



رَبِّ زِدْنِي عِلْمًا، وَارْزُقْنِي فَهْمًا وَاجْعَلْنِي مِنَ الصَّالِحِينَ

Rabbii Zidnii 'Ilmaa, Warzuqnii Fahmaa, Waj'alnii Minash-Shaalihiin.

Artinya : "Ya Allah... Tambahkanlah aku ilmu, dan berilah aku karunia untuk dapat memahaminya, dan jadikanlah aku termasuk golongannya orang-orang yang shaleh."

CSH3E3

Implementasi dan Pengujian

Tim Dosen

KK SIDE



Pertemuan #5

Implementasi Hasil Desain OO



Sub Bahasan

- ▶ Review hasil Analisis & Perancangan OO
- ▶ Coding dari Class Diagram
- ▶ Coding hubungan antar Class
- ▶ Coding dari Sequence Diagram
- ▶ Tugas eksplorasi



Sub Bahasan 1

Review hasil Analisis & Perancangan OO

Hasil Analisis & Perancangan OO

- ▶ **Activity Diagram** : menunjukkan proses bisnis “manual” yang akan digantikan/dibantu dengan software yang akan dibangun
- ▶ **Use Case Diagram** : menunjukkan fitur apa saja dari software yang dapat dipakai pengguna
- ▶ **Use Case Scenario** : urutan interaksi user dengan system di setiap use case
- ▶ **Class Diagram** : menunjukkan class-class yang akan dibangun
- ▶ **Object Diagram** : mensimulasikan interaksi sampel object-object Class Diagram

- ▶ **Sequence Diagram atau Collaboration Diagram** : menunjukkan class-class yang berinteraksi di dalam suatu Use Case Scenario
- ▶ **State Transition Diagram** : perubahan status objek dari awal transaksi hingga akhir transaksi
- ▶ **Component atau Package Diagram** : menunjukkan komponen atau package program yang berhasil dibangun
- ▶ **Deployment Diagram** : menunjukkan lokasi hardware tempat instalasi software yang dibangun, khususnya jika software client server.



- ▶ Rancangan IMK
- ▶ Rancangan persistensi objek dalam basis data relational atau OO



Diagram yang mana untuk Code?

- ▶ Some UML diagrams close to be transformed into programming code
 - Class Diagram
 - Sequence Diagram



Sub Bahasan 2

Coding dari Class Diagram



Class

Cat
-name : String -age : int
+run() +mew() +eat() +sleep()

```
class Cat{  
    private String name;  
    private int age;  
  
    //accsesor method  
  
    public void run(){  
        System.out.print("I am running");  
    }  
  
    public void mew(){  
        System.out.print("miawww...");  
    }  
  
    //other methods...  
  
}
```



Buat Objectnya

Cat
-name : String -age : int
+run() +mew() +eat() +sleep()

```
public static void main(String[] args){  
    Cat persia = new Cat();  
    persia.run();  
    persia.mew();  
    persia.mew();  
    persia.sleep();  
}
```

