

# **LAPORAN FINAL PROJECT**

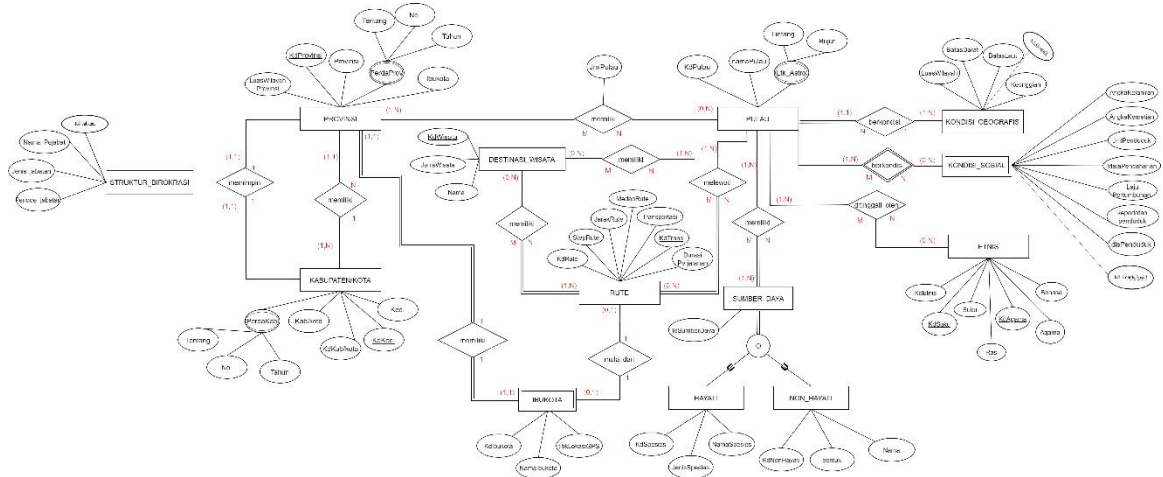
## **“DATABASE KEPENDUDUKAN”**

### **I. DATABASE REQUIREMENTS**

Database ini diharapkan mampu menyajikan informasi mengenai wilayah yang ada di kepulauan Indonesia yang kemudian informasi ini bisa digunakan oleh beberapa pihak seperti warga non lokal atau wisatawan untuk lebih mengetahui daerah tujuan wisatanya serta pemerintah untuk mendukung pelaksanaan pelayanan publik dan pengelolaan data kependudukan dan kewilayahan. Berikut informasi penting dalam pembuatan Database Kependudukan :

- Tidak semua pulau dihuni oleh manusia.  
Setiap pulau yang dihuni oleh manusia mempunyai kondisi sosial yang meliputi: angka kelahiran, angka kematian, jumlah penduduk, mata pencaharian, laju pertumbuhan penduduk, kepadatan penduduk dan distribusi penduduk.  
Penduduk yang menghuni suatu pulau terdiri dari satu atau lebih etnis yang terbagi dalam suku, ras, agama dan bahasa yang berbeda-beda.
- Untuk menuju ke suatu pulau kita bisa melewati rute, namun ada juga pulau yang tidak bisa dijangkau melalui rute manapun. Dari database ini dirincikan rute menuju suatu pulau mulai dari step rutenya, jarak yang ditempuh (dihitung dari Ibukota Provinsi), medan yang dilalui, transportasi yang dapat digunakan serta durasi perjalanan.
- Setiap pulau memiliki kondisi geografis yang meliputi : luas wilayah pulau, batas darat, batas laut dan ketinggian permukaan pulau.
- Setiap pulau memiliki sumber daya yang dapat dibedakan menjadi sumber daya hayati dan non hayati.
- Sumber daya hayati adalah sumber daya yang berasal dari makhluk hidup, dari database ini bisa diketahui nama spesies dan jenis spesies dari sumber daya hayati yang ada di suatu pulau. Sedangkan sumber daya non hayati adalah sumber daya yang berasal dari makhluk tak hidup, dari database ini dapat diketahui nama dan jenis dari sumber daya non hayati yang ada di suatu pulau.
- Terdapat pulau yang memiliki destinasi wisata. Dari database ini bisa diketahui jenis wisata dan nama tempat wisata yang akan dituju. Untuk menuju destinasi wisata juga melalui rute yang dihitung dari Ibukota Provinsi.
- Pulau dibagi menjadi beberapa Provinsi, contohnya seperti Pulau Jawa yang di dalamnya terdapat provinsi Jawa Tengah, Jawa Barat, Jawa Timur dan DI Yogyakarta. Dari database ini ditampilkan nama provinsi, nama ibukota provinsi, luas wilayah, dan peraturan daerah yang ada di provinsi tersebut. Setiap Provinsi memiliki ibukota provinsi. Dari database ini kita bisa mengetahui nama ibukota suatu provinsi dan titik lokasi GPS dari ibukota provinsi.
- Provinsi dibagi lagi menjadi beberapa Kabupaten/Kota. Dari database ini ditampilkan nama kabupaten yang ada di suatu provinsi, nama kecamatan yang ada di kabupaten tersebut, luas wilayah kabupaten, dan peraturan daerah yang ada di kabupaten tersebut.
- Setiap Provinsi dan Kabupaten/Kota dipimpin oleh suatu pemerintahan yang memimpin penyelenggaraan pemerintahan.

