.

Apa Itu Event-Driven Programming?

Sebelum membahas lebih lanjut mengenai penerapannya, penting untuk memahami terlebih dahulu apa yang dimaksud dengan *event-driven programming*. *Event-driven programming* adalah suatu paradigma pemrograman yang menitikberatkan pada kejadian atau peristiwa (*event*) sebagai pemicu alur program. Dalam konteks aplikasi antarmuka grafis seperti Windows Forms, *event* bisa berupa tindakan pengguna seperti mengklik tombol, memilih opsi dari *combo box*, menutup jendela, atau mengetik dalam kolom input. Setiap event tersebut dapat dihubungkan dengan sebuah fungsi tertentu yang akan dijalankan secara otomatis saat event tersebut terjadi. Fungsi ini biasa dikenal dengan istilah *event handler*.

Paradigma ini sangat sesuai untuk aplikasi yang memiliki banyak interaksi dengan pengguna, seperti aplikasi kuis yang saya buat. Karena aplikasi ini sepenuhnya dikendalikan oleh tindakan pengguna, maka penggunaan *event-driven programming* menjadi pondasi utama dalam perancangan dan pengembangannya.

Desain Aplikasi dan Struktur Komponen

Aplikasi kuis ini terdiri dari beberapa form utama, yang masing-masing memiliki fungsinya sendiri. Komponen-komponen yang digunakan antara lain:

- **MainForm.cs:** Form utama yang menampilkan daftar soal dan menyediakan fitur untuk menambah, mengedit, menghapus soal (CRUD), serta tombol untuk memulai kuis.
- **QuestionForm.cs:** Form yang digunakan untuk menambah atau mengedit soal kuis.
- **QuizForm.cs:** Form yang digunakan oleh pengguna untuk menjawab soal secara berurutan dan mendapatkan skor akhir.
- **Database.cs:** Berfungsi untuk melakukan koneksi dan manipulasi data soal pada database lokal (SQLite).

Setiap form dirancang dengan prinsip *event-driven*, yaitu setiap tombol yang diklik akan mengeksekusi sebuah fungsi yang telah ditentukan sebagai *event handler*.

Contoh Penerapan Event Handler: Menambahkan Soal Baru

Ketika pengguna ingin menambahkan soal baru, mereka dapat mengklik tombol "Tambah" pada form utama. Aksi ini akan memicu event `btnAdd_Click`, yang akan membuka form `QuestionForm` untuk menginput soal baru.

```
private void btnAdd_Click(object sender, EventArgs e)
{
    var form = new QuestionForm();
    if (form.ShowDialog() == DialogResult.OK)
        LoadQuestions();
}
```

Penjelasan:

- Fungsi `btnAdd_Click` dipanggil secara otomatis ketika tombol "Tambah" diklik oleh pengguna.
- Program akan menampilkan form input soal baru (`QuestionForm`) dan menunggu hingga pengguna selesai mengisi data.
- Jika pengguna menekan tombol simpan (yang mengembalikan `DialogResult.OK`), maka fungsi `LoadQuestions()` akan dipanggil untuk memperbarui daftar soal pada

tampilan utama.

Dengan pendekatan ini, aplikasi menjadi lebih interaktif dan pengguna merasa terlibat langsung dalam proses pengelolaan soal.

Event Driven pada Form Kuis (*QuizForm.cs*)

Bagian terpenting dari aplikasi kuis adalah form untuk menjawab soal. Form ini memiliki komponen berupa label pertanyaan, pilihan jawaban (menggunakan radio button), dan tombol “Next”. Ketika pengguna memilih jawaban dan menekan tombol “Next”, maka sistem akan mencatat jawaban, memeriksa kebenarannya, dan melanjutkan ke soal berikutnya.

Berikut ini adalah cuplikan kode fungsi event handler dari tombol “Next”:

```
private void btnNext_Click(object sender, EventArgs e)
{
    var q = questions[currentIndex];
    string selectedAnswer = "";

    if (rbA.Checked) selectedAnswer = "A";
    else if (rbB.Checked) selectedAnswer = "B";
    else if (rbC.Checked) selectedAnswer = "C";
    else if (rbD.Checked) selectedAnswer = "D";

    if (selectedAnswer == q.Answer)
    {
        score++;
    }

    currentIndex++;
    ShowQuestion();
}
```

Penjelasan:

- Fungsi ini secara otomatis dijalankan ketika pengguna mengklik tombol “Next”.
- Pertama, program memeriksa jawaban yang dipilih pengguna melalui radio button.
- Jika jawaban sesuai dengan kunci jawaban (*q.Answer*), maka skor ditambah.
- Setelah itu, *currentIndex* ditambah untuk menampilkan soal selanjutnya.

- Fungsi `ShowQuestion()` bertugas menampilkan pertanyaan baru di form.

Pendekatan ini sangat efektif dalam menciptakan pengalaman kuis yang mengalir dan adaptif, serta memungkinkan pengguna untuk mendapatkan umpan balik secara langsung melalui skor akhir.

