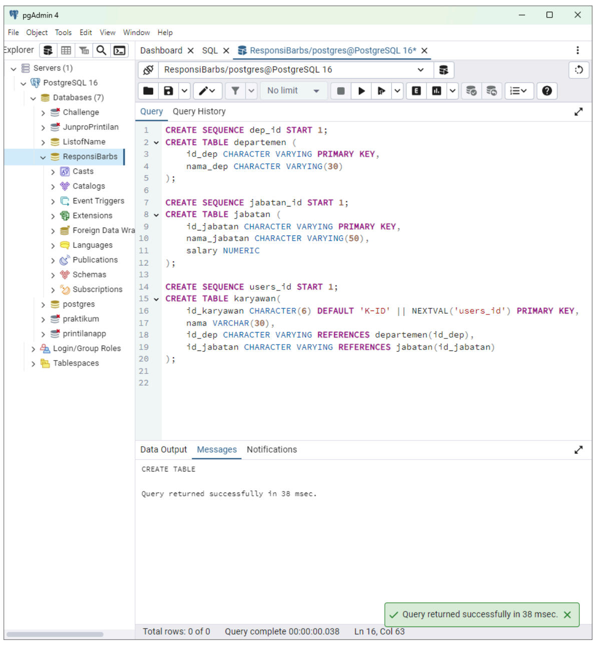
DOKUMENTASI RESPONSI JUNIOR PROJECT

Barbara Neanake Ajiesti
22/494495/TK/54238

1. Database & ERD

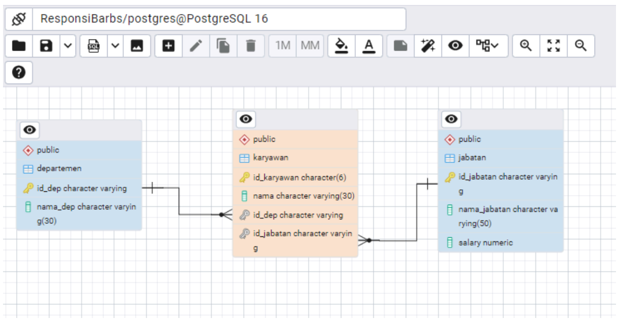
Beserta Code query database lengkap



```
CREATE SEQUENCE dep_id START 1;
CREATE TABLE departemen (
    id_dep CHARACTER VARYING PRIMARY KEY,
    nama_dep CHARACTER VARYING(30)
);

CREATE SEQUENCE jabatan_id START 1;
CREATE TABLE jabatan (
    id_jabatan CHARACTER VARYING PRIMARY KEY,
    nama_jabatan CHARACTER VARYING(50),
    salary NUMERIC
);

CREATE SEQUENCE users_id START 1;
CREATE TABLE karyawan(
    id_karyawan CHARACTER(6) DEFAULT 'K-ID' || NEXTVAL('users_id') PRIMARY KEY,
    nama VARCHAR(30),
    id_dep CHARACTER VARYING REFERENCES departemen(id_dep),
    id_jabatan CHARACTER VARYING REFERENCES jabatan(id_jabatan)
);
```

```
CREATE TABLE

Query returned successfully in 38 msec.
```

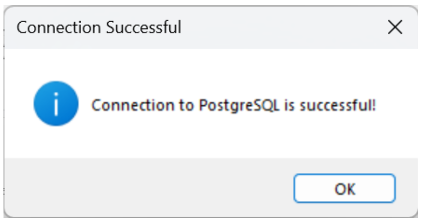
```
Query returned successfully in 38 msec.
Total rows: 0 of 0    Query complete 00:00:00.038    Ln 16, Col 63
```


