

Dokumentasi Tugas Akhir Praktek

Kukuh Setya Arumansyah

A12.2020.06495

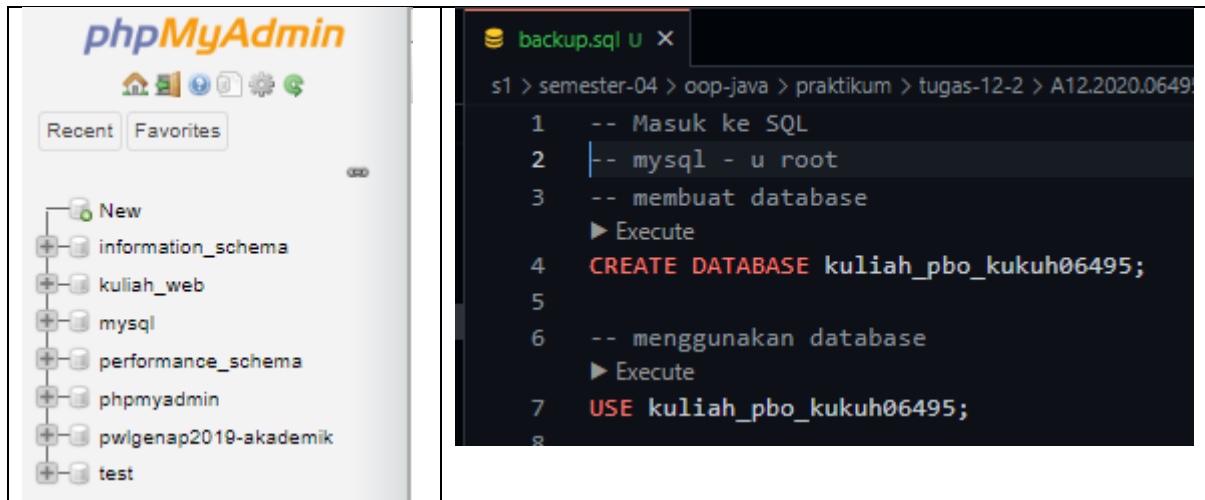
Daftar Isi

Dokumentasi Tugas Akhir Praktek	1
Daftar Isi.....	1
Struktur Folder	2
Impor backup.sql.....	3
Menjalankan File application.jar	4
Pengaturan Aplikasi File application.properties.....	5
Login Aplikasi	6
Perbedaan Project Saya dengan Contoh yang Diberikan	7

Struktur Folder

```
A12202006495
├── backup-sql
│   └── backup.sql // import "backup.sql" pada phpMyAdmin atau database client lainnya
├── doc // tempat menyimpan dokumentasi
│   ├── documentation-project.docx
│   └── documentation-project.pdf
├── driver-jdbc-mysql-terbaru
│   └── mysql-connector-java-8.0.29.jar // driver-jdbc-mysql yang digunakan.
├── jar // git bash here pada folder ini lalu jalankan perintah "java -jar application.jar"
├── application.jar // file java archive yang dapat dieksekusi langsung
├── khusus.properties // file konfigurasi application contoh lainnya jangan dihapus
├── application.properties // file konfigurasi application jangan dihapus
├── project // tempat menyimpan source code application
│   └── src
│       ├── entity // model atau representasi tabel database
│       │   ├── Barang.java
│       │   ├── Pegawai.java
│       │   └── Transaction.java
│       ├── repository // tempat menyimpan kode yang melakukan query ke database
│       │   ├── BarangRepository.java
│       │   ├── BarangRepositoryImpl.java
│       │   ├── PegawaiRepository.java
│       │   ├── PegawaiRepositoryImpl.java
│       │   ├── TransactionRepository.java
│       │   └── TransactionRepositoryImpl.java
│       ├── service // controller atau logic program
│       │   ├── BarangServiceImpl.java
│       │   ├── CetakBarangServiceImpl.java
│       │   ├── LoginServiceImpl.java
│       │   ├── PegawaiServiceImpl.java
│       │   ├── TableBarangService.java
│       │   ├── TablePegawaiService.java
│       │   ├── TableTransactionService.java
│       │   └── TransactionServiceImpl.java
│       ├── util // alat
│       │   ├── ButtonUtil.java
│       │   ├── DatabaseUtil.java // KoneksiDBMS.java
│       │   ├── ExitUtil.java
│       │   ├── FieldUtil.java
│       │   ├── LabelUtil.java
│       │   └── TableUtil.java
│       ├── view // tampilan
│       │   ├── FrameBarang.java // Fbarang.java
│       │   ├── FrameCetakBarangMin.java // cetaklaporan.java
│       │   ├── FrameJual.java // FJual.java
│       │   ├── FrameLogin.java // Login.java
│       │   ├── FrameMenu.java // FMenu.java
│       │   ├── FramePegawai.java // FPegawai.java
│       │   └── PanelCetakBarang.java
│       ├── App.java // main program
│       ├── AppConfig.java // digunakan untuk load application.properties
│       ├── application.properties // file konfigurasi application untuk source code jangan dihapus
│       └── readme.txt // catatan sederhana
```

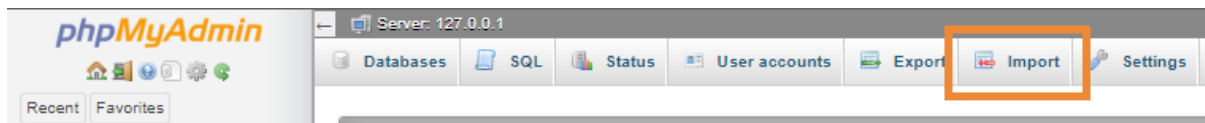
Impor backup.sql



Pastikan anda **tidak memiliki database yang namanya sama** dengan yang ada di backup.sql karena pada backup.sql ada perintah:

“CREATE DATABASE kuliah_pbo_kukuh06495”

Dimana perintah tersebut akan membuat database dengan nama **kuliah_pbo_kukuh06495**.



