Laboratorium Informatika

Junior Project (Teknik Basis Data, Pemrograman Berbasis Objek)

Worksheet Responsi

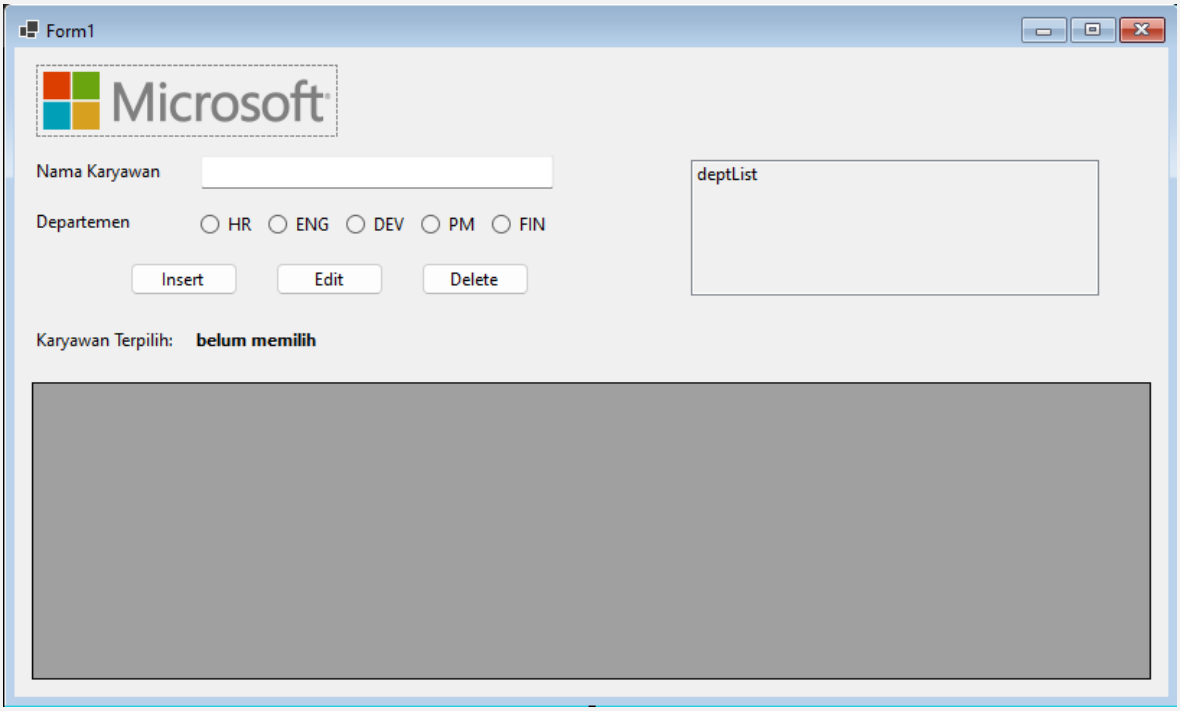
RESPONSI

TUGAS WAJIB

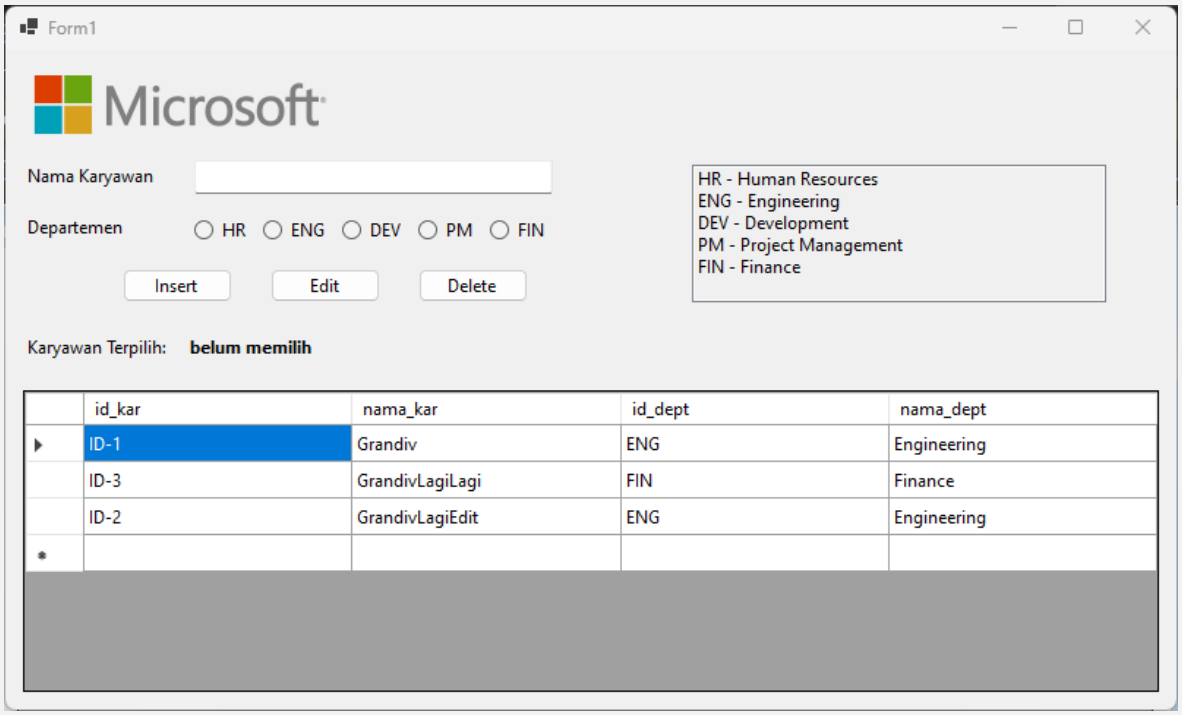
Link Github: https://github.com/grandiv/493242-MuhGrandivLP-ResponsiJuniorProject

