

PRAKTIKUM REKAYASA PERANGKAT LUNAK 2



MANUAL BOOK

“Toko Aksesoris Mobil” Menggunakan Netbeans

Nama Anggota :

- | | |
|----------------------------|------------|
| 1. Fajar Satriatna | (52416566) |
| 2. Herdi Susanto | (53416278) |
| 3. Muhammad Ficki | (54416852) |
| 4. R Ridwan Darmawan | (55416892) |
| 5. Trie Fauzan Febriansyah | (57416443) |

Kelas	: 4IA01
Fakultas	: Teknologi Industri
Jurusan	: Teknik Informatika
PJ	: Muhammad Naufaldi

**Ditulis Guna Melengkapi Sebagian Syarat Praktikum Rekayasa Perangkat Lunak 2
Jenjang S1
Universitas Gunadarma
2020**

DAFTAR ISI

DAFTAR ISI.....	ii
BAB I.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan	1
1.3 Batasan Masalah	1
BAB II	2
2.1 Netbeans	2
2.1.1 Fitur Java Netbeans	2
2.1.2 Fungsi Netbeans	2
2.1.3 Kelebihan Netbeans	3
2.1.4 Kekurangan Netbeans	3
2.2 OOP (<i>Object Oriented Programming</i>)	4
2.3 ERP (<i>Enterprise Resource Planning</i>)	6
2.4 ORM (<i>Object Relational Mapping</i>).....	6
2.4.1 Kelebihan ORM	6
2.5 Hibernate.....	7
2.6 Spring	8
2.6.1 Fitur-Fitur Spring Framework.....	9
2.6.2 Kelebihan Spring Framework	10
BAB III.....	11
OUTPUT	29
BAB IV	32
4.1 Kesimpulan	32
4.2 Saran.....	32

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dengan kemajuan teknologi sekarang, pengolahan data perusahaan lebih tertata dengan adanya penggunaan database. Dengan adanya pengolahan data menggunakan database, perusahaan akan lebih mudah untuk membuat ataupun memeriksa pelaksanaan bisnisnya dengan melihat data yang ada pada database.

Toko aksesoris mobil telah banyak bermunculan seiring dengan banyaknya orang yang mampu untuk memiliki mobil. Tetapi pencatatan transaksinya masih banyak yang menggunakan cara manual, dimana dengan menggunakan cara manual, adanya kesalahan input data dan juga melakukan perubahan data menjadi tidak efektif.

Dalam mengatasi hal tersebut, penulis membangun sebuah aplikasi berbasis desktop pengolahan data untuk digunakan pada toko aksesoris mobil. Pembuatan aplikasi tersebut menggunakan Netbeans dan Java sebagai bahasa pemrograman serta Spring sebagai frameworknya.

1.2 Tujuan

Dalam penulisan ini bertujuan membuat aplikasi desktop untuk mempermudah dalam mengolah data pada toko aksesoris mobil.

1.3 Batasan Masalah

Pembuatan aplikasi dalam penulisan ini dibatasi pada operasi simpan, ubah dan hapus

BAB II

PEMBAHASAN

2.1 Netbeans

Netbeans adalah suatu serambi pengembangan perangkat lunak yang ditulis dalam bahasa pemrograman Java. Serambi Pada NetBeans, pengembangan suatu aplikasi dapat dilakukan dimulai dari setelan perangkat lunak modular bernama *modules*. Netbeans adalah suatu aplikasi IDE atau Integrated Development Environment yang berbasis bahasa Java dan berjalan diatas Swing. Swing adalah suatu teknologi yang memungkinkan pengembangannya dijalankan pada *Microsoft Windows, Mac OS X, Linux, Solaris* dan serambi-serambi lainnya yang mendukung JVM yang sepadan. Sedangkan Integrated Development Environment suatu sistem pemrograman atau development dan diintegrasikan kedalam suatu perangkat lunak. Netbeans menyediakan beberapa tools seperti Graphic User Interface (GUI), kode editor atau text, suatu compiler serta debugger.

2.1.1 Fitur Java Netbeans

1. Smart Code Completion, untuk mengusulkan nama variabel dari suatu tipe, melengkapi keyword dan mengusulkan tipe parameter dari sebuah method.
2. Bookmarking, fitur yang digunakan untuk menandai baris yang suatu saat hendak kita modifikasi.
3. Go to commands, fitur yang digunakan untuk jump ke deklarasi variabel, source code atau file yang ada pada project yang sama.
4. Code generator, jika kita menggunakan fitur ini kita dapat meng-generate constructor, setter and getter method dan yang lainnya.
5. Error stripe, fitur yang akan menandai baris yang eror dengan memberi highlight merah.

2.1.2 Fungsi Netbeans

Suatu aplikasi Netbeans memiliki fungsi yang sangat penting dan wajib ada bagi setiap programmer, yakni sebagai media untuk menulis, melakukan compile, mencari error pada program yang berbasis bahasa Java, C/C++, dan bahkan dynamic

languages seperti PHP, JavaScript, Groovy, serta Ruby. Dengan fungsinya yang multi language ini Netbeans juga bisa dimanfaatkan untuk menghasilkan program yang berjalan dekstop, web, enterprise, serta mobile.

2.1.3 Kelebihan Netbeans

1. Aplikasi Netbeans bersifat *open source* gratis yang dapat didownload pada laman website resminya.
2. Sun Microsystem yang merupakan pengembang Swing dapat membuat aplikasi Netbeans ini sangat kompatibel dengan Swing. Sehingga menjadi salah satu aplikasi development penghasil program yang dapat berjalan multi platform.
3. Netbeans juga termasuk aplikasi development yang multi bahasa, karena selain compatible dengan Java Netbeans juga dapat digunakan untuk membangun program dengan bahasa lain seperti C/C++, Ruby, dan PHP.
4. Netbeans juga cocok untuk pengembangan sistem dengan skala Enterprise. Serta pada paket tertentu juga menyertakan GlassFish V2 UR2 dan Apache Tomcat 6.0.16 yang tentunya menjadi nilai tambah bagi para developer.

2.1.4 Kekurangan Netbeans

1. Sebuah keuntungan ketika Netbeans mensupport salah satu pengembangan Java GUI yakni Swing. Namun sayangnya diluar itu sebenarnya ada juga Java GUI yang dikembangkan oleh Eclipse yakni SWT dan JFace yang sudah cukup populer namun belum disupport oleh Netbeans.
2. Source code untuk Java GUI sudah dipatenkan oleh Netbeans dalam suatu Generated Code, sehingga para programmer dan developer tidak dapat merubah atau memodifikasi isinya secara manual.
3. Dalam hal spesifikasi Netbeans juga tergolong cukup tinggi. Perlu ruang memory dan harddisk yang cukup agar dapat menggunakannya, selain itu penggunaan processornya juga harus diperhatikan agar dapat dijalankan secara optimal.

2.2 OOP (*Object Oriented Programming*)

OOP (*Object Oriented Programming*) atau dalam bahasa Indonesia dikenal dengan pemrograman berorientasikan objek (PBO) merupakan sebuah paradigma atau teknik pemrograman yang berorientasikan Objek.

Pada OOP, Fungsi dan variabel dibungkus dalam sebuah objek atau class yang dapat saling berinteraksi, sehingga membentuk sebuah program. Dengan demikian, tidak akan ada lagi kode yang “berantakan.”

Semua akan kita bungkus dalam objek. Sebelum membuat objek, terlebih dahulu membuat class. Class bertugas untuk mengumpulkan prosedur / fungsi dan variabel dalam satu tempat. Sedangkan objek adalah sebuah variabel yang merupakan instance atau perwujudan dari Class.

Dalam Object Oriented Programming :

- Variabel disebut atribut atau properti;
- Fungsi disebut method.

Konsep Object Oriented Programming :

- Class (Kelas) adalah sebuah rancangan (mirip seperti struct) untuk mendefinisikan karakter dan perilaku dari object. yang merupakan kumpulan atas definisi dan fungsi-fungsi dalam suatu unit, untuk suatu tujuan tertentu.
- Object atau instance adalah dasar dari modularitas dan struktur pada OOP. dan merupakan representasi dari class, object akan memiliki sifat dan perilaku dari class yang digunakan.
- Encapsulation adalah konsep dalam implementasi untuk membungkus data dan fungsi menjadi satu entitas, dan membatasi akses dari luar class.
- Inheritance adalah konsep pewarisan class. Class juga dapat menurun dan memiliki apa yang dimiliki oleh class lainnya.
- Abstraction adalah konsep untuk mendisain sebuah object, teknik dalam menyembunyikan detail suatu proses dalam object tersebut. dengan tujuan untuk memfokuskan pengguna pada fungsi inti object.
- Polymorphism berasal dari bahasa Yunani yang berarti memiliki banyak bentuk, adalah kemampuan dalam menyampaikan pesan tertentu keluar dari hirarki objectnya, dimana

object yang berbeda memberikan tanggapan atau respon terhadap pesan yang sama sesuai dengan sifat masing-masing object.