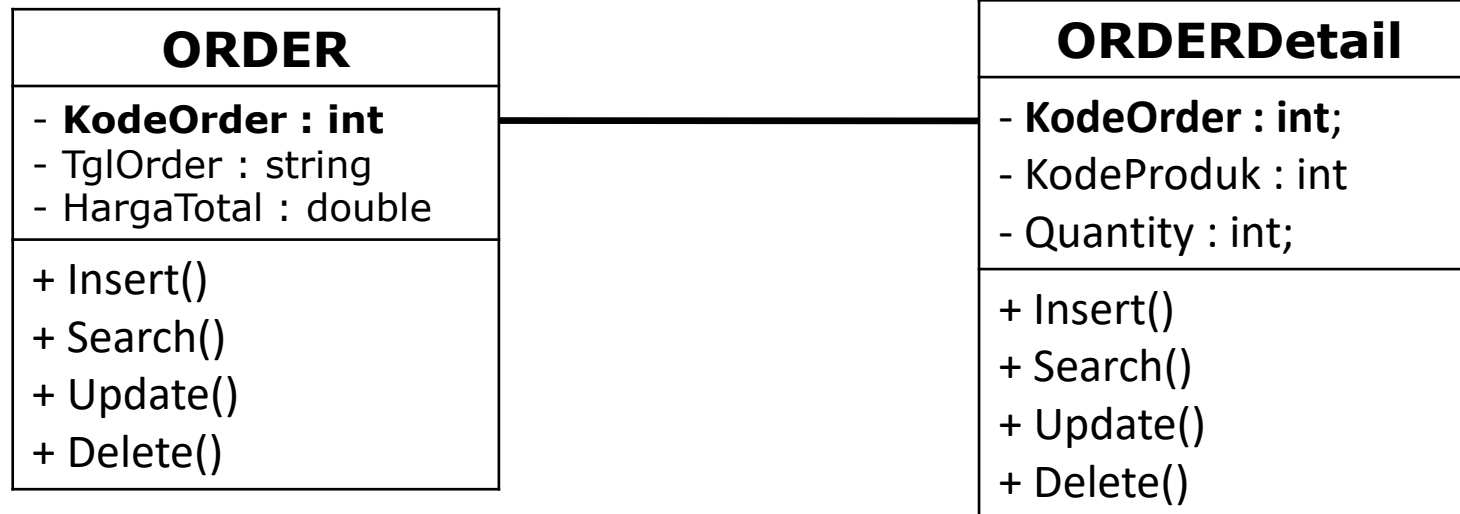


Sub Bahasan 3

Coding Hubungan Antar Class

Coding Hubungan Antar Class

- ▶ Studi Kasus : hubungan class ORDER dan ORDERDetail
- ▶ Kedua class tersebut dapat terhubung secara **Asosiasi**, dalam kedua class ada atribut pengait, semacam primary key-foreign key
- ▶ Misal :





Perhatikan variable KodeOrder ada di kedua Class, sbb:

```
class ORDER{  
    int KodeOrder;  
    String TglOrder;  
    double HargaTotal;
```

```
void Insert(){}  
void Search(){}  
void Update(){}  
void Delete(){}  
}
```

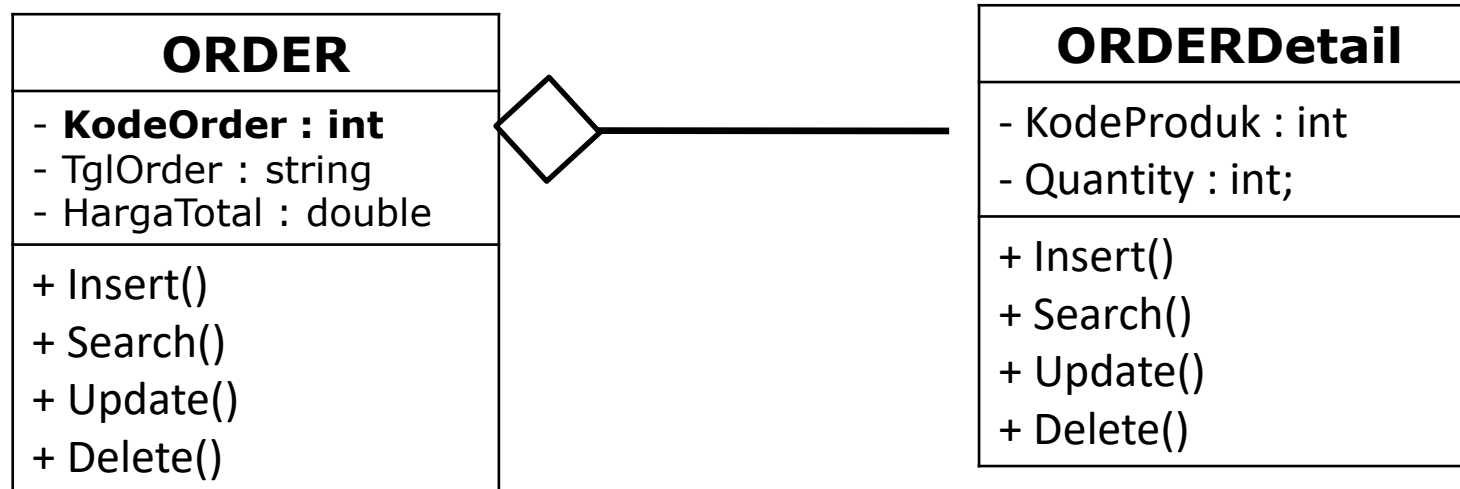
```
class ORDERDetail{  
    int KodeOrder;  
    int KodeProduk;  
    int quantity;
```

```
void Insert(){}  
void Search(){}  
void Update(){}  
void Delete(){}  
}
```

