

DOKUMENTASI KELAS Pembayaran

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package finalproject; // lokasi pada package finalproject
```

```
import java.sql.Connection; // mengimport class Connection dalam package java.sql
```

```
import java.sql.DriverManager; // mengimport class DriverManager dalam package java.sql
```

```
import java.sql.ResultSet; // mengimport class ResulSet dalam package java.sql
```

```
import java.sql.SQLException; // mengimport class SQLException dalam package java.sql
```

```
import java.sql.Statement; // mengimport class Statement dalam package java.sql
```

```
import java.text.ParseException; // mengimport class ParseException dalam package java.text
```

```
import java.text.SimpleDateFormat; // mengimport class SimpleDateFormat dalam package  
java.text
```

```
import java.util.Calendar; // mengimport class Calendar dalam package java.util
```

```
import java.util.Date; // mengimport class Date dalam package java.util
```

```
/**
```

```
*
```

```
* @author Faza Abdi
```

```
*/
```

```
public class Pembayaran {
```

```
    // Deklarasi variabel lama, temp sama dengan 0 berupa integer
```

```
    // Deklarasi variabel jenis, CI, CO berupa String
```

```
    // Deklarasi variabel bayar_hari, bonus, denda berupa double
```

```
    // Deklarasi variabel hari, jam, menit berupa long
```

```
    private int lama, temp = 0;
```

```

private String jenis, CI, CO;

private double bayar_hari, bonus, denda;

private long hari, jam, menit;


// Method untuk mengeset nilai dari variabel bayar_hari

// Memasukkan syntax untuk mengakses data base

// Deklarasi statement

// Memasukkan statement berupa syntax SQL untuk mengambil data lama dan jenis
berdasarkan NIK sama dengan nilai dari variabel nik

// Assignment data lama dan jenis dari database kedalam variabel lama dan jenis

// Proses seleksi kondisi berdasarkan jenis kamar yang memproses nilai dari variabel lama dan
nilai dari method getHarga sesuai jenis

public void bayar_hari(String nik) throws SQLException {

    Connection conn = DriverManager.getConnection("jdbc:derby://localhost:1527/Hotel",
"faza", "aaaaa");

    Statement stmt = conn.createStatement();

    {

        ResultSet bayar = stmt.executeQuery("select lama, jenis from faza.DATAKAMAR where
nik=" + nik);

        while (bayar.next()) {

            lama = bayar.getInt("lama");

            jenis = bayar.getString("jenis");

        }

        if (jenis.equalsIgnoreCase("reguler")) {

            Reguler r = new Reguler("reguler");

            bayar_hari = lama * r.getHarga();

        } else if (jenis.equalsIgnoreCase("premium")) {

            Premium p = new Premium("premium");

            bayar_hari = lama * p.getHarga();

        } else if (jenis.equalsIgnoreCase("suite")) {

            Suite s = new Suite("suite");

            bayar_hari = lama * s.getHarga();

        }
    }
}

```

```

    } else if (jenis.equalsIgnoreCase("vip")) {
        Vip v = new Vip("vip");
        bayar_hari = lama * v.getHarga();
    }
}
}

```

```

// Method untuk mengambil harga akumulasi lama menginap dan harga kamar
// Mengembalikan nilai dari variabel bayar_hari
public double getBayar_hari() {
    return bayar_hari;
}

```

```

// Method untuk menghitung selisih antara dua tanggal
// Memasukkan syntax untuk mengakses data base
// Deklarasi statement
// Memasukkan statement berupa syntax SQL untuk mengambil data check in berdasarkan NIK
sama dengan variabel nik

// Assignment data check in ke dalam variabel CI
// Assignment variabel co dari parameter kedalam variabel CO
// Deklarasi kelas Calendar dengan objek cal
// Instansiasi kelas SimpleDateFormat dengan objek sdf dengan format parameter "dd-MM-
yyyy HH:mm"

// Parse String kedalam date diassignment-kan kedalam variabel a
// Parse String kedalam date diassignment-kan kedalam variabel b
// Perhitungan selisih antara dua tanggal
// Assignment proses mendapatkan menit kedalam variabel selisihMenit
// Assignment proses mendapatkan Jam kedalam variabel selisihJam
// Assignment proses mendapatkan hari kedalam variabel selisihHari
// Assignment selisihHari kedalam variabel hari
// Assignment selisihJam kedalam variabel jam
// Assignment selisihMenit kedalam variabel menit

```

```

public void selisihDateTime(String nik, String co) throws ParseException, SQLException {
    Connection conn = DriverManager.getConnection("jdbc:derby://localhost:1527/Hotel",
"faza", "aaaaa");
    Statement stmt = conn.createStatement();
    {
        ResultSet rset = stmt.executeQuery("select CI from faza.DATAKAMAR where nik=" + nik);
        while (rset.next()) {
            CI = rset.getString("CI");
        }
        CO = co;
        Calendar cal = Calendar.getInstance();
        SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy HH:mm");
        Date a = sdf.parse(CI);
        Date b = sdf.parse(CO);
        long selisihMS = Math.abs(b.getTime() - a.getTime());
        long selisihMenit = selisihMS / (60 * 1000) % 60;
        long selisihJam = selisihMS / (60 * 60 * 1000) % 24;
        long selisihHari = selisihMS / (24 * 60 * 60 * 1000);
        hari = selisihHari;
        jam = selisihJam;
        menit = selisihMenit;
    }
}

public long getHari() {
    return hari;
}

public long getJam() {
    return jam;
}

```

```

public long getMenit() {
    return menit;
}

// Method untuk mendapatkan bonus
// Memasukkan syntax untuk mengakses data base
// Deklarasi statement
// Memasukkan statement berupa syntax SQL untuk mengambil data lama berdasarkan NIK
sama dengan variabel nik
// Proses perulangan untuk mengassignment-kan nilai data lama kedalam array total_inap[]
// Proses perulangan untuk menjumlahkan seluruh nilai dari array total_inap[] kedalam variabel
temp
// Proses seleksi kondisi untuk menentukan bonus berdasarkan seberapa sering menginap
public double bonus(String nik) throws SQLException {
    int total_inap[] = new int[1000];