ResponsiBarbs/postgres@PostgreSQL 16
public
departemen
id_dep character varying
nama_dep character varying(30)
public
karyawan
id_karyawan character(6)
nama character varying(30)
id_dep character varying
id_jabatan character varying
public
jabatan
id_jabatan character varying
nama_jabatan character varying(50)
salary numeric

```
CREATE SEQUENCE dep_id START 1;
CREATE TABLE departemen (
    id_dep CHARACTER VARYING PRIMARY KEY,
    nama_dep CHARACTER VARYING(30)
);

CREATE SEQUENCE jabatan_id START 1;
CREATE TABLE jabatan (
    id_jabatan SERIAL DEFAULT 'K-' ||
NEXTVAL('users_id') PRIMARY KEY,
    nama_jabatan VARCHAR(30),
    salary NUMERIC(12,2)
);

CREATE SEQUENCE users_id START 1;
CREATE TABLE karyawan (
    id_karyawan CHARACTER(6) DEFAULT 'K-'
|| NEXTVAL('users_id') PRIMARY KEY,
    nama VARCHAR(30),
    id_dep CHARACTER VARYING REFERENCES
departemen(id_dep)
);

--KaryawanSelect
CREATE OR REPLACE FUNCTION kr_select()
RETURNS TABLE (
    _id_karyawan CHARACTER,
    _nama CHARACTER VARYING,
    _id_dep CHARACTER VARYING,
    _nama_dep CHARACTER VARYING
)
LANGUAGE plpgsql AS $$
BEGIN
    RETURN QUERY
    SELECT
        karyawan.id_karyawan,
        karyawan.nama,
        departemen.id_dep,
        departemen.nama_dep
    FROM departemen
    JOIN karyawan ON karyawan.id_dep =
departemen.id_dep;
END;
$$;

--KaryawanInsert
CREATE OR REPLACE FUNCTION kr_insert(
    _nama CHARACTER VARYING,
    _id_dep CHARACTER VARYING
)
RETURNS INT AS $$
BEGIN
    INSERT INTO public.karyawan (nama,
id_dep)
    VALUES (_nama, _id_dep);

    RETURN 1;
EXCEPTION
    WHEN OTHERS THEN
        RETURN 0;
END;
$$ LANGUAGE plpgsql;

--KaryawanUpdate
CREATE OR REPLACE FUNCTION kr_update(
    _id_karyawan CHARACTER,
    _nama CHARACTER VARYING,
    _id_dep CHARACTER VARYING
)
RETURNS INT AS $$
BEGIN
    UPDATE karyawan
    SET
        nama = _nama,
        id_dep = _id_dep
    WHERE id_karyawan = _id_karyawan;

    RETURN 1;
EXCEPTION
    WHEN OTHERS THEN
        RETURN 0;
END;
$$ LANGUAGE plpgsql;

--DepartemenInsert
INSERT INTO departemen (id_dep, nama_dep)
VALUES
('HR', 'Human Resources'),
('ENG', 'Engineer'),
('DEV', 'Developer'),
('PM', 'Product Manager'),
('FIN', 'Finance');

--KaryawanDelete
CREATE OR REPLACE FUNCTION kr_delete(
    _id_karyawan CHARACTER VARYING
)
RETURNS INT AS $$
BEGIN
    DELETE FROM public.karyawan
    WHERE id_karyawan = _id_karyawan;

    RETURN 1;
EXCEPTION
    WHEN OTHERS THEN
        RETURN 0;
END;
$$ LANGUAGE plpgsql;
```

2. Fungsionalitas

A. Pop-up database terhubung

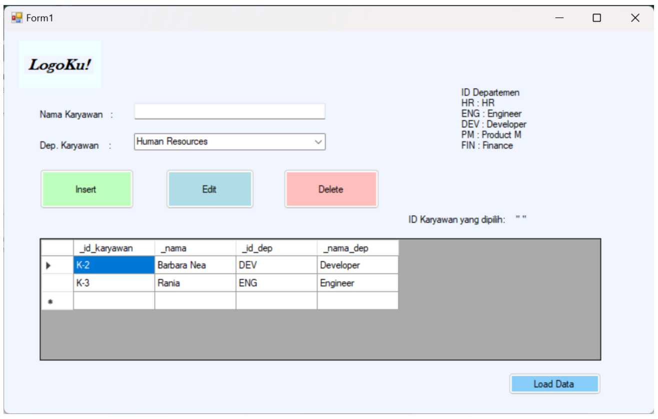


Connection Successful

Connection to PostgreSQL is successful!

OK

B. Form utama



Form1

LogoKu!