Hubungan Agregasi

- ▶ Class ORDER dan ORDERDetail juga dapat terhubung **Agregasi**, Suatu object Order dapat berisi sejumlah objek OrderDetail.

- ▶ Misal :





Perhatikan variable rincianProduk[] yang bertipe ORDERDetail, sbb:

```
class ORDER{  
  
    private int KodeOrder;  
    private String TglOrder;  
    private double HargaTotal;  
    private ORDERDetail rincianProduk[];  
  
    void Insert{}  
    void Search{}  
    void Update{}  
    void Delete{}  
}
```

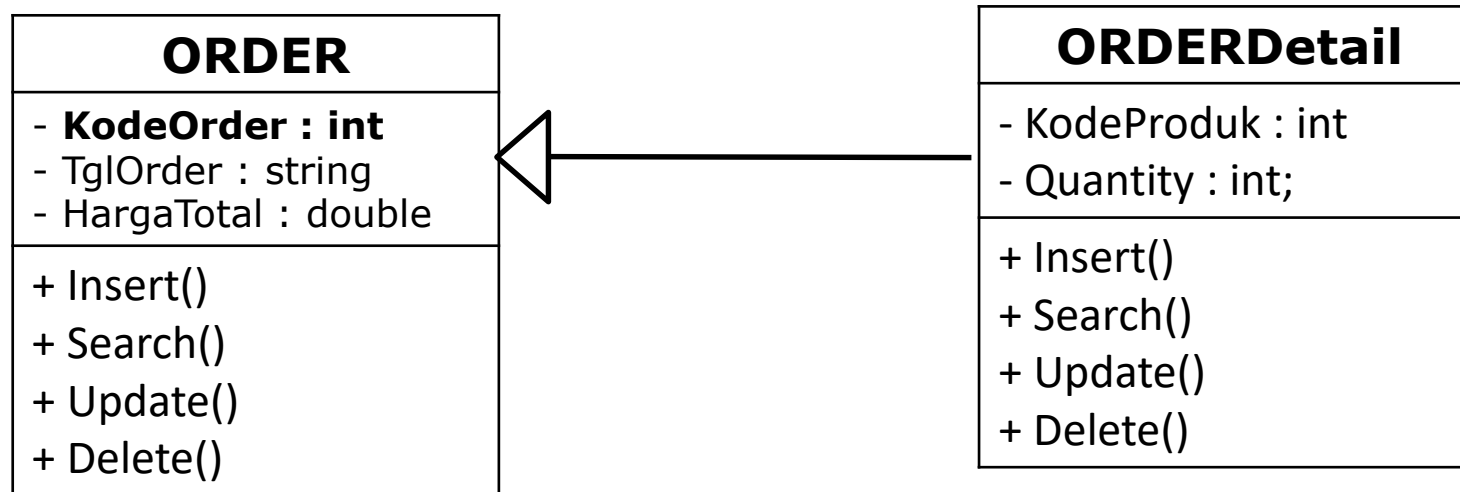
```
class ORDERDetail{  
    private int KodeProduk;  
    private int quantity;  
  
    void Insert{}  
    void Search{}  
    void Update{}  
    void Delete{}  
}
```


Hubungan Inheritance

▶ Class ORDER dan ORDERDetail juga dapat terhubung **Inheritance**,

Suatu object OrderDetail mewarisi atribut & method object Order.

