**Event-Driven Programming dalam Aplikasi**

**Toko Parfum**

* **Pendahuluan**

Aplikasi toko parfum ini dikembangkan dengan pendekatan event-driven programming untuk mendukung operasional administrasi toko secara efisien. Dengan paradigma ini, setiap interaksi pengguna, seperti login admin, menambahkan produk, mengedit data, atau menghapus item, dipicu oleh suatu kejadian (event) yang kemudian ditangani oleh sistem secara responsif.

* **Alur aplikasi**

1. **Login sebagai admin**
   * Masuk menggunakan akun yang sudah terdaftar.
2. **Menambahkan Parfum**
   * Setelah login, dapat melakukan penambahan parfum
3. **Update Parfum**
   * Kita juga dapat mengupdate parfum yang telah ditambahkan sebelumnya.
4. **Hapus Parfum**
   * Kita juga dapat menghapus daftar parfum

* **Struktur**

1. **FormLogin**

Event-driven programming digunakan untuk menangani interaksi pengguna, yaitu ketika tombol "Login" diklik.

**A. Event Source:**

Sumber event: Pengguna mengklik tombol "Login" setelah memasukkan username dan password.

Interaksi lain: Input pengguna pada textbox **txtUsername** dan **txtPassword**.

**B. Event Listener:**

Tombol "Login" memiliki event listener untuk event **Click**. Ini ditangani oleh metode **btnLogin\_Click**.

**C. Event Handler:**

Metode **btnLogin\_Click** adalah event handler yang dieksekusi ketika tombol "Login" diklik. Dalam metode ini:

* + Ambil nilai dari textbox **txtUsername** dan **txtPassword**.
  + Validasi kredensial admin dengan query ke database MySQL.
  + Jika valid, buka form dashboard admin (**AdminDashboard**) dan sembunyikan form login.
  + Jika tidak valid, tampilkan pesan kesalahan.

1. **CRUD**

* **A. Event Source:** Sumber event adalah interaksi pengguna, seperti klik tombol atau pemilihan baris di DataGridView.
* **B. Event Listener:** Tombol-tombol memiliki listener untuk event Click, yang memicu eksekusi metode tertentu.
* **C. Event Handler:** Metode-metode seperti btnAdd\_Click, btnUpdate\_Click, dan btnDelete\_Click adalah handler yang menangani logika bisnis terkait operasi CRUD. Berikut penjelasannya.

**a. Create (Tambah Parfum)**

**Event Source:**

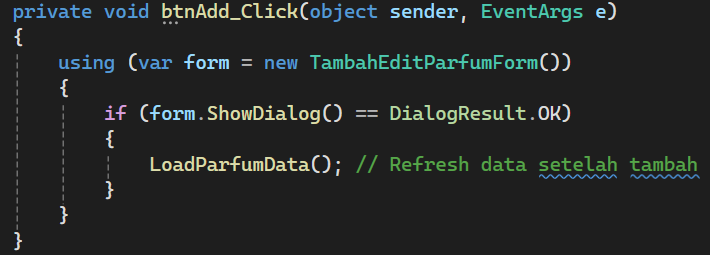
* Sumber Event: Pengguna mengklik tombol "Tambah" (**btnAdd**).

**Event Listener:**

* Tombol "Tambah" memiliki listener untuk event **Click**. Ini diimplementasikan melalui binding antara tombol dan metode **btnAdd\_Click**.

**Event Handler:**

* Metode **btnAdd\_Click** adalah handler yang dieksekusi ketika tombol "Tambah" diklik. Berikut adalah langkah-langkahnya:
  1. Buka Form Tambah Parfum:
     + Form **TambahEditParfumForm** dibuka dalam mode dialog menggunakan **ShowDialog()**.
  2. Refresh Data:
     + Jika pengguna menyelesaikan proses tambah dengan menekan tombol "OK", data di-refresh menggunakan **LoadParfumData()**.



**2. Read (Menampilkan Data Parfum)**

**Event Source:**

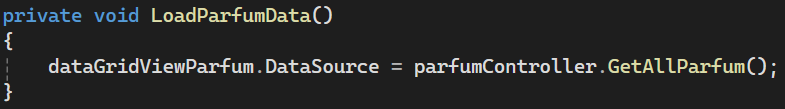
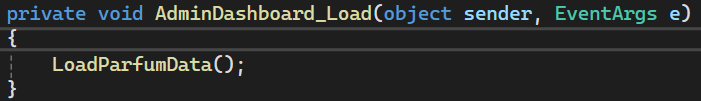
* Sumber Event: Form **AdminDashboard** dimuat atau data perlu di-refresh.

**Event Listener:**

* Event **Load** pada form mendengarkan event saat form pertama kali dibuka. Ini diimplementasikan melalui metode **AdminDashboard\_Load**.

**Event Handler:**

* Metode **AdminDashboard\_Load** adalah handler yang dieksekusi ketika form dimuat. Berikut adalah langkah-langkahnya:
  1. Muat Data dari Database:
     + Data parfum diambil dari database menggunakan **parfumController.GetAllParfum()**.
  2. Tampilkan Data di DataGridView:
     + Data diset ke properti **DataSource** dari **dataGridViewParfum**.