Screenshot file Tampilan:

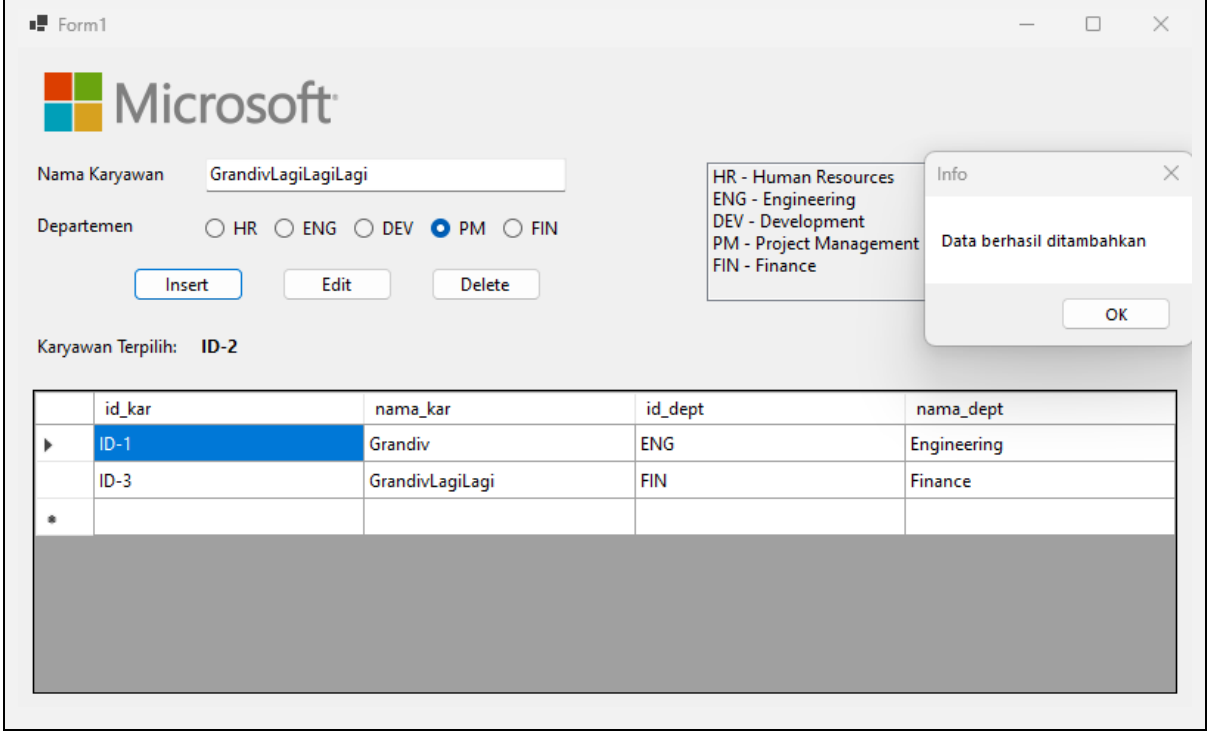
Tampilan Awal:

