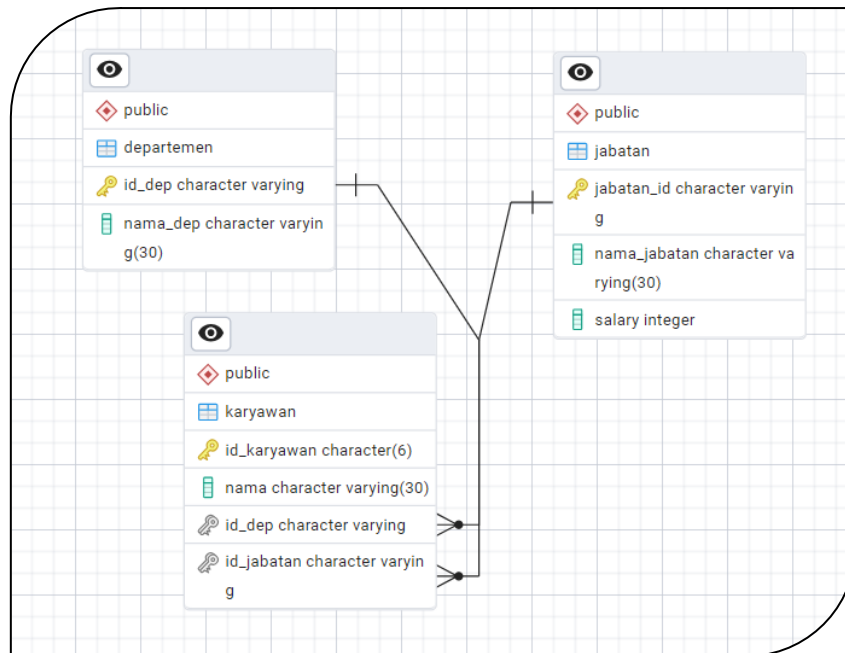


Nama : Septian Eka Rahmadi

NIM : 22/496732/TK/54442

1. ERD DIAGRAM



2. SQL Function

2 Function in SQL pgAdmin and 1 Function in Back-End C# WinForm (completed request from additional question)

a. Function SQL to a new Insert Data in pgAdmin

```
2024 07.54.30
Rows affected 35 msec
Duration
Copy Copy to Query Editor

CREATE OR REPLACE FUNCTION add_karyawan(
    in_nama text,
    in_id_dep character varying,
    in_id_jabatan character varying
)
RETURNS integer AS $$
DECLARE
    new_id_karyawan char(6);
BEGIN
    -- Membuat ID Karyawan baru menggunakan sequence
    new_id_karyawan := 'K' || nextval('users_id');

    -- Memeriksa apakah karyawan sudah ada
    IF EXISTS (
        SELECT 1 FROM karyawan
        WHERE id_karyawan = new_id_karyawan
    ) THEN
        RETURN 409; -- Karyawan sudah ada
    ELSE
        -- Menambahkan data karyawan
        INSERT INTO karyawan (id_karyawan, nama, id_dep, id_jabatan)
        VALUES (new_id_karyawan, in_nama, in_id_dep, in_id_jabatan);
        RETURN 201; -- Berhasil
    END IF;
END;
$$ LANGUAGE plpgsql;

Messages
Query returned successfully in 35 msec.
```

b. Function SQL to Delete Data in pgAdmin

```
2/2024 07.54.58
ate Rows affected 34 msec
Duration
Copy Copy to Query Editor

CREATE OR REPLACE FUNCTION delete_karyawan(
    in_id_karyawan char(6)
)
RETURNS INTEGER AS $$
BEGIN
    IF EXISTS (
        SELECT 1
        FROM karyawan
        WHERE id_karyawan = in_id_karyawan
    )
    THEN
        DELETE FROM karyawan
        WHERE id_karyawan = in_id_karyawan;
        RETURN 204;
    ELSE
        RETURN 404;
    END IF;
END;
$$ LANGUAGE plpgsql;

Messages
Query returned successfully in 34 msec.
```

c. Function SQL to Update Data in Back-End C# btnUpdate_Click

```
// Query untuk memperbarui data
string query = $"UPDATE karyawan SET nama = @nama, id_dep = @id_dep, id_jabatan = @id_jabatan WHERE id_karyawan = '{idKaryawan}'";
using (var cmd = new NpgsqlCommand(query, connection))
{
    cmd.Parameters.AddWithValue("@nama", namaBaru);
    cmd.Parameters.AddWithValue("@id_dep", idDepBaru);
    cmd.Parameters.AddWithValue("@id_jabatan", idJabatanBaru);
    cmd.Parameters.AddWithValue("@id_karyawan", idKaryawan);

    // Eksekusi query
    connection.Open();
    int rowsAffected = cmd.ExecuteNonQuery(); // Mengembalikan jumlah baris yang diubah
    connection.Close();

    // Tampilkan pesan berdasarkan hasil
    if (rowsAffected > 0)
    {
        LoadKaryawan(); // Perbarui tampilan DataGridView
        MessageBox.Show("Data karyawan berhasil diperbarui.");
    }
    else
    {
        MessageBox.Show("Data karyawan tidak ditemukan atau tidak ada perubahan.");
    }
}
```

3. Bukti Fungsionalitas Aplikasi

a. Insert Data

Responsi CRUD Database Karyawan

UGM

Nama Karyawan: Septian Eka

Nama Departemen: HR

Jabatan: Team Leader

Insert Edit Delete Reload

Keterangan ID
ID DEP:
HR: HR
ENG: Engineer
DEV: Developer
PM: Product Manager
FIN: Finance
ID JABATAN:
LD: Leader Departemen
TM: Team Leader
SF: Staff

	id_karyawan	nama	id_dep	id_jabatan
▶	K5	Marchel	DEV	SF
	K2	Eka Rahmadi	ENG	LD
	K4	Ilham	HR	TM
*				

Responsi CRUD Database Karyawan

UGM

Nama Karyawan: Edo Bagus

Nama Departemen: Developer

Jabatan: Staff

Insert Edit Delete Reload

Keterangan ID
ID DEP:
HR: HR
ENG: Engineer

Karyawan berhasil ditambahkan.