Dalam PBO dasar kita harus memahami apa itu class, method, dan attribute, berikut penjelasannya :

- Class (Kelas) digunakan untuk membuat objek, dan berperan sebagai tipe data dari objek. Class merupakan sarana pengkapsulan kumpulan data dan kumpulan method yang mengoperasikan kumpulan data tersebut.
- Attribute adalah segala sesuatu (data) yang berhubungan dengan karakteristik objek dalam setiap kelas. Tiap atribut memiliki nilai berbeda untuk setiap objek.
- Method merupakan tingkah laku dari suatu objek, jika bersifat static berarti tingkah laku semua objek dalam kelas tersebut.

Modifier untuk Method dan Attribute :

1. Public

Menyatakan bahwa kelas/method/attribute tersebut dapat diakses oleh kelas lain dimanapun.

2. Protected

Menyatakan bahwa kelas/method/attribute tersebut dapat diakses oleh kelas lain yang berada dalam satu package atau kelas lain tersebut merupakan turunannya.

3. Private

Menyatakan bahwa kelas tersebut tidak dapat diakses sama sekali oleh kelas lain bahkan juga tidak dapat diturunkan. Berarti attribute-attribute yang private hanya dapat diakses oleh method-method dalam kelas yang sama.

4. Final (no extended, no overrided)

Menyatakan bahwa suatu kelas tidak dapat menurunkan (extend) kelas lain.

Menyatakan bahwa suatu method tidak dapat dioverride oleh method lain.

5. Static (no need instantiation, no overrided)

Method dan attribute ada dua jenis, yaitu method dan attribute milik kelas serta method dan attribute milik suatu objek.

6. Abstract (no instantiation, should be overrided)

Abstract class adalah kelas murni (tanpa objek) dan tidak boleh memiliki objek (tidak boleh ada instansiasi).

Dan lain sebagainya.

2.3 ERP (*Enterprise Resource Planning*)

ERP (Enterprise Resource Planning) adalah salah satu aplikasi perangkat lunak yang biasa digunakan untuk membantu dan memudahkan pekerjaan dengan lebih cepat, efisien, dan tepat. ERP bersifat *open source*.

2.4 ORM (*Object Relational Mapping*)

ORM (*Object Relational Mapping*) adalah suatu metode/teknik pemrograman yang digunakan untuk mengkonversi data dari lingkungan bahasa pemrograman berorientasi objek (OOP) dengan lingkungan database relasional. Seperti kita ketahui, dalam aplikasi enterprise kedua lingkungan tersebut berada pada sistem yang berbeda, yaitu OOP berada pada sisi pemrograman aplikasi, sedangkan database relasional berada pada sisi sistem database. Misi utama dari ORM ini adalah menjembatani kedua sistem yang berbeda tersebut.

ORM memiliki kemampuan untuk menciptakan objek database virtual, yaitu suatu model database yang di representasikan kedalam sebuah objek pada bahasa pemrograman OOP.

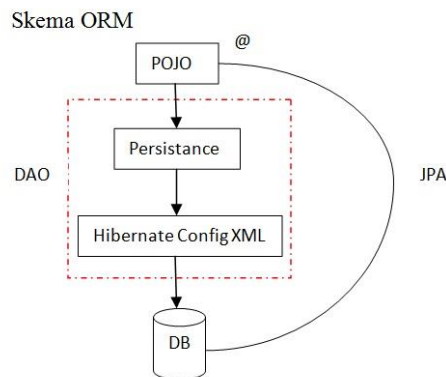
2.4.1 Kelebihan ORM

- Fasilitas penerapan pola Domain Model (Pattern Domain Model). Menggunakan pola ini berarti model entitas berdasarkan konsep bisnis yang nyata dan bukan berdasarkan struktur database anda. ORM menyediakan fungsi ini melalui pemetaan antara model bisnis logis dan model penyimpanan fisik.
- Mempercepat pengembangan program. Contohnya, mengurangi perulangan kode query, memudahkan pemakaian karena tabel-tabel ter-representasikan dalam bentuk objek.
- Membuat akses data menjadi lebih abstrak dan portable. Hal ini dikarenakan ORM menghandle pen-generate-an syntax SQL berdasarkan vendor database-nya.

- Mensupport pengkapsulan business rule pada lapisan Data Access.
- Mengenerate boilerplate code (unit kode yang reusable) untuk fungsi dasar CRUD (Create,Read,Update,Delete).

2.5 Hibernate

Hibernate adalah Framework untuk *Object Relation Mapping* (ORM) yang mana pada hibernate memetakan objek-objek kedalam database. Object Relational Mapping merupakan teknik pemrograman yang memetakan sebuah objek dengan database. ORM merupakan 'jembatan' penghubung antara objek yang dibuat dalam program dengan database sehingga dalam implementasinya tidak perlu lagi menggunakan query SQL dalam memanipulasi data di database karena objek yang dibuat dimapping sehingga ORM yang akan mengeksekusi ke dalam database.



Yang harus dilakukan dalam membuat Hibernate Project :

1. Buat Project baru, buat package pada folder src yaitu package : domain, dao, main, util.
2. Import Library untuk dapat menjalankan hibernate, lalu add Jar ke dalam library, klik kanan di project pilih properties dan pilih Java Built Path dan pilih add Jar.
3. Pada package domain buat POJO dari kelas diagram misalkan akan membuat kelas Karyawan yang memiliki atribut nip, nama, gaji dan buat juga table di database dengan field yang sama dengan atribut class.

4. Buat class dan JPA ny yang mana JPA ini merupakan mapping antara atribut dalam kelas di pojo dengan field di database dengan menambahkan notasi @.
5. Buat folder baru dengan nama resources dan pada folder tersebut buat hibernate config (xml) dengan nama “hibernate.cfg.xml” di project tersebut untuk konfigurasi koneksi ke database, menentukan dialect dan cara eksekusi dari objek ke database.
6. Pada package util buat file “HibernateUtil.java” yang berisi sessionFactory untuk mengatur eksekusi yang akan dijalankan ke dalam database ketika menjalankan fungsi insert, delete, update dll.
7. Pada package dao, buat DAO Interface dengan nama “DAOInt.java” yang mana pada interface ini berisi fungsi-fungsi yang harus ada untuk dapat mengeksekusi objek ke dalam database. Interface : Kontrak terhadap kelas-kelas yang implements ke interface tersebut sehingga harus memiliki fungsi-fungsi yang ada dalam interface.
8. Buat kelas object DAO yang implements terhadap DAOInt dan harus me-override fungsi yang ada dalam DAOInt, fungsi kelas objek dao ini adalah untuk fungsi eksekusi ke dalam database yaitu save, edit, delete dll. Buat kelas, contoh : “Karyawan Dao.java”.
9. Daftarkan kelas yang mengandung annotated pada POJO kedalam hebernate.cfg.xml. contoh : <mapping class=“bnis.domain.Karyawan”/>.
10. Buat kelas utama untuk menjalankan class-class yang sudah diimplementasikan pada package main.

2.6 Spring

Framework Spring pada Java adalah framework open source berbasis Java yang menyediakan infrastruktur yang komprehensif dalam mengembangkan aplikasi Java dengan mudah dan cepat. Spring pertama kali ditulis dan dirilis oleh Rod Johnson dengan lisensi Apache 2.0 pada bulan Juni 2003. Spring akan membantu pembaca dalam pengembangan aplikasi dengan build yang sederhana, portable, cepat dan sistem berbasis JVM yang fleksibel.

Spring bertujuan untuk mengatasi masalah desain sistem dalam pengembangan aplikasi enterprise. Spring juga mengimplementasikan beberapa teknologi IoC (Inversion of Control) kedalam sebuah MVC (Model-View-Controller).

Dengan menggunakan Spring Framework, developer dapat membuat aplikasi enterprise ataupun web. Selain itu juga, para developer dapat membuat aplikasi untuk keamanan dan aplikasi yang terkait dengan big data. Spring termasuk portabel karena aplikasi yang dikembangkan dapat berjalan pada JVM manapun. Untuk menggunakan Spring, developer dapat menggunakan Eclipse atau Netbeans.

Tapi jika developer menggunakan Eclipse, bisa menggunakan Spring Tool Suites yang merupakan tool pengembangan berbasis Eclipse yang disesuaikan untuk mengembangkan aplikasi Spring. Ini merupakan tool yang siap digunakan, debug, run dan deploy aplikasi Spring termasuk integrasi untuk Pivotal tc Server, Pivotal Cloud Foundry, Git, Maven, AspectJ dan lainnya.

