

## Laboratorium Informatika

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

#### Worksheet Responsi

#### RESPONSI

##### IDENTITAS

Nama : Chaira Nastya Warestri

NIM : 23/514942/TK/56550

Ruang : Labdas B1

##### TUGAS WAJIB BAGIAN A

Link Github (**Wajib Publik dan bisa dibuka**): [chairanastya/514942-ChairaNastyaW-ResponsiJuniorProject](https://github.com/chairanastya/514942-ChairaNastyaW-ResponsiJuniorProject)

Screenshot aplikasi (dengan data tertampil):

Form1



## Meong App

### Developer Team Performance Tracker

**Data Developer**

Nama Developer	<input type="text" value="lala"/>
Pilih Proyek	<input type="text" value="Sistem Parkir QR"/>
Status Kontrak	<input type="text" value="Freelance"/>

**Data Kinerja**

Jumlah Fitur Selesai	<input type="text" value="3"/>	Masukkan angka jumlah fitur selesai
Jumlah Bug	<input type="text" value="0"/>	Masukkan angka jumlah bug yang terjadi

[Insert](#)   [Update](#)   [Delete](#)

**Daftar Performa Tim**

	Nama	Proyek	Status	Fitur	Bug	Skor
▶	hayo	Web Company ...	Full Time	5	1	45
▶	<b>lala</b>	Sistem Parkir QR	<b>Freelance</b>	<b>3</b>	<b>0</b>	<b>100</b>
▶	oya	IoT Agriculture ...	Full Time	8	2	70
▶	hai	AI Fraud Detect...	Full Time	6	1	55
▶	miau	Web Company ...	Freelance	8	2	83.33
▶	uwawaw	Sistem Parkir QR	Full Time	10	2	90
...				16	8	160

Screenshot Konsep OOP (Enkapsulasi dan *Inheritance*):

```

20 // ENCAPSULATION
21 1 reference
22
23 private void Form1_Load(object sender, EventArgs e)
24 {
25     LoadProyekCombo();
26     LoadDataDeveloper();
27
28     cbStatus.Items.Clear();
29     cbStatus.Items.Add("Full Time");
30     cbStatus.Items.Add("Freelance");
31     cbStatus.SelectedIndex = -1;
32 }
A reference

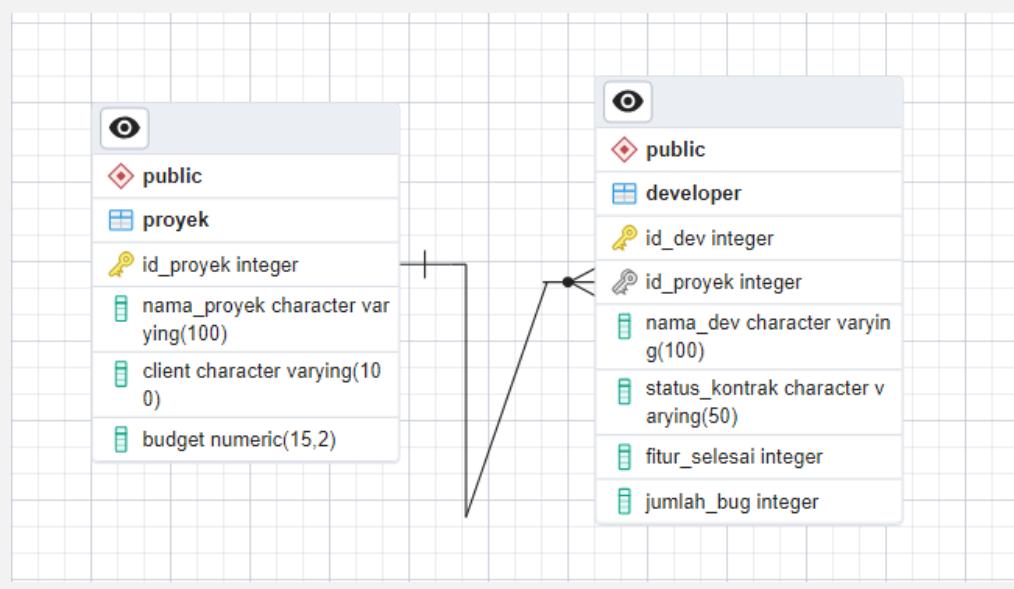
```

```

26 // 2. INHERITANCE & POLYMORPHISM
27 4 references
28 public class FullTimeDeveloper : Developer
29 {
30     3 references
31     public FullTimeDeveloper() { Status = "Full Time"; }
32     3 references
33     public override double HitungSkor()
34     {
35         double skor = (10 * Fitur) - (5 * Bug);
36         return skor < 0 ? 0 : skor;
37     }
38     5 references
39     public override decimal HitungGaji()
40     {
41         decimal gajiPokok = 5000000;
42         decimal bonus = (decimal)HitungSkor() * 20000;
43         return gajiPokok + bonus;
44     }
45
46 // 3. INHERITANCE & POLYMORPHISM
47 4 references
48 public class FreelanceDeveloper : Developer
49 {
50     3 references
51     public FreelanceDeveloper() { Status = "Freelance"; }
52     3 references
53     public override double HitungSkor()
54     {
55         if (Fitur == 0) return 0;
56         double ratio = (2.0 * Bug) / (3.0 * Fitur);
57         double skor = 100 * (1 - ratio);
58         return skor < 0 ? 0 : skor;
59     }
60     5 references
61     public override decimal HitungGaji()
62     {
63         double skor = HitungSkor();
64         decimal tarifPerFitur;
65         if (skor >= 80) tarifPerFitur = 500000;
66         else if (skor >= 50) tarifPerFitur = 400000;
67         else tarifPerFitur = 250000;
68         return tarifPerFitur * Fitur;
69     }
70 }