## II. MODEL DATA ERD



Gambar detail model data ERD dapat dilihat di [bit.ly/ERDKependudukan](https://bit.ly/ERDKependudukan)

## III. MAPPING DAN RELATIONAL MODEL

### Step 1 : Mapping of Regular Entity Types

PROVINSI (KdProvinsi, Nama, Ibukota, LuasWilayah, PerdaProv)

STRUKTUR\_BIROKRASI (KdJabatan, NamaPejabat, JenisJabatan, PeriodeJabatan)

KABUPATEN\_KOTA (KdKabupaten, KdKecamatan, NamaKabupaten, NamaKecamatan, PerdaKab)

DESTINASI\_WISATA (KdDestinasiWisata, JenisWisata, NamaDestinasi)

RUTE (KdRute, KdTransportasi, StepRute, JarakRute, MedanRute, Transportasi, DurasiPerjalanan)

PULAU (KdPulau, NamaPulau, Lintang, Bujur)

KONDISI\_GEOGRAFIS (KdGeografis, BatasLaut, BatasDarat, LuasWilayah, Ketinggian)

ETNIS (KdEtnis, KdSuku, KdAgama, Suku, Ras, Agama, Bahasa)

### Step 2 : Mapping of Weak Entity Types

IBUKOTA (KdProvinsi, KdIbukota, NamaIbukota, TitikLokasiGPS)

KONDISI\_SOSIAL (KdPulau, KdKondisiSosial, AngkaKelahiran, AngkaKematian, JumlahPenduduk, MataPencarian, LajuPertumbuhanPenduduk, KepadatanPenduduk, DistribusiPenduduk)

### Step 3 : Mapping Binary 1:1 Relationship Types

KABUPATEN\_KOTA (KdProvinsi, KdKabupaten, KdKec, ProvIbukota, NamaProvinsi, LuasWilayah, PerdaProv, PerdaKab, KabupatenKota, Kecamatan)

IBUKOTA (KdProvinsi, KdIbukota, NamaIbukota, TitikLokasiGPS, NamaProvinsi, LuasWilayah, PerdaProv, ProvIbukota)

STRUKTUR\_BIROKRASI (KdProvinsi, KdJabatan, NamaProvinsi, LuasWilayah, PerdaProv, NamaIbukota, NamaPenjabat, JenisJabatan, PeriodeJabatan)

RUTE (KdRute, KdTransportasi, KdIbukota, StepRute, JarakRute, MedanRute, Transportasi, DurasiPerjalanan, NamaIbukota, TitikLokasiGPS)

### Step 4 : Mapping of Binary 1:N Relationship Types

PROVINSI (KdProvinsi, Nama, Ibukota, LuasWilayah, PerdaProv, KdKabupaten, KdKec)

PULAU (KdPulau, NamaPulau, Lintang, Bujur, KdGeografis)

### Step 5 : Mapping of Binary M:N Relationship Types

PROV\_MEMILIKI\_PULAU (KdProvinsi, KdPulau, JmlPulau)

DES\_MEMILIKI\_PULAU (KdWisata, KdPulau)

PUL\_MELEWATI\_RUTE (KdTrans, KdPulau)  
 PUL\_BERKONDISI\_SOSIAL (KdPulau, KdKondisiSosial)  
 PULAU\_DITINGGALI\_ETNIS (KdPulau, KdEtnis, KdSuku, KdAgama)  
 PUL\_MEMILIKI\_SUM (KdPulau, IdSumberDaya)

#### Step 6 : Mapping of Multivalued Attribute

LTK\_ASTRO (KdPulau, PLtkAstro)

#### Step 7 : Mapping of N-ary Relationship Types

MEMIMPIN (KdProv, KdJabatan, KdKabupaten, KdKecamatan, Nama, Ibukota, LuasWilayah, PerdaProv, NamaPejabat, JenisJabatan, PeriodeJabatan, NamaKabupaten, NamaKecamatan, PerdaKab)