▶ Misal :





Perhatikan class ORDERDetail yang mengextends class ORDER, sbb:

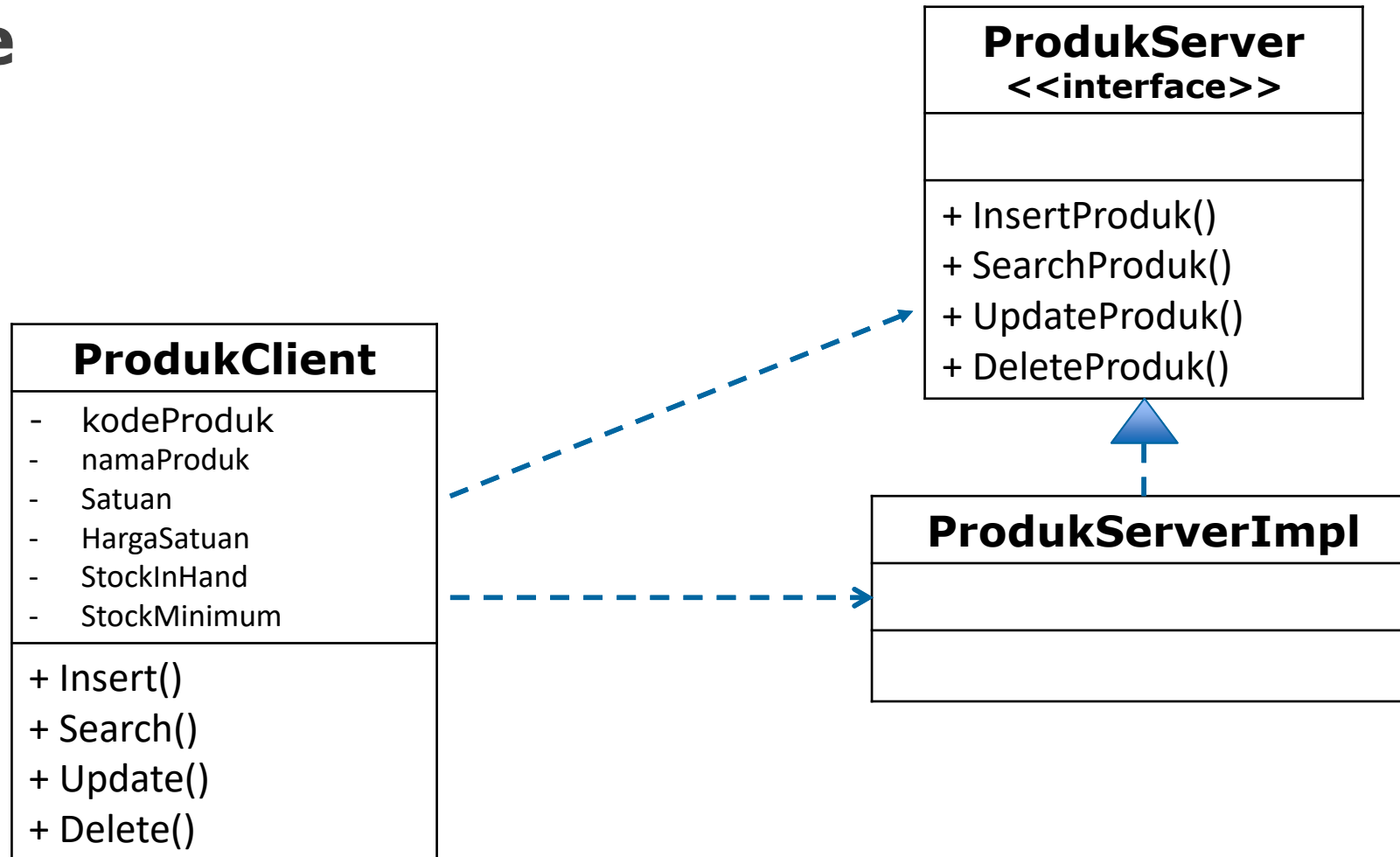
```
class ORDER{  
  
    private int KodeOrder;  
    private String TglOrder;  
    private double HargaTotal;  
  
    void Insert{}  
    void Search{}  
    void Update{}  
    void Delete{}  
}
```

```
class ORDERDetail extends ORDER{  
    private int KodeProduk;  
    private int quantity;  
  
    void Insert{}  
    void Search{}  
    void Update{}  
    void Delete{}  
}
```



