

## Laboratorium Informatika

### Junior Project (Teknik Basis Data, Pemrograman Berbasis Objek)

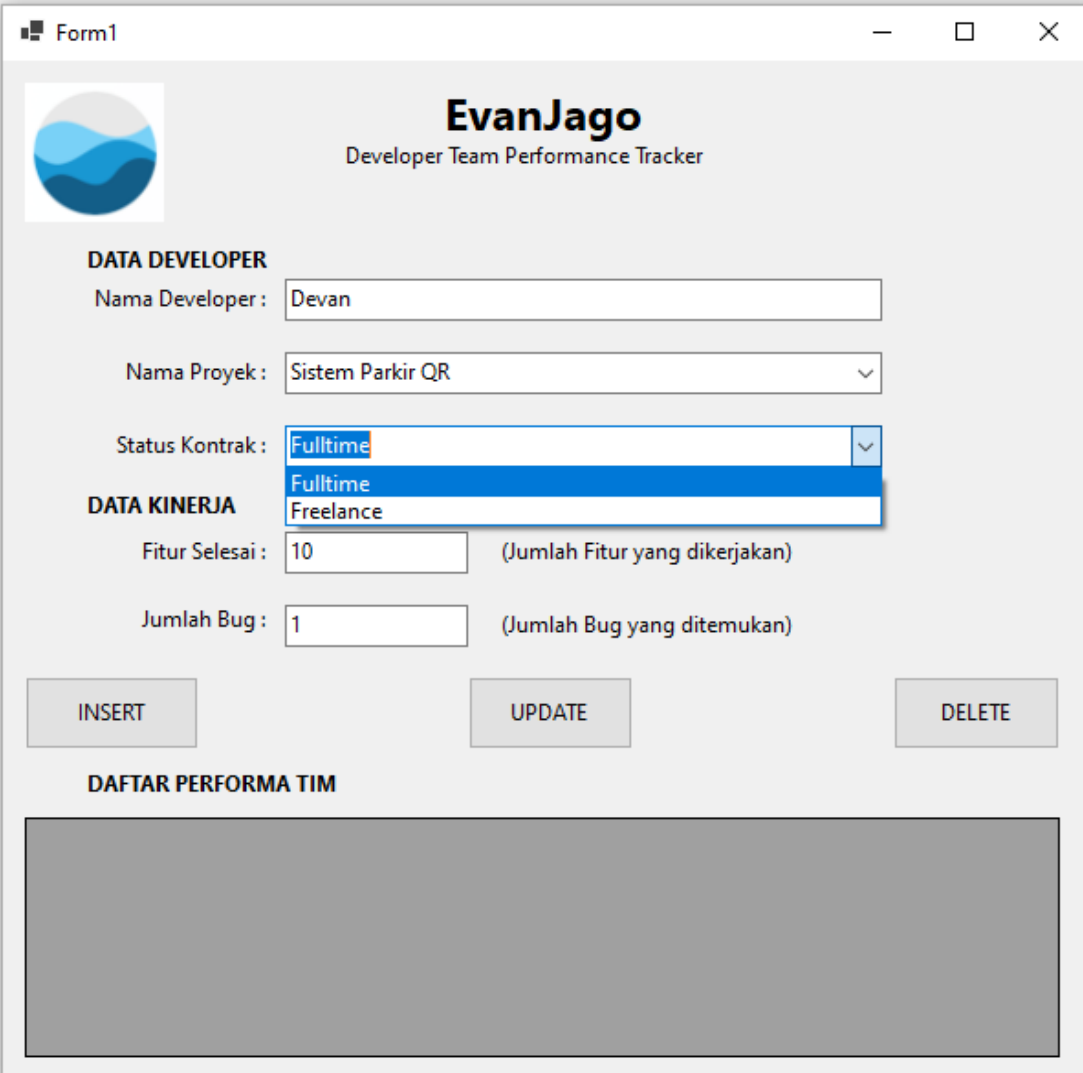
#### Worksheet Responsi

## RESPONSI

### TUGAS WAJIB

Link Github: [DevanWestley/ResponsiJunpro](https://github.com/DevanWestley/ResponsiJunpro)

Screenshot file Tampilan:



The screenshot shows a Windows application window titled "Form1". The application is "EvanJago Developer Team Performance Tracker". It features a profile picture of a person with blue hair. The form is divided into two main sections: "DATA DEVELOPER" and "DATA KINERJA".

