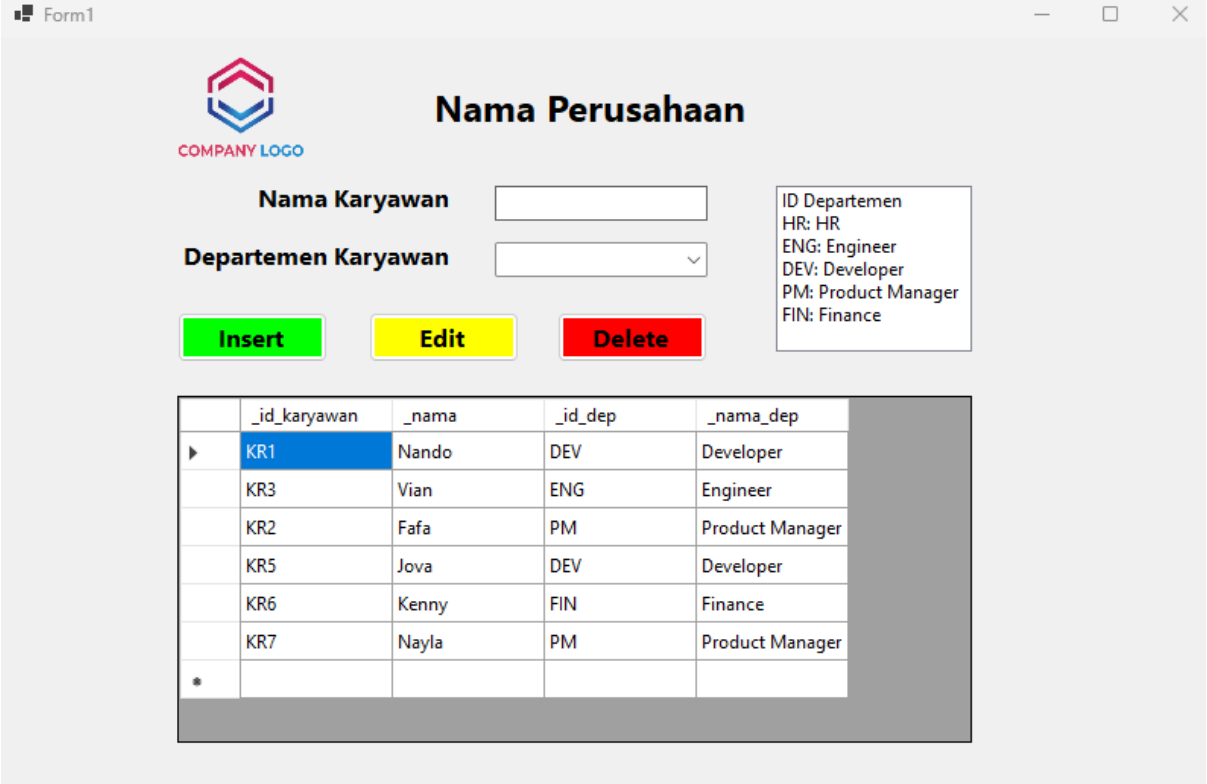



Agustinus Angelo Christian Fernando

21/473804/TK/52235

- Tampilan Keseluruhan Program



Form1


COMPANY LOGO

Nama Perusahaan

Nama Karyawan

Departemen Karyawan

Insert **Edit** **Delete**

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

	_id_karyawan	_nama	_id_dep	_nama_dep
▶	KR1	Nando	DEV	Developer
	KR3	Vian	ENG	Engineer
	KR2	Fafa	PM	Product Manager
	KR5	Jova	DEV	Developer
	KR6	Kenny	FIN	Finance
	KR7	Nayla	PM	Product Manager
*				

Penjelasan: Ini adalah tampilan program yang saya buat. Lengkap sesuai ketentuan dari soal. Saya juga memberi warna pada button supaya tampilan lebih interaktif. Selain itu, data akan otomatis di-load dari database sehingga tidak membutuhkan button load. Semua fungsionalitas sudah berhasil dijalankan. Akan saya berikan buktinya pada poin-poin di bawah.

- Pembuatan Tabel Karyawan

karyawan

General Columns Advanced Constraints Parameters Security SQL

Inherited from table(s)

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	id_karyawan	character varying	6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	('KR':text
	nama	character varying	30		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	id_dep	character varying			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Info: id_karyawan saya buat menggunakan sequence sehingga bisa auto increment setiap meng-insert karyawan baru. Formatnya adalah: KR1, KR2, KR3, dan seterusnya.