Interface

Misal :





Code Interface : ProdukServer

```
import java.rmi.*;
import java.util.*;

public interface ProdukServer extends Remote
{
    /* Declare the remote method. */
    String insertProduk(String kp, String np, String st, String hs, String sih, String sm)
    throws RemoteException;
    Vector searchProduk(String kp) throws RemoteException;
    String deleteProduk(String kp) throws RemoteException;
    String updateProduk(String kp, String np, String st, String hs, String sih, String sm)
    throws RemoteException;
}
```



```
import java.rmi.*;    import java.rmi.server.UnicastRemoteObject;    import java.sql.*;    import java.util.*;    import javax.swing.*;

public class ProdukServerImpl extends UnicastRemoteObject implements ProdukServer {
    static ResultSet rs;    static Connection con;    static PreparedStatement stat;

    /* Define the default constructor of the Impl class */
    public ProdukServerImpl() throws RemoteException { super(); }
    /* Define the remote method */
    public String insertProduk(String kp, String np, String st, String hs, String sih, String sm) throws RemoteException {
        int rowsAffected = 0;
        String sReturn = "fail";

        try {
            /* Register the database driver. */
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            /* Create the connection. */
            con=DriverManager.getConnection("jdbc:odbc:DBToserba", "", "");

            /* Create the preparedStatement to insert the value in the database. */
            stat=con.prepareStatement("insert into produk values (?, ?, ?, ?, ?, ?)");
            stat.setString(1, kp); stat.setString(2, np); stat.setString(3, st); stat.setString(4, hs); stat.setString(5, sih); stat.setString(6, sm);
            rowsAffected = stat.executeUpdate();

            if(rowsAffected>0) { sReturn = "success"; }
            else{ System.out.println("Gagal insert" + v); }
        }
        catch(Exception v) { System.out.println("Error at value insert" + v); }
        return sReturn;
    }
}
```