**DATA DEVELOPER**

- Nama Developer :
- Nama Proyek :
- Status Kontrak :  (Dropdown menu is open showing "Fulltime" and "Freelance")

**DATA KINERJA**

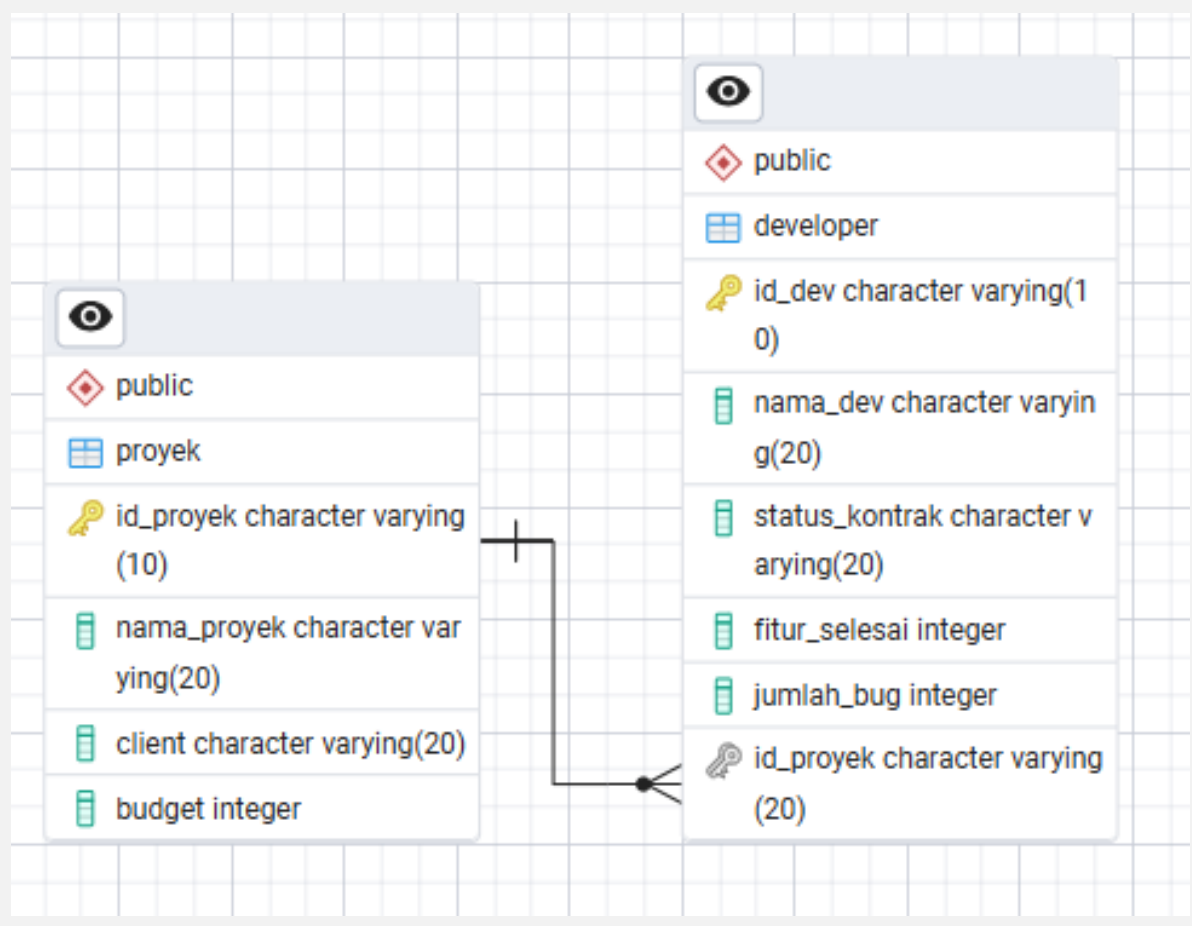
- Fitur Selesai :  (Jumlah Fitur yang dikerjakan)
- Jumlah Bug :  (Jumlah Bug yang ditemukan)

At the bottom of the form are three buttons: "INSERT", "UPDATE", and "DELETE". Below these buttons is a section titled "DAFTAR PERFORMA TIM" which contains a large, empty gray rectangular area.

Screenshot Konsep OOP (Enkapsulasi dan *Inheritance*):

```
namespace EvanJago
{
    1 reference
    public class Developer
    {
        private string nama;
        private string proyek;
        private string status;
        private string fitur;
        private string bug;
        0 references
        public string Nama { get { return nama; } set { nama = value; } }
        0 references
        public string Proyek { get { return proyek; } set { proyek = value; } }
        0 references
        public string Status { get { return status; } set { status = value; } }
        0 references
        public string Fitur { get { return fitur; } set { fitur = value; } }
        0 references
        public string Bug { get { return bug; } set { bug = value; } }
    }
}
```

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



## Implementasi Kode CRUD:

### Insert

```
// Insert
0 references
public void Insert(TextBox tbNama, ComboBox cbProyek, ComboBox cbStatus, TextBox tbFitur, TextBox tbBug)
{
    connection = new NpgsqlConnection(conn);
    try
    {
        connection.Open();
        string sql = @"SELECT * FROM emp_insert(:_nama, :_proyek, :_status, :_fitur, :_bug)";
        cmd = new NpgsqlCommand(sql, connection);
        cmd.Parameters.AddWithValue("_nama", tbNama.Text);
        cmd.Parameters.AddWithValue("_proyek", cbProyek.Text);
        cmd.Parameters.AddWithValue("_status", cbStatus.Text);
        cmd.Parameters.AddWithValue("_fitur", tbFitur.Text);
        cmd.Parameters.AddWithValue("_bug", tbBug.Text);
        if ((int)cmd.ExecuteScalar() == 1)
        {
            MessageBox.Show("Data berhasil ditambahkan");
            connection.Close();
            tbNama.Text = null;
            cbProyek.Text = null;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
        connection.Close();
    }
    finally { connection.Close(); }
}
```

