**PROPOSAL UAS**

**DATABASE SEDERHANA KULI ONLINE**



Disusun Oleh :

Angga Maulana Yusup (20210040002)

Andhika Oktasandira (20210040054)

Eril Mahditira (20210040073)

Heru Subakti (20210040071)

Reza Ramdani Irawan (20210040040)

**PROGRAM STUDI TEKNIK INFORMATIKA**

**UNIVERSITAS NUSA PUTRA SUKABUMI**

**Jl. Raya Cibatu Cisaat No.21, Cibolang Kaler, Kec. Cisaat, Kabupaten Sukabumi, Jawa Barat 43152**

2023

### **KATA PENGANTAR**

Puji dan syukur saya panjatkan kepada Allah Swt. atas ridho-Nya saya dapat menyelesaikan penyusunan Proposal ini. Adapun judul Propsal yang kami buat adalah "Database Sederhana Kuli Online".  
 Propsal ini diajukan untuk memenuhi tugas Proyek mata kuliah Pemprograman Berorientasi Objek program studi Teknik Informatika Universitas Nusa Putra Sukabumi.   
 Semoga segala kebaikan dan pertolongan semuanya mendapat berkah dari Allah Swt. dan akhirnya kami menyadari bahwa propsal ini masih jauh dari kata sempurna, karena keterbatasan ilmu yang saya miliki. Untuk itu saya dengan kerendahan hati mengharapkan saran dan kritik yang sifatnya membangun dari semua pihak untuk dijadikan pembelajaran oleh kami

Sukabumi, 24 Januari 2023

Penulis

### **ABSTRAK**

Kami saat berdiskusi menyadari bahwa dalam dunia kerja pertukangan yang ada di setiap kota maupun kampung kesulitan akan mencari pekerjaan dalam keahliannya sehingga kami memiliki sebuah ide untuk membangun sebuah aplikasi yang bertujuan untuk membantu para tukang tersebut. Kami namakan aplikasi tersebut KULON (Kuli Online), untuk sekarang kami membuat sebuah database sederhana untuk media latih kami dan untuk memenuhi tugas proyek ulangan akhir semester. Dalam basis data yang kami buat ini berisi tentang pengguna, pekerja dan pembayaran. Dimana semua itu bertujuan agar pengguna aplikasi yang membutuhkan jasa tukang dapat mendapatkan tukang dengan mudah dan bagi pekerja atau tukang dapat mencari pekerjaan dengan mudah juga. Kuli Online adalah sebuah aplikasi yang memungkinkan pengguna untuk menemukan dan memesan kuli dan jasa lainnya.

Aplikasi ini memiliki basis data yang mencakup informasi tentang kuli, jasa, pengguna, pemesanan, dan lainnya. Ini membutuhkan database yang dapat menangani operasi CRUD seperti membuat, membaca, memperbarui, dan menghapus data. Database ini harus menyimpan data dalam struktur yang efisien dan handal untuk memastikan bahwa aplikasi berjalan dengan lancar. Database ini harus menggunakan enkripsi untuk menjaga keamanan data dan menyediakan pembaruan secara berkala untuk menjaga keamanan dan kinerja aplikasi. Ini juga harus memiliki dukungan yang kuat untuk penggunaan masa depan.

### **DAFTAR ISI**

[**KATA PENGANTAR** 2](#_Toc126038710)

[**ABSTRAK** ........................................................................................................................3](#_Toc126038711)

[**DAFTAR ISI** 4](#_Toc126038712)

[**BAB I** 5](#_Toc126038713)

[**PENDAHULUAN** 5](#_Toc126038714)

[**1.1** **Latar Belakang** 5](#_Toc126038715)

[**1.2** **Rumusan Masalah** 5](#_Toc126038716)

[**1.4** **Tujuan** 5](#_Toc126038717)

[**BAB II** 6](#_Toc126038718)

[**STUDI PUSTAKA** 6](#_Toc126038719)

[**2.1** **Bahasa Pemrograman java** 6](#_Toc126038720)

[**2.2** **Database** 6](#_Toc126038721)

[**2.3** **Metode CRUD** 6](#_Toc126038722)

[**2.4** **My SQL** 6](#_Toc126038723)

[**BAB III** 7](#_Toc126038724)

[**METHODOLOGI** 7](#_Toc126038725)

[**3.1** **Metode Pembuatan Program** 7](#_Toc126038726)

[**BAB IV** 8](#_Toc126038727)

[**HASIL DAN DISKUSI** 8](#_Toc126038728)

[**4.1 Kode Program** 8](#_Toc126038729)

[**4.2** **Pengujian Program** 23](#_Toc126038730)

[**BAB V** 27](#_Toc126038731)

[**PENUTUP** 27](#_Toc126038732)

[**5.1** **Kesimpulan** 27](#_Toc126038733)

[**DAFTAR PUSTAKA** 28](#_Toc126038734)

# **BAB I**

## **PENDAHULUAN**

### **Latar Belakang**

Java adalah bahasa pemrograman yang populer, yang digunakan untuk mengembangkan aplikasi berbasis web, desktop dan mobile. Salah satu fitur yang umum diimplementasikan dalam aplikasi Java adalah operasi CRUD (Create, Read, Update, Delete). CRUD memungkinkan pengguna untuk membuat, membaca, memperbarui dan menghapus data dari aplikasi. Hal ini sangat penting ketika membangun aplikasi berbasis data, seperti sistem manajemen penjualan atau sistem manajemen inventori.