Foreign Key

karyawan

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Foreign Key

	Name	Columns	Referenced Table
	karyawan_fkey	(id_dep) -> (id_dep)	public.departemen

Info: id_dep adalah foreign key pada tabel Karyawan yang nantinya akan menghubungkan ke tabel Departemen.

- Pembuatan Tabel Departemen

departemen

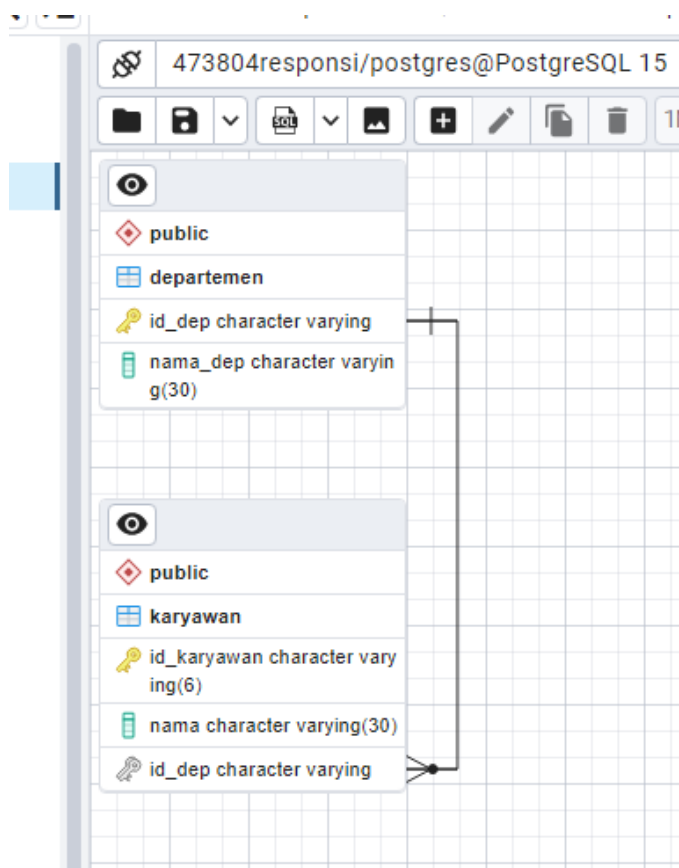
General Columns Advanced Constraints Parameters Security SQL

Inherited from table(s)

Columns							
	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	id_dep	character varying			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	nama_dep	character varying	30		<input checked="" type="checkbox"/>	<input type="checkbox"/>	

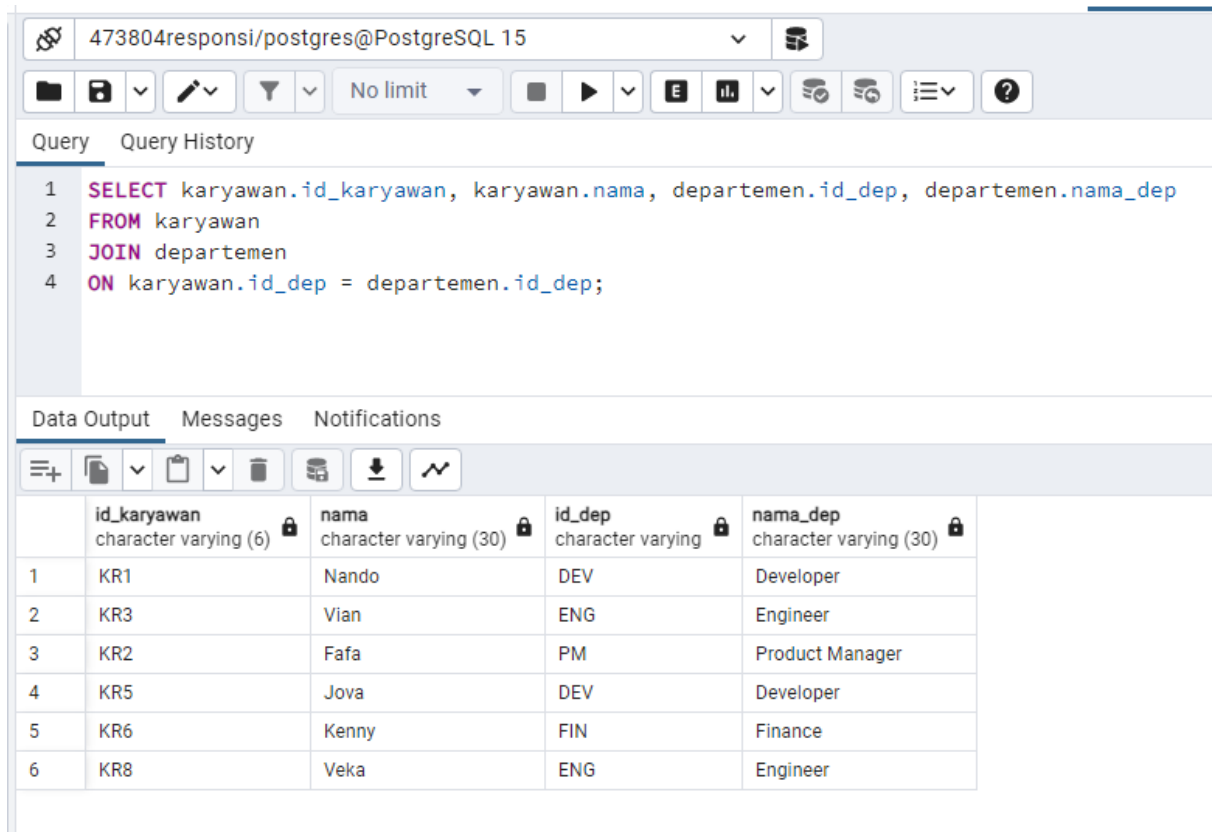
Info: id_dep adalah primary key dari tabel Departemen. id_dep berupa singkatan dari setiap posisi pekerjaannya. Contoh: HR, FIN, DEV, ENG, PM.