2.6.1 Fitur-Fitur Spring Framework

1. Java Inversion of Control (IoC) Container atau disebut juga Dependency Injection. Spring membantu pembuatan kontainer dalam menampung object-object yang memiliki ketergantungan pada object lain dalam melakukan transaksi. IoC berupa layer abstrak yang menampung object-object yang akan digunakan secara singleton (singleton adalah object yang dibuat sekali dan ditampung terus dalam memory selama aplikasi berjalan).
2. Basis Web
 - Web MVC Framework Spring MVC menyediakan fitur pengembangan web yang memisahkan proses bisnis pada User Interface (Client Layer), Kontroller or Remote Server (Server Layer) dan Data Access Layer kedalam Model-View-Controller (MVC).
 - Web Enterprise-Java-Beans (EJB) Integration, yaitu teknologi component oriented yang memudahkan Konversi/Interaksi User Interface melalui Java Beans.
3. Transaction Management

Spring framework menyediakan sebuah layer abstrak yang generik untuk manajemen transaksi, sehingga memudahkan para developer dalam melakukan manajemen transaksi.

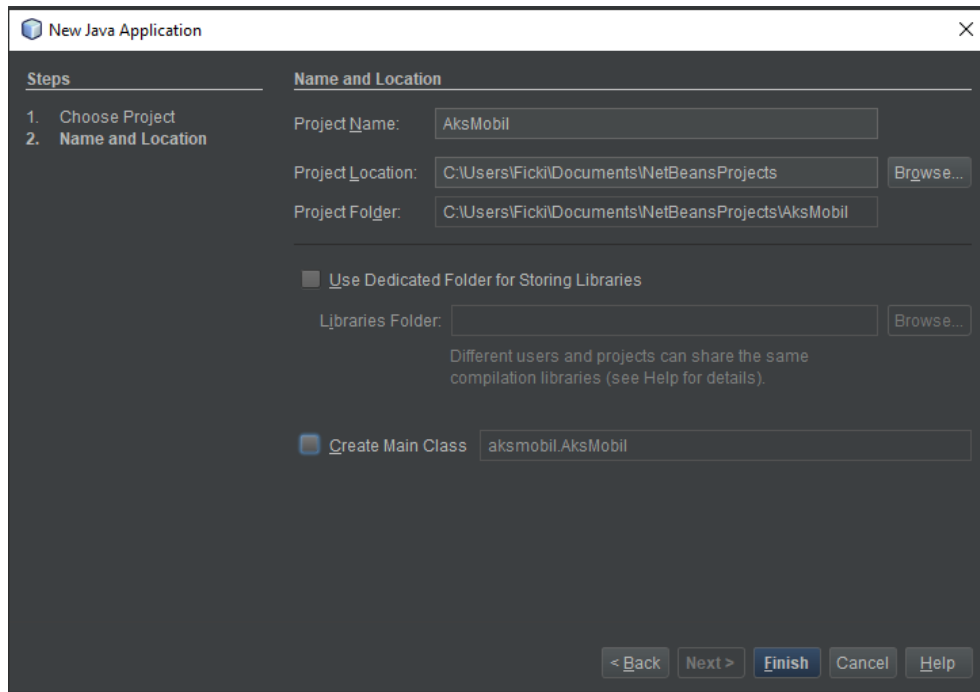
2.6.2 Kelebihan Spring Framework

1. Build Anything, yaitu menulis kode yang bersih, dapat diuji terhadap komponen infrastruktur pilihan dan lakukan tugas apa pun.
2. Run Anywhere, yaitu aplikasi spring berbasis berjalan dimana saja yang menggunakan JVM.
3. Rest Assured, yaitu kode dengan percaya diri. Spring menyediakan model pemrograman terbuka yang komprehensif, kohesif, dipahami secara luas dan didukung dengan baik.
4. Pola MVC(*Model View Controller*)
5. Modul Generator
6. ORM (*Object Relational Mapping*)
7. Ketersediaan Plugin

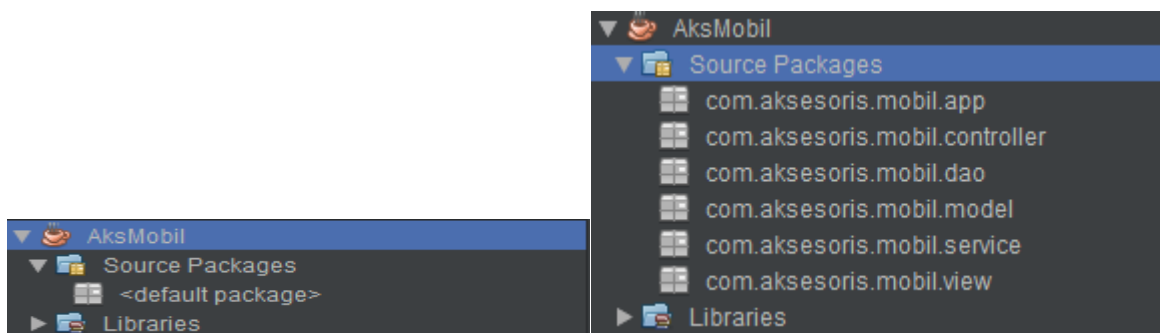
BAB III

ANALISA DAN PERANCANGAN

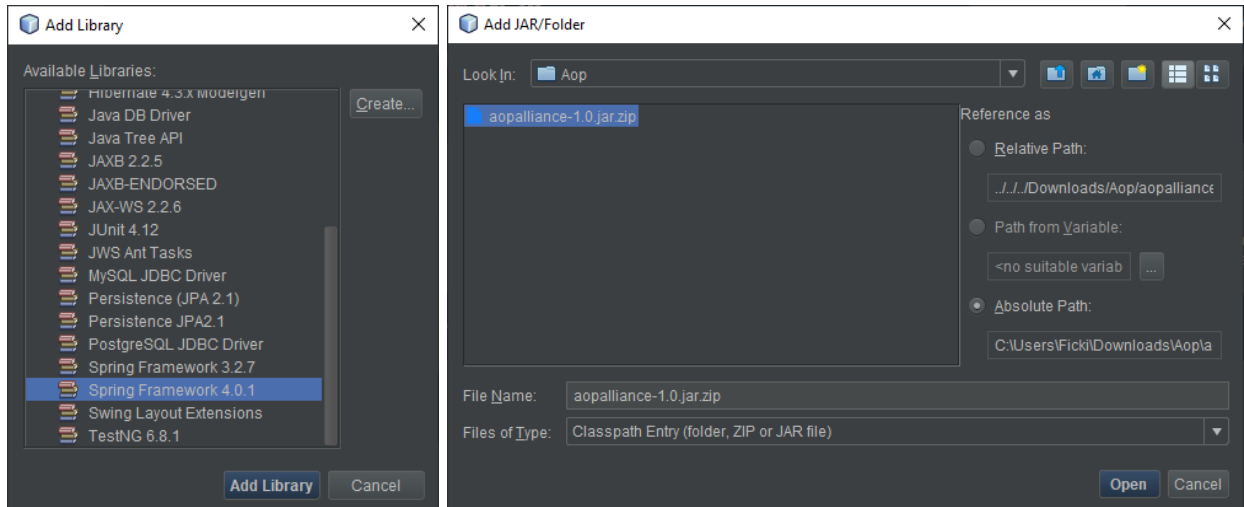
Pada proyek ini, kelompok kami mengambil tema Toko Aksesoris Mobil, dengan menggunakan framework Spring. Langkah pertama, buka netbeans buat project baru, kemudian beri nama “AksMobil”. Uncheck box Create Main Class, kemudian finish.



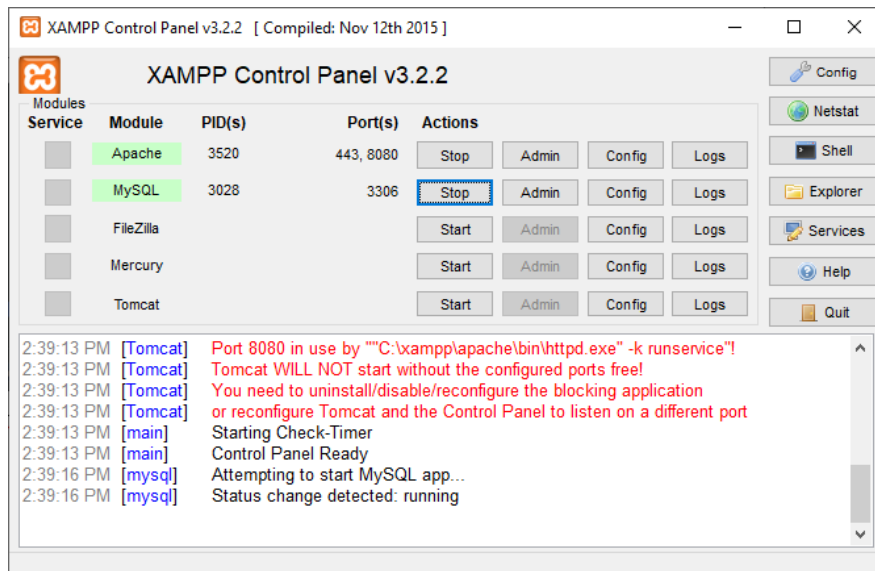
Jika sudah, proyek yang terbentuk akan seperti ini (kiri), maka langkah selanjutnya buat 6 package baru yaitu com.aksesoris.mobil.app, com.aksesoris.mobil.controller, com.aksesoris.mobil.dao, com.aksesoris.mobil.model, com.aksesoris.mobil.service, com.aksesoris.mobil.view (kanan).



