NAMA : CANDRA SURYA S (2213020107) 3E

Event-Driven dalam Page CRUD Staking Crypto Berbasis WinForms

Pendahuluan

Aplikasi ini di buat menggunakan pendekatan event-driven programming, yang merupakan paradigma utama dalam pengembangan antarmuka pengguna (Graphical User Interface atau GUI). Dalam pendekatan ini, alur program tidak berjalan secara linear, tetapi merespons berbagai peristiwa (event) seperti klik tombol, perubahan input, pemilihan data, dan lain sebagainya.

Pada aplikasi ini, pengguna dapat melakukan operasi Create, Read, Update, dan Delete (CRUD) terhadap data staking coin crypto dengan mengisi form dan berinteraksi langsung dengan tombol dan elemen visual lainnya. Semua interaksi tersebut dikendalikan oleh event handler, yang secara otomatis dijalankan saat sebuah event terjadi.

Konsep Event-Driven Programming

Setiap komponen dalam aplikasi, seperti tombol (Button), input teks (TextBox), atau tabel (DataGridView), memiliki event-event tertentu yang dapat ditangani melalui *event handler*. Misalnya, saat tombol "Add" diklik, akan terjadi event “btnAdd_Click” yang kemudian memicu serangkaian aksi seperti validasi input dan eksekusi perintah SQL ke database.

Contoh Event dan Implementasinya

Berikut adalah beberapa event yang diterapkan dalam aplikasi beserta penjelasan:

1. **Event: Klik Tombol “Add” (btnAdd_Click)**

Event ini menangani proses penambahan data staking ke dalam database. Aksi ini termasuk validasi input, eksekusi perintah INSERT, dan refresh tampilan DataGridView

```
1 reference
private void btnAdd_Click(object sender, EventArgs e)
{
    string coin = cmbCoinName.SelectedItem.ToString();

    if (!decimal.TryParse(txtAmount.Text, out decimal amount))
    {
        MessageBox.Show("Fill Number", "Wrong Input", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    using (var conn = new MySqlConnection(connStr))
    {
        conn.Open();
        var cmd = new MySqlCommand("INSERT INTO liquidity (coin_name, amount) VALUES (@coin, @amount)", conn);
        cmd.Parameters.AddWithValue("@coin", coin);
        cmd.Parameters.AddWithValue("@amount", amount);
        cmd.ExecuteNonQuery();
    }

    LoadData();
    txtAmount.Clear();
    cmbCoinName.SelectedIndex = 0;

    MessageBox.Show("Staking Added Succesfully.", "Succes", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
```

Penjelasan :

- **btnAdd_Click** adalah event handler yang dipicu ketika tombol "Add" ditekan oleh pengguna.

- **cmbCoinName.Selected**

mengambil nam coin dari Dropdown Combobox yang dipilih user.

- **IF** Validasi input: memastikan bahwa nilai yang dimasukkan di txtAmount dapat dikonversi ke decimal.Jika gagal (input bukan angka), pengguna akan mendapatkan peringatan dan proses dibatalkan.

- **USING** Membuka koneksi ke database menggunakan string koneksi connStr, Menyusun perintah **SQL INSERT** untuk menambahkan data ke tabel liquidity dan Parameterisasi digunakan (@coin, @amount) untuk mencegah SQL Injection.
- Menyegarkan tabel (dengan **LoadData())** agar data baru langsung muncul.
- **TxtAmount** Membersihkan form input agar siap untuk entri berikutnya.

2. **Event: btnEdit_Click – Mengedit Data yang Dipilih**

Event ini digunakan untuk memperbarui data staking yang telah dipilih sebelumnya.

```
1 reference
private void btnEdit_Click(object sender, EventArgs e)
{
    if (selectedId == -1)
    {
        MessageBox.Show("Select data before edit.", "Alert", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        return;
    }

    if (!decimal.TryParse(txtAmount.Text, out decimal amount))
    {
        MessageBox.Show("Fill Number", "Wrong Input", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    using (var conn = new MySqlConnection(connStr))
    {
        conn.Open();
        var cmd = new MySqlCommand("UPDATE liquidity SET coin_name = @coin, amount = @amount WHERE id = @id", conn);
        cmd.Parameters.AddWithValue("@coin", cmbCoinName.SelectedItem.ToString());
        cmd.Parameters.AddWithValue("@amount", amount);
        cmd.Parameters.AddWithValue("@id", selectedId);
        cmd.ExecuteNonQuery();
    }

    LoadData();
    txtAmount.Clear();
    cmbCoinName.SelectedIndex = 0;
    selectedId = -1;

    MessageBox.Show("Staking Update Succesfully", "Succes", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
```