#### Step 8 : Mapping Specialization

SUMBER\_DAYA (IdSumberDaya)  
 HAYATI (IdSumberDaya, KdSpesies, NamaSpesies, JenisSpesies)  
 NON\_HAYATI (IdSumberDaya, KdNonHayati, Bentuk, Nama)

#### Final

STRUKTUR\_BIROKRASI (KdJabatan, NamaPejabat, JenisJabatan, PeriodeJabatan)  
 DESTINASI\_WISATA (KdDestinasiWisata, JenisWisata, NamaDestinasi)  
 KONDISI\_GEOGRAFIS (KdGeografis, BatasLaut, BatasDarat, LuasWilayah, Ketinggian)  
 ETNIS (KdEtnis, KdSuku, KdAgama, Suku, Ras, Agama, Bahasa)  
 KONDISI\_SOSIAL (KdPulau, KdKondisiSosial, AngkaKelahiran, AngkaKematian, JumlahPenduduk, MataPencarian, LajuPertumbuhanPenduduk, KepadatanPenduduk, DistribusiPenduduk)  
 KABUPATEN\_KOTA (KdProvinsi, KdKabupaten, KdKec, ProvIbukota, NamaProvinsi, LuasWilayah, PerdaProv, PerdaKab, KabupatenKota, Kecamatan)  
 IBUKOTA (KdProvinsi, KdIbukota, NamaIbukota, TitikLokasiGPS, NamaProvinsi, LuasWilayah, PerdaProv, ProvIbukota)  
 STRUKTUR\_BIROKRASI (KdProvinsi, KdJabatan, NamaProvinsi, LuasWilayah, PerdaProv, NamaIbukota, NamaPenjabat, JenisJabatan, PeriodeJabatan)  
 RUTE (KdRute, KdTransportasi, KdIbukota, StepRute, JarakRute, MedanRute, Transportasi, DurasiPerjalanan, NamaIbukota, TitikLokasiGPS)  
 PROVINSI (KdProvinsi, Nama, Ibukota, LuasWilayah, PerdaProv, KdKabupaten, KdKec)  
 PULAU (KdPulau, NamaPulau, Lintang, Bujur, KdGeografis)  
 PROV\_MEMILIKI\_PULAU (KdProvinsi, KdPulau, JmlPulau)  
 DES\_MEMILIKI\_PULAU (KdWisata, KdPulau)  
 PUL\_MELEWATI\_RUTE (KdTrans, KdPulau)  
 PUL\_BERKONDISI\_SOSIAL (KdPulau, KdKondisiSosial)  
 PULAU\_DITINGGALI\_ETNIS (KdPulau, KdEtnis, KdSuku, KdAgama)  
 PUL\_MEMILIKI\_SUM (KdPulau, IdSumberDaya)  
 LTK\_ASTRO (KdPulau, PLtkAstro)  
 MEMIMPIN (KdProv, KdJabatan, KdKabupaten, KdKecamatan, Nama, Ibukota, LuasWilayah, PerdaProv, NamaPejabat, JenisJabatan, PeriodeJabatan, NamaKabupaten, NamaKecamatan, PerdaKab)  
 SUMBER\_DAYA (IdSumberDaya)  
 HAYATI (IdSumberDaya, KdSpesies, NamaSpesies, JenisSpesies)  
 NON\_HAYATI (IdSumberDaya, KdNonHayati, Bentuk, Nama)

