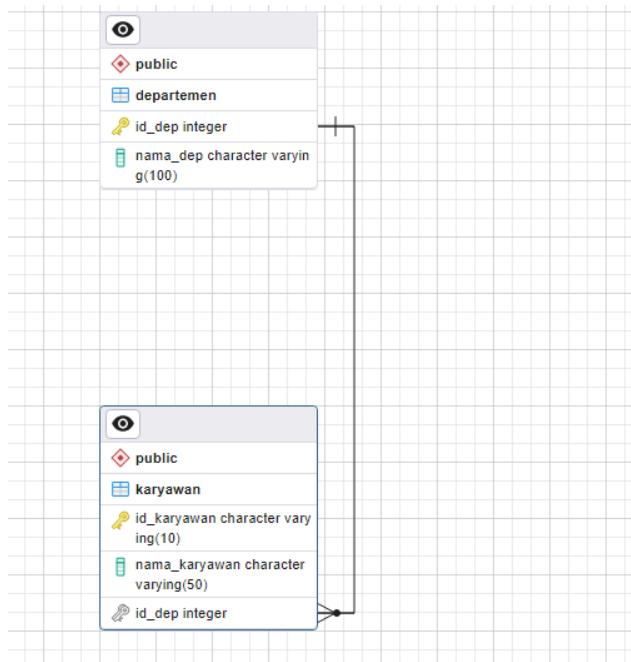
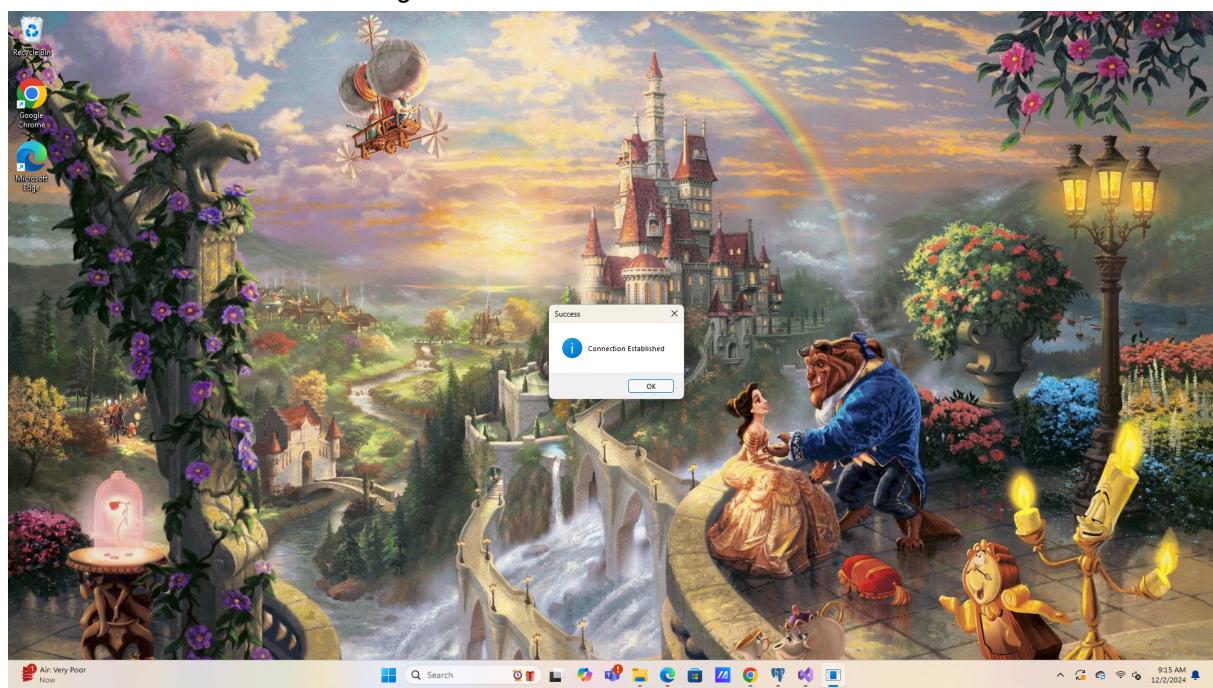


A. ERD Database “fikaresponsi”



B. Operasi dalam Aplikasi

Inisiasi keberhasilan connect dengan database local:



- Load:

Form1



Nama Karyawan:

Dep. Karyawan:

ID Departemen

HR: HR
ENG: Engineer
DEV: Developer
PM: Product Manager
FIN: Finance

Insert **Update** **Delete**

	ID Karyawan	Nama	ID Departemen	Nama Departemen
▶	3	Satama Safika	3	DEV
	2	Danella Zefanya Siahaan	5	FIN
	5	Danu Firda	1	HR
	6	Sastraa Wintana	2	ENG

Load

- **Insert:**

Form1



Nama Karyawan:

Dep. Karyawan:

ID Departemen

HR: HR
ENG: Engineer
DEV: Developer
PM: Product Manager
FIN: Finance

Insert

Sukses

Data berhasil ditambahkan!

OK

	ID Karyawan	Nama	ID Departemen	Nama Departemen
▶	3	Satama Safika	3	DEV
	2	Danella Zefanya Siahaan	5	FIN
	5	Danu Firda	1	HR
	6	Sastraa Wintana	2	ENG

Load

Load setelah Insert:

Form1



Nama Karyawan:	<input type="text"/>	ID Departemen
Dep. Karyawan:	<input type="text"/>	HR: HR ENG: Engineer DEV: Developer PM: Product Manager FIN: Finance

Insert Update Delete

	ID Karyawan	Nama	ID Departemen	Nama Departemen
5	Danu Firda	1	HR	
6	Sastrawintana	2	ENG	
7	Faiz Rahmawan	2	ENG	
*				

Load

- Update:

Form1



Nama Karyawan:	<input type="text" value="Satama Safika"/>	ID Departemen
Dep. Karyawan:	<input type="text" value="FIN"/>	HR: HR ENG: Engineer DEV: Developer PM: Product Manager FIN: Finance

Well Done!