Penjelasan :

- **selectedId** menyimpan ID dari data yang dipilih pengguna di tabel. Jika tidak ada data yang dipilih (nilai -1), edit tidak boleh dilakukan.
- **IF** Validasi lagi terhadap input angka seperti saat proses tambah.
- **USING** Membuka koneksi ke database, lalu menjalankan SQL UPDATE berdasarkan id data yang dipilih kemudian Data coin dan amount baru dikirim dari form.
- **Load data()** Mengembalikan form ke keadaan kosong dan tidak memilih apa pun lagi.

3. **Event: Klik pada DataGridView (dgvLiquidity_CellClick)**
Saat pengguna memilih baris dalam tabel, event ini akan menampilkan data tersebut ke input form untuk proses edit atau delete.

```
1 reference
private void dgvLiquidity_CellClick(object sender, DataGridViewCellEventArgs e)
{
    if (e.RowIndex >= 0)
    {
        selectedId = Convert.ToInt32(dgvLiquidity.Rows[e.RowIndex].Cells["id"].Value);
        cmbCoinName.SelectedItem = dgvLiquidity.Rows[e.RowIndex].Cells["Coin"].Value.ToString();
        txtAmount.Text = dgvLiquidity.Rows[e.RowIndex].Cells["Amount"].Value.ToString();
    }
}
```

- Event ini dipicu saat pengguna mengklik sel di dalam DataGridView.
- **selectedId** akan menyimpan id untuk referensi saat edit/hapus.
- Nilai coin dan amount dari baris yang diklik akan otomatis di *set* ke dalam form.

NAMA : CANDRA SURYA S (2213020107) 3E

4. **Event: Klik Tombol “Withdraw/Delete” (btnDelete_Click)**
   Event ini digunakan untuk menghapus data staking.

```
1 reference
private void btnDelete_Click(object sender, EventArgs e)
{
    if (selectedId == -1)
    {
        MessageBox.Show("Select Data Before Edit.", "Alert", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        return;
    }

    var confirm = MessageBox.Show("Withdraw?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (confirm != DialogResult.Yes) return;

    using (var conn = new MySqlConnection(connStr))
    {
        conn.Open();
        var cmd = new MySqlCommand("DELETE FROM liquidity WHERE id = @id", conn);
        cmd.Parameters.AddWithValue("@id", selectedId);
        cmd.ExecuteNonQuery();
    }

    LoadData();
    txtAmount.Clear();
    cmbCoinName.SelectedIndex = 0;
    selectedId = -1;

    MessageBox.Show("Withdraw Succesfully.", "Succes", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
```

Penjelasan :

- **If selected** Validasi bahwa pengguna telah memilih data untuk dihapus**.**
- **Confirm** Menampilkan kotak dialog konfirmasi sebelum menjalankan penghapusan dan jika pengguna menekan “No”, proses berhenti.
- **USING** Menjalankan SQL DELETE berdasarkan ID data yang dipilih.
- **LoadData()** Membersihkan form dan menyegarkan tabel agar data yang sudah dihapus tidak muncul lagi.

5. **Event: Validasi Input Angka (txtAmount_TextChanged)**
   Event ini terjadi saat pengguna mengetik di TextBox. Validasi dilakukan secara real-time.

```
1 reference
private void txtAmount_TextChanged(object sender, EventArgs e)
{
    if (!decimal.TryParse(txtAmount.Text, out _) && !string.IsNullOrWhiteSpace(txtAmount.Text))
    {
        txtAmount.BackColor = Color.MistyRose;
    }
    else
    {
        txtAmount.BackColor = Color.SkyBlue;
    }
}
```

Penjelasan :

- Event ini akan aktif setiap kali pengguna mengetik di dalam textbox jumlah.
- Jika input bukan angka, maka warna background akan menjadi merah muda sebagai peringatan.
- Jika valid, warnanya kembali ke biru langit.