**3. Update (Edit Parfum)**

**Event Source:**

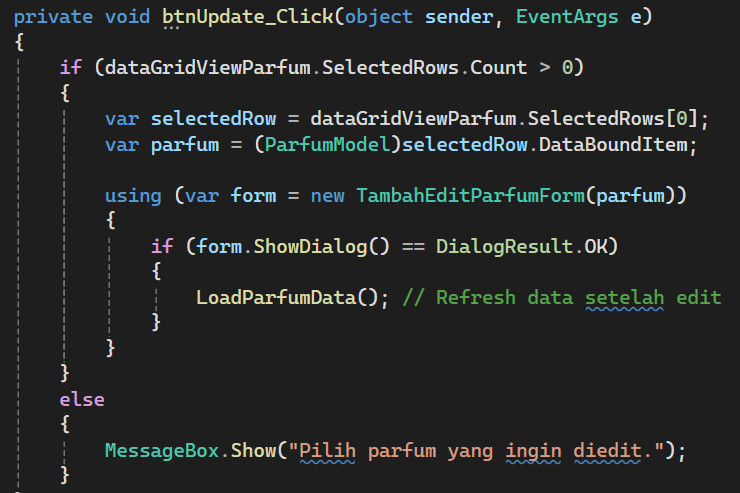
* Sumber Event: Pengguna memilih baris tertentu di **DataGridView** lalu mengklik tombol "Edit" (**btnUpdate**).

**Event Listener:**

* Tombol "Edit" memiliki listener untuk event **Click**. Ini diimplementasikan melalui binding antara tombol dan metode **btnUpdate\_Click**.

**Event Handler:**

* Metode **btnUpdate\_Click** adalah handler yang dieksekusi ketika tombol "Edit" diklik. Berikut adalah langkah-langkahnya:
  1. Periksa Baris yang Dipilih:
     + Pastikan ada baris yang dipilih di **DataGridView**.
  2. Ambil Data Parfum yang Dipilih:
     + Ambil objek **ParfumModel** dari baris yang dipilih.
  3. Buka Form Edit Parfum:
     + Form **TambahEditParfumForm** dibuka dalam mode dialog dengan data parfum yang dipilih.
  4. Refresh Data:
     + Jika pengguna menyelesaikan proses edit dengan menekan tombol "OK", data di-refresh menggunakan **LoadParfumData()**.



**4. Delete (Hapus Parfum)**

**Event Source:**

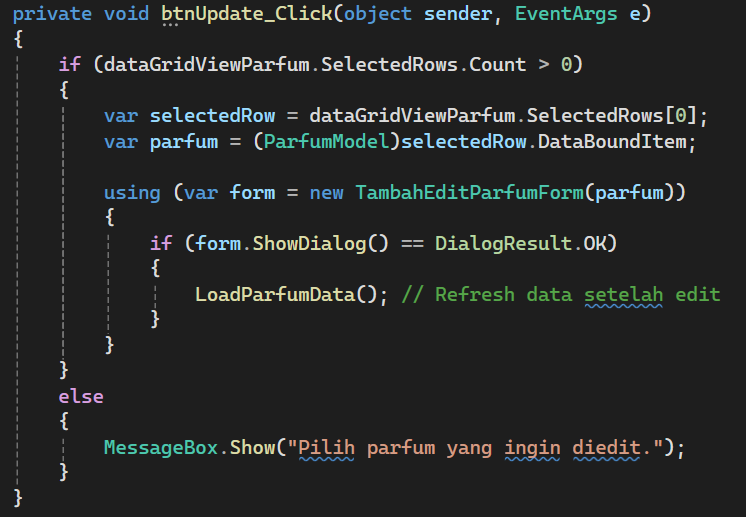
* Sumber Event: Pengguna memilih baris tertentu di **DataGridView** lalu mengklik tombol "Hapus" (**btnDelete**).

**Event Listener:**

* Tombol "Hapus" memiliki listener untuk event **Click**. Ini diimplementasikan melalui binding antara tombol dan metode **btnDelete\_Click**.

**Event Handler:**

* Metode **btnDelete\_Click** adalah handler yang dieksekusi ketika tombol "Hapus" diklik. Berikut adalah langkah-langkahnya:
  1. Periksa Baris yang Dipilih:
     + Pastikan ada baris yang dipilih di **DataGridView**.
  2. Konfirmasi Penghapusan:
     + Tampilkan konfirmasi kepada pengguna sebelum menghapus.
  3. Hapus Data dari Database:
     + Panggil metode **parfumController.DeleteParfum()** untuk menghapus data dari database.
  4. Refresh Data:
     + Data di-refresh menggunakan **LoadParfumData()**.



* **Penutup**

Aplikasi toko parfum dengan pendekatan event-driven programming telah berhasil menyediakan solusi yang responsif dan modular untuk mendukung manajemen toko.

Dengan memanfaatkan prinsip event-driven, aplikasi ini mampu memberikan respons cepat terhadap setiap interaksi pengguna, mulai dari login admin hingga manajemen produk seperti penambahan, pengeditan, dan penghapusan data. Pendekatan ini tidak hanya meningkatkan efisiensi kerja tetapi juga menciptakan pengalaman pengguna yang lebih intuitif dan dinamis.