Database Kuli Online Sederhana Menggunakan Metode CRUD." Makalah ini menyajikan ikhtisar database freelancer online sederhana yang dirancang menggunakan metode CRUD (Create, Read, Update, and Delete). Tujuan dari database ini adalah untuk menyediakan platform bagi freelancer untuk memamerkan keahlian mereka dan terhubung dengan klien potensial.

Metode CRUD digunakan untuk melakukan operasi basis data dasar, seperti membuat dan mengelola profil, mencari freelancer, dan memperbarui informasi. Penelitian ini bertujuan untuk menunjukkan keefektifan dan manfaat penggunaan metode CRUD dalam membuat database freelancer online yang fungsional dan user-friendly.

### **Rumusan Masalah**

Database kuli Online Sederhana Menggunakan Metode CRUD." Makalah ini membahas masalah membuat database yang sederhana, efisien, dan ramah pengguna untuk freelancer online. Basis data harus memungkinkan pengguna untuk membuat dan mengelola profil mereka, mencari freelancer berdasarkan keterampilan, dan mengakses informasi yang relevan tentang pengalaman kerja mereka.

1. **Batasan Masalah**
2. Aplikasi database sederhana untuk kuli online.
3. Menggunakan metode CRUD (Create, Read, Update, Delete)
4. Tidak ada spesifikasi tambahan yang diberikan.

### **Tujuan**

Tujuan dari proposal ini adalah untuk mengembangkan aplikasi Java yang melakukan operasi CRUD. Aplikasi ini akan mampu membuat, membaca, memperbarui dan menghapus data dari berbagai sumber data, seperti basis data relasional, basis data MySQL Localhost. Aplikasi ini akan memungkinkan pengguna untuk dengan cepat dan mudah mengakses dan memanipulasi data yang tersimpan di sumber data.

# **BAB II**

## **STUDI PUSTAKA**

### **Bahasa Pemrograman java**

Bahasa Java adalah tahap penting dalam proses pengembangan aplikasi menggunakan bahasa Java. Studi ini bertujuan untuk memahami konsep dan aplikasi Java dalam pemrograman.

### **Database**

Database merupakan tahap penting dalam proses pengembangan aplikasi database. Ini melibatkan penelitian dan analisis sumber daya yang ada untuk memahami konsep dan teknologi yang relevan dengan aplikasi yang akan dibuat.

### **Metode CRUD**

metode CRUD merupakan bagian dari studi pustaka dalam pengembangan aplikasi database. Studi ini bertujuan untuk memahami konsep dan aplikasi metode CRUD dalam pengelolaan data.

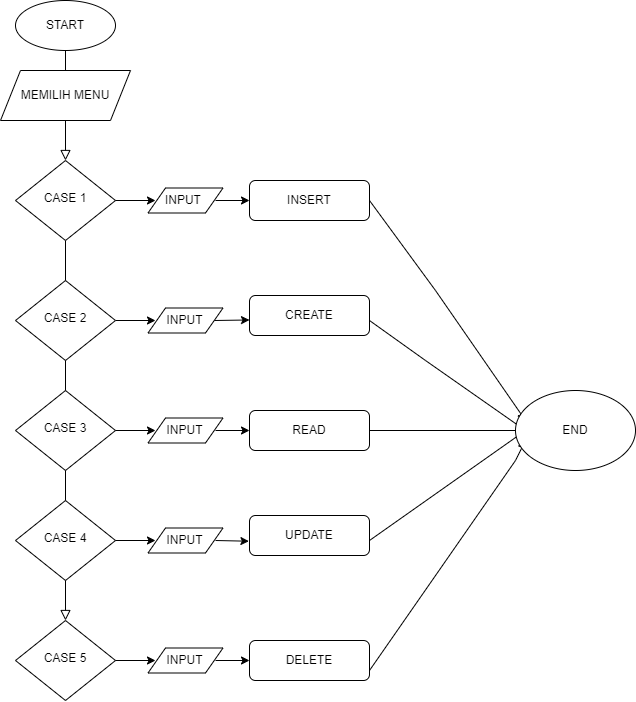
### **My SQL**

MySQL adalah tahap penting dalam proses pengembangan aplikasi database menggunakan MySQL. Studi ini bertujuan untuk memahami konsep dan aplikasi MySQL dalam pengelolaan data.

# **BAB III**

## **METHODOLOGI**

### **Metode Pembuatan Program**

1. Metode Pembuatan Program kita menggunakan Metode CRUD adalah akronim untuk Create, Read, Update, dan Delete, yaitu operasi dasar dalam pengelolaan data pada basis data. Dalam pembuatan program, metode CRUD dapat digunakan sebagai salah satu metodologi untuk membuat aplikasi yang berinteraksi dengan basis data. Metode ini mengikuti tahapan-tahapan berikut:
2. Menentukan Entitas: Identifikasi entitas yang akan diterima, disimpan, dan dikelola oleh aplikasi.
3. Penentuan Atribut: Penentuan atribut yang akan disimpan untuk setiap entitas.
4. Implementasi Create: Implementasi proses untuk menambahkan entitas baru ke basis data.
5. Implementasi Read: Implementasi proses untuk membaca entitas dari basis data.
6. Implementasi Update: Implementasi proses untuk memperbarui entitas yang ada di basis data.
7. Implementasi Delete: Implementasi proses untuk menghapus entitas dari basis data.
8. Pengujian dan Validasi: Uji aplikasi untuk memastikan bahwa setiap operasi CRUD berfungsi dengan benar.
9. Flowchart