Kemudian import library Spring framework 4.0.1, Hibernate 4.3.x (JPA2.1), dan Mysql JDBC Driver dengan mengklik kanan pada library kemudian Add library (kiri). Setelah itu import juga aopalliance-1.0 dengan mengklik kanan pada library dan pilih add JAR/folder, dan pilih filenya(kanan).



Karena project ini menggunakan database, nyalakan xampp dan check box mysql, kemudian buka localhost/phpmyadmin pada browser.



Kemudian buat database beri nama db_aks mobil, dengan nama table barang, dan ketentuan field di dalamnya seperti berikut.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	kode_barang	varchar(8)	latin1_swedish_ci		No	None			Change Drop More
2	nama_barang	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
3	merek_barang	varchar(10)	latin1_swedish_ci		No	None			Change Drop More
4	stok_barang	int(4)			No	None			Change Drop More
5	garansi_barang	varchar(4)	latin1_swedish_ci		No	None			Change Drop More
6	harga_barang	int(11)			No	None			Change Drop More

Setelah membuat database, maka langkah selanjutnya yaitu membuat java class pada package com.aksesoris.mobil.model dengan nama class **AksMobil.java**, seperti diketahui pada

framework website fungsi dari model yaitu untuk untuk mapping tabel ke dalam database. Untuk code pada class tersebut dapat dilihat pada gambar berikut.

```
package com.aksesoris.mobil.model;

import ...5 lines

@Entity
@Table(name="barang")
public class AksMobil implements Serializable{
    @Id
    @Column(name="kode_barang", length=8)
    private String kode_barang;

    @Column(name="nama_barang", length=30)
    private String nama_barang;

    @Column(name="merek_barang", length=10)
    private String merek_barang;

    @Column(name="stok_barang")
    private int stok_barang;

    @Column(name="garansi_barang", length=4)
    private String garansi_barang;

    @Column(name="harga_barang")
    private int harga_barang;
}
```

@Table(name="barang") => berfungsi untuk mapping nama table.

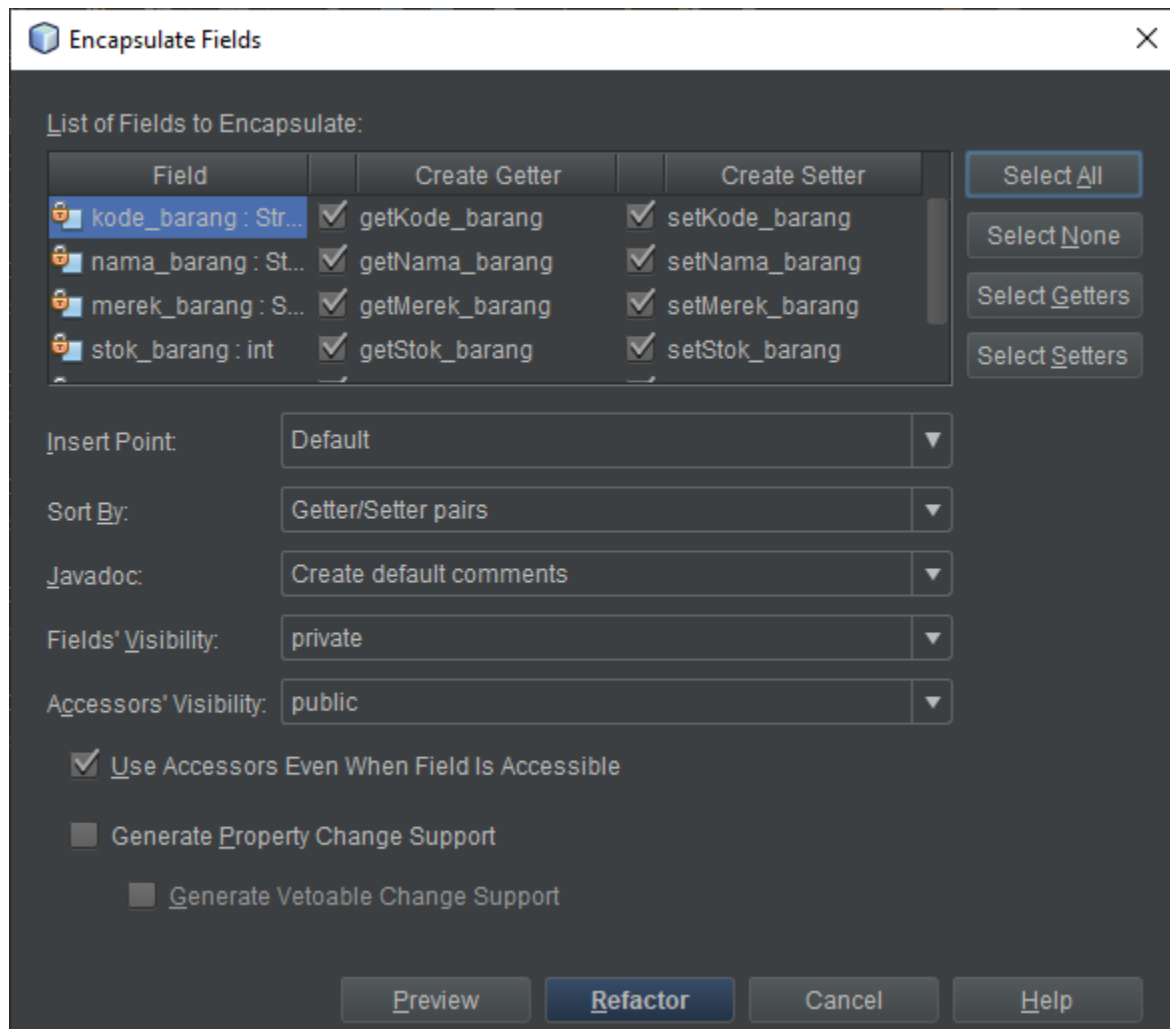
Lalu ada class Mobil yang di implements Serializable, berfungsi untuk mewariskan interface Serializable.

Lalu selanjutnya ada @Column yang berfungsi untuk memberikan nama-nama column, terdapat kode_barang, nama_barang, merek_barang, stok_barang, garansi_barang, harga_barang dengan ketentuan seperti pada database yang telah dibuat.

Anotasi @Entity berguna untuk menandai bahwa kelas tersebut adalah sebuah kelas entity/entitas.

Anotasi @Id menandakan bahwa field/property/attribut yang dibawahnya merupakan id dari kelas tersebut atau yang biasa disebut dengan primary key.

Kemudian melakukan Encapsulate Fields dengan cara mengklik Refactor -> Encapsulate Fields
-> Select All->Refactor



Setelah melakukan encapsulate field maka akan muncul function-function seperti berikut.


```

public String getKode_barang() {
    return kode_barang;
}

public void setKode_barang(String kode_barang) {
    this.kode_barang = kode_barang;
}

public String getNama_barang() {
    return nama_barang;
}

public void setNama_barang(String nama_barang) {
    this.nama_barang = nama_barang;
}

public String getMerek_barang() {
    return merek_barang;
}

public void setMerek_barang(String merek_barang) {
    this.merek_barang = merek_barang;
}

public int getStok_barang() {
    return stok_barang;
}

```

Selanjutnya membuat java interface pada package com.aksesoris.mobil.dao dengan cara klik kanan pada package dao dan pilih add new java interface, kemudian beri nama **AksMobilDao.java**. Pada file interface ini berfungsi untuk memberikan batasan, jadi class harus mempunyai method yang ada pada interface seperti (create, read, update, delete). Harus bisa melakukan proses seperti interface.

```

package com.aksesoris.mobil.dao;

import com.aksesoris.mobil.model.AksMobil;
import java.util.List;

public interface AksMobilDao {
    public void save(AksMobil aksmobil);
    public void delete(AksMobil aksmobil);
    public void update(AksMobil aksmobil);
    public List<AksMobil> getList();
}

```

Langkah berikutnya yaitu membuat java class bernama **AksMobilDaoImpl.java** pada package `com.aksesoris.mobil.dao`. class ini berfungsi sebagai penyedia `SessionFactory` yang akan digunakan sebagai kendaraan untuk berhubungan dengan database.

```
package com.aksesoris.mobil.dao;

import ...5 lines

@Repository
public class AksMobilDaoImpl implements AksMobilDao{
    @Autowired
    private SessionFactory sessionFactory;

    @Override
    public void save(AksMobil aksmobil) {
        sessionFactory.getCurrentSession().save(aksmobil);
    }

    @Override
    public void delete(AksMobil aksmobil) {
        sessionFactory.getCurrentSession().delete(aksmobil);
    }

    @Override
    public void update(AksMobil aksmobil) {
        sessionFactory.getCurrentSession().update(aksmobil);
    }

    @Override
    public List<AksMobil> getList() {
        return sessionFactory.getCurrentSession().createCriteria(AksMobil.class).list();
    }
}
```