OK

	id_karyawan	nama	id_dep	id_jabatan
▶	K5	Marchel	DEV	SF
	K2	Eka Rahmadi	ENG	LD
	K4	Ilham	HR	TM
*				

b. Select Data from Data Grid View to TextBox and ComboBox

Responsi CRUD Database Karyawan

UGM

Nama Karyawan: Ilham

Nama Departemen: HR

Jabatan: Team Leader

Insert Edit Delete Reload

Keterangan ID
ID DEP:
HR: HR
ENG: Engineer
DEV: Developer
PM: Product Manager
FIN: Finance
ID JABATAN:
LD: Leader Departemen
TM: Team Leader
SF: Staff

	id_karyawan	nama	id_dep	id_jabatan
	K5	Marchel	DEV	SF
	K2	Eka Rahmadi	ENG	LD
▶	K4	Ilham	HR	TM
	K6	Edo Bagus	DEV	SF
*				

c. Update Data

Responsi CRUD Database Karyawan

UGM

Nama Karyawan:

Nama Departemen:

Jabatan:

Keterangan ID

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

ID JABATAN:
LD: Leader Departemen
TM: Team Leader
SF: Staff

	id_karyawan	nama	id_dep	id_jabatan
	K2	Eka Rahmadi	ENG	LD
	K4	Ilham	HR	TM
	K6	Edo Bagus	DEV	SF
▶	K5	Marchel	DEV	SF

Responsi CRUD Database Karyawan

UGM

Nama Karyawan:

Nama Departemen:

Jabatan:

Keterangan ID

ID DEP:
HR: HR

Data karyawan berhasil diperbarui.

	id_karyawan	nama	id_dep	id_jabatan
▶	K2	Eka Rahmadi	ENG	LD
	K4	Ilham	HR	TM
	K6	Edo Bagus	DEV	SF
	K5	Marchel untuk Tangsel	DEV	SF

Responsi CRUD Database Karyawan

UGM

Nama Karyawan:

Nama Departemen:

Jabatan:

Keterangan ID

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

ID JABATAN:
LD: Leader Departemen
TM: Team Leader
SF: Staff

	id_karyawan	nama	id_dep	id_jabatan
▶	K2	Eka Rahmadi	ENG	LD
	K4	Ilham	HR	TM
	K6	Edo Bagus	DEV	SF
	K5	Marchel untuk Tangsel	DEV	SF

d. Delete Data

Responsi CRUD Database Karyawan

UGM

Nama Karyawan: Septian Eka

Nama Departemen: HR

Jabatan: Team Leader

Buttons: Insert, Edit, Delete, Reload

	id_karyawan	nama	id_dep	id_jabatan
	K5	Marchel	DEV	SF
	K2	Eka Rahmadi	ENG	LD
	K4	Ilham	HR	TM
	K1	Septian Eka	HR	TM

Keterangan ID

ID DEP:
HR: HR

Karyawan berhasil dihapus.

OK

Responsi CRUD Database Karyawan

UGM

Nama Karyawan: Septian Eka

Nama Departemen: HR

Jabatan: Team Leader

Buttons: Insert, Edit, Delete, Reload

	id_karyawan	nama	id_dep	id_jabatan
	K5	Marchel	DEV	SF
	K2	Eka Rahmadi	ENG	LD
	K4	Ilham	HR	TM

Keterangan ID

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

ID JABATAN:
LD: Leader Departemen
TM: Team Leader
SF: Staff

4. Penerapan OOP

a. Encapsulation

Fungsi connectionString, connection, dan selectedRow hanya bisa digunakan dari dalam kelas Form1

```
// Koneksi database PostgreSQL
private string connectionString = "Host=localhost;Port=5432;Username=postgres;Password=root;Database=responsi";
private NpgsqlConnection connection; // Objek koneksi ke database
private DataGridViewRow selectedRow; // Menyimpan baris yang dipilih pada DataGridView
```

b. Abstraction

Method LoadKaryawan dan LoadDepartemen menyembunyikan logika pengambilan data dari database

```
// Konstruktor Form
public Form1()
{
    InitializeComponent();
    connection = new NpgsqlConnection(connectionString);

    // Muat data karyawan dan departemen saat form dibuka
    LoadKaryawan();
    LoadDepartemen();
    LoadJabatan();
}
```

Berikut beberapa contoh implementasinya pada btnInsert, btnUpdate, btnDelete, dan btnReload

```
// Event untuk tombol Load
private void btnLoad_Click(object sender, EventArgs e)
{
    LoadKaryawan();
}

// Perbarui data karyawan pada DataGridView
LoadKaryawan();
```

c. Inheritance

Kelas Form1 mewarisi dari Form, yang merupakan bagian dari framework Windows Forms. Ini memberikan akses ke properti dan metode standard seperti InitializeComponent, Event Handler, dan Kontrol UI / Toolbox

```
namespace Responsi_Junpro2024
{
    public partial class Form1 : Form
    {
    }
```

d. Polymorphism

Dalam kode saya tidak memiliki contoh Polymorphism secara eksplisit. Namun btnUpdate_Click dan btnInsert_Click dapat dianggap sebagai bentuk overloading dalam konteks event