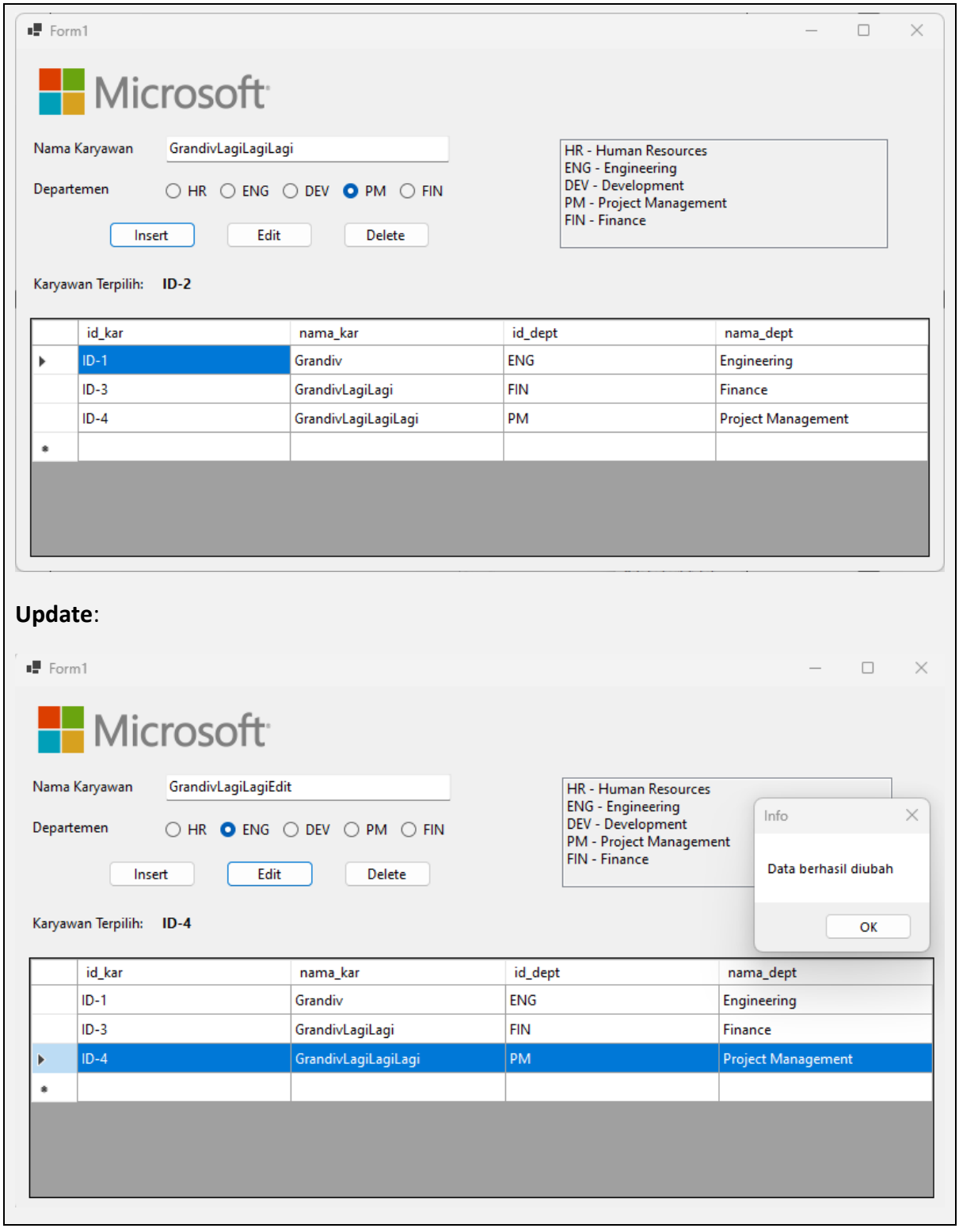


Load (secara otomatis setelah aplikasi dijalankan):


Form1
Microsoft
Nama Karyawan
Departemen
HR ENG DEV PM FIN
Insert
Edit
Delete
HR - Human Resources
ENG - Engineering
DEV - Development
PM - Project Management
FIN - Finance
Karyawan Terpilih: belum memilih
id_kar
nama_kar
id_dept
nama_dept
ID-1
Grandiv
ENG
Engineering
ID-3
GrandivLagiLagi
FIN
Finance
ID-2
GrandivLagiEdit
ENG
Engineering

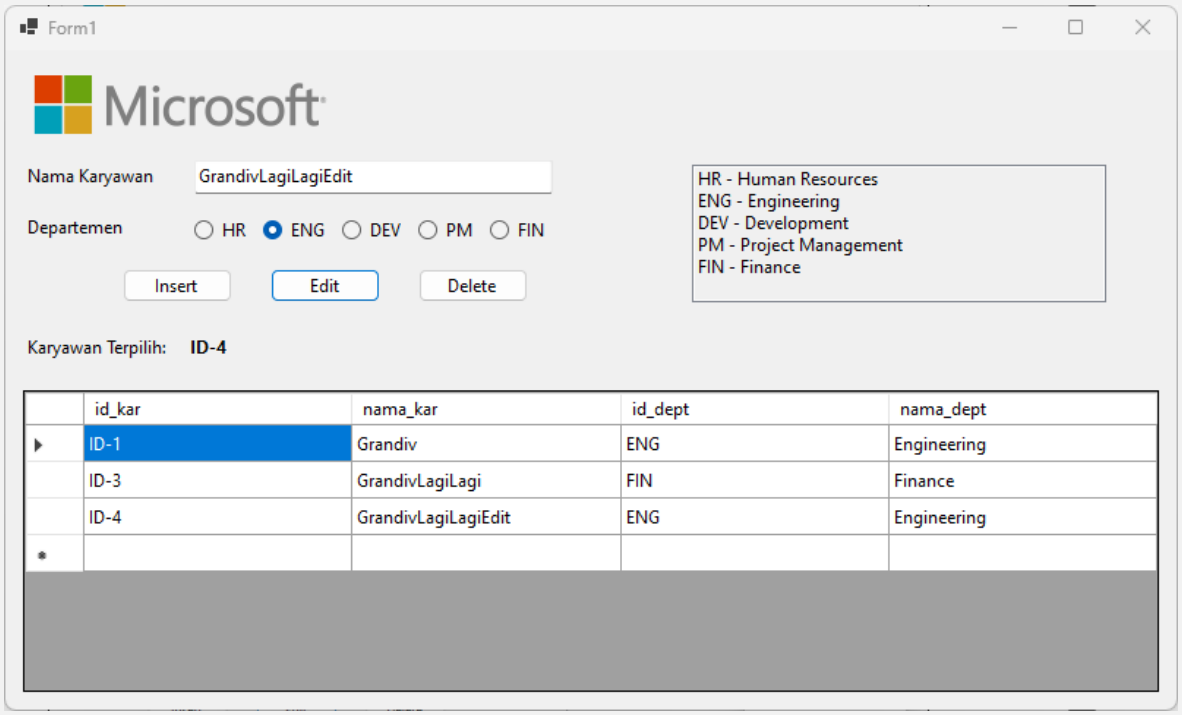
Create:


