

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 Cibat 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 Pemrograman 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
ABSTRAK	3
DAFTAR ISI.....	4
BAB I.....	5
PENDAHULUAN	5
1.1 Latar Belakang	5
1.2 Rumusan Masalah	5
1.4 Tujuan	5
BAB II	6
STUDI PUSTAKA	6
2.1 Bahasa Pemrograman java	6
2.2 Database	6
2.3 Metode CRUD	6
2.4 My SQL	6
BAB III.....	7
METHODOLOGI.....	7
3.1 Metode Pembuatan Program	7
BAB IV	8
HASIL DAN DISKUSI	8
4.1 Kode Program	8
4.2 Pengujian Program	23
BAB V	27
PENUTUP.....	27
5.1 Kesimpulan	27
DAFTAR PUSTAKA	28

BAB I

PENDAHULUAN

1.1 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.

1.2 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.

1.3 Batasan Masalah

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

1.4 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

2.1 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.

2.2 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.

2.3 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.

2.4 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.

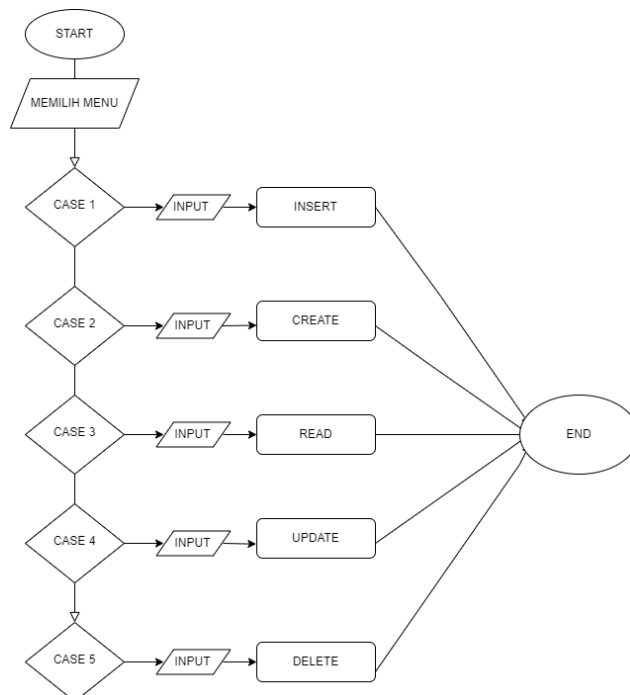
BAB III METHODOLOGI

3.1 Metode Pembuatan Program

A. 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:

1. Menentukan Entitas: Identifikasi entitas yang akan diterima, disimpan, dan dikelola oleh aplikasi.
2. Penentuan Atribut: Penentuan atribut yang akan disimpan untuk setiap entitas.
3. Implementasi Create: Implementasi proses untuk menambahkan entitas baru ke basis data.
4. Implementasi Read: Implementasi proses untuk membaca entitas dari basis data.
5. Implementasi Update: Implementasi proses untuk memperbarui entitas yang ada di basis data.
6. Implementasi Delete: Implementasi proses untuk menghapus entitas dari basis data.
7. Pengujian dan Validasi: Uji aplikasi untuk memastikan bahwa setiap operasi CRUD berfungsi dengan benar.

B. 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();  
    }  
}  
}
```

2. **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();
    }
}
}

```

3. 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("|    MENYEWAKAN 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));