- ERD



Penjelasan: Tampak bahwa kedua tabel berhasil terhubung karena tabel departemen memiliki primary key id_dep dan tabel karyawan memiliki foreign key id_dep.

Hasil Query Join SQL



The screenshot shows a PostgreSQL query editor interface. At the top, the connection is identified as '473804responsi/postgres@PostgreSQL 15'. Below the connection bar is a toolbar with icons for saving, editing, and running queries. The 'Query' tab is active, displaying the following SQL query:

```
1 SELECT karyawan.id_karyawan, karyawan.nama, departemen.id_dep, departemen.nama_dep
2 FROM karyawan
3 JOIN departemen
4 ON karyawan.id_dep = departemen.id_dep;
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query in a table format. The table has four columns: id_karyawan, nama, id_dep, and nama_dep. The results are as follows:

	id_karyawan character varying (6)	nama character varying (30)	id_dep character varying	nama_dep character varying (30)
1	KR1	Nando	DEV	Developer
2	KR3	Vian	ENG	Engineer
3	KR2	Fafa	PM	Product Manager
4	KR5	Jova	DEV	Developer
5	KR6	Kenny	FIN	Finance
6	KR8	Veka	ENG	Engineer

Penjelasan: Tampak bahwa berhasil melakukan query kedua tabel menggunakan join karena keduanya sudah saling berelasi.

- Fungsi Select SQL

473804responsi/postgres@PostgreSQL 15

Query Query History

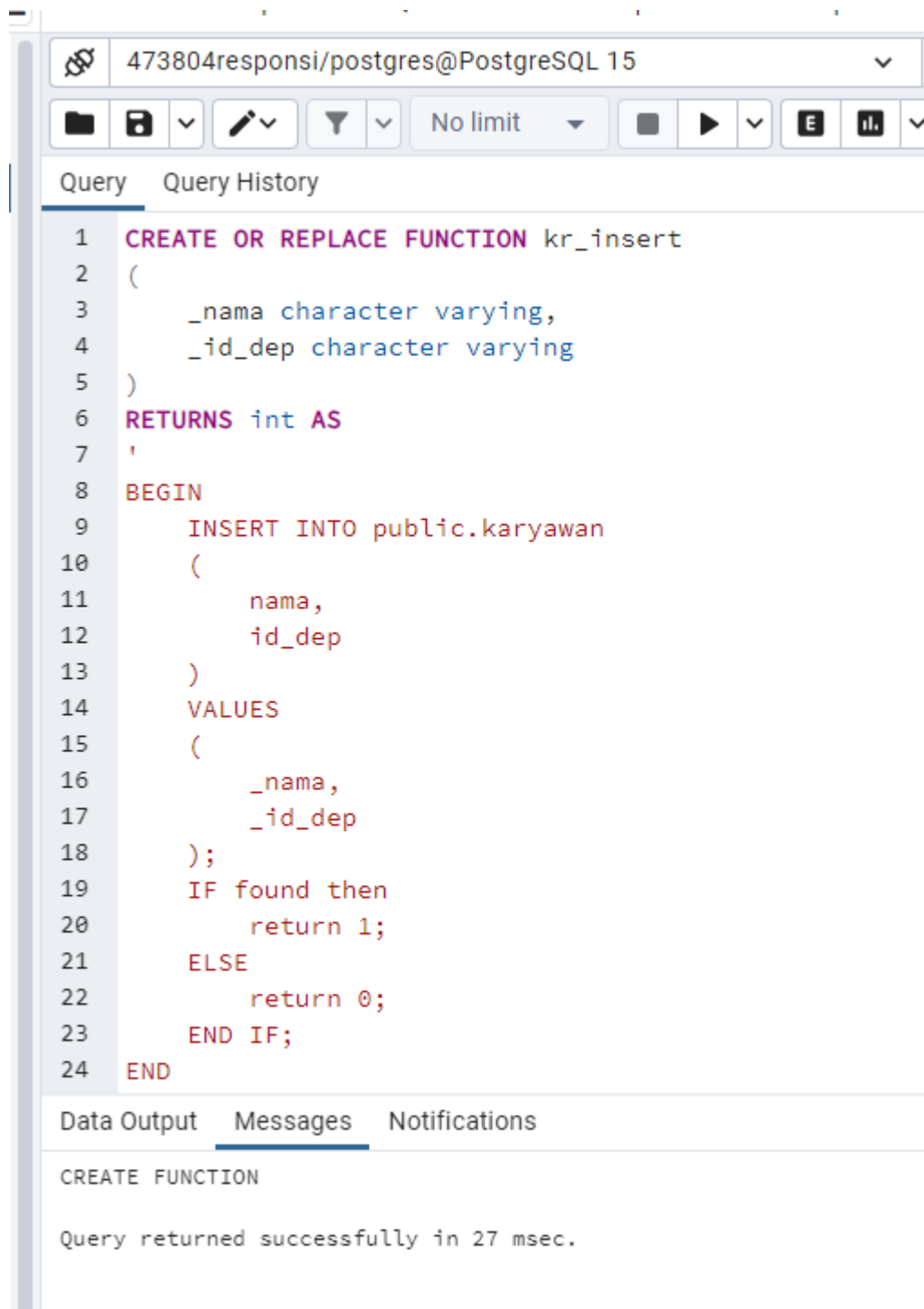
```
1 CREATE OR REPLACE FUNCTION kr_select()
2 RETURNS TABLE
3 (