```

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



Implementasi Kode CRUD (Insert, Update, Delete):

Insert

Form1



## Meong App

### Developer Team Performance Tracker

#### Data Developer

Nama Developer

fufufafa

Pilih Proyek

Sistem Parkir QR

Status Kontrak

Full Time

#### Data Kinerja

Jumlah Fitur Selesai

9

Masukkan angka jumlah fitur selesai

Jumlah Bug

1

Masukkan angka jumlah bug yang terjadi

Insert

Update

Delete

#### Daftar Performa Tim

	Nama	Proyek	Status	Fitur	Bug	Skor
	lala	Sistem Parkir QR	Freelance	3	0	100
	oya	IoT Agriculture ...	Full Time	8	2	70
	hai	AI Fraud Detect...	Full Time	6	2	50
	miau	Web Company ...	Freelance	8	2	83.33
	uwawaw	Sistem Parkir QR	Full Time	10	2	90
▶	fufufafa	Sistem Parkir QR	Full Time	9	1	85

Update:

Form1



## Meong App

### Developer Team Performance Tracker

**Data Developer**

Nama Developer	<input type="text"/>
Pilih Proyek	<input type="text"/>
Status Kontrak	<input type="text"/>

**Data Kinerja**

Jumlah Fitur Selesai	<input type="text"/>	Masukkan angka jumlah fitur selesai
Jumlah Bug	<input type="text"/>	Masukkan angka jumlah bug yang terjadi

**Daftar Performa Tim**

	Nama	Proyek	Status	Fitur	Bug	Skor
1	hayo	Web Company ...	Full Time	5	1	45
2	lala	Sistem Parkir QR	Freelance	3	0	100
3	oya	IoT Agriculture ...	Full Time	8		
4	hai	AI Fraud Detect...	Full Time	6		
5	miau	Web Company ...	Freelance	8		
6	uwawaw	Sistem Parkir QR	Full Time	10		
7	E.E.E.E.	AI Fraud Detect...	Full Time	16		

Sukses

i Data berhasil diupdate!

OK

Insert      Update      Delete

Delete:

Form1



## Meong App

### Developer Team Performance Tracker

**Data Developer**

Nama Developer	fufufafa
Pilih Proyek	AI Fraud Detection
Status Kontrak	Full Time

**Data Kinerja**

Jumlah Fitur Selesai	16	Masukkan
Jumlah Bug	0	Masukkan

Confirm

Hapus?