# **BAB IV**

### **HASIL DAN DISKUSI**

### **4.1 Kode Program**

1. **Pengguna**

package S3UAS;

import java.sql.\*;

import java.io.BufferedReader;

import java.io.InputStreamReader;

public class pengguna{

static final String JDBC\_DRIVER = "com.mysql.jdbc.Driver";

static final String DB\_URL = "jdbc:mysql://localhost/kuli\_online";

static final String USER = "root";

static final String PASS = "";

static Connection conn;

static Statement stmt;

static ResultSet rs;

static InputStreamReader inputStreamReader = new InputStreamReader(System.in);

static BufferedReader input = new BufferedReader(inputStreamReader);

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

try {

// register driver

Class.forName(JDBC\_DRIVER);

conn = DriverManager.getConnection(DB\_URL, USER, PASS);

stmt = conn.createStatement();

while (!conn.isClosed()) {

showMenu();

}

stmt.close();

conn.close();

} catch (Exception e) {

e.printStackTrace();

}

}

static void showMenu() {

System.out.println("+=========== MENU UTAMA =========+");

System.out.println("| 1. Masukan Data Pengguna |");

System.out.println("| 2. Menampilkan Data Pengguna |");

System.out.println("| 3. Edit Data Pengguna |");

System.out.println("| 4. Hapus Data Pengguna |");

System.out.println("| 0. Keluar |");

System.out.println("| |");

System.out.println("+--------------------------------+");

System.out.print("Masukan No > ");

try {

int pilihan = Integer.parseInt(input.readLine());

switch (pilihan) {

case 0:

System.exit(0);

break;

case 1:

insertPengguna();

break;

case 2:

showData();

break;

case 3:

updatePengguna();

break;

case 4:

deletePengguna();

break;

default:

System.out.println("Pilihan salah!");

}

} catch (Exception e) {

e.printStackTrace();

}

}

static void showData() {

String sql = "SELECT \* FROM pengguna";

try {

rs = stmt.executeQuery(sql);

System.out.println("+--------------------------------+");

System.out.println("| DATA PENGGUNA KULI ONLINE |");

System.out.println("+--------------------------------+");

while (rs.next()) {

int id\_pengguna = rs.getInt("id\_pengguna");

String nama\_pengguna = rs.getString("nama\_pengguna");

String alamat\_pengguna = rs.getString("alamat\_pengguna");

System.out.println(String.format("%d, %s ,%s)", id\_pengguna, nama\_pengguna, alamat\_pengguna));

}

} catch (Exception e) {

e.printStackTrace();

}

}

static void insertPengguna() {

try {

// ambil input dari user

System.out.print("id pengguna: ");

String id\_pengguna = input.readLine().trim();

System.out.print("nama pengguna: ");

String nama\_pengguna = input.readLine().trim();

System.out.print("alamat pengguna: ");

String alamat\_pengguna = input.readLine().trim();

System.out.print("email pengguna: ");

String email\_pengguna = input.readLine().trim();

// query simpan

String sql = "INSERT INTO pengguna (id\_pengguna,nama\_pengguna, alamat\_pengguna, email\_pengguna) VALUE('%s', '%s', '%s','%s')";

sql = String.format(sql,id\_pengguna, nama\_pengguna, alamat\_pengguna, email\_pengguna);

// simpan buku

stmt.execute(sql);

} catch (Exception e) {

e.printStackTrace();

}

}

static void updatePengguna() {

try {

// ambil input dari user

System.out.print("ID yang mau diedit: ");

int id\_pengguna = Integer.parseInt(input.readLine());

System.out.print("nama pengguna: ");

String nama\_pengguna = input.readLine().trim();

System.out.print("alamat\_pengguna: ");

String alamat\_pengguna = input.readLine().trim();

System.out.print("email\_pengguna: ");

String email\_pengguna = input.readLine().trim();

// query update

String sql = "UPDATE pengguna SET nama\_pengguna='%s', alamat\_pengguna='%s', email\_pengguna='%s' WHERE id\_pengguna=%d";

sql = String.format(sql, nama\_pengguna, alamat\_pengguna, email\_pengguna, id\_pengguna);

// update data buku

stmt.execute(sql);

} catch (Exception e) {

e.printStackTrace();

}

}

static void deletePengguna() {

try {

// ambil input dari user

System.out.print("ID yang mau dihapus: ");

int id\_pengguna = Integer.parseInt(input.readLine());

// buat query hapus

String sql = String.format("DELETE FROM pengguna WHERE id\_Pengguna=%d", id\_pengguna);

// hapus data

stmt.execute(sql);

System.out.println("Data telah terhapus...");

} catch (Exception e) {

e.printStackTrace();

}

}

}

1. **Pekerja**

package S3UAS;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.ResultSet;

public class pekerja{

static final String JDBC\_DRIVER = "com.mysql.jdbc.Driver";

static final String DB\_URL = "jdbc:mysql://localhost/kuli\_online";

static final String USER = "root";

static final String PASS = "";

static Connection conn;

static Statement stmt;

static ResultSet rs;

static InputStreamReader inputStreamReader = new InputStreamReader(System.in);

static BufferedReader input = new BufferedReader(inputStreamReader);

private static String sq3;

private static String sq2;

public static void main(String[] args) {

try {

// register driver

Class.forName(JDBC\_DRIVER);

conn = DriverManager.getConnection(DB\_URL, USER, PASS);

stmt = conn.createStatement();

while (!conn.isClosed()) {

showMenu();

