

Penerapan Event-Driven Programming dalam Aplikasi MathRush

Pendahuluan

Event-driven programming adalah paradigma pemrograman yang mendasarkan alur eksekusinya pada kejadian-kejadian atau peristiwa (event) yang terjadi selama aplikasi dijalankan. Dalam konteks aplikasi MathRush yang dikembangkan menggunakan bahasa C# dan platform Windows Forms, paradigma ini sangat dominan, karena sebagian besar interaksi pengguna ditangani melalui berbagai event seperti klik tombol, timer, serta perubahan input.

1. Event-Driven pada MainForm.cs

File MainForm.cs merupakan antarmuka utama aplikasi MathRush, yang memuat fitur untuk memulai permainan, menjawab soal, dan mengelola leaderboard. Berikut ini adalah penjelasan beberapa event penting yang terdapat dalam file ini:

1.1 Event: btnStart_Click

Event ini menangani aksi ketika pengguna mengklik tombol 'Mulai'. Fungsinya adalah mengatur ulang skor, mengatur waktu, mengaktifkan input jawaban, dan memulai soal baru serta timer permainan. Event ini dipicu saat pengguna mengklik tombol 'Mulai' pada form utama.

Saat tombol ditekan, berbagai aksi dijalankan:

- Skor di-reset ke 0.
- Timer diatur ulang ke 10 detik.
- Tombol Submit dan input jawaban diaktifkan.
- Tombol 'Mulai' dan 'Kelola' dinonaktifkan untuk menghindari gangguan selama permainan.
- Soal pertama dibuat menggunakan fungsi GenerateQuestion().
- Timer dimulai dengan StartTimer().

```
private void btnStart_Click(object sender, EventArgs e)  
{  
    score = 0;  
    timeLeft = 10;  
    lblScore.Text = "Skor: 0";  
    btnSubmit.Enabled = true;  
}
```

```

txtAnswer.Enabled = true;
btnStart.Enabled = false;
btnKelola.Enabled = false;
lstLeaderboard.Items.Clear();
GenerateQuestion();
StartTimer();
}

```

1.2 Event: btnKelola_Click

Event ini membuka form CRUD yang digunakan untuk mengelola data skor yang tersimpan di database. Setelah form ditutup, leaderboard diperbarui.

Tombol 'Kelola' digunakan untuk membuka form CRUD (Create, Read, Update, Delete) data skor. Saat tombol ini diklik:

- FormCRUD ditampilkan sebagai dialog.
- Setelah ditutup, leaderboard diperbarui dengan LoadLeaderboard().

```

private void btnKelola_Click(object sender, EventArgs e)
{
    FormCRUD formCRUD = new FormCRUD();
    formCRUD.ShowDialog();
    LoadLeaderboard();
}

```

1.3 Event: btnSubmit_Click

Event ini dipicu saat pengguna mengirimkan jawaban dari soal matematika. Jika jawaban benar, skor bertambah dan soal baru dihasilkan.

Digunakan untuk memverifikasi jawaban dari soal yang ditampilkan:

- Jawaban dibandingkan dengan variabel correctAnswer.
- Jika benar, skor bertambah dan waktu di-reset.
- Soal baru dihasilkan.

```

private void btnSubmit_Click(object sender, EventArgs e)
{
    if (txtAnswer.Text.Trim() == correctAnswer)
    {
        score++;
        lblScore.Text = $"Skor: {score}";
        timeLeft = 10;
        GenerateQuestion();
    }
}

```

```
txtAnswer.Clear();  
}
```

2. Event-Driven pada GameForm.cs

2.1 Event: btnStartGame_Click

Event ini digunakan untuk memulai permainan pada form terpisah. Waktu dan skor disiapkan, dan soal pertama dimunculkan.

Tombol ini memulai game pada form terpisah dengan waktu 60 detik:

- Timer dan skor di-reset.
- Soal pertama dihasilkan.

```
private void btnStartGame_Click(object sender, EventArgs e)  
{  
    gameTime = 60;  
    playerScore = 0;  
    lblScoreGame.Text = "Score: 0";  
    StartGameTimer();  
    GenerateGameQuestion();  
    btnStartGame.Enabled = false;}  
}
```

2.2 Event: btnSubmitGame_Click

Event ini menangani jawaban pemain di form game. Jika benar, skor bertambah dan soal baru diberikan.

Memeriksa jawaban yang diberikan dalam form game:

- Jika benar, skor bertambah 10 poin.
- Soal baru ditampilkan.

```
private void btnSubmitGame_Click(object sender, EventArgs e)  
{  
    if (txtAnswerGame.Text.Trim() == gameCorrectAnswer)  
    {  
        playerScore += 10;  
        lblScoreGame.Text = $"Score: {playerScore}";  
        GenerateGameQuestion();  
    }  
    txtAnswerGame.Clear();  
}
```

3. Event-Driven pada FromCRUD.cs

3.1 Event: *btnAdd_Click*

Event ini menangani penambahan data skor baru ke database berdasarkan input pengguna.

Menambahkan data baru ke database:

- Validasi input nama dan skor.
- Menyimpan data dengan SaveScore().

```
private void btnAdd_Click(object sender, EventArgs e)  
{  
    string name = txtName.Text.Trim();  
    if (int.TryParse(txtScore.Text.Trim(), out int score) &&  
!string.IsNullOrEmpty(name))  
    {  
        dbHelper.SaveScore(name, score);  
        LoadData();  
    }  
}
```

3.2 Event: *btnUpdate_Click*

Event ini memperbarui data skor yang dipilih dari daftar skor di database.

Memperbarui data skor berdasarkan pilihan pengguna:

- Validasi input dan ambil ID dari data yang dipilih.
- Jalankan UpdateScore().

```
private void btnUpdate_Click(object sender, EventArgs e)  
{  
    if (lstScores.SelectedItem != null)  
    {  
        string selectedItem = lstScores.SelectedItem.ToString();  
        int id = int.Parse(selectedItem.Split('.')[0]);  
        string name = txtName.Text.Trim();  
        if (int.TryParse(txtScore.Text.Trim(), out int score) &&  
!string.IsNullOrEmpty(name))  
        {  
            dbHelper.UpdateScore(id, name, score);  
            LoadData();  
        }  
    }  
}
```

```
}
```

3.3 Event: *btnDelete_Click*

Event ini digunakan untuk menghapus data skor dari database berdasarkan pilihan pengguna.

Menghapus data skor berdasarkan data yang dipilih:

- Ambil ID dari data yang dipilih.
- Jalankan DeleteScore().

```
private void btnDelete_Click(object sender, EventArgs e)
{
    if (lstScores.SelectedItem != null)
    {
        string selectedItem = lstScores.SelectedItem.ToString();
        int id = int.Parse(selectedItem.Split('.')[0]);
        dbHelper.DeleteScore(id);
        LoadData();
    }
}
```

Kesimpulan

Aplikasi *MathRush* merupakan contoh nyata dari penerapan paradigma *event-driven programming*, di mana setiap interaksi pengguna seperti menekan tombol, memasukkan jawaban, atau memilih data memicu suatu aksi spesifik dalam sistem. Seluruh logika alur program dirancang untuk merespons berbagai event ini melalui metode-metode handler seperti *btnStart_Click*, *btnSubmit_Click*, dan *btnAdd_Click*.

Penerapan event handler tersebut memungkinkan aplikasi untuk:

- Memberikan respons instan terhadap input pengguna,
- Mengelola alur permainan secara dinamis (misalnya, memulai permainan dan menghitung skor),
- Menangani interaksi dengan database untuk menyimpan, memperbarui, dan menghapus data.