Kemudian tambahkan java interface pada package `com.aksesoris.mobil.service` dan beri nama **AksMobilService.java**, dengan codenya seperti berikut

```
package com.aksesoris.mobil.service;

import com.aksesoris.mobil.model.AksMobil;
import java.util.List;

public interface AksMobilService {
    public void save(AksMobil aksmobil);
    public void delete(AksMobil aksmobil);
    public void update(AksMobil aksmobil);
    public List<AksMobil> getList();
}
```

Tambahkan juga java class **AksMobilServiceImpl.java** pada `com.aksesoris.mobil.service` dengan codenya seperti berikut

```

@Override
public void save(AksMobil aksmobil) {
    aksmobilDao.save(aksmobil);
}

@Transactional
@Override
public void delete(AksMobil aksmobil) {
    aksmobilDao.delete(aksmobil);
}

@Transactional
@Override
public void update(AksMobil aksmobil) {
    aksmobilDao.update(aksmobil);
}

@Override
public List<AksMobil> getList() {
    return aksmobilDao.getList();
}

```

Selanjutnya buat JFrame form pada **package com.aksesoris.mobil.view** dengan nama **MainView.java**. Di dalam form terdapat textfield untuk memasukan nilai pada kolom dan terdapat 4 button untuk save, update, delete, dan clear. Kemudian tambahkan sebuah table untuk menampilkan data dari database.

TOKO AKSESORIS MOBIL

KODE BARANG

NAMA BARANG

MEREK BARANG

STOK BARANG

GARANSI BARANG

HARGA BARANG

CLEAR

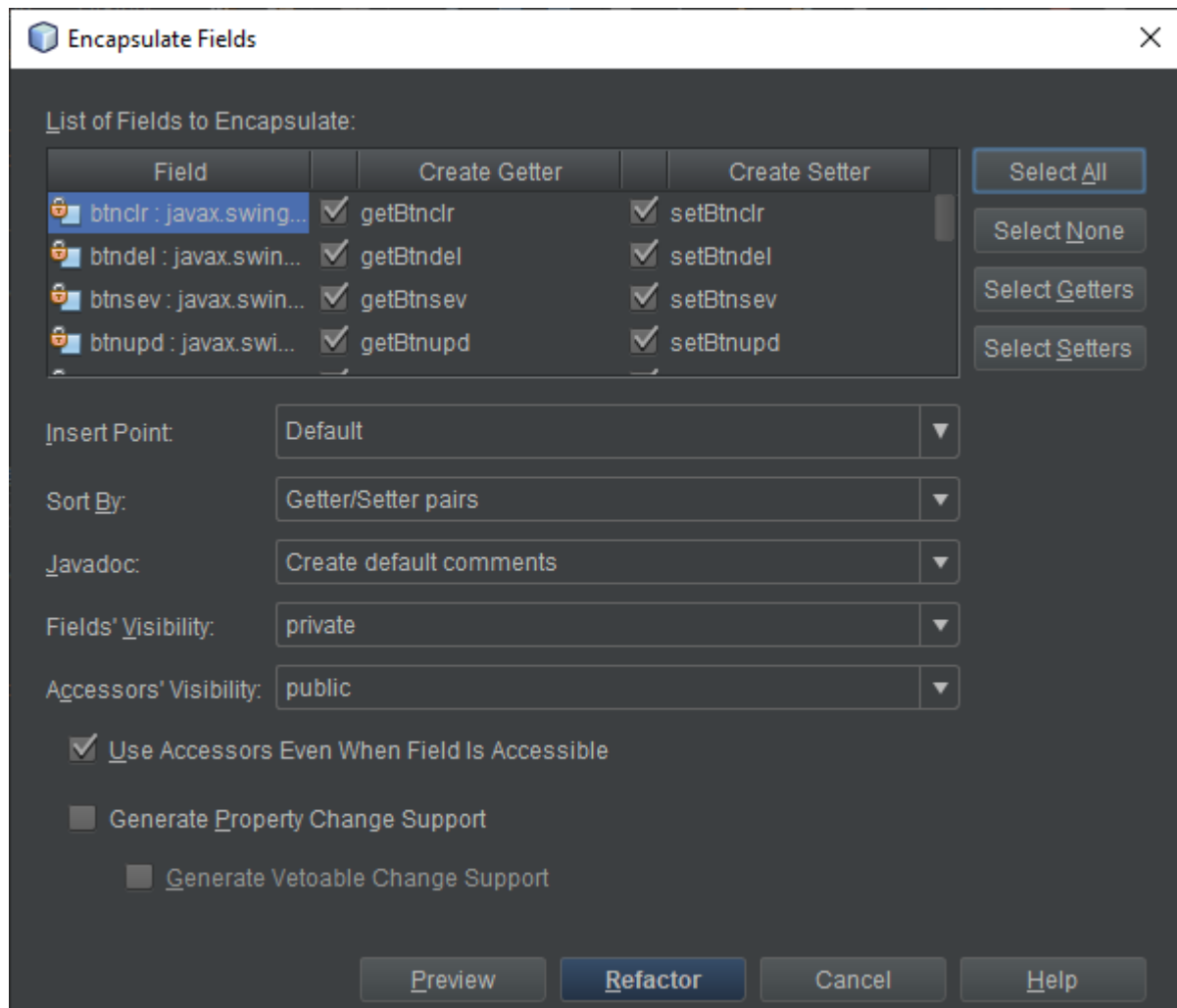
SAVE

DELETE

UPDATE

Title 1	Title 2	Title 3	Title 4

Kemudian lakukan encapsulate field, pilih semua textfield dan table.



Langkah berikutnya yaitu membuat java class baru pada package com.aksesoris.mobil.model dengan nama **AksMobilTableModel.java** class ini berfungsi untuk mengatur table dari program

```

package com.aksesoris.mobil.model;

import java.util.ArrayList;
import java.util.List;
import javax.swing.table.AbstractTableModel;

public class AksMobilTableModel extends AbstractTableModel{

    private List<AksMobil> listAksMobil = new ArrayList();
    private final String HEADER[]={"Kode", "Nama Barang", "Merek Barang", "Stok Barang", "Garansi Barang", "Harga Barang"}

    public AksMobilTableModel(List<AksMobil>listAksMobil){
        this.listAksMobil = listAksMobil;
    }

    @Override
    public int getRowCount() {
        return listAksMobil.size();
    }

    @Override
    public int getColumnCount() {
        return HEADER.length;
    }

    @Override
    public Object getValueAt(int rowIndex, int columnIndex) {
        AksMobil aksmobil = listAksMobil.get(rowIndex);

```

```

@Override
public int getRowCount() {
    return listAksMobil.size();
}

@Override
public int getColumnCount() {
    return HEADER.length;
}

@Override
public Object getValueAt(int rowIndex, int columnIndex) {
    AksMobil aksmobil = listAksMobil.get(rowIndex);
    switch(columnIndex){
        case 0:
            return aksmobil.getKode_barang();
            //break tdk perlu karna sudah di return
        case 1:
            return aksmobil.getNama_barang();
        case 2:
            return aksmobil.getMerek_barang();
        case 3:
            return aksmobil.getStok_barang();
        case 4:
            return aksmobil.getGaransi_barang();
        case 5:
            return aksmobil.getHarga_barang();
        default:

```

```

        return aksmobil.getGaransi_barang();
    case 4:
        return aksmobil.getGaransi_barang();
    case 5:
        return aksmobil.getHarga_barang();
    default:
        return null;
    }
}

@Override
public String getColumnName(int index) {
    return HEADER[index]; //agar tabel bisa nmpilin judul kolom di view
}
}

```

Tambahkan java class **AksMobilApp.java** pada com.aksesoris.mobil.app, nantinya class ini akan berfungsi sebagai Main class (entry point) dari aplikasi aksesoris mobil. ClassPathXmlApplicationContext merupakan class bertanggung jawab untuk meload konfigurasi spring.

```

package com.aksesoris.mobil.app;

import com.aksesoris.mobil.service.AksMobilService;
import com.aksesoris.mobil.view.MainView;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class AksMobilApp {
    private static ApplicationContext appContext;

    public static void main(String[] args) {
        appContext = new ClassPathXmlApplicationContext("classpath:spring-config.xml");
        new MainView().setVisible(true);
    }

    public static AksMobilService getAksMobilService() {
        return (AksMobilService) appContext.getBean("AksMobilService");
    }
}

```

Langkah selanjutnya adalah membuat java class pada package com.aksesoris.mobil.controller dengan nama **AksMobilController.java**. Class ini menyediakan function-function yang akan digunakan untuk menjalankan fungsi button pada form MainView.

```

package com.aksesoris.mobil.controller;

import ...6 lines

public class AksMobilController {
    private final MainView mainView;
    private List<AksMobil> listAksMobil;
    private AksMobilTableModel aksmobiltablemodel;

    public AksMobilController (MainView mainview){
        this.mainView = mainView;
    }
}