Form1
Microsoft
Nama Karyawan
GrandivLagiLagiLagi
Departemen
HR ENG DEV PM FIN
Insert
Edit
Delete
HR - Human Resources
ENG - Engineering
DEV - Development
PM - Project Management
FIN - Finance
Info
Data berhasil ditambahkan
OK
Karyawan Terpilih: ID-2
id_kar
nama_kar
id_dept
nama_dept
ID-1
Grandiv
ENG
Engineering
ID-3
GrandivLagiLagi
FIN
Finance

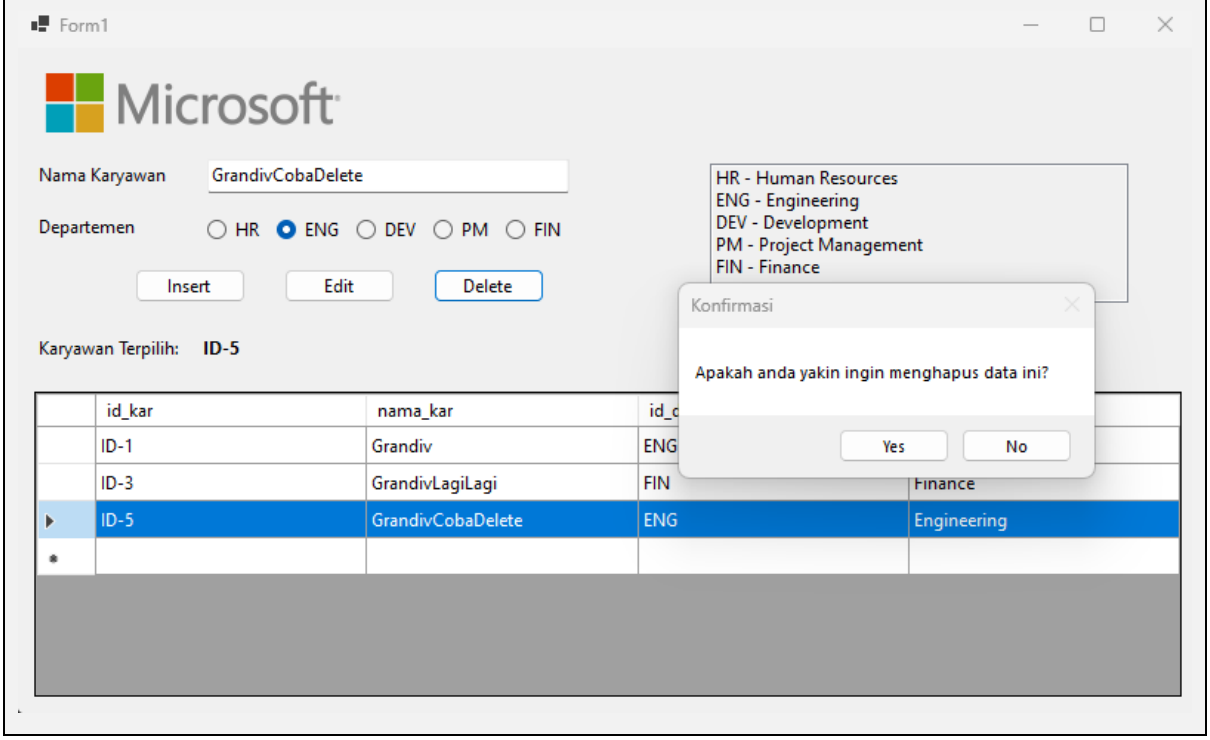

Form1
Microsoft
Nama Karyawan
GrandivLagiLagiLagi
Departemen
HR ENG DEV PM FIN
Insert
Edit
Delete
HR - Human Resources
ENG - Engineering
DEV - Development
PM - Project Management
FIN - Finance
Karyawan Terpilih: ID-2
id_kar nama_kar id_dept nama_dept
ID-1 Grandiv ENG Engineering
ID-3 GrandivLagiLagi FIN Finance
ID-4 GrandivLagiLagiLagi PM Project Management