}

stmt.close();

conn.close();

} catch (Exception e) {

e.printStackTrace();

}

}

static void showMenu() {

System.out.println("+--------------------------------+");

System.out.println("\n========= MENU UTAMA =========");

System.out.println("1. Masukan Data Pekerja");

System.out.println("2. Menampilkan Data Pekerja");

System.out.println("3. Edit Data Pekerja");

System.out.println("4. Hapus Data Pekerja");

System.out.println("0. Keluar");

System.out.println("");

System.out.println("+--------------------------------+");

System.out.print("Masukan No > ");

try {

int pilihan = Integer.parseInt(input.readLine());

switch (pilihan) {

case 0:

System.exit(0);

break;

case 1:

insertPekerja();

break;

case 2:

showData();

break;

case 3:

updatePekerja();

break;

case 4:

deletePekerja();

break;

default:

System.out.println("Pilihan salah!");

}

} catch (Exception e) {

e.printStackTrace();

}

}

static void showData() {

String sql = "SELECT \* FROM pekerja";

try {

rs = stmt.executeQuery(sql);

System.out.println("+--------------------------------+");

System.out.println("| DATA PEKERJA KULI ONLINE |");

System.out.println("+--------------------------------+");

while (rs.next()) {

int id\_pekerja = rs.getInt("id\_pekerja");

String nama\_pekerja = rs.getString("nama\_pekerja");

String harga\_jasa = rs.getString("harga\_jasa");

String alamat\_pekerja = rs.getString("alamat\_pekerja");

String no\_hp = rs.getString("no\_hp");

String email\_pekerja = rs.getString("email\_pekerja");

String rekening\_pekerja = rs.getString("rekening\_pekerja");

System.out.println(String.format("%d. %s,%s,%s,%s,%s",id\_pekerja ,nama\_pekerja, harga\_jasa,alamat\_pekerja, no\_hp, email\_pekerja, rekening\_pekerja));

}

} catch (Exception e) {

e.printStackTrace();

}

}

static void insertPekerja() {

try {

// ambil input dari user

System.out.print("id\_pekerja: ");

String id\_pekerja = input.readLine().trim();

System.out.print("nama\_pekerja: ");

String nama\_pekerja = input.readLine().trim();

System.out.print("harga\_jasa: ");

String harga\_jasa = input.readLine().trim();

System.out.print("alamat\_pekerja: ");

String alamat\_pekerja = input.readLine().trim();

System.out.print("no\_hp: ");

String no\_hp = input.readLine().trim();

System.out.print("email\_pekerja: ");

String email\_pekerja = input.readLine().trim();

System.out.print("rekening\_pekerja: ");

String rekening\_pekerja= input.readLine().trim();

// query simpan

String sql = "INSERT INTO pekerja (id\_pekerja, nama\_pekerja, harga\_jasa, alamat\_pekerja,no\_hp, email\_pekerja, rekening\_pekerja) VALUE('%s','%s','%s','%s','%s','%s','%s')";

sql = String.format(sql,id\_pekerja, nama\_pekerja, harga\_jasa, alamat\_pekerja, no\_hp, email\_pekerja, rekening\_pekerja );

stmt.execute(sql);

} catch (Exception e) {

e.printStackTrace();

}

}

static void updatePekerja() {

try {

// ambil input dari user

System.out.print("ID yang mau diedit: ");

int id\_pekerja = Integer.parseInt(input.readLine());

System.out.print("nama pekerja: ");

String nama\_pekerja = input.readLine().trim();

System.out.print("harga jasa: ");

String harga\_jasa = input.readLine().trim();

System.out.print("alamat pekerja : ");

String alamat\_pekerja = input.readLine().trim();

System.out.print("no hp :");

String no\_hp = input.readLine().trim();

System.out.print("email pekerja : ");

String email\_pekerja = input.readLine().trim();

System.out.println("rekening pekerja :");

String rekening\_pekerja =input.readLine().trim();

// query update

String sql = "UPDATE pekerja SET nama\_pekerja='%s', harga\_jasa='%s',alamat\_pekerja='%s', no\_hp='%s', email\_pekerja='%s', rekening\_pekerja='%s' WHERE id\_pekerja=%d";

sql = String.format(sql, nama\_pekerja, harga\_jasa,alamat\_pekerja, no\_hp, email\_pekerja, rekening\_pekerja, id\_pekerja );

// update data buku

stmt.execute(sql);

} catch (Exception e) {

e.printStackTrace();

}

}

static void deletePekerja() {

try {

// ambil input dari user

System.out.print("ID yang mau dihapus: ");

int id\_pekerja = Integer.parseInt(input.readLine());

// buat query hapus

String sql = String.format("DELETE FROM pekerja WHERE id\_pekerja=%d", id\_pekerja);

// hapus data

stmt.execute(sql);

System.out.println("Data telah terhapus...");

} catch (Exception e) {

e.printStackTrace();

}

}

}

1. **Pembayaran**

package S3UAS;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

public class pembayaran {

public static void main(String[] args) {

try {

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost/kuli\_online", "root", "");

Statement stmt = con.createStatement();