```

Function Read yang berfungsi untuk menampilkan data ke dalam table

```

//READ
public void getAllData() {
    listAksMobil = getAksMobilService().getList();
    aksmobiltablemodel = new AksMobilTableModel(listAksMobil);
    this.mainView.getTblbrg().setModel(aksmobiltablemodel);
}

```

Function save yang berfungsi untuk menyimpan data yang diinput pada textfield ke dalam database dan tabel

```

//SAVE
public void saveAksMobil() {
    AksMobil aksmobil = new AksMobil();
    aksmobil.setKode_barang(this.mainView.getTxtKode().getText());
    aksmobil.setNama_barang(this.mainView.getTxtnama().getText());
    aksmobil.setMerek_barang(this.mainView.getTxtMerek().getText());
    aksmobil.setStok_barang(Integer.parseInt(this.mainView.getStk().getText()));
    aksmobil.setGaransi_barang(this.mainView.getGrn().getText());
    aksmobil.setHarga_barang(Integer.parseInt(this.mainView.getTxtHrg().getText()));

    try{
        getAksMobilService().save(aksmobil);
        JOptionPane.showMessageDialog(null, "Berhasil menyimpan Data Mobil", "Success",
            JOptionPane.INFORMATION_MESSAGE);
        clear();
        getAllData();
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null, "Error saat menyimpan data mobil", "Error",
            JOptionPane.ERROR_MESSAGE);
        clear();
    }
}

```

Function update berfungsi untuk mengupdate data pada tabel


```
//UPDATE
public void updateAksMobil() {
    AksMobil aksmobil = new AksMobil();
    aksmobil.setKode_barang(this.mainView.getTxtKode().getText());
    aksmobil.setNama_barang(this.mainView.getTxtnama().getText());
    aksmobil.setMerek_barang(this.mainView.getTxtMerek().getText());
    aksmobil.setStok_barang(Integer.parseInt(this.mainView.getStk().getText()));
    aksmobil.setGaransi_barang(this.mainView.getGrn().getText());
    aksmobil.setHarga_barang(Integer.parseInt(this.mainView.getTxtHrg().getText()));

    try{
        getAksMobilService().update(aksmobil);
        JOptionPane.showMessageDialog(null, "Berhasil menyimpan Data Mobil", "Success",
            JOptionPane.INFORMATION_MESSAGE);
        clear();
        getAllData();
    }catch (Exception ex){
        JOptionPane.showMessageDialog(null, "Error saat menyimpan data mobil", "Error",
            JOptionPane.ERROR_MESSAGE);
        clear();
    }
}
```

Function delete berfungsi untuk menghapus data berdasarkan field kode.

```
//DELETE
public void deleteAksMobil(){
    //untuk ngapus cuma butuh ngeset primary key aja
    if (this.mainView.getTxtKode().getText() == null){
        //cek kode, kalo belom muncul error
        JOptionPane.showMessageDialog(null, "Data yang akan dihapus belum dipilih", "Error",
            JOptionPane.ERROR_MESSAGE);
    } else{
        AksMobil aksmobil = new AksMobil();
        aksmobil.setKode_barang(this.mainView.getTxtKode().getText());
        //buat ngehold data yang dipilih user buat diapus
        int option = JOptionPane.showConfirmDialog(null, "Apakah anda ingin menghapus data?", "Warning",
            JOptionPane.YES_OPTION, JOptionPane.WARNING_MESSAGE);
        if(option == JOptionPane.YES_OPTION){
            try{
                getAksMobilService().delete(aksmobil);
                JOptionPane.showMessageDialog(null, "Berhasil menghapus data mobil", "Success",
                    JOptionPane.INFORMATION_MESSAGE);
                clear();
                getAllData();
            } catch (Exception ex){
                JOptionPane.showMessageDialog(null, "Error saat menghapus data mobil", "Error",
                    JOptionPane.ERROR_MESSAGE);
                clear();
            }
        }
    }
}
```

Function Clear yang berfungsi untuk menghilangkan inputan pada textfield dan function getData yang berfungsi untuk mengambil data yang dipilih dari table dan dimasukan ke dalam textfield.

```
//CLEAR
public void clear() {
    this.mainView.getTxtKode().setText("");
    this.mainView.getTxtnama().setText("");
    this.mainView.getTxtMerek().setText("");
    this.mainView.getStk().setText("");
    this.mainView.getGrn().setText("");
    this.mainView.getTxtHrg().setText("");
}

public void getData() {
    int rowIndex = this.mainView.getTblbrg().getSelectedRow();
    this.mainView.getTxtKode().setText(String.valueOf(this.mainView.getTblbrg().getValueAt(rowIndex, 0)));
    this.mainView.getTxtnama().setText(String.valueOf(this.mainView.getTblbrg().getValueAt(rowIndex, 1)));
    this.mainView.getTxtMerek().setText(String.valueOf(this.mainView.getTblbrg().getValueAt(rowIndex, 2)));
    this.mainView.getStk().setText(String.valueOf(this.mainView.getTblbrg().getValueAt(rowIndex, 3)));
    this.mainView.getGrn().setText(String.valueOf(this.mainView.getTblbrg().getValueAt(rowIndex, 4)));
    this.mainView.getTxtHrg().setText(String.valueOf(this.mainView.getTblbrg().getValueAt(rowIndex, 5)));
}
}
```

Kemudian tambahkan code berikut pada source code MainView

```
package com.aksesoris.mobil.view;

import com.aksesoris.mobil.controller.AksMobilController;

public class MainView extends javax.swing.JFrame {

    private final AksMobilController aksmobilController = new AksMobilController(this);

    public MainView() {
        initComponents();
        aksmobilController.getAllData();
    }

    @SuppressWarnings("unchecked")
    Generated Code
}
```