4     _id_karyawan character varying,
5     _nama character varying,
6     _id_dep character varying,
7     _nama_dep character varying
8 )
9 LANGUAGE plpgsql
10 AS
11 '
12 BEGIN
13     RETURN query
14     SELECT karyawan.id_karyawan, karyawan.nama, departemen.id_dep, departemen.nama_dep
15     FROM karyawan
16     JOIN departemen
17     ON karyawan.id_dep = departemen.id_dep;
18 END
19 '
20
```

Data Output Messages Notifications

CREATE FUNCTION

Query returned successfully in 24 msec.

- Fungsi Insert SQL



The screenshot shows a PostgreSQL query editor interface. At the top, the connection string is '473804responsi/postgres@PostgreSQL 15'. Below the connection bar is a toolbar with icons for file operations, query execution, and settings. The main area is divided into 'Query' and 'Query History' tabs. The 'Query' tab is active, displaying a SQL script for creating or replacing a function named 'kr_insert'. The script takes two parameters, '_nama' and '_id_dep', both of type 'character varying'. It returns an integer. The function body begins with an 'INSERT INTO public.karyawan' statement, followed by an 'IF found then' block that returns 1, and an 'ELSE' block that returns 0. The script ends with 'END IF;' and 'END'. Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is selected, showing the output 'CREATE FUNCTION' and a status message 'Query returned successfully in 27 msec.'