// Menampilkan tabel 1

ResultSet rs1 = stmt.executeQuery("SELECT \* FROM pengguna");

System.out.println("+--------------------------------+");

System.out.println("| DATA PENGGUNA |");

System.out.println("+--------------------------------+");

while (rs1.next()) {

System.out.println("id\_pengguna :"+ rs1.getInt(1));

System.out.println("nama\_pengguna :"+ rs1.getString(2));

}

// Menampilkan tabel 2

ResultSet rs2 = stmt.executeQuery("SELECT \* FROM pekerja");

System.out.println("+--------------------------------+");

System.out.println("| MENYEWA PEKERJA |");

System.out.println("+--------------------------------+");

while (rs2.next()) {

System.out.println("id\_pekerja :"+ rs2.getInt(1));

System.out.println("nama pengguna :"+ rs2.getString(2));

}

ResultSet rs3 = stmt.executeQuery("SELECT \* FROM pembayaran");

System.out.println("+--------------------------------+");

System.out.println("| DETAIL PEMBAYARAN |");

System.out.println("+--------------------------------+");

while (rs3.next()) {

System.out.println("id pengguna :"+rs3.getInt(1));

System.out.println("id pekerja :" +rs3.getString(2));

System.out.println("total harga :"+rs3.getString(3));

System.out.println("tanggal pembayaran :"+rs3.getString(4));

}

con.close();

} catch (Exception e) {

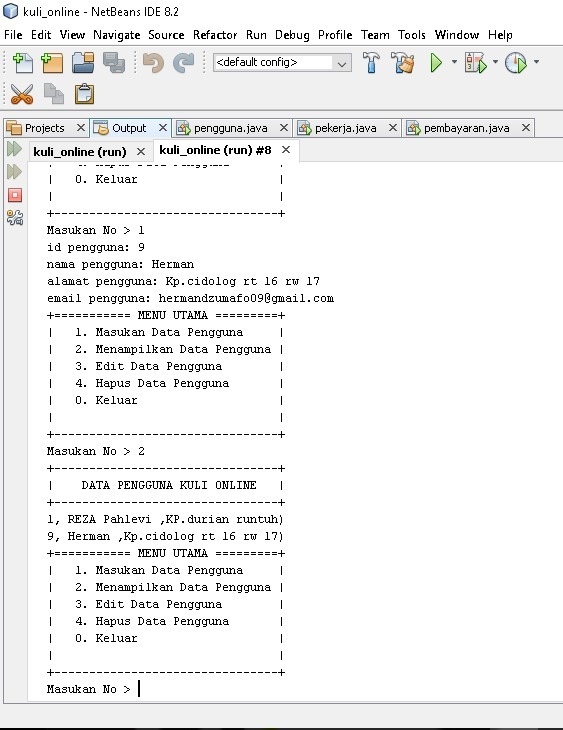
System.out.println(e);