Daftar Performa Tim

	Nama	Proyek	Status	Fitur	Bug	Skor
	lala	Sistem Parkir QR	Freelance	3	0	100
	oya	IoT Agriculture ...	Full Time	8	2	70
	hai	AI Fraud Detect...	Full Time	6	2	50
	miau	Web Company ...	Freelance	8	2	83.33
	uwawaw	Sistem Parkir QR	Full Time	10	2	90
▶	fufufafa	AI Fraud Detect...	Full Time	16	0	160

## TUGAS CHALLENGE BAGIAN B

### Screenshot kode *abstraction*

```
// 1. ABSTRACTION
8 references
public abstract class Developer
{
    5 references
    public int Id { get; set; }
    3 references
    public string Nama { get; set; }
    3 references
    public int IdProyek { get; set; }
    2 references
    public string NamaProyek { get; set; }
    4 references
    public string Status { get; set; }
    11 references
    public int Fitur { get; set; }
    9 references
    public int Bug { get; set; }

    // Abstract Methods (Polymorphism)
    5 references
    public abstract double HitungSkor();
    6 references
    public abstract decimal HitungGaji();
}
```

### Screenshot kode *polymorphism* implementasi hitung skor dan gaji (serta *error handling*)

```
// 2. INHERITANCE & POLYMORPHISM
4 references
public class FullTimeDeveloper : Developer
{
    3 references
    public FullTimeDeveloper() { Status = "Full Time"; }
    3 references
    public override double HitungSkor()
    {
        double skor = (10 * Fitur) - (5 * Bug);
        return skor < 0 ? 0 : skor;
    }
    5 references
    public override decimal HitungGaji()
    {
        decimal gajiPokok = 5000000;
        decimal bonus = (decimal)HitungSkor() * 20000;
        return gajiPokok + bonus;
    }
}

// 3. INHERITANCE & POLYMORPHISM
4 references
public class FreelanceDeveloper : Developer
{
    3 references
    public FreelanceDeveloper() { Status = "Freelance"; }
    3 references
    public override double HitungSkor()
    {
        if (Fitur == 0) return 0;
        double ratio = (2.0 * Bug) / (3.0 * Fitur);
        double skor = 100 * (1 - ratio);
        return skor < 0 ? 0 : skor;
    }
    5 references
    public override decimal HitungGaji()
    {
        double skor = HitungSkor();
        decimal tarifPerFitur;
        if (skor >= 80) tarifPerFitur = 500000;
        else if (skor >= 50) tarifPerFitur = 400000;
        else tarifPerFitur = 250000;
        return tarifPerFitur * Fitur;
    }
}
```

## Screenshot main code yang terubah untuk menampilkan skor dan gaji

```
private void LoadDataDeveloper()
{
    listDevs.Clear();
    try
    {
        using (NpgsqlConnection conn = new NpgsqlConnection(connString))
        {
            conn.Open();
            string sql = @"SELECT d.*, p.nama_proyek
                           FROM developer d
                           JOIN proyek p ON d.id_proyek = p.id_proyek
                           ORDER BY d.id_dev ASC";

            using (NpgsqlCommand cmd = new NpgsqlCommand(sql, conn))
            using (NpgsqlDataReader reader = cmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    string status = reader["status_kontrak"].ToString();
                    Developer dev;

                    if (status == "Full Time") dev = new FullTimeDeveloper();
                    else dev = new FreelanceDeveloper();

                    // Isi data dari DB
                    dev.Id = Convert.ToInt32(reader["id_dev"]);
                    dev.IdProyek = Convert.ToInt32(reader["id_proyek"]);
                    dev.NamaProyek = reader["nama_proyek"].ToString();
                    dev>Nama = reader["nama_dev"].ToString();
                    dev.Fitur = Convert.ToInt32(reader["fitur_selesai"]);
                    dev.Bug = Convert.ToInt32(reader["jumlah_bug"]);

                    listDevs.Add(dev);
                }
            }
            var displayList = new List<object>();
            foreach (var d in listDevs)
            {
                displayList.Add(new
                {
                    ID = d.Id,
                    Nama = d>Nama,
                    Proyek = d.NamaProyek,
                    Status = d.Status,
                    Fitur = d.Fitur,
                    Bug = d.Bug,
                    Skor = d.HitungSkor().ToString("0.##"),
                    TotalGaji = d.HitungGaji().ToString("N0")
                });
            }
            dgvData.DataSource = null;
            dgvData.DataSource = displayList;

            dgvData.Columns["ID"].Visible = false;
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error Load: " + ex.Message);
        }
    }
}

2 references
private bool IsBudgetSafe(int idProyek, decimal gajiDeveloperBaru, int ignoreDevId = -1)
{
    decimal totalGajiExisting = 0;
    decimal budgetProyek = 0;

    using (NpgsqlConnection conn = new NpgsqlConnection(connString))
    {
        conn.Open();

        string sqlBudget = "SELECT budget FROM proyek WHERE id_proyek = @id";
        using (NpgsqlCommand cmd = new NpgsqlCommand(sqlBudget, conn))
        {
            cmd.Parameters.AddWithValue("@id", idProyek);
            object result = cmd.ExecuteScalar();
            if (result != null) budgetProyek = Convert.ToDecimal(result);
        }

        foreach (var dev in listDevs)
        {
            // Hanya hitung developer di proyek yang sama, KECUALI developer yang sedang diedit
            if (dev.IdProyek == idProyek && dev.Id != ignoreDevId)
            {
                totalGajiExisting += dev.HitungGaji();
            }
        }
    }

    decimal totalPerkiraan = totalGajiExisting + gajiDeveloperBaru;

    if (totalPerkiraan > budgetProyek)
    {
        MessageBox.Show($"GAGAL! Project Overbudget.\nBudget: {budgetProyek:N0}\nTotal Gaji (termasuk baru): {totalPerkiraan:N0}\nSelisih: {(totalPerkiraan - budgetProyek):N0}");
        return false;
    }
    return true;
}
```