Nama Karyawan :

Dep. Karyawan : Human Resources

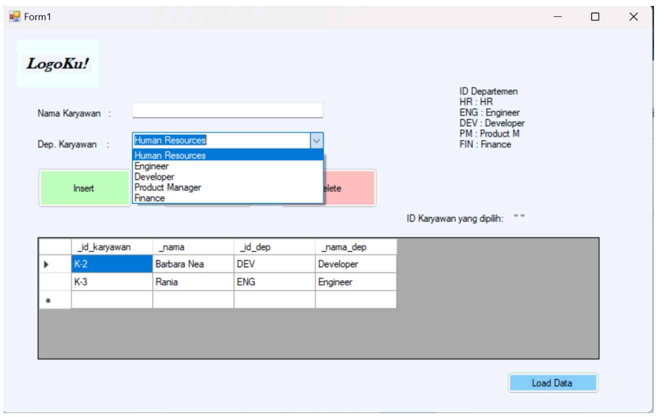
ID Departemen
HR : HR
ENG : Engineer
DEV : Developer
PM : Product M
FIN : Finance

Insert | Edit | Delete

ID Karyawan yang dipilih: " "

	_id_karyawan	_nama	_id_dep	_nama_dep
▶	K-2	Barbara Nea	DEV	Developer
	K-3	Rania	ENG	Engineer
*				

Load Data



Form1

LogoKu!

Nama Karyawan :

Dep. Karyawan : Human Resources

Human Resources
Engineer
Developer
Product Manager
Finance

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

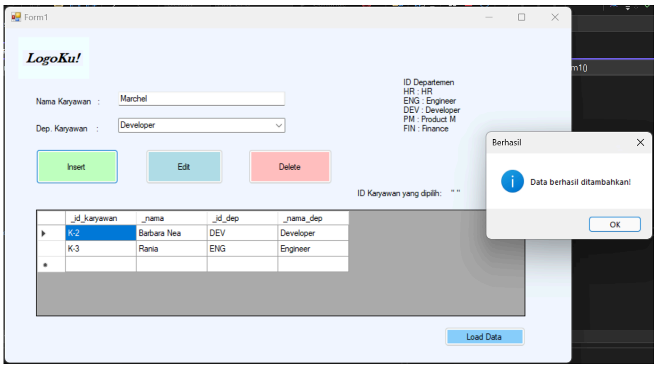
Insert | elete

ID Karyawan yang dipilih: " "

	_id_karyawan	_nama	_id_dep	_nama_dep
▶	K-2	Barbara Nea	DEV	Developer
	K-3	Rania	ENG	Engineer
*				

Load Data

C. Insert Data



Form1

LogoKu!

Nama Karyawan : Marchel

Dep. Karyawan : Developer

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

Insert | Edit | Delete

ID Karyawan yang dipilih: " "

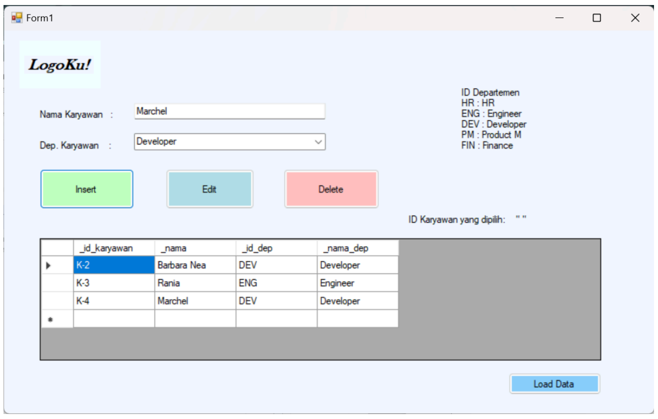
	_id_karyawan	_nama	_id_dep	_nama_dep
▶	K-2	Barbara Nea	DEV	Developer
	K-3	Rania	ENG	Engineer
*				

Load Data

Berhasil

Data berhasil ditambahkan!

OK



Form1

LogoKu!

