

**Nama** : Krisna Dzakyya Darajat

**NPM** : 2113020029

**Kelas** : 3B

**Mata Kuliah** : Pemrograman Visual

### **Event Driven Program di DuckFarmApps**

DuckFarmApps adalah aplikasi saya buat untuk memenuhi kebutuhan produktivitas para peternak. Mulai dari tombol untuk mengarahkan ke halaman menghitung bobot panen secara live(saat itu juga), menghitung FCR(on progress), maupun manajemen data karyawan ABK(Anak Buah Kandang).

#### **Halaman1.cs**

Halaman ini merupakan halaman utama dari sebuah aplikasi bernama DuckFarmApps, aplikasi saya buat untuk memenuhi kebutuhan produktivitas para peternak. Mulai dari tombol untuk mengarahkan ke halaman menghitung bobot panen secara live(saat itu juga), menghitung FCR(on progress), maupun manajemen data karyawan ABK(Anak Buah Kandang).

Pada halaman ini terdapat 4 tombol. Yaitu:

1. SQLConnect, tombol ini digunakan untuk mengecek apakah aplikasi sudah terhubung MySQL atau belum, notifnya ditandai dengan Message Box bertuliskan Data: admin.
2. Hitung Bobot, tombol ini digunakan untuk mengarahkan kita ke halaman menghitung bobot bebek. Saat tombol HitungBobot(button2\_Click\_1) diklik, maka user akan diarahkan menuju halaman2.cs.
3. Hitung FCR, tombol ini digunakan untuk mengarahkan kita ke halaman menghitung bobot bebek. Saat tombol HitungFCR(button1\_Click) diklik, maka user akan diarahkan menuju halaman3.cs.
4. Data Karyawan, tombol ini digunakan untuk mengarahkan kita ke halaman manajemen data karyawan dengan metode Create, Update, & Delete(CRUD). Saat tombol Data Karyawan(button4\_Click) diklik, maka user akan diarahkan menuju FormDataKaryawan.cs.

#### **Halaman2.cs**

Halaman ini merupakan halaman ke 2 yaitu Hitung Bobot, halaman ini digunakan untuk menghitung Bobot ternak secara live dengan pengambilan per karung, pengambilannya bisa 12 ekor per karung sampai unggas habis di panen. Halaman hitung bobot pada aplikasi peternak bebek pedaging sangat dibutuhkan karena memungkinkan peternak untuk memantau dan mengelola pertumbuhan bebek secara akurat. Dengan fitur ini, peternak dapat memasukkan data bobot bebek secara teratur dan mendapatkan informasi tentang perkembangan berat badan bebek.

Beberapa contoh event yang ditangani:

1. button1\_Click: event yang terjadi ketika tombol pertama diklik, yang membuka form Hitung\_FCR.
2. button2\_Click\_1: event yang terjadi ketika tombol kedua diklik, yang membuka form HitungBobot.
3. button3\_Click: event yang terjadi ketika tombol ketiga diklik, yang memanggil method CekKoneksiDatabase() untuk memeriksa koneksi database.

4. `button4_Click`: event yang terjadi ketika tombol keempat diklik, yang membuka form `FormDataKaryawan`.

Dalam setiap event handler, kode program melakukan aksi yang sesuai, seperti membuka form lain, memeriksa koneksi database, atau menampilkan data.

Perlu diperhatikan bahwa ada beberapa event handler yang kosong, seperti `label1_Click`, `label2_Click`, dan `label3_Click`, yang tidak melakukan apa-apa ketika label-label tersebut diklik. Selain itu, ada juga event handler `button2_Click` yang kosong dan tidak digunakan, sedangkan `button2_Click_1` digunakan sebagai event handler untuk tombol kedua.

#### Halaman3.cs

Halaman Food Conversion Ratio (FCR) pada aplikasi desktop umumnya dirancang untuk memberikan kemudahan bagi pengguna dalam mengelola dan menganalisis data efisiensi pakan. Halaman ini biasanya menyertakan fitur input data, di mana pengguna dapat memasukkan jumlah total pakan yang digunakan dan berat biomassa yang dihasilkan, seperti daging atau ikan. Setelah data diinput, aplikasi akan menghitung FCR secara otomatis dan menyajikan hasilnya dalam bentuk angka yang mudah dipahami.

Penerapan Food Conversion Ratio (FCR) dalam aplikasi desktop memiliki berbagai fungsi yang bermanfaat, terutama dalam bidang agrikultur, akuakultur, dan peternakan. Aplikasi ini memungkinkan pengguna untuk memantau efisiensi penggunaan pakan dengan menghitung FCR secara otomatis melalui data yang dimasukkan, seperti jumlah pakan yang digunakan dan berat biomassa yang dihasilkan. Proses perhitungan yang terotomatisasi tidak hanya mengurangi risiko kesalahan, tetapi juga menghemat waktu. Lebih jauh lagi, aplikasi ini mendukung upaya keberlanjutan dengan membantu pengguna mengurangi pemborosan pakan, menjadikannya alat yang penting untuk operasional yang lebih efisien.