```
1 CREATE OR REPLACE FUNCTION kr_insert
2 (
3     _nama character varying,
4     _id_dep character varying
5 )
6 RETURNS int AS
7 '
8 BEGIN
9     INSERT INTO public.karyawan
10    (
11        nama,
12        id_dep
13    )
14    VALUES
15    (
16        _nama,
17        _id_dep
18    );
19    IF found then
20        return 1;
21    ELSE
22        return 0;
23    END IF;
24 END
```

Data Output Messages Notifications

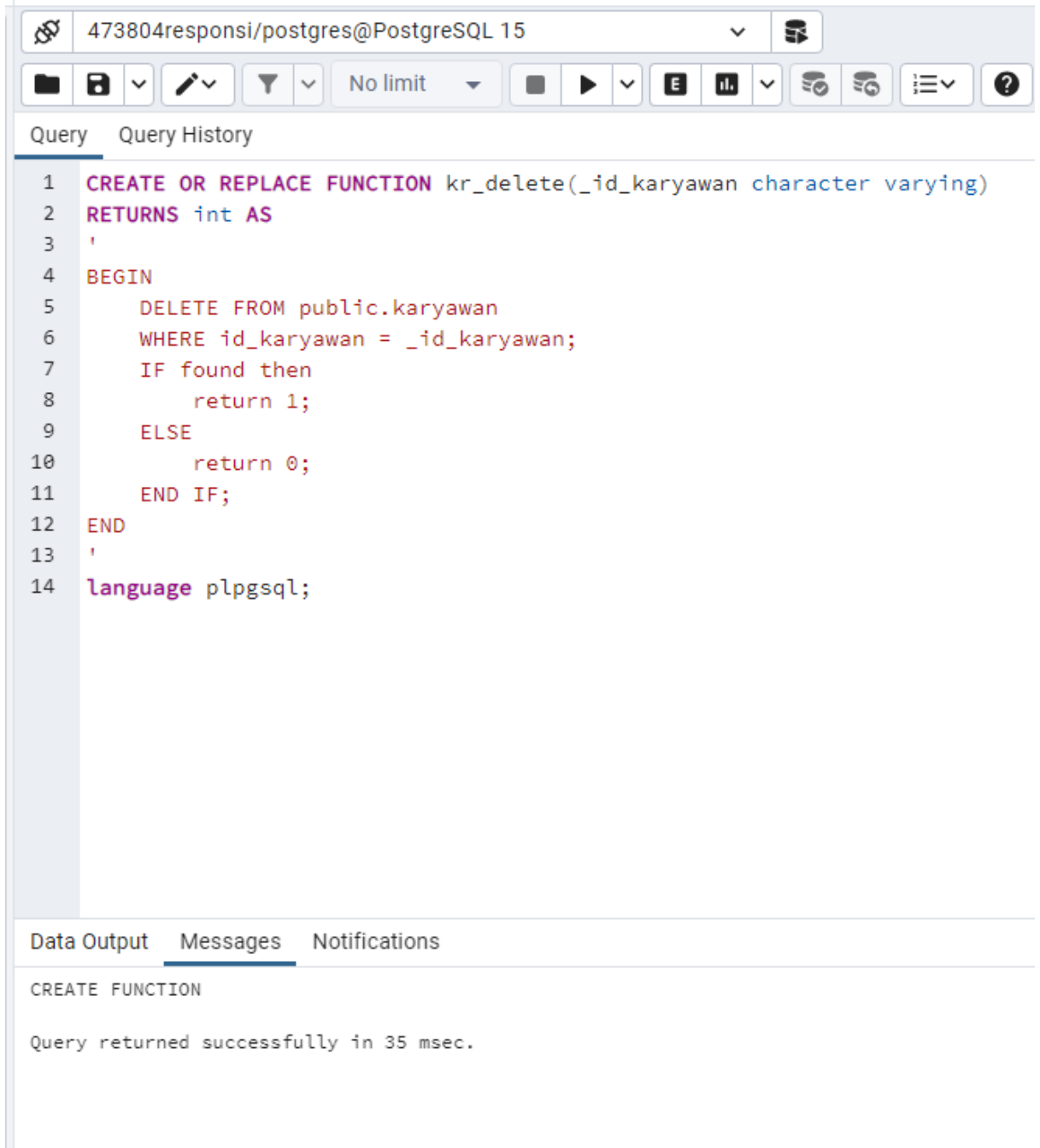
CREATE FUNCTION

Query returned successfully in 27 msec.

- **Fungsi Update SQL**



- Fungsi Delete SQL



- Koneksi Windows Form ke PostgreSQL

```
private NpgsqlConnection conn;  
string connstring = "Host=localhost;Port=5432;Username=postgres;Password=informatika;Database=473804responsi";  
public DataTable dt;  
public static NpgsqlCommand cmd;  
private string sql = null;  
private DataGridViewRow r;
```

Info: berhasil melakukan koneksi dari windows form ke PostgreSQL dengan menuliskan host, port, username, password, dan nama database yang saya buat.

- Koneksi Select / Load Data dari Windows Form ke PostgreSQL