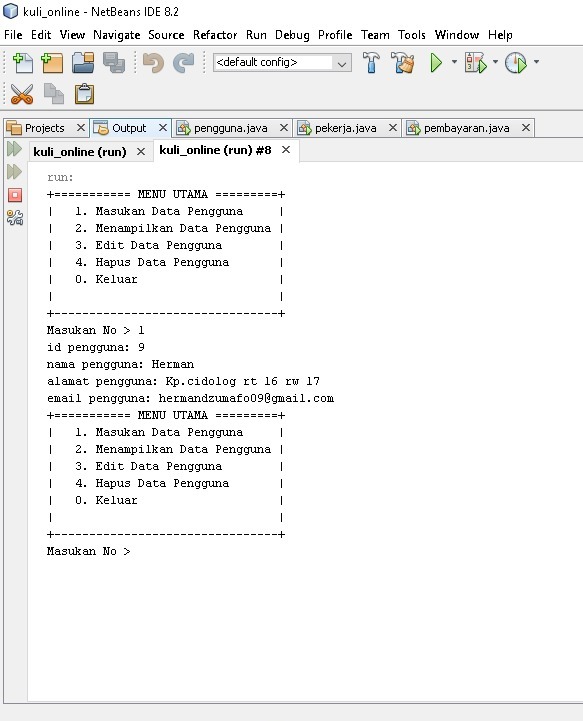
}

}

}

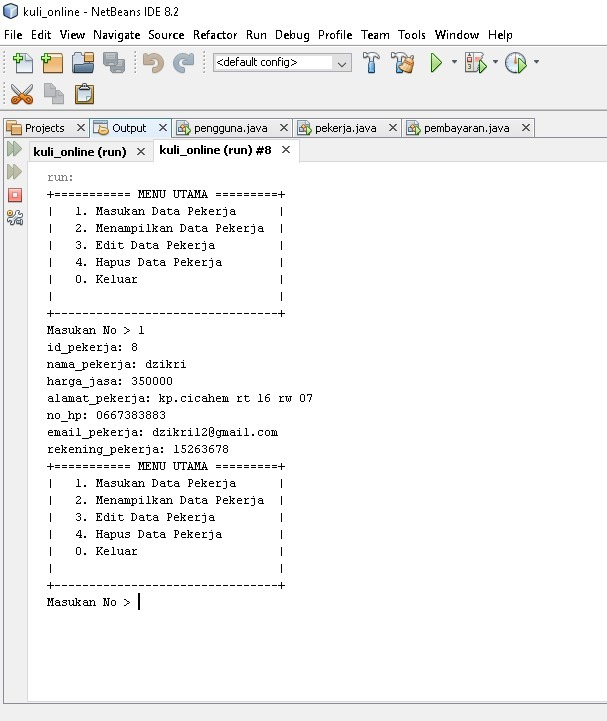
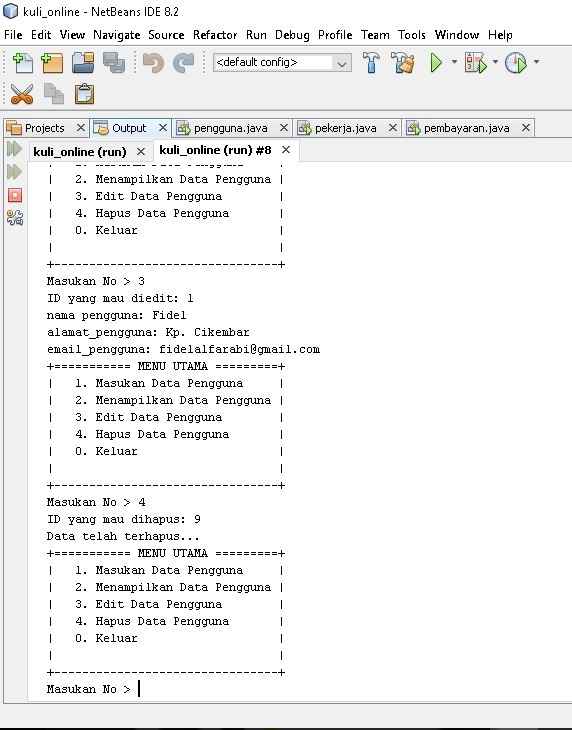
### **Pengujian Program**