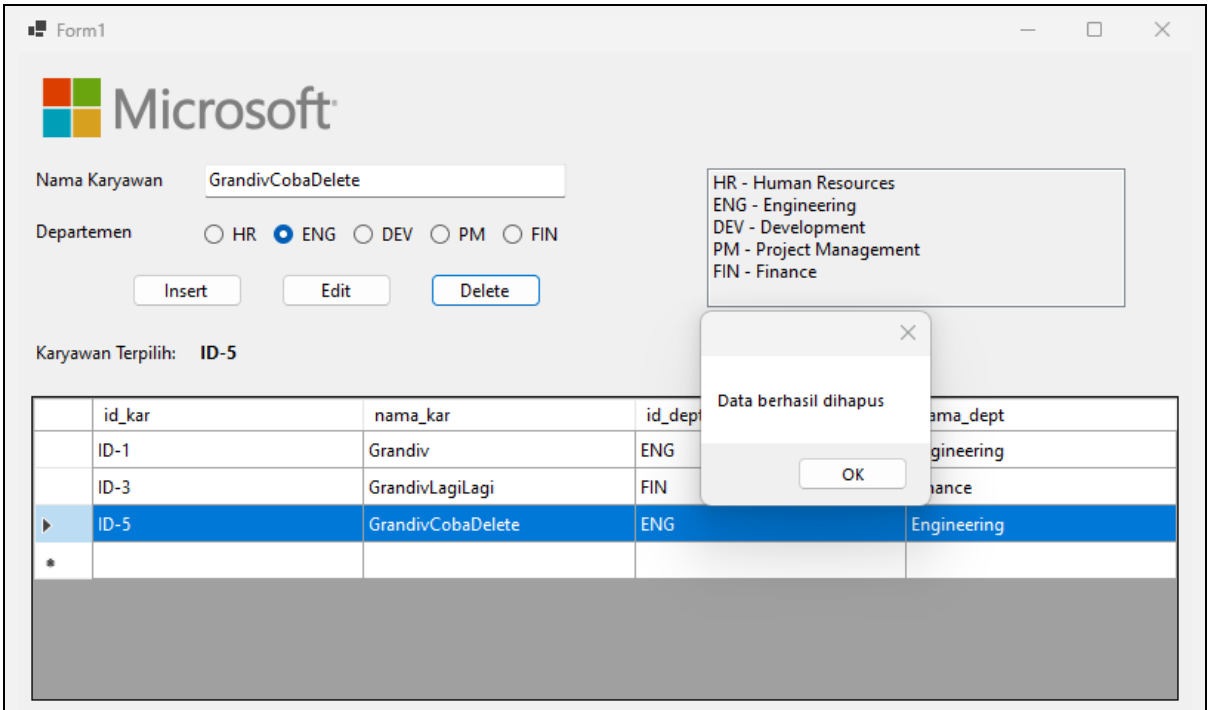
Update:

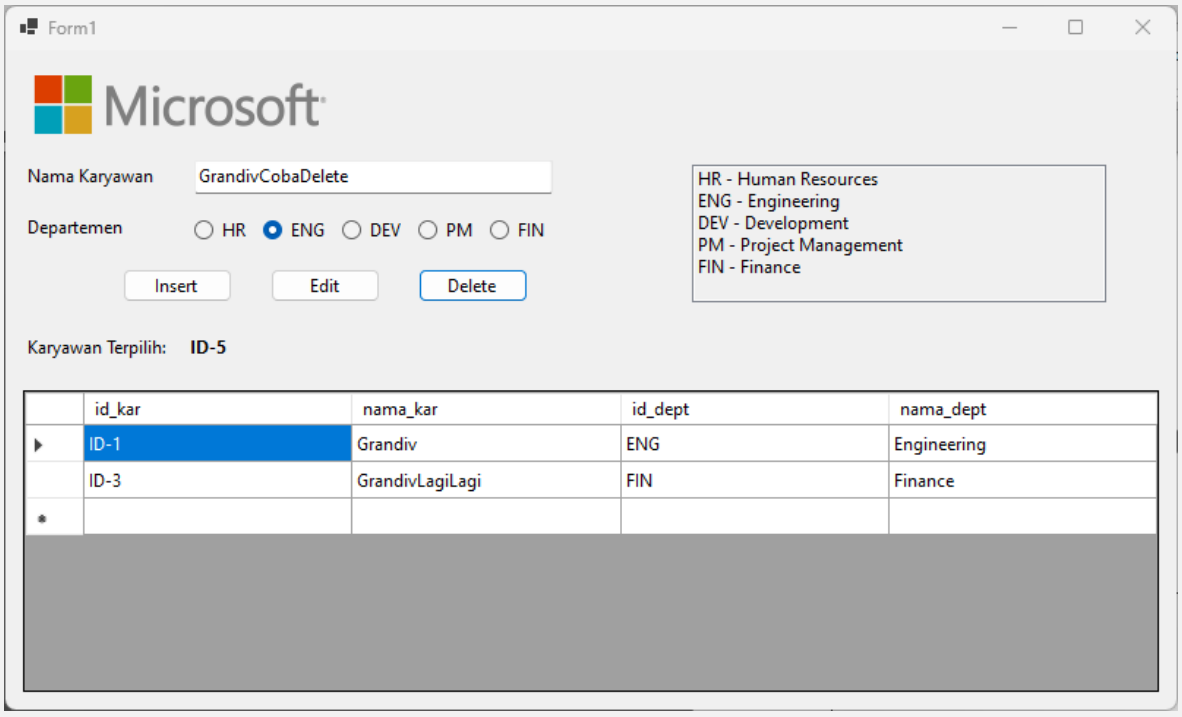
Form1
Microsoft
Nama Karyawan
GrandivLagiLagiEdit
Departemen
HR ENG DEV PM FIN
Insert
Edit
Delete
HR - Human Resources
ENG - Engineering
DEV - Development
PM - Project Management
FIN - Finance
Info
Data berhasil diubah
OK
Karyawan Terpilih: ID-4
id_kar nama_kar id_dept nama_dept
ID-1 Grandiv ENG Engineering
ID-3 GrandivLagiLagi FIN Finance
ID-4 GrandivLagiLagiLagi PM Project Management