Setelah itu tambahkan fungsi untuk masing masing tombol dengan mengklik 2 kali pada button dan tambahkan event mouseclicked pada table agar dapat digunakan, setelah itu memanggil fungsi yang dibuat pada Controller untuk masing masing button.

```

private void btnsevActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    aksmobilController.saveAksMobil();
}

private void btndelActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    aksmobilController.deleteAksMobil();
}

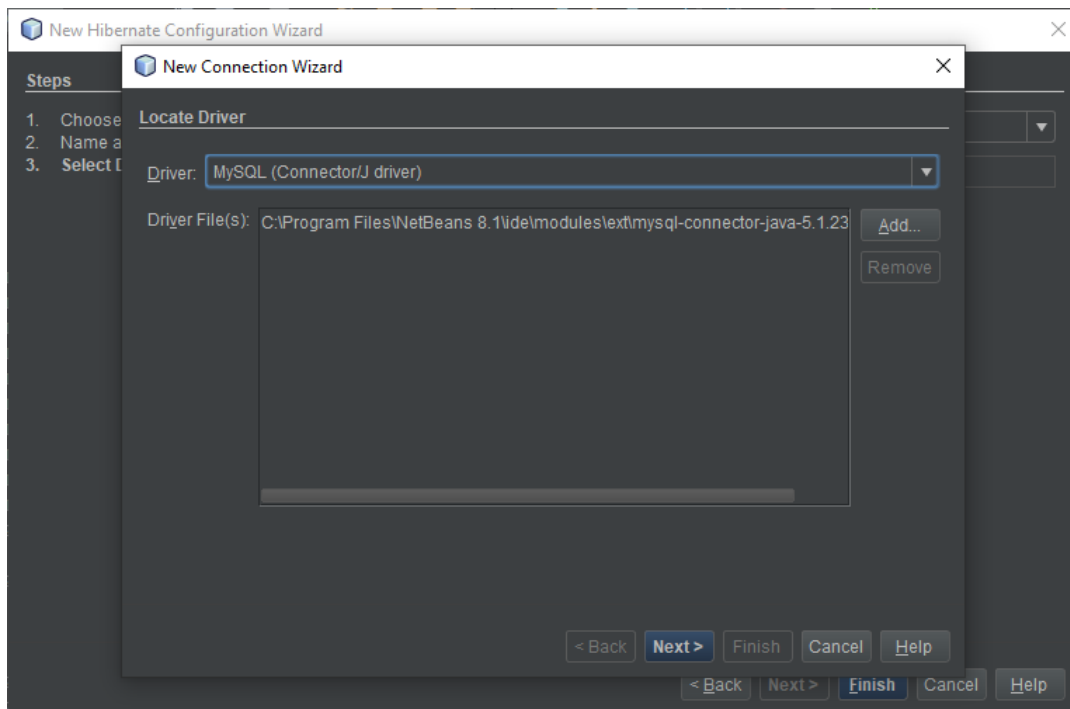
private void btnupdActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    aksmobilController.updateAksMobil();
}

private void btncclrActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    aksmobilController.clear();
}

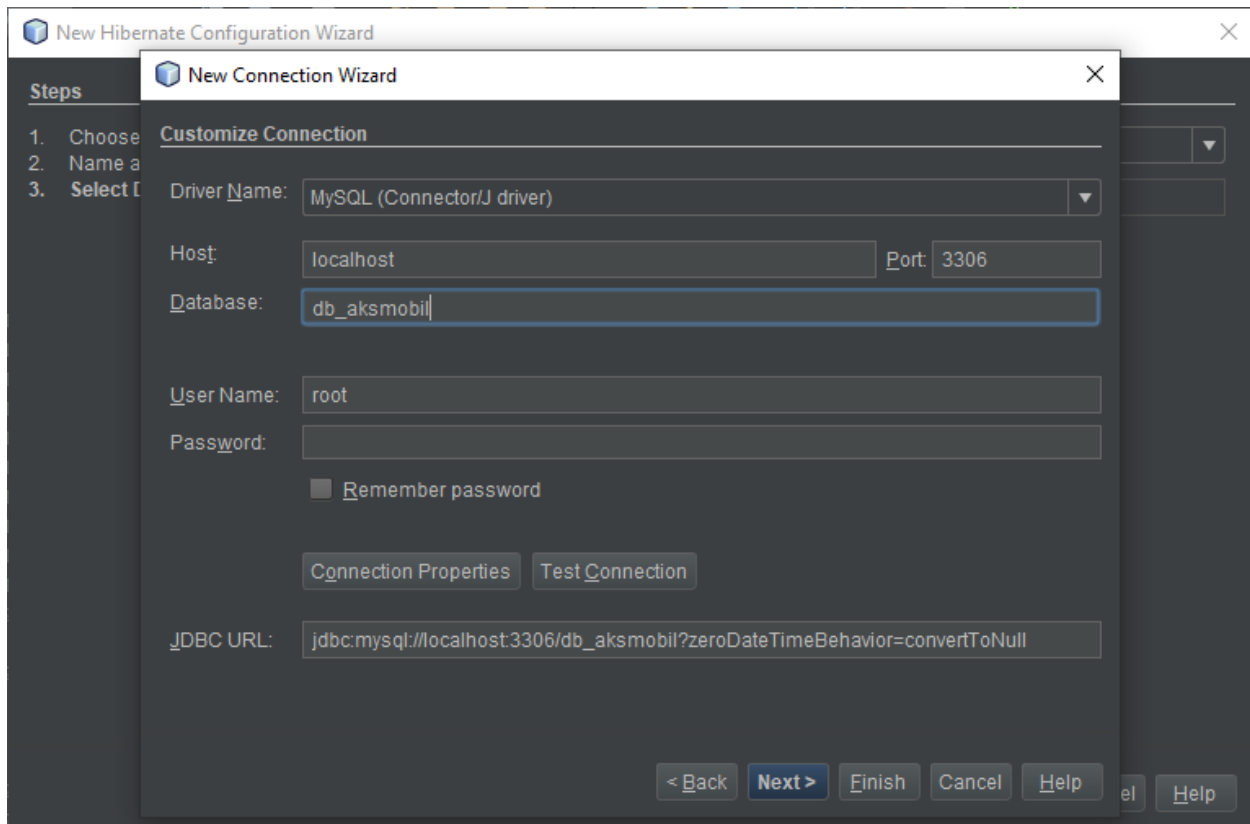
private void tblbrgMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    aksmobilController.getData();
}

```

Selanjutnya membuat **Hibernate Configuration Wizard** pada default package. Hibernate => Hibernate Configuration Wizard, lalu next => berikan nama filenya Hibernate.cfg. kemudian pilih Database Connection , New Database Connection => Pilih Driver MySql, lalu next.



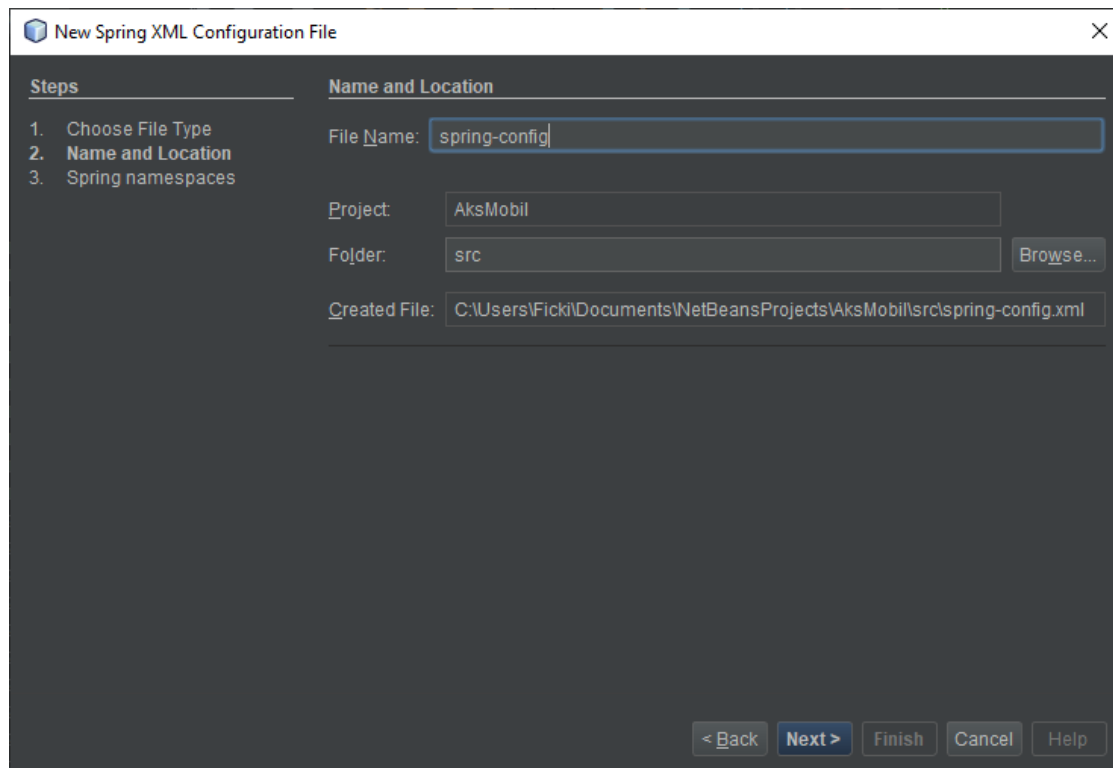
beri nama database dengan db_aksmobil seperti yang telah dibuat sebelumnya di localhost/phpmyadmin. Setelah itu klik finish.



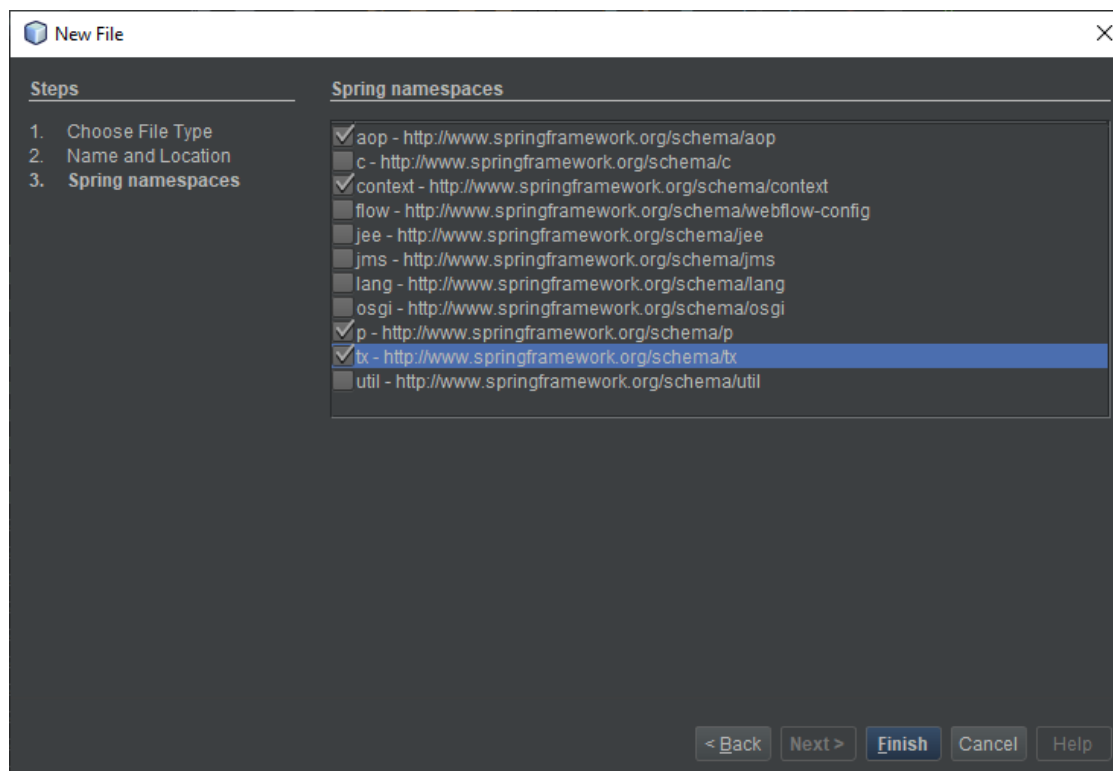
Tambahkan code di bawah ini pada **hibernate.cfg.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN" "http://
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
    <!--<property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="hibernate.connection.url">
      jdbc:mysql://localhost:3306/db_aksmobil?zeroDateTimeBehavior=convertToNull</property>
    <property name="hibernate.connection.username">root</property>-->
    <property name="hibernate.show_sql">true</property>
    <property name="hibernate.format_sql">true</property>
    <property name="hibernate.hbm2ddl.auto">update</property>
    <mapping class="com.aksesoris.mobil.model.AksMobil" />
  </session-factory>
</hibernate-configuration>
```

Spring-config.xml untuk konfigurasi spring.



ceklis pada bagian aop, context, p, dan tx



Tambahkan spring config