1. Tampilan Menu Tabel Pengguna

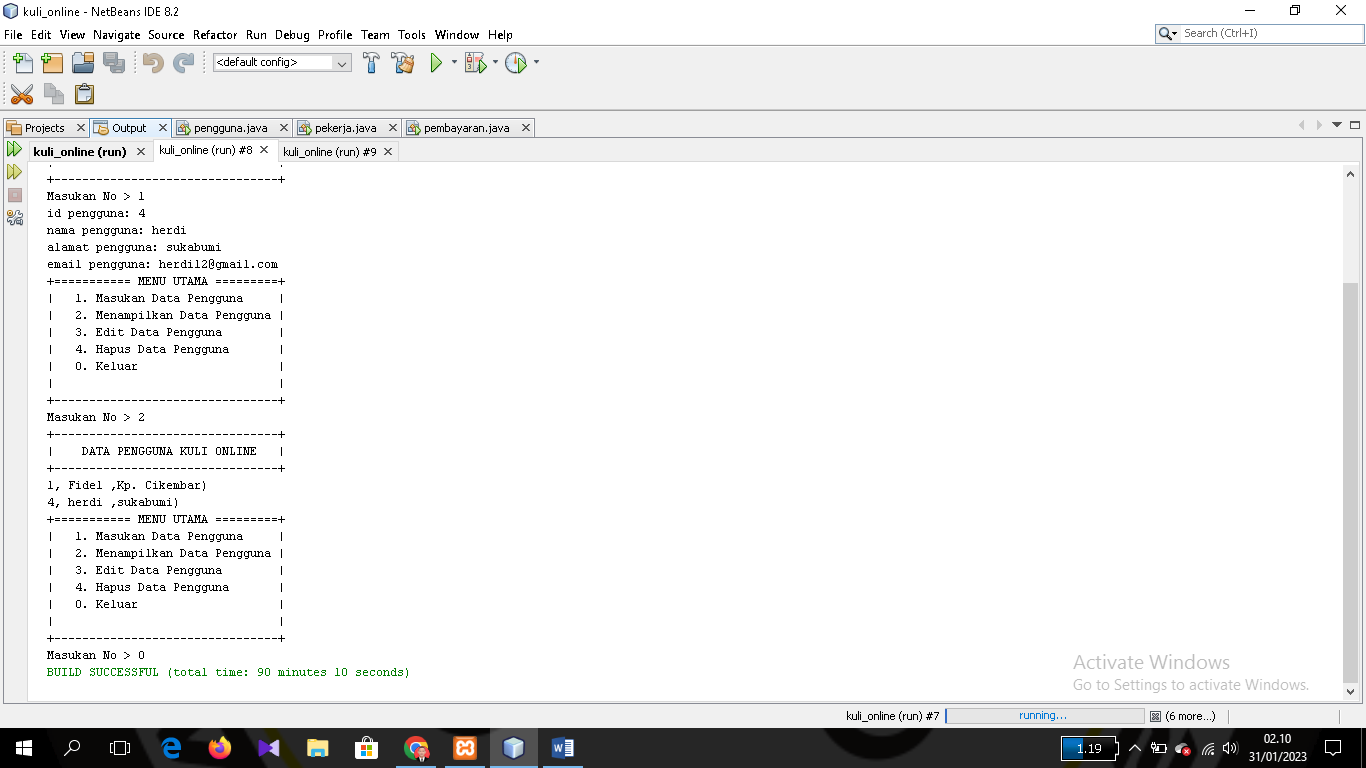


Gambar .2 tampilan tabel menu pengguna pilihan no 2

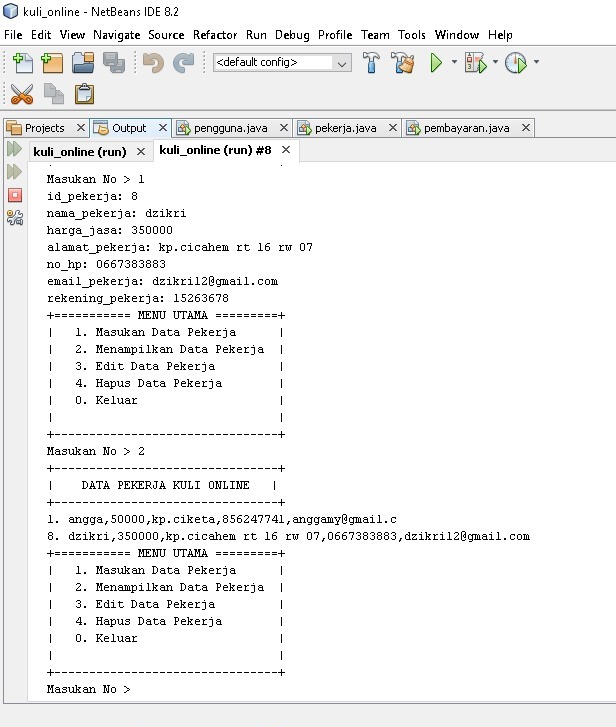
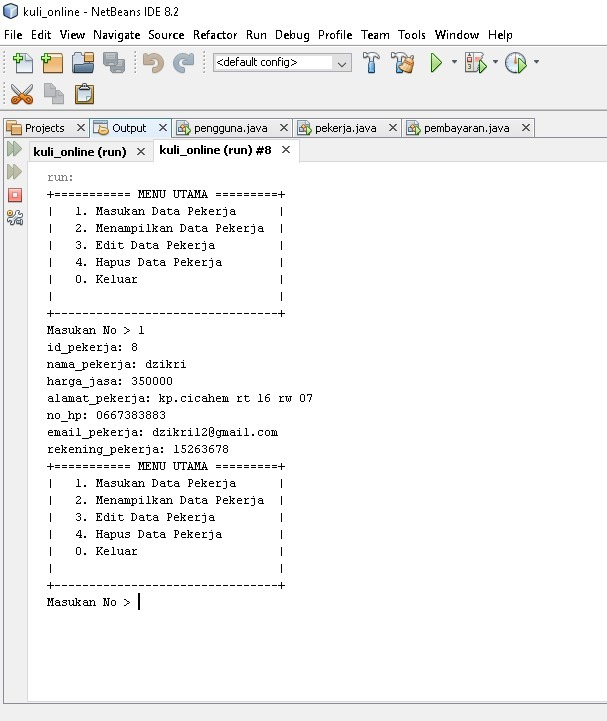
**Gambar 1.1Menu tabel pengguna pilihan menu no 1**

****

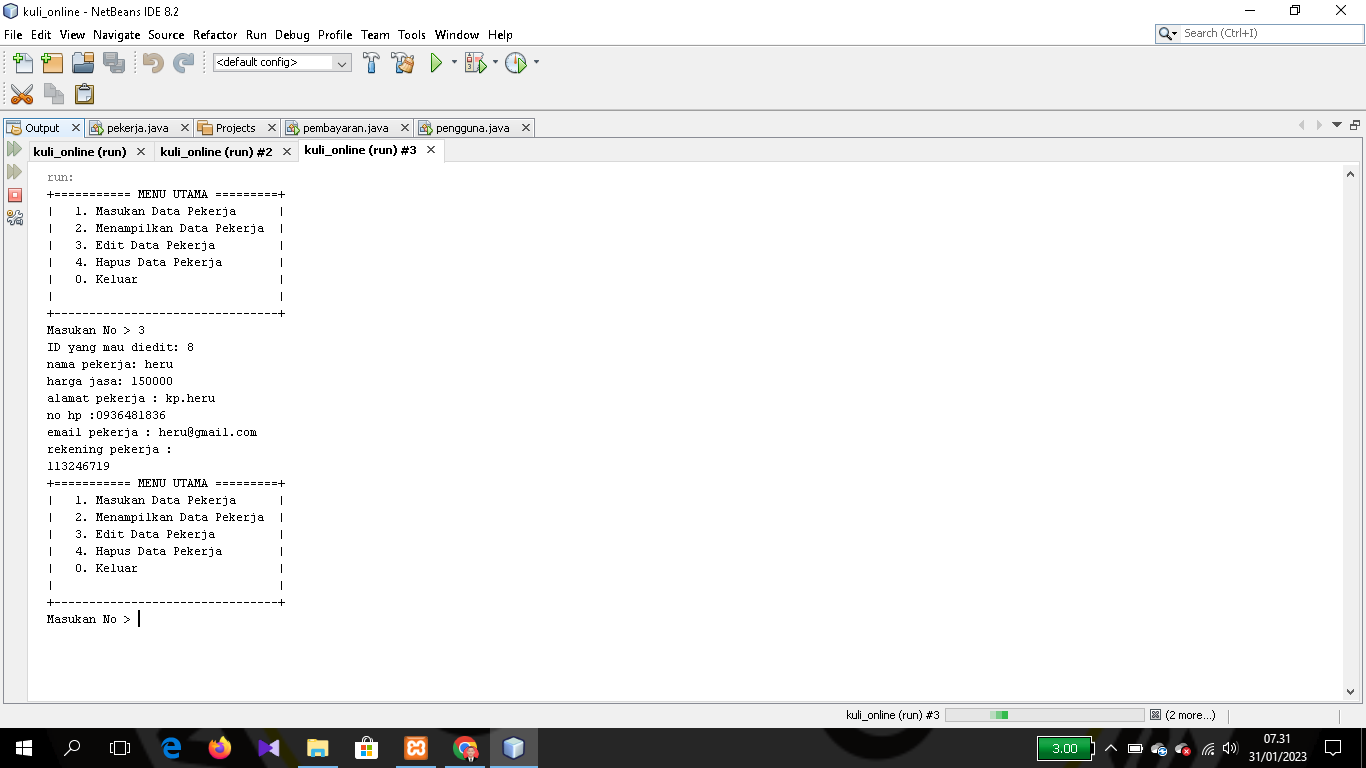
**Gambar 1.3 Menu tabel pengguna pilihan menu no 3 Gambar 1.4 Menu tabel pengguna pilihan menu no 4**