```

4 references
void LoadData()
{
    try
    {
        conn.Open();
        dgvData.DataSource = null;
        sql = "select * from kr_select()";
        cmd = new NpgsqlCommand(sql, conn);
        dt = new DataTable();
        NpgsqlDataReader rd = cmd.ExecuteReader();
        dt.Load(rd);
        dgvData.DataSource = dt;
        conn.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error: " + ex.Message, "FAIL!!", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

1 reference
private void Form1_Load(object sender, EventArgs e)
{
    conn = new NpgsqlConnection(connstring);
    // Supaya otomatis load data tanpa perlu tombol load
    LoadData();
}

```

	_id_karyawan	_nama	_id_dep	_nama_dep
▶	KR1	Nando	DEV	Developer
	KR3	Vian	ENG	Engineer
	KR2	Fafa	PM	Product Manager
	KR5	Jova	DEV	Developer
	KR6	Kenny	FIN	Finance
	KR7	Nayla	PM	Product Manager
*				

Penjelasan: Tampak bahwa berhasil melakukan load / select data tanpa perlu tombol load. Ketika program di-run, maka program akan otomatis menampilkan data yang ada.

- **Koneksi Insert Data dari Windows Form ke PostgreSQL**

```

1 reference
private void btnInsert_Click(object sender, EventArgs e)
{
    try
    {
        conn.Open();
        sql = @"select * from kr_insert(:_nama, :_id_dep)";
        cmd = new NpgsqlCommand(sql, conn);
        cmd.Parameters.AddWithValue("_nama", txtNamaKaryawan.Text);
        cmd.Parameters.AddWithValue("_id_dep", cbDepartemenKaryawan.SelectedItem.ToString());
        if ((int)cmd.ExecuteScalar() == 1)
        {
            MessageBox.Show("Data Berhasil Diinputkan", "Success!", MessageBoxButtons.OK, MessageBoxIcon.Information);
            conn.Close();
            LoadData();
            txtNamaKaryawan.Text = cbDepartemenKaryawan.Text = null;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error: " + ex.Message, "Insert FAIL!!", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

Ketika melakukan insert:

Nama Perusahaan

Nama Karyawan Veka

Departemen Karyawan ENG

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


Insert **Edit** **Delete**

	_id_karyawan	_nama	_id_dep	_nama_dep
▶	KR1	Nando	DEV	Developer
	KR3	Vian	ENG	Engineer
	KR2	Fafa	PM	Product Manag
	KR5	Jova	DEV	Developer
	KR6	Kenny	FIN	Finance
	KR7	Nayla	PM	Product Manager
*				

Success!
 Data Berhasil Diinputkan
 OK

Setelah berhasil melakukan insert:

Form1



Nama Perusahaan

Nama Karyawan

Departemen Karyawan

Insert
Edit
Delete

ID Departemen

HR: HR

ENG: Engineer

DEV: Developer

PM: Product Manager

FIN: Finance

	_id_karyawan	_nama	_id_dep	_nama_dep
▶	KR1	Nando	DEV	Developer
	KR3	Vian	ENG	Engineer
	KR2	Fafa	PM	Product Manager
	KR5	Jova	DEV	Developer
	KR6	Kenny	FIN	Finance
	KR7	Nayla	PM	Product Manager
	KR8	Veka	ENG	Engineer
*				

Penjelasan: Tampak bahwa saya berhasil menambah karyawan baru bernama Veka dengan departemen Engineer.

- Koneksi Edit Data dari Windows Form ke PostgreSQL

```

1 reference
private void dgvData_CellClick(object sender, DataGridViewCellEventArgs e)
{
    if (e.RowIndex >= 0)
    {
        r = dgvData.Rows[e.RowIndex];
        txtNamaKaryawan.Text = r.Cells["_nama"].Value.ToString();
        cbDepartemenKaryawan.Text = r.Cells["_id_dep"].Value.ToString();
    }
}