## IV. IMPLEMENTASI

Implementasi program java :

```
package net.codejava.sql;

import java.util.*;
import java.sql.*;

public class dbms {

    public static void main(String[] args) {
        Connection connection = null;
        try {
            String url = "jdbc:sqlserver://LAPTOP-C2TTN0LB;databaseName= KEPENDUDUKAN4";
            String User = "bismillah";
            String Password = "1234";
            connection = DriverManager.getConnection(url,User,Password);

            PreparedStatement state ;

            Scanner inputCommand = new Scanner(System.in);
            Scanner inputTabel= new Scanner(System.in);
            Scanner inputKolom = new Scanner(System.in);
            Scanner inputData = new Scanner(System.in);
            Scanner stopKolomChoose = new Scanner(System.in);
            System.out.println("Masukkan command (insert, update, delete,exit)?");

            String sql_execute = "";
            String command_dipilih = inputCommand.nextLine(); // select , insert , update , delete
            switch(command_dipilih) {
                case "insert":
                    sql_execute = sql_execute + "INSERT INTO ";
                    System.out.println("Masukkan tabel yang akan diinput!");
                    String tabel_dipilih = inputTabel.nextLine();
                    sql_execute = sql_execute + tabel_dipilih + "(";

                    ArrayList<String> columns = new ArrayList<String>();
                    String stop_kolom = "yes";
                    while (stop_kolom.equals("yes")) {
                        System.out.println("Masukkan kolom yang akan diinput");
                        String kolom_dipilih = inputKolom.nextLine();
                        sql_execute = sql_execute + kolom_dipilih;
                        columns.add(kolom_dipilih);
                        System.out.println("ada lagi? (yes/no)");
                        stop_kolom = stopKolomChoose.nextLine();

                        if (stop_kolom.equals("yes")) {
                            sql_execute = sql_execute + ",";
                        }
                    }

                    sql_execute = sql_execute + ") VALUES ("; // sampai sini sql yang terbentuk "INSERT INTO nama_table ("nama_kolom") VALUES ("
                    // loopin untuk data dari kolom yang akan diinput
                    ArrayList<String> datas = new ArrayList<String>();
                    for (int i=1; i<columns.size(); i++) {
                        System.out.println("Masukkan data untuk kolom - " + columns.get(i));
                        String data = inputData.nextLine();
                        sql_execute = sql_execute + "?";
                        datas.add(data);

                        if (i != columns.size()-1) {
                            sql_execute = sql_execute + ",";
                        }
                    }
                    sql_execute = sql_execute + ")";
                    System.out.println(sql_execute);

                    // persiapan input
                    state= connection.prepareStatement(sql_execute);
                    for (int i=0; i<datas.size(); i++) {
                        state.setString(i+1, datas.get(i));
                    }

                    int rowsInserted = state.executeUpdate();
                    if (rowsInserted > 0) {
                        System.out.println("The new data was inserted successfully!");
                    }
                    break;
                case "update":
                    sql_execute = sql_execute + "UPDATE ";
                    System.out.println("Masukkan tabel yang ingin dipilih");
                    String tabel_pilih = inputTabel.nextLine();

                    sql_execute = sql_execute + tabel_pilih + " SET ";
                    String stop_kolom1 = "yes";
                    while (stop_kolom1.equals("yes")) {
                        System.out.println("Masukkan kolom yang akan diubah");
                        String kolom_dipilih = inputKolom.nextLine();
                        sql_execute = sql_execute + kolom_dipilih + "=";
                        System.out.println("Masukkan data untuk kolom - " + kolom_dipilih);
                        String data = inputData.nextLine();
                        sql_execute = sql_execute + data;

                        System.out.println("ada lagi? (yes/no)");
                        stop_kolom1 = stopKolomChoose.nextLine();
                        if (stop_kolom1.equals("yes")) {
                            sql_execute = sql_execute + " , ";
                        }
                    }

                    sql_execute = sql_execute + " WHERE ";
                    String stop = "yes";
            }
        }
    }
}
```

```

sql_execute = sql_execute + " WHERE ";
String stop = "yes";
while (stop.equals("yes")) {
    System.out.println("Masukkan kolom kondisi (where)");
    String kolom_pilih = inputKolom.nextLine();
    sql_execute = sql_execute + kolom_pilih + "=";
    System.out.println("Masukkan data untuk kolom - " + kolom_pilih);
    String data = inputData.nextLine();
    sql_execute = sql_execute + data;

    System.out.println("ada lagi? (yes/no)");
    stop = stopKolomChoose.nextLine();
    if (stop.equals("yes")) {
        sql_execute = sql_execute + " AND ";
    }
}

sql_execute = sql_execute + ";";
state = connection.prepareStatement(sql_execute);
System.out.println(sql_execute);
int rowsInserted = state.executeUpdate();
if (rowsInserted > 0) {
    System.out.println("Data updated");
}

break;
case "delete":
    sql_execute = sql_execute + "DELETE FROM ";
    System.out.println("Masukkan tabel yang ingin dipilih");
    String tabel_dipilih1 = inputTabel.nextLine();
    sql_execute = sql_execute + tabel_dipilih1 + " WHERE ";
    System.out.println("Masukkan kolom data yang ingin dipilih");
    String data_pilih = inputData.nextLine();
    sql_execute = sql_execute + data_pilih + "=";
    System.out.println("Masukkan data yang ingin dihapus");
    String data2_dipilih = inputData.nextLine();
    sql_execute = sql_execute + data2_dipilih;
    state = connection.prepareStatement(sql_execute);

    int rowsDeleted = state.executeUpdate();
    if (rowsDeleted > 0) {
        System.out.println("the Data was deleted successfully!");
    } else System.out.println("data not found!");
    break;
}

} catch (SQLException e) {
    System.out.println("oops");
    e.printStackTrace();
}
}
}

```

Jalannya program :

Insert

Masukkan command (insert, select, update, delete)?

insert

Masukkan tabel yang akan diinput!