Form1
Microsoft
Nama Karyawan
GrandivLagiLagiEdit
Departemen
HR ENG DEV PM FIN
Insert
Edit
Delete
HR - Human Resources
ENG - Engineering
DEV - Development
PM - Project Management
FIN - Finance
Karyawan Terpilih: ID-4
id_kar nama_kar id_dept nama_dept
ID-1 Grandiv ENG Engineering
ID-3 GrandivLagiLagi FIN Finance
ID-4 GrandivLagiLagiEdit ENG Engineering

Delete:


Form1
Microsoft
Nama Karyawan
GrandivCobaDelete
Departemen
HR ENG DEV PM FIN
Insert
Edit
Delete
HR - Human Resources
ENG - Engineering
DEV - Development
PM - Project Management
FIN - Finance
Konfirmasi
Apakah anda yakin ingin menghapus data ini?
Yes
No
Karyawan Terpilih: ID-5
id_kar nama_kar
ID-1 Grandiv ENG
ID-3 GrandivLagiLagi FIN Finance
ID-5 GrandivCobaDelete ENG Engineering





Screenshot Konsep OOP (Enkapsulasi dan *Inheritance*):

Enkapsulasi:

```
public DataTable dt;
private DataGridViewRow row;

8 references
public DataGridViewRow Row { get => row; set => row = value; }
```

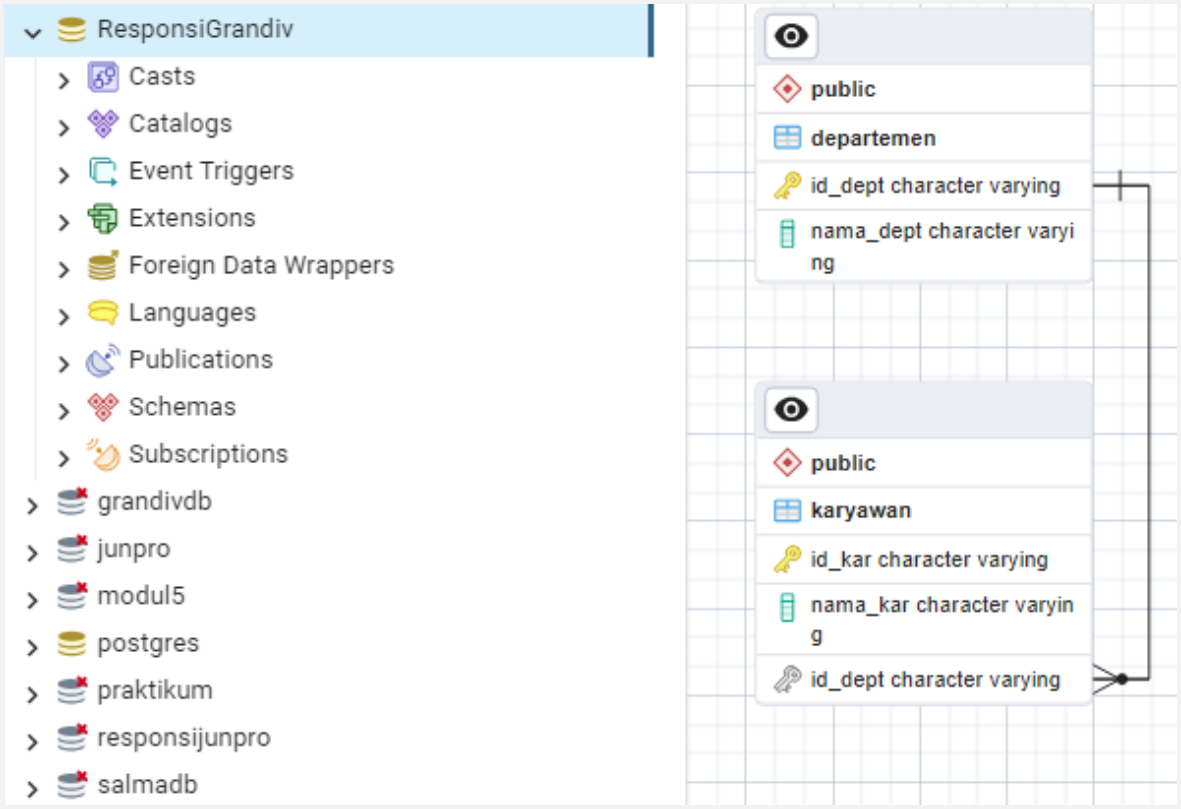
Enkapsulasi ditandai dengan adanya access-modifier yang membedakan kelas-kelas mana saja yang bisa mengakses variabel atau metode tersebut. Misalnya pada screenshot, variabel `dt` akan dapat diakses oleh semua kelas, sedangkan variabel `row` hanya dapat diakses oleh kelas itu saja, yang dalam kasus ini adalah Form1. `row` kemudian dibuat constructure-nya dengan `get` dan `set`

Inheritance:

```
3 references
public partial class Form1 : Form
{
```

Inheritance atau turunan berarti suatu kelas akan mewarisi kelas-kelas yang menjadi children-nya. Pada screenshot maka semua variabel atau metode dalam `Form` akan diwarisi ke `Form1`

Screenshot ERD Databases pada PgAdmin (klik kanan databases -> ERD For Database):