Food Conversion Ratio (FCR) adalah ukuran efisiensi dalam mengubah pakan menjadi biomassa, seperti daging atau ikan. FCR dihitung dengan membandingkan berat total pakan yang diberikan selama periode tertentu dengan berat total hasil biomassa yang dihasilkan.

Sebagai contoh, jika Anda memberikan 200 kg pakan dan menghasilkan 150 kg ikan, maka FCR-nya adalah  $200/150 = 1,33$ . Semakin kecil nilai FCR, semakin efisien prosesnya, yang berarti lebih sedikit pakan yang diperlukan untuk menghasilkan satu kilogram biomassa.

FCR sering digunakan dalam budidaya ikan dan ternak untuk mengukur keberhasilan teknis dan finansial.

#### Fitur Aplikasi:

1. Form dengan beberapa kontrol: Aplikasi ini memiliki form dengan beberapa kontrol, seperti tombol (button), label, dan lain-lain.
2. Event handler: Kode ini memiliki beberapa event handler untuk menangani event yang terjadi pada kontrol, seperti klik tombol atau label.

#### Fungsi yang Sudah Diimplementasikan:

1. Membuka form lain: Pada event handler `button5_Click`, kode ini membuka form lain (`Form1`) dan menampilkan form tersebut. Selain itu, kode ini juga menutup form saat ini (`this.Close()`).

#### Fungsi yang Belum Diimplementasikan:

1. Tombol lain: Beberapa tombol lain, seperti `button1`, `button6`, `button7`, `button8`, `button9`, dan `button10`, belum memiliki kode yang diimplementasikan untuk menangani event klik.
2. Label: Beberapa label, seperti `label2`, `label3`, `label10`, `label11`, dan `label12`, memiliki event handler untuk menangani event klik, tetapi kode di dalamnya kosong.

#### FormDataKaryawan.cs

Fungsi CRUD pada FormDataKaryawan merupakan operasi dasar yang digunakan untuk mengelola data karyawan dalam database. Dengan menggunakan fungsi ini, pengguna dapat melakukan berbagai aksi pada data karyawan, seperti menambah data baru, mengambil data yang sudah ada, mengubah data yang sudah ada, dan menghapus data yang tidak diperlukan lagi.

Fungsi Create memungkinkan pengguna untuk menambah data karyawan baru ke dalam database dengan mengisi formulir pendaftaran karyawan baru. Fungsi Read memungkinkan pengguna untuk mengambil data karyawan dari database dan menampilkannya dalam bentuk tabel atau daftar. Fungsi Update memungkinkan pengguna untuk mengubah data karyawan yang sudah ada dalam database, seperti memperbarui informasi alamat atau nomor telepon. Fungsi Delete memungkinkan pengguna untuk menghapus data karyawan dari database yang sudah tidak diperlukan lagi.

Dengan menggunakan fungsi CRUD, FormDataKaryawan dapat mengelola data karyawan dengan efektif dan efisien. Pengguna dapat dengan mudah mengakses dan mengubah data karyawan, sehingga data dapat tetap akurat dan konsisten. Selain itu, fungsi CRUD juga membantu meminimalisir kesalahan dalam pengolahan data dan meningkatkan keamanan data karyawan.

Kode ini adalah contoh aplikasi Windows Forms menggunakan C# yang terhubung dengan database MySQL. Untuk penjelasan lengkap dan detail tentang kode ini, adalah sebagai berikut:

#### **Fungsi Aplikasi**

Aplikasi ini memiliki beberapa fungsi utama, yaitu:

1. Menampilkan data karyawan: Aplikasi ini dapat menampilkan data karyawan dari database MySQL dalam bentuk tabel.
2. Menambah data karyawan: Aplikasi ini dapat menambah data karyawan baru ke database MySQL.
3. Mengedit data karyawan: Aplikasi ini dapat mengedit data karyawan yang sudah ada di database MySQL.
4. Menghapus data karyawan: Aplikasi ini dapat menghapus data karyawan dari database MySQL.

#### **Komponen Aplikasi**

Aplikasi ini memiliki beberapa komponen utama, yaitu:

1. Form: Aplikasi ini memiliki satu form utama yang menampilkan data karyawan dan beberapa tombol untuk melakukan aksi tertentu.
2. DataGridView: Aplikasi ini menggunakan DataGridView untuk menampilkan data karyawan dalam bentuk tabel.
3. Tombol: Aplikasi ini memiliki beberapa tombol untuk melakukan aksi tertentu, seperti menambah, mengedit, dan menghapus data karyawan.

#### **Koneksi Database**

Aplikasi ini menggunakan MySQL Connector/NET untuk terhubung dengan database MySQL. Koneksi database diatur dalam variabel `connection` dengan string koneksi yang berisi informasi tentang server, database, username, dan password.