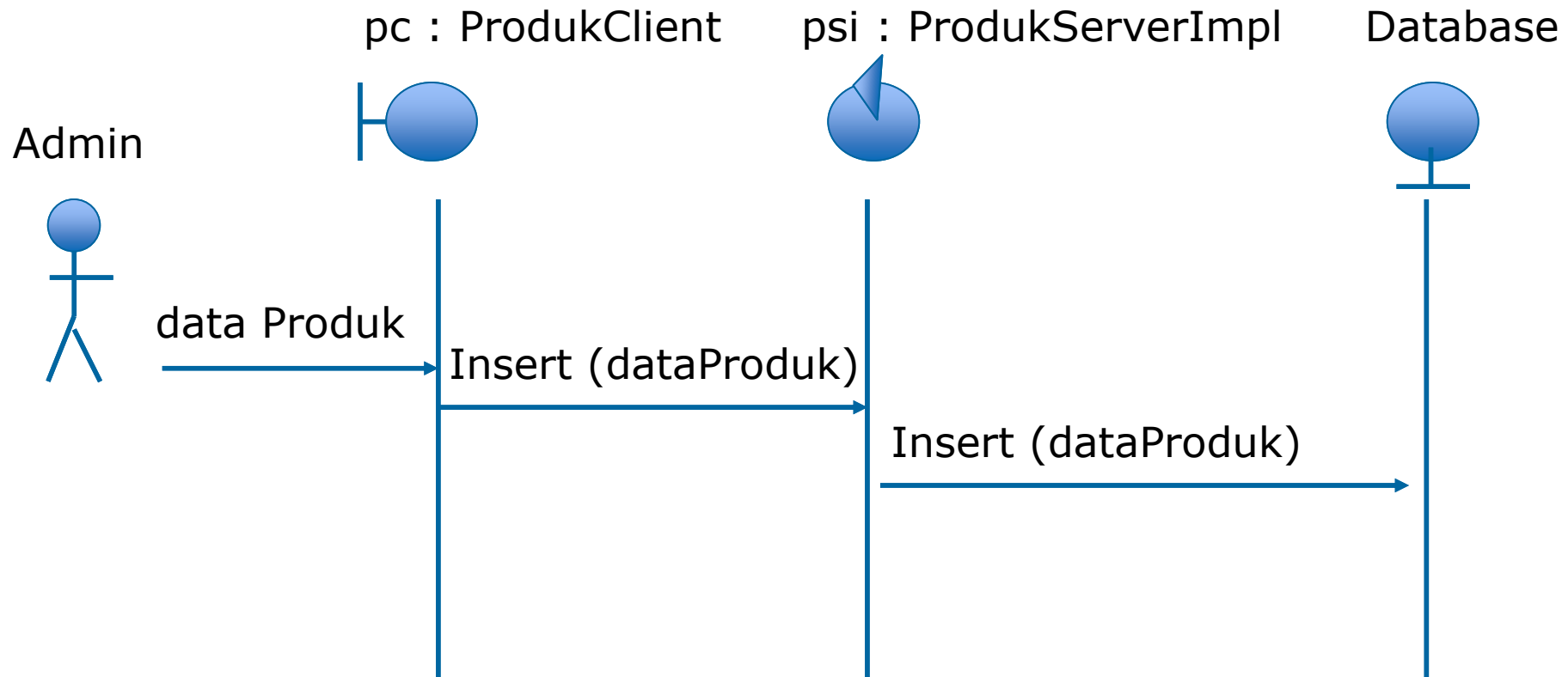


Sub Bahasan 4

Code dari Sequence Diagram

Code dari Sequence Diagram

Misal :





Implementasi dalam bentuk aplikasi Java RMI (Remote Method Invocation)

1. Siapkan basis datanya di [Microsoft Access](#) atau yang lain.
Simpan di tempat public (di Windows 7) di laptop Anda.
2. Buat connector di ODBC (Control Panelnya Windows - System & Security - Administrative Tools - ODBC - Add DSN - pilih Driver, beri nama, pilih basis data)
3. Siapkan Server.
 - kompilasi interface [Produk Server](#) : `prompt> javac ProdukServer.java`
 - kompilasi [program server](#) (yang mengimplement interface) : `prompt> javac ProdukServerImpl.java`
 - siapkan konektor ke client : `rmic ProdukServerImpl`
 - siapkan server rmi nya Windows : `start rmiregistry`
(Kalau diblock oleh Windows, buka dulu di `prompt> policytool`
ubah kebijakan security : [All Permission](#),
simpan di [.java.policy](#) di folder [user](#) di Laptop Anda)
 - jalankan program server : `java ProdukServerImpl`
4. Siapkan Client
 - kompilasi [Produk Client](#) : `prompt> javac ProdukClient`
 - jalankan : `prompt> java ProdukClient`
 - uji koneksi program : `Client -- Server -- ODBC -- McAccess`



TUGAS Eksplorasi

Start Your Code In OOP :

- Java
- PHP 5 ke atas
- Code Igniter
- Laravel
- Atau yang lainnya

Simpan Code Di Github

Thank you for your continued support
and for making our community a better place.



THANK YOU