```

<context:annotation-config/>
<tx:annotation-driven transaction-manager="transactionManager"/>
<context:component-scan base-package="com.aksesoris.mobil"/>
<context:property-placeholder location="classpath:jdbc.properties"/>

<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
<property name="driverClassName" value="${jdbc.driver}"/>
<property name="url" value="${jdbc.url}"/>
<property name="username" value="${jdbc.username}"/>
</bean>
<bean id="sessionFactory" class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">
<property name="dataSource" ref="dataSource"/>
<property name="configLocation" value="classpath:hibernate.cfg.xml"/>
</bean>
<bean id="transactionManager" class="org.springframework.orm.hibernate4.HibernateTransactionManager">
<property name="sessionFactory" ref="sessionFactory"/>
</bean>

```

Jdbc.properties

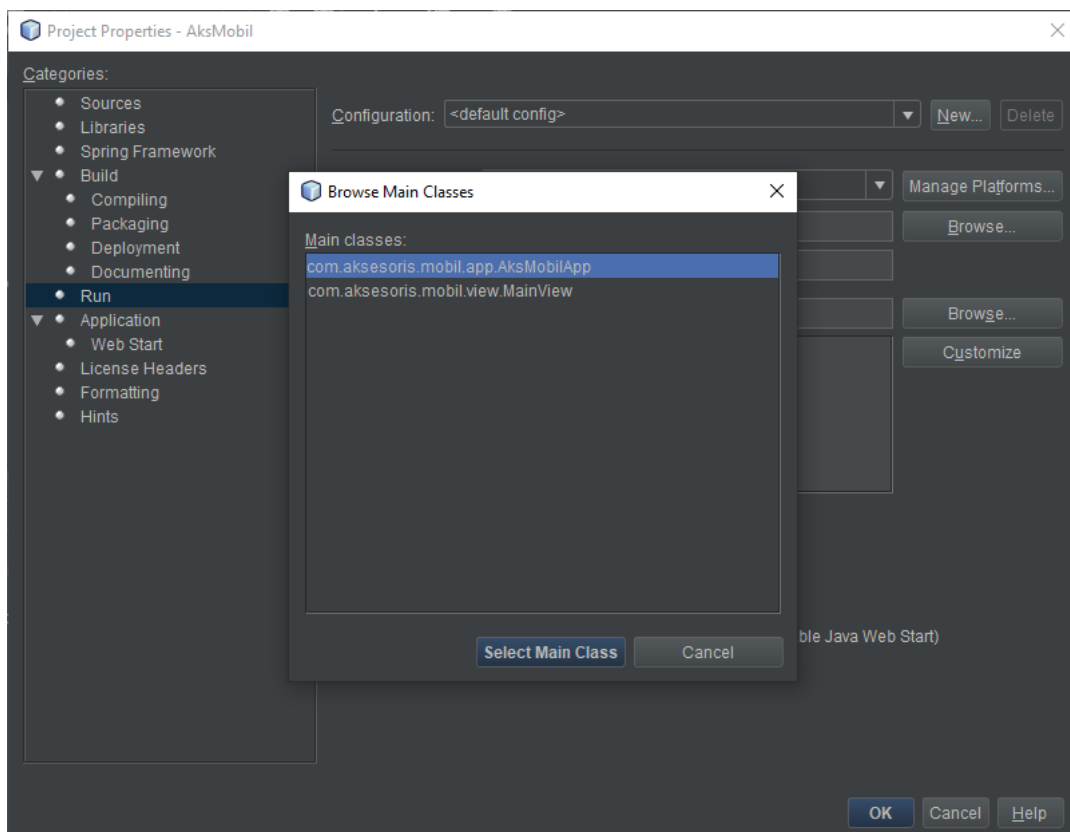
Jdbc.properties yang menjembatani *Database Relasional* dengan Java untuk mengakses data di dalam database relasional dengan menggunakan Spesifikasi Bahasa SQL.

```

jdbc.driver = com.mysql.jdbc.Driver
jdbc.url = jdbc:mysql://localhost:3306/db_aksmobil?zeroDateTimeBehavior=convertToNull
jdbc.username = root

```

Terakhir jadikan AksMobilApp sebagai main class dengan klik kanan pada package utama yaitu AksMobil lalu pilih properties



OUTPUT

Tampilan awal

The screenshot shows the initial state of the 'TOKO AKSESORIS MOBIL' application. The window has a title bar with standard Windows controls. The main area is divided into two sections. On the left, there are six input fields for product details: 'KODE BARANG', 'NAMA BARANG', 'MEREK BARANG', 'STOK BARANG', 'GARANSI BARANG', and 'HARGA BARANG'. Each field is currently empty. Below these fields are three buttons: 'CLEAR', 'SAVE', 'DELETE', and 'UPDATE'. On the right, there is a table with six columns: 'Kode', 'Nama Bara...', 'Merek Bara...', 'Stok Barang', 'Garansi Ba...', and 'Harga Bara...'. The table is currently empty.

Coba menambahkan data

The screenshot shows the application after data has been entered. The input fields on the left now contain the following values: 'KODE BARANG' is 'AO 1', 'NAMA BARANG' is 'KNALPOT RACING BURNTIP', 'MEREK BARANG' is 'MUGEN', 'STOK BARANG' is '10', 'GARANSI BARANG' is '2023', and 'HARGA BARANG' is '2000000'. The 'SAVE' button is highlighted. A modal dialog box titled 'Success' is displayed in the center of the window. It contains an information icon, the text 'Berhasil menyimpan Data Mobil', and an 'OK' button. The table on the right remains empty.

Membaca data

Kode	Nama Bara...	Merek Bara...	Stok Barang	Garansi Ba...	Harga Bara...
A01	KNALPOT ...	MUGEN	10	2023	2000000
A02	KNALPOT ...	KHS	10	2024	2300000

Update data

Success

Berhasil menyimpan Data Mobil

OK

Hapus Data

TOKO AKSESORIS MOBIL

KODE BARANG

NAMA BARANG

MEREK BARANG

STOK BARANG

GARANSI BARANG

HARGA BARANG

Kode	Nama Bara...	Merek Bara...	Stok Barang	Garansi Ba...	Harga Bara...
A01	KNALPOT ...	MUGEN	10	2023	2000000
A02	KNALPOT ...	KHS	8	2024	2300000

Warning

⚠️ Apakah anda ingin menghapus data?

TOKO AKSESORIS MOBIL

KODE BARANG

NAMA BARANG

MEREK BARANG

STOK BARANG

GARANSI BARANG

HARGA BARANG

Kode	Nama Bara...	Merek Bara...	Stok Barang	Garansi Ba...	Harga Bara...
A01	KNALPOT ...	MUGEN	10	2023	2000000
A02	KNALPOT ...	KHS	8	2024	2300000

Success

ℹ️ Berhasil menghapus data mobil

Data yang tersimpan pada database

+ Options						
	kode_barang	nama_barang	merek_barang	stok_barang	garansi_barang	harga_barang
<input type="checkbox"/>	A01	Knalpot Racing Burntip	Mugen	10	2023	2000000
↑ <input type="checkbox"/> Check all With selected: <input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/> <input type="button" value="Export"/>						

BAB IV

PENUTUP

4.1 Kesimpulan

Dengan adanya praktikum Rekayasa Perangkat Lunak 2, penulis dapat membangun aplikasi pengolahan data yang memiliki fungsi menyimpan, mengubah dan menghapus data. Dimana aplikasi ini berbasis desktop dengan menggunakan Java dan juga framework Spring.

4.2 Saran

Untuk pengembangan aplikasi, dapat dilakukan dengan menambahkan fitur lain seperti laporan bulanan ataupun membuat tampilan aplikasi menjadi lebih baik lagi supaya terlihat menarik.