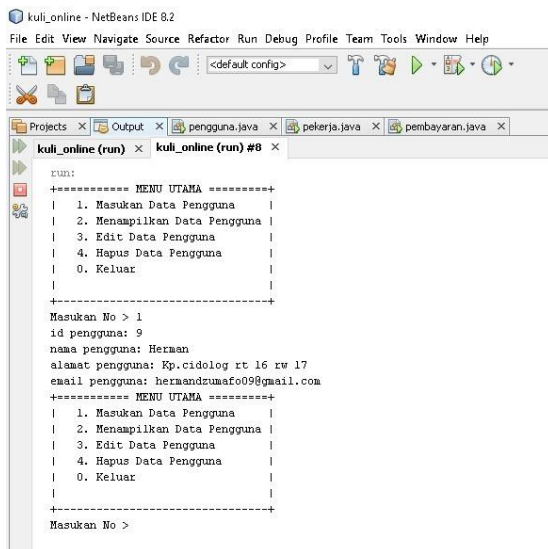
        }

        con.close();
    } catch (Exception e) {
        System.out.println(e);
    }
}

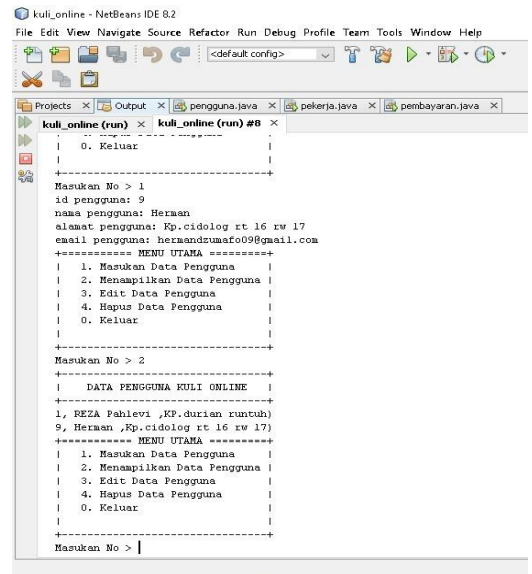
```

4.2 Pengujian Program

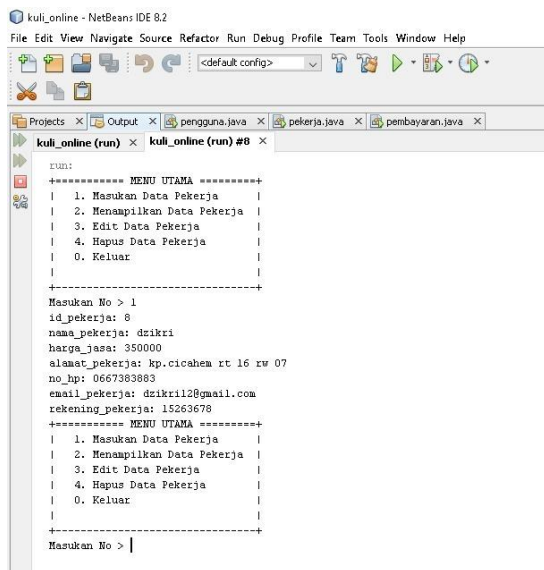
A. Tampilan Menu Tabel Pengguna



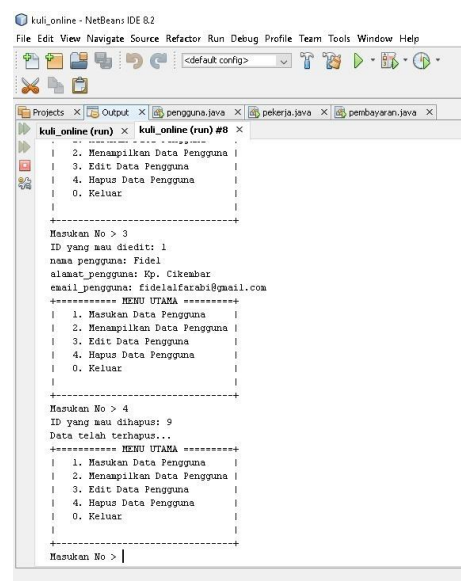
Gambar 1.1 Menu tabel pengguna pilihan menu no 1



Gambar 1.2 tampilan tabel menu pengguna pilihan menu no 2



Gambar 1.3 Menu tabel pengguna pilihan menu no 3



Gambar 1.4 Menu tabel pengguna pilihan menu no 4

```

kuli_online - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects Output pengguna.java pekerja.java pembayaran.java
kuli_online (run) kuli_online (run) #8 kuli_online (run) #9
+-----+
Masukan No > 1
id pengguna: 4
nama pengguna: herdi
alamat pengguna: sukabumi
email pengguna: herdi12@gmail.com
+-----+
MENU UTAMA
+-----+
1. Masukkan Data Pengguna
2. Menampilkan Data Pengguna
3. Edit Data Pengguna
4. Hapus Data Pengguna
0. Keluar
+-----+
Masukan No > 2
+-----+
DATA PENGGUNA KULI ONLINE
+-----+
1, Fidel ,Kp. Cikembar)
4, herdi ,sukabumi)
+-----+
MENU UTAMA
+-----+
1. Masukkan Data Pengguna
2. Menampilkan Data Pengguna
3. Edit Data Pengguna
4. Hapus Data Pengguna
0. Keluar
+-----+
Masukan No > 0
BUILD SUCCESSFUL (total time: 90 minutes 10 seconds)

```

Gambar 1.4 Menu tabel pengguna pilihan menu no 4

B. Tampilan Menu tabel Pekerja

```

kuli_online - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects Output pengguna.java pekerja.java pembayaran.java
kuli_online (run) kuli_online (run) #8
+-----+
MENU UTAMA
+-----+
1. Masukkan Data Pekerja
2. Menampilkan Data Pekerja
3. Edit Data Pekerja
4. Hapus Data Pekerja
0. Keluar
+-----+
Masukan No > 1
id pekerja: 8
nama pekerja: dzikri
harga_jasa: 350000
alamat pekerja: kp.cicahem rt 16 rw 07
no_hp: 0667383883
email pekerja: dzikri12@gmail.com
rekening pekerja: 15263678
+-----+
MENU UTAMA
+-----+
1. Masukkan Data Pekerja
2. Menampilkan Data Pekerja
3. Edit Data Pekerja
4. Hapus Data Pekerja
0. Keluar
+-----+
Masukan No > 1

```

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