### Update

```
// Edit
0 references
public void Edit(TextBox tbNama, ComboBox cbProyek, ComboBox cbStatus, TextBox tbFitur, TextBox tbBug)
{
    connection = new NpgsqlConnection(conn);
    if (row == null)
    {
        MessageBox.Show("Mohon pilih baris data yang ingin diubah");
    }

    try
    {
        connection.Open();
        string sql = @"SELECT * FROM emp_update(:_id, :_nama, :_proyek, :_status, :_fitur, :_bug)";
        cmd = new NpgsqlCommand(sql, connection);
        cmd.Parameters.AddWithValue("_id", row.Cells["_id_dev"].Value.ToString());
        cmd.Parameters.AddWithValue("_nama", tbNama.Text);
        cmd.Parameters.AddWithValue("_proyek", cbProyek.Text);
        cmd.Parameters.AddWithValue("_status", cbStatus.Text);
        cmd.Parameters.AddWithValue("_fitur", tbFitur.Text);
        cmd.Parameters.AddWithValue("_bug", tbBug.Text);

        if ((int)cmd.ExecuteScalar() == 1)
        {
            MessageBox.Show("Data berhasil diubah");
            connection.Close();
            tbNama.Text = null;
            cbProyek.Text = null;
            row = null;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
        connection.Close();
    }
    finally { connection.Close(); }
}
```

## Delete

```
// Delete
0 references
public void Delete(TextBox tbNama, ComboBox cbProyek, ComboBox cbStatus, TextBox tbFitur, TextBox tbBug)
{
    connection = new NpgsqlConnection(conn);
    if (row == null)
    {
        MessageBox.Show("Mohon pilih baris data yang ingin dihapus");
    }

    try
    {
        connection.Open();
        string sql = @"select * from emp_delete(:_id)";
        cmd = new NpgsqlCommand(sql, connection);
        cmd.Parameters.AddWithValue("_id", row.Cells["_id_dev"].Value.ToString());

        if ((int)cmd.ExecuteScalar() == 1)
        {
            MessageBox.Show("Data berhasil dihapus");
            connection.Close();
            tbNama.Text = null;
            cbProyek.Text = null;
            row = null;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
        connection.Close();
    }
    finally { connection.Close(); }
}
```

## TUGAS OPSIONAL

### Screenshot Konsep OOP (Polimorfisme dan Abstraksi):

```
namespace EvanJago
{
    1 reference
    internal class DeveloperRepository : Developer
    {
        private const string conn = "Host=localhost;Port=5432;Username=postgres;Password=usti;Database=EvanJago";
        private static NpgsqlConnection connection;
        private static NpgsqlCommand cmd;
        private static DataTable dt;

        private DataGridView dgvData;
        private DataGridViewRow row;

        0 references
        public DataGridViewRow Row { set { row = value; } }

        0 references
        public DeveloperRepository(DataGridView _dgvData)
        {
            dgvData = _dgvData;
        }
    }
}
```

### Screenshot Fungsi pada Postgres:

Query Query History

```
1 INSERT INTO developer (id_dev, nama_dev, status_kontrak, fitur_selesai, jumlah_bug)
2 VALUES
3 ('1', 'Devan', 'Fulltime', '10', '2'),
4 ('2', 'Westley', 'Freelance', '5', '3'),
5 ('3', 'Mulat', 'Fulltime', '9', '1'),
6 ('4', 'Moses', 'Fulltime', '3', '7'),
7 ('5', 'Nia', 'Freelance', '10', '10');
```

Data Output Messages Notifications

INSERT 0 5

Query returned successfully in 61 msec.

Query Query History

```
1 SELECT *
2 FROM developer
3 WHERE fitur_selesai > 3;
```

Data Output Messages Notifications

≡+ 📄 ▼ 📋 ▼ 🗑️ 🗄️ ⬇️ 📈 SQL

	id_dev [PK] character varying (10) ✎	nama_dev character varying (20) ✎	status_kontrak character varying (20) ✎	fitur_selesai integer ✎	jumlah_bug integer ✎
1	1	Devan	Fulltime	10	2
2	2	Westley	Freelance	5	3
3	3	Mulat	Fulltime	9	1
4	5	Nia	Freelance	10	10

Query Query History

```
1 SELECT *
2 FROM developer
3 WHERE status_kontrak = 'Freelance'
4 AND jumlah_bug > 5;
```

Data Output Messages Notifications

≡+ 📄 ▼ 📋 ▼ 🗑️ 🗄️ ⬇️ 📈 SQL

	id_dev [PK] character varying (10) ✎	nama_dev character varying (20) ✎	status_kontrak character varying (20) ✎	fitur_selesai integer ✎	jumlah_bug integer ✎
1	5	Nia	Freelance	10	10