Mengelola Soal: Event Edit dan Delete

Tidak hanya menambah soal, pengguna juga bisa mengedit atau menghapus soal yang ada. Fungsionalitas ini juga dibangun dengan *event-driven*. Berikut contoh event handler untuk mengedit soal:

```
private void btnEdit_Click(object sender, EventArgs e)
{
    if (dataGridViewQuestions.CurrentRow != null)
    {
        var q = (Question)dataGridViewQuestions.CurrentRow.DataBoundItem;
        var form = new QuestionForm(q);
        if (form.ShowDialog() == DialogResult.OK)
            LoadQuestions();
    }
}
```

Sedangkan untuk menghapus soal:

```
private void btnDelete_Click(object sender, EventArgs e)
{
    if (dataGridViewQuestions.CurrentRow != null)
    {
        var q = (Question)dataGridViewQuestions.CurrentRow.DataBoundItem;
        Database.DeleteQuestion(q.Id);
        LoadQuestions();
    }
}
```

Penjelasan:

- Event `btnEdit_Click` akan mengambil data dari baris yang dipilih di `DataGridView` dan membuka form edit.
- Event `btnDelete_Click` akan langsung menghapus data dari database berdasarkan ID soal.

- Setelah edit atau hapus, data akan dimuat ulang untuk memperbarui tampilan.

```
using System;  
using System.Data;  
using System.Windows.Forms;  
using MySql.Data.MySqlClient;
```

Berikut penjelasan dari kodingan di gambar baris demi baris:

1. **using System** : Ini adalah pustaka dasar dari .NET. Berisi fungsi-fungsi umum seperti *Console.WriteLine*, operasi *string*, manipulasi waktu (*DateTime*), dll.
2. **using System.Data** : *Namespace* ini menyediakan kelas-kelas yang dibutuhkan untuk mengakses dan mengelola data, biasanya dalam konteks *DataTable*, *DataSet*, dan lainnya. Digunakan saat kita bekerja dengan data baik dari database maupun sumber lain.
3. **using System.Windows.Forms** : Ini adalah *namespace* utama untuk membuat aplikasi Windows Forms (antarmuka grafis di Windows). Berisi elemen seperti *Form*, *Button*, *TextBox*, dan sebagainya.
4. **using MySql.Data.MySqlClient** : Ini adalah namespace dari library **MySQL Connector/NET**. Digunakan untuk menghubungkan aplikasi *C#* ke database **MySQL**. Di sini terdapat kelas seperti:
 - **MySqlConnection** (untuk membuka koneksi ke database),
 - **MySqlCommand** (untuk menjalankan query SQL),
 - **MySqlDataReader** (untuk membaca hasil query).

```
// MySQL connection string  
string connStr = "server=localhost;user=root;database=db_loginvs;port=3306";  
  
using (MySqlConnection conn = new MySqlConnection(connStr))  
{  
    try  
    {  
        conn.Open();  
        string query = "SELECT * FROM user WHERE username = @username AND password = @password";  
        MySqlCommand cmd = new MySqlCommand(query, conn);  
        cmd.Parameters.AddWithValue("@username", username);  
        cmd.Parameters.AddWithValue("@password", password);  
  
        using (MySqlDataReader reader = cmd.ExecuteReader())  
        {  
            if (reader.HasRows)  
            {  
                this.Hide();  
                FormStart start = new FormStart(username);  
                start.Show();  
            }  
            else  
            {  
                MessageBox.Show("Login failed. Invalid username or password.");  
            }  
        }  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show($"An error occurred: {ex.Message}");  
    }  
}
```

Penjelasan:

Kode diatas merupakan implementasi proses login pada aplikasi **C#** yang terhubung dengan database MySQL. Kode ini memverifikasi kredensial pengguna (username dan password) dengan mencocokkannya terhadap data di tabel **user**. Jika cocok, aplikasi akan menyembunyikan form login dan membuka form berikutnya (**FormStart**). Jika tidak cocok, akan ditampilkan pesan kesalahan.

Kesimpulan

Melalui aplikasi kuis ini, saya telah mempelajari secara langsung dan mendalam bagaimana konsep *event-driven programming* bekerja dalam konteks pengembangan perangkat lunak berbasis GUI. Dengan mengandalkan event seperti klik tombol dan perubahan nilai pada komponen UI, aplikasi dapat berjalan dengan responsif, fleksibel, dan interaktif.

Pendekatan *event-driven* sangat cocok digunakan dalam aplikasi pendidikan karena mampu menyesuaikan alur program berdasarkan tindakan pengguna. Hal ini mencerminkan prinsip pembelajaran yang aktif dan partisipatif, di mana siswa atau pengguna menjadi pusat dari pengalaman belajar itu sendiri.

Penerapan ini menjadi bukti nyata bahwa penguasaan konsep dasar dalam pemrograman, seperti *event-driven programming*, sangat penting bagi calon pendidik di era digital agar mampu merancang media pembelajaran yang inovatif dan efektif.