#### **Metode yang Digunakan**

Aplikasi ini memiliki beberapa metode yang digunakan untuk melakukan operasi tertentu, yaitu:

1. ``getDataKaryawan()``: Metode ini digunakan untuk mengambil data karyawan dari database MySQL dan mengembalikan data dalam bentuk DataTable.
2. ``fillDataKaryawan()``: Metode ini digunakan untuk mengisi DataGridView dengan data karyawan yang diambil dari database MySQL.
3. ``btnSimpan_Click()``: Metode ini digunakan untuk menangani event klik tombol "Simpan" dan menambah data karyawan baru ke database MySQL.
4. ``dataGridViewABK_CellContentClick()``: Metode ini digunakan untuk menangani event klik cell di DataGridView dan melakukan aksi tertentu berdasarkan kolom yang diklik.
5. ``btnUpdate_Click()``: Metode ini digunakan untuk menangani event klik tombol "Update" dan mengedit data karyawan yang sudah ada di database MySQL.

### **Penanganan Error**

Aplikasi ini menggunakan try-catch block untuk menangani error yang mungkin terjadi saat melakukan operasi database. Jika error terjadi, aplikasi akan menampilkan pesan error kepada pengguna.

### **Kelebihan dan Kekurangan**

Kelebihan aplikasi ini adalah:

1. Mudah digunakan: Aplikasi ini memiliki antarmuka yang sederhana dan mudah digunakan.
2. Fleksibel: Aplikasi ini dapat melakukan operasi CRUD (Create, Read, Update, Delete) pada data karyawan.

### **Kekurangan aplikasi ini adalah:**

1. Tidak ada validasi data: Aplikasi ini tidak memiliki validasi data yang baik, sehingga pengguna dapat memasukkan data yang tidak valid.
2. Tidak ada keamanan: Aplikasi ini tidak memiliki keamanan yang baik, sehingga data karyawan dapat diakses oleh orang yang tidak berwenang.

### **DatabaseConnection.cs**

Kode ini adalah contoh kelas DatabaseConnection yang digunakan untuk mengelola koneksi ke database MySQL menggunakan C#. Berikut adalah penjelasan lengkap dan detail tentang kode ini:

### **Fungsi Kelas**

Kelas DatabaseConnection memiliki beberapa fungsi utama, yaitu:

1. Mengelola koneksi database: Kelas ini digunakan untuk mengelola koneksi ke database MySQL, termasuk membuka dan menutup koneksi.
2. Menyediakan koneksi yang stabil: Kelas ini memastikan bahwa koneksi ke database selalu stabil dan dapat digunakan oleh aplikasi.

### **Properti dan Variabel**

Kelas DatabaseConnection memiliki beberapa properti dan variabel, yaitu:

1. `connectionString`: Variabel ini digunakan untuk menyimpan string koneksi ke database MySQL.

2. connection: Variabel ini digunakan untuk menyimpan objek MySqlConnection yang digunakan untuk terhubung ke database.

### **Metode**

Kelas DatabaseConnection memiliki beberapa metode, yaitu:

1. GetConnection(): Metode ini digunakan untuk membuka koneksi ke database MySQL jika koneksi belum terbuka. Jika koneksi sudah terbuka, metode ini akan mengembalikan objek koneksi yang sudah terbuka.
2. CloseConnection(): Metode ini digunakan untuk menutup koneksi ke database MySQL jika koneksi masih terbuka.

### **Cara Kerja**

Berikut adalah cara kerja kelas DatabaseConnection:

1. Ketika objek DatabaseConnection dibuat, objek MySqlConnection dibuat dengan string koneksi yang telah ditentukan.
2. Ketika metode GetConnection() dipanggil, koneksi ke database MySQL akan dibuka jika koneksi belum terbuka.
3. Ketika metode CloseConnection() dipanggil, koneksi ke database MySQL akan ditutup jika koneksi masih terbuka.

### **Kelebihan**

Kelebihan kelas DatabaseConnection adalah:

1. Mudah digunakan: Kelas ini menyediakan metode yang sederhana untuk mengelola koneksi ke database MySQL.
2. Stabil: Kelas ini memastikan bahwa koneksi ke database selalu stabil dan dapat digunakan oleh aplikasi.

### **Kekurangan**

Kekurangan kelas DatabaseConnection adalah:

1. Tidak ada penanganan error: Kelas ini tidak memiliki penanganan error yang baik, sehingga aplikasi mungkin akan mengalami error jika terjadi masalah dengan koneksi database.
2. Tidak ada keamanan: Kelas ini tidak memiliki keamanan yang baik, sehingga string koneksi dapat diakses oleh orang yang tidak berwenang.

### **Contoh Penggunaan**

Berikut adalah contoh penggunaan kelas DatabaseConnection:

```
DatabaseConnection db = new DatabaseConnection();
MySqlConnection connection = db.GetConnection();
// Gunakan koneksi untuk melakukan operasi database
db.CloseConnection();
```