    Connection conn = DriverManager.getConnection("jdbc:derby://localhost:1527/Hotel",
"faza", "aaaaa");

    Statement stmt = conn.createStatement();

    {
        ResultSet rset = stmt.executeQuery("select lama from faza.DATAPELANGGAN where nik=" +
nik);

        for (int i = 0; i < 1000; i++) {
            if (rset.next()) {
                total_inap[i] = rset.getInt("lama");
            }
        }

        for (int i = 0; i < 1000; i++) {
            temp += total_inap[i];
        }

        if (temp < 10) {
            return bonus = 0;
        }
    }
}

```

```

    } else if (temp >= 10 && temp <= 30) {
        return bonus = bayar_hari * 10 / 100;
    } else if (temp > 30 && temp <= 60) {
        return bonus = bayar_hari * 15 / 100;
    } else {
        return bonus = bayar_hari * 25 / 100;
    }
}
}
}

```

// Method untuk menghitung denda

// Proses seleksi kondisi apakah variabel hari kurang dari variabel lama, apabila benar maka akan mengembalikan nilai dari variabel denda

// Proses seleksi kondisi apakah variabel hari sama dengan variabel lama, apabila benar maka akan melakukan proses seleksi kondisi apakah

// nilai dari variabel jam tidak sama dengan 0, apabila benar akan mengembalikan nilai dari variabel jam dikalikan dengan 30000 dan apabila

// salah maka akan mengembalikan nilai 0

// Proses seleksi kondisi apakah variabel hari lebih besar dari variabel lama, apabila benar maka nilai dari variabel hari dikurangi nilai dari

// variabel lama dikalikan 24 kemudian diassignmentkan kedalam variabel sel, proses seleksi kondisi apakah nilai dari variabel jam

// tidak sama dengan 0, apabila benar akan mengembalikan nilai dari variabel sel ditambah nilai dari variabel jam kemudian dikalikan dengan 30000

// dan apabila salah maka akan mengembalikan nilai variabel sel dikalikan 30000

// Proses seleksi kondisi selain kondisi sebelumnya, apabila benar maka akan mengembalikan nilai 0

```

public double denda() {
    if (hari < lama) {
        return denda = 0;
    } else if (hari == lama) {
        if (jam != 0) {
            return denda = jam * 30000;
        }
    }
}

```

```

    } else {
        return denda = 0;
    }
} else if (hari > lama) {
    long sel = (hari - lama) * 24;
    if (jam != 0) {
        return denda = (sel + jam) * 30000;
    } else {
        return denda = sel * 30000;
    }
} else {
    return 0;
}
}

```

// Method untuk mendapatkan total bayar

// Mengembalikan nilai dari perhitungan antara nilai dari variabel bayar_hari, bonus dan denda

```

public double totalBayar() {
    return bayar_hari - bonus + denda;
}

```

// Method untuk menentukan harga kamar

// Proses seleksi kondisi berdasarkan variabel jenis kemudian mengembalikan nilai dari method getHarga yang dipanggil oleh object masing-masing class

// Proses seleksi kondisi selain kondisi sebelumnya, apabila benar akan mengembalikan "Jenis Tidak ditemukan"

```

public String hargaKamar() {
    if (jenis.equalsIgnoreCase("reguler")) {
        Reguler r = new Reguler("reguler");
        return String.valueOf(r.getHarga());
    } else if (jenis.equalsIgnoreCase("premium")) {
        Premium p = new Premium("premium");
    }
}

```

```

        return String.valueOf(p.getHarga());
    } else if (jenis.equalsIgnoreCase("suite")) {
        Suite s = new Suite("suite");
        return String.valueOf(s.getHarga());
    } else if (jenis.equalsIgnoreCase("vip")) {
        Vip v = new Vip("vip");
        return String.valueOf(v.getHarga());
    } else {
        return "Jenis Tidak ditemukan";
    }
}

```

```

// Method untuk menginputkan pemasukan kedalam database
// Memasukkan syntax untuk mengakses data base
// Deklarasi statement
// Proses untuk memisahkan String CO yang mengandung tanda "-" kedalam variabel tgl, bln,
sis

```

```

// Proses untuk memisahkan Strig sisa yang mengandung " " kedalam variabel thn dan jamm
// Memasukkan statement berupa syntax SQL untuk memasukkan data tanggal, bulan, tahun
dan pendapatan

```

```

public void data_keuangan() throws SQLException {
    Connection conn = DriverManager.getConnection("jdbc:derby://localhost:1527/Hotel",
"faza", "aaaaa");

    Statement stmt = conn.createStatement();

    {
        String[] parts = CO.split("-");

        String tgl = parts[0];

        String bln = parts[1];

        String sisa = parts[2];

        String[] part = sisa.split(" ");

        String thn = part[0];

        String jamm = part[1];
    }
}

```



```
        int inputD = stmt.executeUpdate("insert into faza.DATAKEUANGAN values ('" + tgl + "', '" +  
bln + "', '" + thn + "', " + totalBayar() + ")");  
    }  
}  
}
```