Screenshot aplikasi dengan hasil perhitungan tertampil

Form1



# Meong App

## Developer Team Performance Tracker

**Data Developer**

Nama Developer	<input type="text"/>
Pilih Proyek	<input type="text"/> Web Company Profile
Status Kontrak	<input type="text"/>

**Data Kinerja**

Jumlah Fitur Selesai	<input type="text"/>	Masukkan angka jumlah fitur selesai
Jumlah Bug	<input type="text"/>	Masukkan angka jumlah bug yang terjadi

**Daftar Performa Tim**

	Nama	Proyek	Status	Fitur	Bug	Skor	TotalGaji
▶	hayo	Web Company ...	Full Time	5	1	45	5,900,000
	lala	Sistem Parkir QR	Freelance	3	0	100	1,500,000
	oya	IoT Agriculture ...	Full Time	8	2	70	6,400,000
	hai	AI Fraud Detect...	Full Time	6	2	50	6,000,000
	miau	Web Company ...	Freelance	8	2	83.33	4,000,000
	uwawaw	Sistem Parkir QR	Full Time	10	2	90	6,800,000
	fufufafa	Sistem Parkir QR	Full Time	9	1	85	6,700,000

Insert      Update      Delete

## Screenshot fungsi pada PostgreSQL

### Fungsi Update Deeveloper

```
49 CREATE OR REPLACE FUNCTION update_developer(
50     _id_dev INT,
51     _id_proyek INT,
52     _nama VARCHAR,
53     _status VARCHAR,
54     _fitur INT,
55     _bug INT
56 )
57 RETURNS VOID AS $$
```

58▼ BEGIN

```
59     UPDATE developer
60     SET id_proyek = _id_proyek,
61         nama_dev = _nama,
62         status_kontrak = _status,
63         fitur_selesai = _fitur,
64         jumlah_bug = _bug
65     WHERE id_dev = _id_dev;
```

```
66 END;
67 $$ LANGUAGE plpgsql;
```

### Fungsi Insert Data Developer

```
CREATE OR REPLACE FUNCTION insert_developer(
    _id_proyek INT,
    _nama VARCHAR,
    _status VARCHAR,
    _fitur INT,
    _bug INT
)
RETURNS VOID AS $$
```

BEGIN

```
    INSERT INTO developer (id_proyek, nama_dev, status_kontrak, fitur_selesai, jumlah_bug)
    VALUES (_id_proyek, _nama, _status, _fitur, _bug);
```

```
END;
$$ LANGUAGE plpgsql;
```