```

kuli_online - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects Output pengguna.java pekerja.java pembayaran.java
kuli_online (run) kuli_online (run) #8
+-----+
MENU UTAMA
+-----+
1. Masukkan Data Pekerja
2. Menampilkan Data Pekerja
3. Edit Data Pekerja
4. Hapus Data Pekerja
0. Keluar
+-----+
Masukan No > 2
+-----+
DATA PEKERJA KULI ONLINE
+-----+
1. anggota,50000,kp.ciketa,856247741,angguny@gmail.c
8. dzikri,350000,kp.cicahem rt 16 rw 07,0667383883,dzikri12@gmail.com
+-----+
MENU UTAMA
+-----+
1. Masukkan Data Pekerja
2. Menampilkan Data Pekerja
3. Edit Data Pekerja
4. Hapus Data Pekerja
0. Keluar
+-----+
Masukan No >

```

Gambar 2.2 Menu tabel pekerja pilihan


```

run:
+===== MENU UTAMA =====+
| 1. Masukan Data Pekerja |
| 2. Menampilkan Data Pekerja |
| 3. Edit Data Pekerja |
| 4. Hapus Data Pekerja |
| 0. Keluar |
+-----+
Masukan No > 3
ID yang mau diedit: 8
nama pekerja: heru
harga jasa: 150000
alamat pekerja : kp.heru
no hp :0936481836
email pekerja : heru@gmail.com
rekening pekerja :
113246719
+===== MENU UTAMA =====+
| 1. Masukan Data Pekerja |
| 2. Menampilkan Data Pekerja |
| 3. Edit Data Pekerja |
| 4. Hapus Data Pekerja |
| 0. Keluar |
+-----+
Masukan No > |

```

Gambar 2.1 Menu tabel pekerja pilihan menu no 3 no 4

```

Masukan No > 4
ID yang mau dihapus: 9
Data telah terhapus...
+===== MENU UTAMA =====+
| 1. Masukan Data Pekerja |
| 2. Menampilkan Data Pekerja |
| 3. Edit Data Pekerja |
| 4. Hapus Data Pekerja |
| 0. Keluar |
+-----+
Masukan No > |

```

Gambar 2.1 Menu tabel pekerja pilihan menu

```

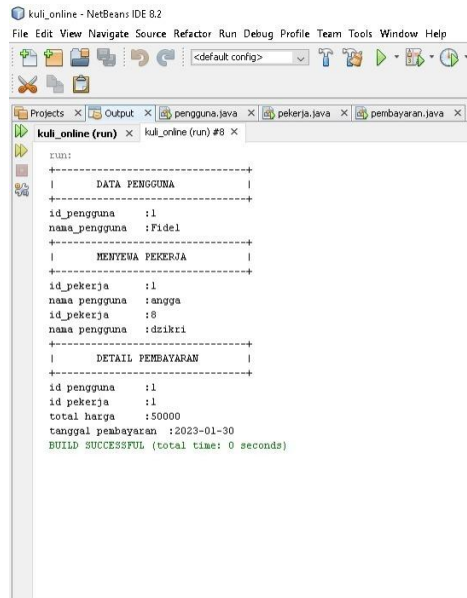
Masukan No > 4
ID yang mau dihapus: 9
Data telah terhapus...
+===== MENU UTAMA =====+
| 1. Masukan Data Pekerja |
| 2. Menampilkan Data Pekerja |
| 3. Edit Data Pekerja |
| 4. Hapus Data Pekerja |
| 0. Keluar |
+-----+
Masukan No > 0
BUILD SUCCESSFUL (total time: 13 minutes 3 seconds)
+-----+
Masukan No > |

```

Gambar 2.1 Menu tabel pekerja pilihan menu no 0

C. Tampilan Tabel Menu Pembayaran

Menampilkan show data keseluruhan Relasi pada Tabel



```
Run:
-----+
|  DATA PENGGUNA  |
+-----+
id_pengguna      :1
nama_pengguna     :Fadel
+-----+
|  MENYEWAKAN PEKERJA  |
+-----+
id_pekerja        :1
nama_pengguna     :angga
id_pekerja        :8
nama_pengguna     :dzikri
+-----+
|  DETAIL PEMBAYARAN  |
+-----+
id pengguna       :1
id pekerja        :1
total harga       :50000
tanggal pembayaran :2023-01-30
BUILD SUCCESSFUL (total time: 0 seconds)
```

Gambar 3.1 Menu tabel pembayaran

BAB V

PENUTUP

5.1 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