Implementasi Kode CRUD:

Create:

```
1 reference
private void InsertData()
{
    string nama = namaKar_TB.Text;
    string id_dept = "";

    if (HR_RB.Checked)
    {
        id_dept = "HR";
    }
    else if (ENG_RB.Checked)
    {
        id_dept = "ENG";
    }
    else if (DEV_RB.Checked)
    {
        id_dept = "DEV";
    }
    else if (PM_RB.Checked)
    {
        id_dept = "PM";
    }
    else if (FIN_RB.Checked)
    {
        id_dept = "FIN";
    }

    if (id_dept == "")
    {
        MessageBox.Show("Pilih departemen terlebih dahulu");
        return;
    }
```

```
    if (nama == "")
    {
        MessageBox.Show("Nama tidak boleh kosong");
        return;
    }
    try
    {
        conn = new NpgsqlConnection(connString);
        conn.Open();
        sql = "SELECT * FROM add_karyawan(:_nama, :_id_dept)";
        cmd = new NpgsqlCommand(sql, conn);
        cmd.Parameters.AddWithValue("_nama", nama);
        cmd.Parameters.AddWithValue("_id_dept", id_dept);
        int statusCode = (int)cmd.ExecuteScalar();
        if (statusCode == 201)
        {
            MessageBox.Show("Data berhasil ditambahkan", "Info");
            LoadData();
            return;
        }
        else if (statusCode == 409)
        {
            throw new Exception("Data sudah ada");
        }
        else
        {
            MessageBox.Show("Data gagal ditambahkan");
        }
    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message);
        return;
    }
    finally { conn.Close(); }
}
```

Read:

```
4 references
private void LoadData()
{
    conn = new NpgsqlConnection(connString);
    try
    {
        conn.Open();

        sql = @"SELECT k.id_kar, k.nama_kar, k.id_dept, d.nama_dept
                FROM karyawan k
                JOIN departemen d ON k.id_dept = d.id_dept";
        cmd = new NpgsqlCommand(sql, conn);
        dt = new DataTable();
        dt.Load(cmd.ExecuteReader());
        dgvDataTable.DataSource = dt;

        sql = "SELECT id_dept, nama_dept FROM departemen";
        cmd = new NpgsqlCommand(sql, conn);
        using (NpgsqlDataReader reader = cmd.ExecuteReader())
        {
            deptList.Items.Clear();
            while (reader.Read())
            {
                deptList.Items.Add(reader["id_dept"].ToString() + " - " + reader["nama_dept"].ToString());
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        conn.Close();
    }
}
```

Update:

```
1 reference
private void EditData()
{
    if (Row == null)
    {
        MessageBox.Show("Pilih data untuk diedit");
        return;
    }
    try
    {
        string nama = namaKar_TB.Text;
        string id_dept = "";

        if (HR_RB.Checked) id_dept = "HR";
        else if (ENG_RB.Checked) id_dept = "ENG";
        else if (DEV_RB.Checked) id_dept = "DEV";
        else if (PM_RB.Checked) id_dept = "PM";
        else if (FIN_RB.Checked) id_dept = "FIN";

        if (id_dept == "")
        {
            MessageBox.Show("Pilih departemen terlebih dahulu");
            return;
        }

        if (nama == "")
        {
            MessageBox.Show("Nama tidak boleh kosong");
            return;
        }

        conn = new NpgsqlConnection(connString);
        conn.Open();
        sql = "SELECT * FROM edit_karyawan(:_id_kar, :_nama, :_id_dept)";
        cmd = new NpgsqlCommand(sql, conn);
        cmd.Parameters.AddWithValue("_id_kar", Row.Cells["id_kar"].Value.ToString());
        cmd.Parameters.AddWithValue("_nama", nama);
        cmd.Parameters.AddWithValue("_id_dept", id_dept);
```

```
        int statusCode = (int)cmd.ExecuteScalar();

        if (statusCode == 200)
        {
            MessageBox.Show("Data berhasil diubah", "Info");
            LoadData();
            return;
        }
        if (statusCode == 404)
        {
            throw new Exception(nama + " tidak ditemukan");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally { conn.Close(); }
}
```

Delete:

```
1 reference
private void DeleteData()
{
    if (Row == null)
    {
        MessageBox.Show("Pilih karyawan untuk dihapus");
        return;
    }
    try
    {
        MessageBox.Show("Apakah anda yakin ingin menghapus data ini?", "Konfirmasi", MessageBoxButtons.YesNo);
        if (DialogResult == DialogResult.No)
        {
            return;
        }
        try
        {
            conn = new NpgsqlConnection(connString);
            conn.Open();
            sql = "SELECT * FROM delete_karyawan(:_id_karyawan)";
            cmd = new NpgsqlCommand(sql, conn);
            cmd.Parameters.AddWithValue("_id_karyawan", Row.Cells["id_kar"].Value.ToString());

            int statusCode = (int)cmd.ExecuteScalar();

            if (statusCode == 200)
            {
                MessageBox.Show("Data berhasil dihapus");
                LoadData();
                return;
            }
            if (statusCode == 404)
            {
                throw new Exception("Data tidak ditemukan");
            }
        }
```

```
            catch (Exception e)
            {
                MessageBox.Show(e.Message);
            }
        }
        catch (Exception e)
        {
            MessageBox.Show(e.Message);
        }
        finally
        {
            conn.Close();
        }
}
```