```

Lanjut ke halaman berikutnya

```

1 reference
private void btnEdit_Click(object sender, EventArgs e)
{
    if (r == null)
    {
        MessageBox.Show("Mohon pilih baris data yang ingin diedit!", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    try
    {
        conn.Open();
        sql = @"select * from kr_update(:_id_karyawan, :_nama, :_id_dep)";
        cmd = new NpgsqlCommand(sql, conn);
        cmd.Parameters.AddWithValue("_id_karyawan", r.Cells["_id_karyawan"].Value.ToString());
        cmd.Parameters.AddWithValue("_nama", txtNamaKaryawan.Text);
        cmd.Parameters.AddWithValue("_id_dep", cbDepartemenKaryawan.SelectedItem.ToString());
        if ((int)cmd.ExecuteScalar() == 1)
        {
            MessageBox.Show("Data Berhasil Diedit", "Success!", MessageBoxButtons.OK, MessageBoxIcon.Information);
            conn.Close();
            LoadData();
            txtNamaKaryawan.Text = cbDepartemenKaryawan.Text = null;
            r = null;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error: " + ex.Message, "Update FAIL!", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

Ketika melakukan edit:

Nama Perusahaan

COMPANY LOGO

Nama Karyawan

Departemen Karyawan

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

Insert **Edit** **Delete**


	_id_karyawan	_nama	_id_dep	_nama_dep
	KR1	Nando	DEV	Developer
	KR3	Vian	ENG	Engineer
	KR2	Fafa	PM	Product Manager
	KR5	Jova	DEV	Developer
	KR6	Kenny	FIN	Finance
▶	KR7	Nayla	PM	Product Manager
	KR8	Veka	ENG	Engineer
*				

Success! Data Berhasil Diedit

OK

Setelah berhasil melakukan edit:

Form1



Nama Perusahaan

Nama Karyawan

Departemen Karyawan

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

Insert
Edit
Delete

	_id_karyawan	_nama	_id_dep	_nama_dep
▶	KR1	Nando	DEV	Developer
	KR3	Vian	ENG	Engineer
	KR2	Fafa	PM	Product Manager
	KR5	Jova	DEV	Developer
	KR6	Kenny	FIN	Finance
	KR8	Veka	ENG	Engineer
	KR7	Tiara	DEV	Developer
*				

Penjelasan: Tampak bahwa saya berhasil mengedit karyawan bernama Nayla menjadi Tiara dan departemen Product Manager menjadi Developer.

- Koneksi Delete Data dari Windows Form ke PostgreSQL


```

1 reference
private void btnDelete_Click(object sender, EventArgs e)
{
    if (r == null)
    {
        MessageBox.Show("Mohon pilih baris data yang ingin dihapus!", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (MessageBox.Show("Apakah yakin ingin menghapus data " + r.Cells["_nama"].Value.ToString() + " ?", "Hapus Data Terkonfirmasi",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question, MessageBoxDefaultButton.Button1) == DialogResult.Yes)
    {
        try
        {
            conn.Open();
            sql = @"select * from kr_delete(:_id_karyawan)";
            cmd = new NpgsqlCommand(sql, conn);
            cmd.Parameters.AddWithValue("_id_karyawan", r.Cells["_id_karyawan"].Value.ToString());
            if ((int)cmd.ExecuteScalar() == 1)
            {
                MessageBox.Show("Data Berhasil Dihapus", "Success!", MessageBoxButtons.OK, MessageBoxIcon.Information);
                conn.Close();
                LoadData();
                txtNamaKaryawan.Text = cbDepartemenKaryawan.Text = null;
                r = null;
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error: " + ex.Message, "Delete FAIL!!", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

Ketika melakukan delete:

Form1



Nama Perusahaan

Nama Karyawan

Departemen Karyawan

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


	_id_karyawan	_nama	_id_dep	_nama_dep
	KR1	Nando	DEV	Developer
	KR3	Vian	ENG	Engineer
	KR2	Fafa	PM	Product Manager
	KR5	Jova	DEV	Developer
	KR6	Kenny	FIN	Finance
	KR8	Veka	ENG	Engineer
▶	KR7	Tiara	DEV	Developer
*				

Success!

Data Berhasil Dihapus

Setelah berhasil melakukan delete:

Form1



Nama Perusahaan

Nama Karyawan

Departemen Karyawan

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

	_id_karyawan	_nama	_id_dep	_nama_dep
▶	KR1	Nando	DEV	Developer
	KR3	Vian	ENG	Engineer
	KR2	Fafa	PM	Product Manager
	KR5	Jova	DEV	Developer
	KR6	Kenny	FIN	Finance
	KR8	Veka	ENG	Engineer
*				

Penjelasan: Tampak bahwa saya berhasil menghapus karyawan bernama Tiara.