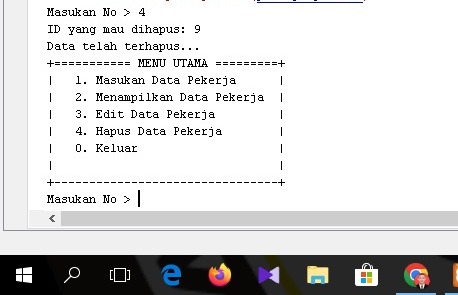
*Gambar 1.4 Menu tabel pengguna pilihan menu no 4*

1. Tampilan Menu tabel Pekerja

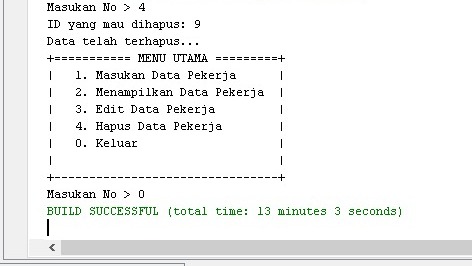
****

*Gambar 2.1 Menu tabel pekerja pilihan menu no 1 Gambar 2.2 Menu tabel pekerja pilihan menu no 2*



****

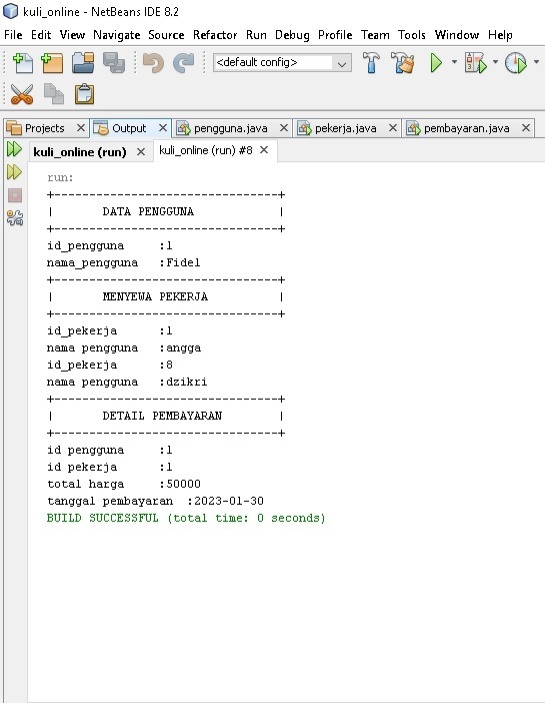
*Gambar 2.1 Menu tabel pkerja pilihan menu no 3 Gambar 2.1 Menu tabel pekerja pilihan menu no 4*

****

*Gambar 2.1 Menu tabel pekerja pilihan menu no 0*

1. Tampilan Tabel Menu Pembayaran

Menampilkan show data keseluruhan Relasi pada Tabel



*Gambar 3.1 Menu tabel pembayaran*

# **BAB V**

## **PENUTUP**

### **Kesimpulan**

Program database Kuli Online berbasis CRUD adalah bahwa program ini memungkinkan pengguna untuk membuat, membaca, memperbarui, dan menghapus data dari database Kuli Online. Program ini menyediakan berbagai fitur untuk membantu pengguna melakukan tugasnya dengan lebih efisien dan lancar. Program ini juga memungkinkan pengguna untuk mengelola data dan informasi dengan lebih mudah, bahkan meskipun mereka tidak punya pengalaman dalam pengembangan aplikasi. Dengan demikian, program ini menawarkan solusi yang efisien dan dapat diandalkan untuk mengelola data dan informasi di database Kuli Online.

# **DAFTAR PUSTAKA**

Muhardian , Muhammad.2017*. Tutorial Java dan MySQL: Membuat Program CRUD Berbasis Teks*. https://www.petanikode.com/java-mysql/. 20 Januari 2023

Fauzan, Falah.2020. *Tutorial Java : Cara Membuat CRUD Java dan MySQL Netbeans.* https://www.codekop.com/read/tutorial-java-cara-membuat-crud-java-dan-mysql-netbeans.html. 22 Januari 2023

Darmayuda, I Ketut.2021. *Belajar MySQL Part#6 - Cara Membuat dan Menjalankan Trigger di MySQL.* https://www.geriatama.com/2021/11/mysql-cara-membuat-trigger-database.html. 25 Januari 2023