TUGAS OPSIONAL

Screenshot Konsep OOP (Polimorfisme dan Abstraksi):

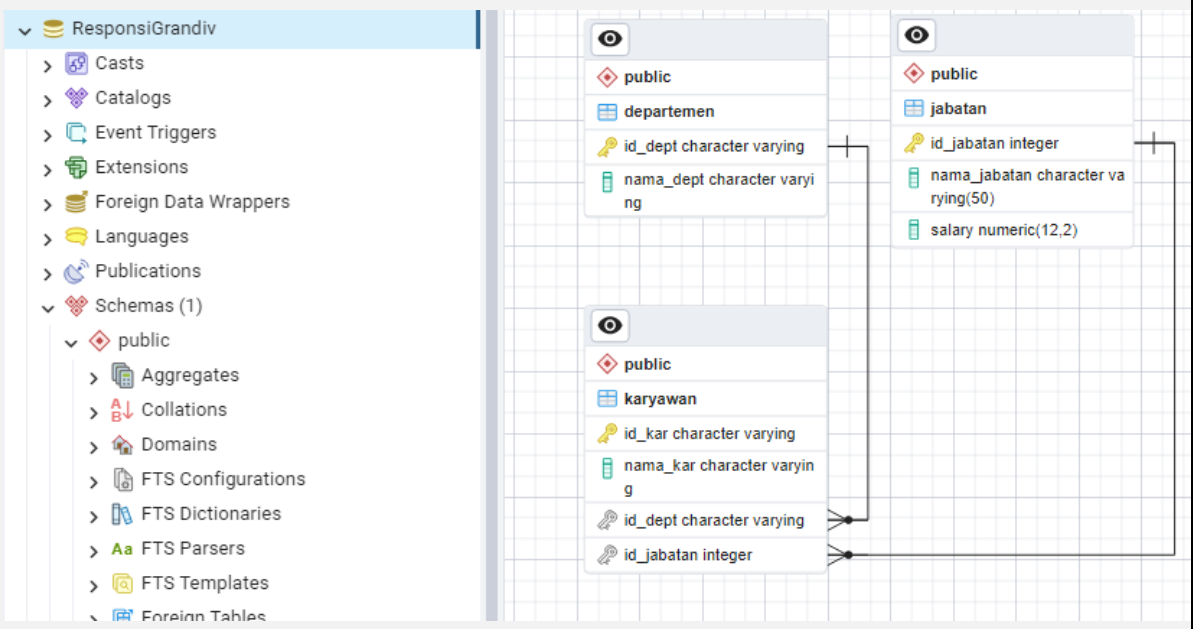
Abstraksi:

```
1 reference
private void InsertData()...

1 reference
private void EditData()...

1 reference
private void DeleteData()...

1 reference
private void insertBt_Click(object sender, EventArgs e)
{
    InsertData();
}

1 reference
private void editBt_Click(object sender, EventArgs e)
{
    EditData();
}

1 reference
private void deleteBt_Click(object sender, EventArgs e)
{
    DeleteData();
}
```

Abstraksi dilakukan untuk menyembunyikan fungsi yang terjadi saat pengguna melakukan aksi pada aplikasi. Contohnya jika pengguna menekan tombol insert, edit, atau delete maka akan memanggil metode InsertData(), EditData(), atau DeleteData()

Screenshot Fungsi pada Postgres:

ERD:



Fungsi:

Create:

```
--- INSERT ----
CREATE OR REPLACE FUNCTION add_jabatan(
    _nama CHARACTER VARYING(50),
    _salary numeric(12, 2)
)
RETURNS int AS
$$
BEGIN
    IF EXISTS (
        SELECT 1 FROM jabatan
        WHERE namajab = _nama AND salary = _salary
    )
    THEN RETURN 409;
    ELSE
        INSERT INTO jabatan(namajab, salary)
        VALUES (_nama, _salary);
        RETURN 201;
    END IF;
END;
$$
LANGUAGE plpgsql;
```

Update:

```sql
--- UPDATE ---
CREATE OR REPLACE FUNCTION edit_jabatan(
    _id_jabatan INTEGER,
    _nama CHARACTER VARYING(50),
    _salary NUMERIC(12, 2)
)
RETURNS int AS
$$
BEGIN
    IF EXISTS (
        SELECT 1 FROM jabatan
        WHERE id_jabatan = _id_jabatan
    )
    THEN
        UPDATE jabatan
        SET namajab = _nama,
            salary = _salary
        WHERE id_jabatan = _id_jabatan;
        RETURN 200;
    ELSE
        RETURN 404;
    END IF;
END;
$$
LANGUAGE plpgsql;
```

Delete:

```sql
--- DELETE ---
CREATE OR REPLACE FUNCTION delete_jabatan(
    _id_jabatan INTEGER
)
RETURNS int AS
$$
BEGIN
    IF EXISTS (
        SELECT 1 FROM jabatan
        WHERE id_jabatan = _id_jabatan
    )
    THEN
        DELETE FROM jabatan
        WHERE id_jabatan = _id_jabatan;
        RETURN 200;
    ELSE
        RETURN 404;
    END IF;
END;
$$
LANGUAGE plpgsql;
```