Nama Karyawan : Marchel

Dep. Karyawan : Developer

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

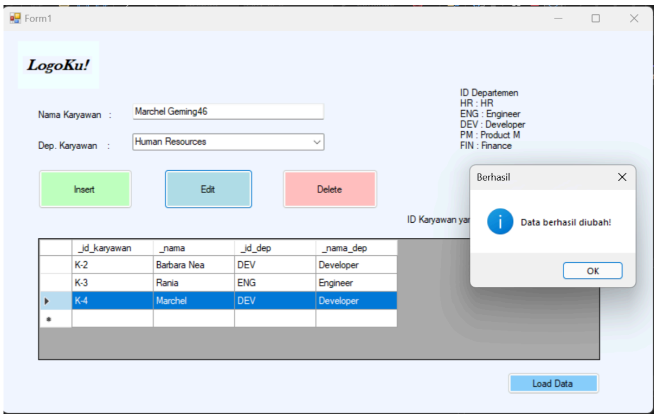
Insert | Edit | Delete

ID Karyawan yang dipilih: " "

	_id_karyawan	_nama	_id_dep	_nama_dep
▶	K-2	Barbara Nea	DEV	Developer
	K-3	Rania	ENG	Engineer
	K-4	Marchel	DEV	Developer
*				

Load Data

D. Edit Data



Form1

LogoKu!

Nama Karyawan : Marchel Geming46

Dep. Karyawan : Human Resources

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

Insert | Edit | Delete

ID Karyawan yar

	_id_karyawan	_nama	_id_dep	_nama_dep
	K-2	Barbara Nea	DEV	Developer
	K-3	Rania	ENG	Engineer
▶	K-4	Marchel	DEV	Developer
*				

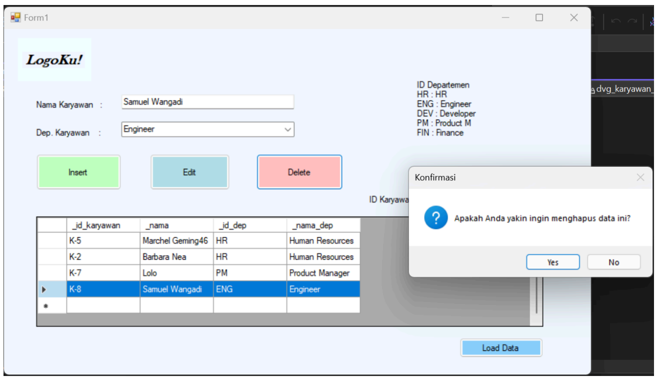
Load Data

Berhasil

Data berhasil diubah!

OK

E. Delete Data



Form1

LogoKu!

Nama Karyawan : Samuel Wangadi

Dep. Karyawan : Engineer

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

Insert | Edit | Delete

ID Karyawa

	_id_karyawan	_nama	_id_dep	_nama_dep
	K-5	Marchel Geming46	HR	Human Resources
	K-2	Barbara Nea	HR	Human Resources
	K-7	Lolo	PM	Product Manager
▶	K-8	Samuel Wangadi	ENG	Engineer
*				

Load Data

dvg_karyawan_

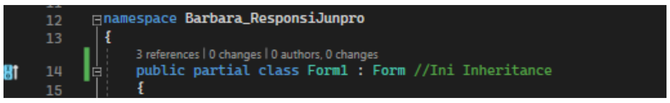
Konfirmasi

Apakah Anda yakin ingin menghapus data ini?

Yes | No

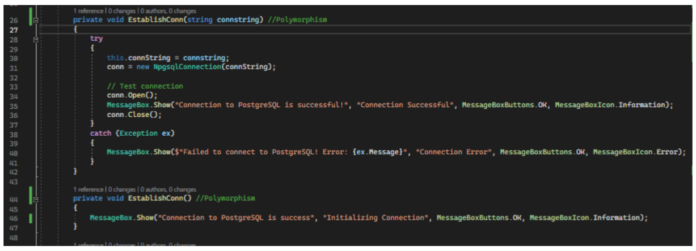
3. OOP Concept

A. Inheritance



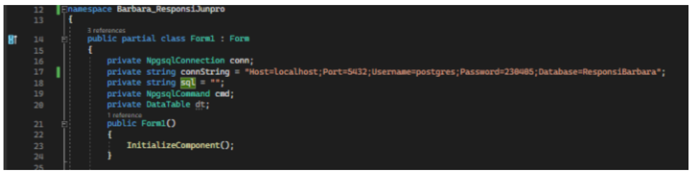
```
namespace Barbara_ResponsiJunpro
{
    3 references | 0 changes | 0 authors, 0 changes
    public partial class Form1 : Form //Ini Inheritance
    {
```