NONHAYATI

Masukkan kolom yang akan diinput

IdSumberDaya

ada lagi? (yes/no)

yes

Masukkan kolom yang akan diinput

KdNonHayati

ada lagi? (yes/no)

yes

Masukkan kolom yang akan diinput

Bentuk

ada lagi? (yes/no)

yes

Masukkan kolom yang akan diinput

NamaSumberDayaNonHayati

ada lagi? (yes/no)

no

Masukkan data untuk kolom - IdSumberDaya

6

Masukkan data untuk kolom - KdNonHayati

12

Masukkan data untuk kolom - Bentuk

Cair

Masukkan data untuk kolom - NamaSumberDayaNonHayati

Air

INSERT INTO NONHAYATI(IdSumberDaya,KdNonHayati,Bentuk,NamaSumberDayaNonHayati) VALUES (?, ?, ?, ?);

The new data was inserted successfully!

## Delete

Masukkan command (insert, update, delete, exit)?

delete

Masukkan tabel yang ingin dipilih

NONHAYATI

Masukkan kolom data yang ingin dipilih

IdSumberDaya

Masukkan data yang ingin dihapus

6

the Data was deleted successfully!

## Update

Masukkan command (insert, update, delete)?

update

Masukkan tabel yang ingin dipilih

KONDISIGEOGRAFIS

Masukkan kolom yang akan diubah

Ketinggian

Masukkan data untuk kolom - Ketinggian

3214

ada lagi? (yes/no)

no

Masukkan kolom kondisi (where)

KdGeografis

Masukkan data untuk kolom - KdGeografis

2

ada lagi? (yes/no)

no

UPDATE KONDISIGEOGRAFIS SET Ketinggian=3214 WHERE KdGeografis=2;

Data updated

Query informasi yang bisa disediakan dari database sebagai VIEW

Dari data :

	KdPulau	NamaPulau	Lintang	Bujur	KdGeografis
1	1	Jawa	7	113	1
2	2	Kalimantan	1	114	2
3	3	Sumatera	1	98	3

	KdKondisiSosial	KdPulau	AngkaKelahiran	AngkaKematian	JumlahPenduduk	DistribusiPenduduk
1	1	1	50000	12000	4500000	padat
2	2	2	3000	600	1258901	tidak padat
3	3	3	43187	5543	5049214	padat
4	4	1	26785	2232	8572401	padat

Pulau dengan kepadatan penduduk terbanyak :

USE KEPENDUDUKAN4

CREATE VIEW KEPADATAN\_TERBANYAK AS

```
SELECT TOP(1) P>NamaPulau FROM PULAU as P
JOIN KONDISISOSIAL AS G ON P.KdPulau = G.KdPulau
where G.DistribusiPenduduk='padat'
ORDER BY G.DistribusiPenduduk DESC
```



Results Messages

	NamaPulau
1	Jawa

## V. REFERENSI

<https://www.codejava.net/java-se/jdbc/jdbc-tutorial-sql-insert-select-update-and-delete-examples>  
<https://youtu.be/1pA64-foiz8>

## VI. KELOMPOK

NIM	Nama/email	Role	Tanggungjawab	TTD
M0519042	Gestina Ratna Pratiwi <a href="mailto:gestina012@student.uns.ac.id">gestina012@student.uns.ac.id</a>	Ketua	<ul style="list-style-type: none"><li>- Membantu menyusun diagram ERD</li><li>- Memperbaiki ERD</li><li>- Membuat program java</li><li>- Membantu membuat script query untuk membuat database</li><li>- Membuat query view informasi database</li></ul>	
M0519049	Indah Paksi Larasati <a href="mailto:ip_larasati@student.uns.ac.id">ip_larasati@student.uns.ac.id</a>	Anggota	<ul style="list-style-type: none"><li>- Menyusun diagram ERD</li><li>- Membuat program java</li><li>- Membuat script query untuk membuat database</li><li>- Membantu membuat query view informasi database</li></ul>	
M0519057	Lutfiana Arifah <a href="mailto:lutfianarfh@student.uns.ac.id">lutfianarfh@student.uns.ac.id</a>	Anggota	<ul style="list-style-type: none"><li>- Menyusun diagram ERD</li><li>- Mapping ERD</li><li>- Membuat program java</li><li>- Membantu membuat script query untuk membuat database</li><li>- Menyusun laporan</li></ul>	