- Penerapan Enkapsulasi

```

4 references
public partial class Form1 : Form
{
    1 reference
    public Form1()
    {
        InitializeComponent();
    }

    private NpgsqlConnection conn;
    string connstring = "Host=localhost;Port=5432;Username=postgres;Password=informatika;Database=473804responsi";
    public DataTable dt;
    public static NpgsqlCommand cmd;
    private string sql = null;
    private DataGridViewRow r;

    4 references
    void LoadData()
    {
        try
        {

```

```

1 reference
private void btnInsert_Click(object sender, EventArgs e)
{
    try
    {
        conn.Open();
        sql = @"select * from kr_insert(:_nama, :_id_dep)";
        cmd = new NpgsqlCommand(sql, conn);
        cmd.Parameters.AddWithValue("_nama", txtNamaKaryawan.Text);
        cmd.Parameters.AddWithValue("_id_dep", cbDepartemenKaryawan.SelectedItem.ToString());
        if ((int)cmd.ExecuteScalar() == 1)
        {
            MessageBox.Show("Data Berhasil Diinputkan", "Success!", MessageBoxButtons.OK, MessageBoxIcon.Information);
            conn.Close();
            LoadData();
            txtNamaKaryawan.Text = cbDepartemenKaryawan.Text = null;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error: " + ex.Message, "Insert FAIL!!", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

Penjelasan: Tampak bahwa saya berhasil menerapkan enkapsulasi pada fungsi-fungsi dan variabel-variabel yang ada di program ini. Variabel dan fungsi yang private hanya bisa digunakan di class Form1.cs ini, tidak bisa digunakan di class yang lain supaya menjaga keamanan datanya.

- Penerapan Inheritance

```

using Npgsql;
using System.Data;
using System.Xml.Linq;

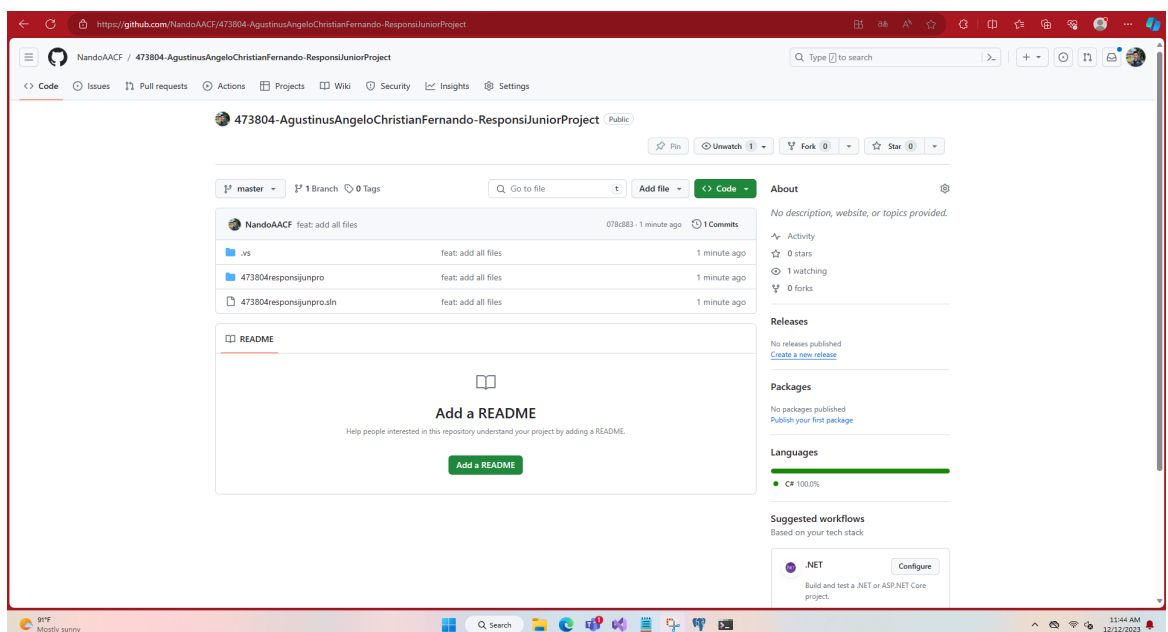
namespace _473804responsijunpro
{
    4 references
    public partial class Form1 : Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();

            private NpgsqlConnection conn;
            string connstring = "Host=localhost;Port
            public DataTable dt;
            public static NpgsqlCommand cmd;
            private string sql = null;
            private DataGridViewRow r;
    }
}
4 references

```

Penjelasan: Tampak bahwa saya berhasil menerapkan inheritance. Form1 merupakan turunan dari Form. Form sebagai induknya berisi property dan method yang sudah dibuat oleh framework .NET supaya bisa sebagai event handler elemen windows form. Inheritance ini bertujuan agar semua method dan property dari Form bisa diturunkan dan digunakan pada Form1.

- Bukti Repository Github



Link repository:

<https://github.com/NandoAACF/473804-AgustinusAngeloChristianFernando-ResponsiJuniorProject>