B. Polymorphism



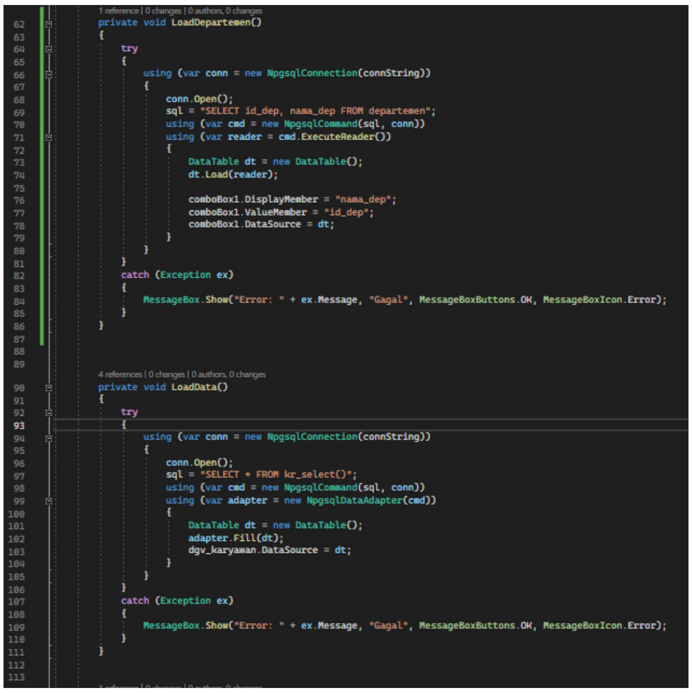
```
        1 reference | 0 changes | 0 authors, 0 changes
        private void EstablishConn(string connstring) //Polymorphism
        {
            try
            {
                this.connString = connstring;
                conn = new NpgsqlConnection(connString);

                // Test connection
                conn.Open();
                MessageBox.Show("Connection to PostgreSQL is successful!", "Connection Successful", MessageBoxButtons.OK, MessageBoxIcon.Information);
                conn.Close();
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Failed to connect to PostgreSQL! Error: {ex.Message}", "Connection Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        1 reference | 0 changes | 0 authors, 0 changes
        private void EstablishConn() //Polymorphism
        {
            MessageBox.Show("Connection to PostgreSQL is success", "Initializing Connection", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
```

C. Encapsulation



```
namespace Barbara_ResponsiJunpro
{
    3 references
    public partial class Form1 : Form
    {
        private NpgsqlConnection conn;
        private string connString = "Host=localhost;Port=5432;Username=postgres;Password=230405;Database=ResponsiBarbara";
        private string sql = "";
        private NpgsqlCommand cmd;
        private DataTable dt;
        1 reference
        public Form1()
        {
            InitializeComponent();
        }
```

D. Polymorphism



```
        1 reference | 0 changes | 0 authors, 0 changes
        private void LoadDepartemen()
        {
            try
            {
                using (var conn = new NpgsqlConnection(connString))
                {
                    conn.Open();
                    sql = "SELECT id_dep, nama_dep FROM departemen";
                    using (var cmd = new NpgsqlCommand(sql, conn))
                    using (var reader = cmd.ExecuteReader())
                    {
                        DataTable dt = new DataTable();
                        dt.Load(reader);

                        comboBox1.DisplayMember = "nama_dep";
                        comboBox1.ValueMember = "id_dep";
                        comboBox1.DataSource = dt;
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Error: " + ex.Message, "Gagal", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }


        4 references | 0 changes | 0 authors, 0 changes
        private void LoadData()
        {
            try
            {
                using (var conn = new NpgsqlConnection(connString))
                {
                    conn.Open();
                    sql = "SELECT * FROM kr_select()";
                    using (var cmd = new NpgsqlCommand(sql, conn))
                    using (var adapter = new NpgsqlDataAdapter(cmd))
                    {
                        DataTable dt = new DataTable();
                        adapter.Fill(dt);
                        dgv_karyawan.DataSource = dt;
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Error: " + ex.Message, "Gagal", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
```