Data Karyawan Berhasil Diperbarui

OK

Insert Delete

	ID Karyawan	Nama	ID Departemen	Nama Departemen
3	Satama Safika	3	DEV	
2	Danella Zefanya Sianaan	2	FIN	
5	Danu Firda	1	HR	
6	Sastrawintana	2	ENG	
*				

Load

Load setelah Update:

Form1



Nama Karyawan:

Dep. Karyawan:

ID Departemen

- HR: HR
- ENG: Engineer
- DEV: Developer
- PM: Product Manager
- FIN: Finance

Insert **Update** **Delete**

	ID Karyawan	Nama	ID Departemen	Nama Departemen
6	Sastrawintana	2	ENG	
7	Faiz Rahmawan	2	ENG	
3	Satama Safika	5	FIN	
*				

Load

- Delete:

Form1



Nama Karyawan:

Dep. Karyawan:

ID Departemen

- HR: HR
- ENG: Engineer
- DEV: Developer
- PM: Product Manager
- FIN: Finance

Insert **Update** **Delete**

	ID Karyawan	Nama	ID Departemen	Nama Departemen
7	Faiz Rahmawan	2	ENG	
3	Satama Safika	5	FIN	
5	Danu Firda	1	HR	
*				

Well Done!

Data Karyawan Berhasil Dihapus

OK

Load

Load setelah Delete:

ID Karyawan	Nama	ID Departemen	Nama Departemen
6	Sastraa Wintana	2	ENG
7	Faiz Rahmawan	2	ENG
3	Satama Safika	5	FIN
*			

C. Konsep OOP dalam Aplikasi

- **Encapsulation**

Encapsulation adalah konsep membungkus data (atribut) dan metode (fungsi) dalam satu unit, yaitu kelas. Dalam kode ini, beberapa contoh encapsulation adalah:

1. **Pembungkusan data dan metode dalam kelas**

- Data dan metode didefinisikan dalam kelas Form1, seperti properti conn, sql, dan metode ConnectToDatabase, LoadDepartments, dan lainnya.
- **Access modifier** seperti private digunakan untuk membatasi akses ke atribut/method tertentu. Misalnya, properti conn, connstring, cmd, dan sql dibuat **private**, sehingga hanya dapat diakses di dalam kelas Form1.

2. **Getter dan Setter (secara implisit)**

- Properti cbDepartemen.DataSource, cbDepartemen.DisplayMember, dan cbDepartemen.ValueMember adalah contoh penggunaan **encapsulation** pada kontrol Windows Forms. Properti ini dikontrol menggunakan mekanisme getter dan setter bawaan framework.

3. **Validasi input sebelum operasi**

- Validasi dilakukan untuk memastikan data yang dimasukkan benar, seperti pada metode btnInsert_Click, di mana validasi dilakukan terhadap txtName.Text dan cbDepartemen.SelectedIndex.

- **Inheritance**

Inheritance adalah konsep di mana sebuah kelas dapat mewarisi sifat (atribut dan metode) dari kelas lain. Dalam kode ini:

1. **Kelas Form1 mewarisi dari kelas Form:**

- Form1 adalah kelas turunan dari kelas Form (sebuah kelas bawaan dari System.Windows.Forms).
- Kelas Form menyediakan semua properti dan metode dasar untuk membuat antarmuka pengguna Windows Forms, seperti Load, FormClosing, dan kontrol seperti DataGridView.

2. Pemanfaatan metode dan properti kelas induk:

- Metode seperti InitializeComponent dipanggil dari kelas Form untuk menginisialisasi komponen.

- Polymorphism

Polymorphism adalah konsep yang memungkinkan satu nama metode memiliki banyak bentuk, bergantung pada konteksnya. Dalam kode ini:

1. Event Handling

- Beberapa metode seperti btnInsert_Click, btnLoad_Click, dan dgvKaryawan_CellClick adalah metode yang diimplementasikan untuk menangani event spesifik. Meski memiliki nama yang berbeda, mereka semuanya adalah implementasi dari polimorfisme metode karena mereka berasal dari event handler yang sama (EventHandler).

2. Overriding

- Metode bawaan seperti Form1_FormClosing (dari kelas Form) di-override untuk menyediakan perilaku khusus (menutup koneksi database saat form ditutup).

- Abstraction

Abstraction adalah konsep menyembunyikan detail implementasi dan hanya menampilkan fungsionalitas penting kepada pengguna. Dalam kode ini, abstraction diterapkan dengan:

1. Pemanfaatan Framework

- Metode bawaan dari kelas Form dan DataGridView menyembunyikan detail implementasi. Contohnya, kita tidak perlu tahu bagaimana DataSource diatur secara internal, cukup dengan menyetel properti seperti DataSource, DisplayMember, dan ValueMember.

2. Keterpisahan Logika

- Fungsi ConnectToDatabase menyembunyikan detail bagaimana koneksi dibuat. Pengguna hanya tahu bahwa fungsi tersebut akan membuka koneksi ke database.
- LoadDepartments menyembunyikan bagaimana data diambil dari database dan diikat ke ComboBox.