Setelah itu klik **“import”** pada nav-bar.

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.

A compressed file's name must end in `.[format].[compression]`. Example: `.sql.zip`

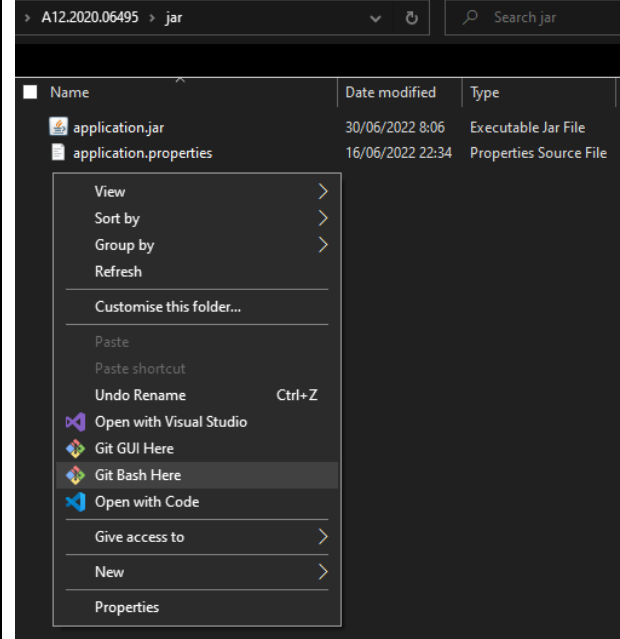
Browse your computer: backup.sql (Max: 40MiB)

You may also drag and drop a file on any page.

Character set of the file:

Lalu **“Choose File backup.sql”** yang sudah saya sediakan dalam folder **backup-sql**. Setelah itu **“Go”**

Menjalankan File application.jar



Klik kanan pada mouse lalu git bash here jika anda menggunakan git bash sebagai terminal.

Jika anda menggunakan cmd silahkan sampai terminal berada pada folder ".../A12.2020.06495/jar".

Jika sudah coba ketik perintah "ls" untuk melihat daftar file dalam folder tersebut.

```
MINGW64/c/xampp/htdocs/_my/kuliah/s1/semester-04/oop-java/praktikum/tugas-12-2/A12.2020.06495/jar
kaesa@Kaesa_Lyrih MINGW64 /c/xampp/htdocs/_my/kuliah/s1/semester-04/oop-java/praktikum/tugas-12-2/A12.2020.06495/jar (main)
$ ls
application.jar  application.properties
```

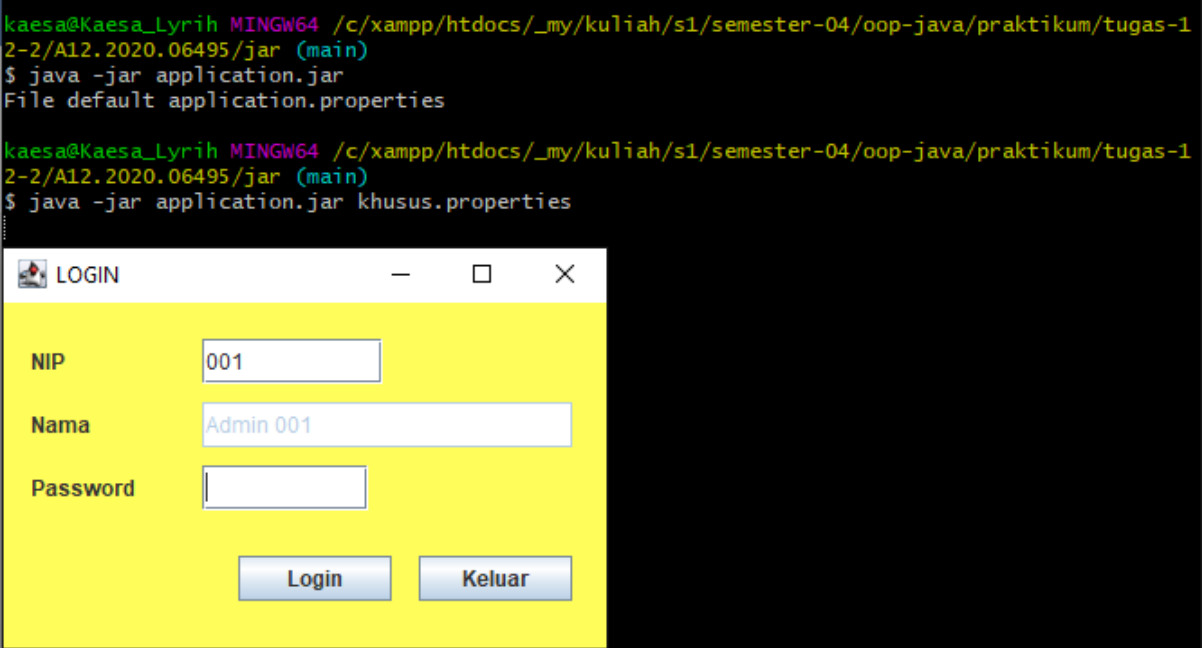
Pastikan terdapat:
application.jar,application.properties.

Jalankan perintah

- "java -jar application.jar" atau
 - "java -jar **applicaton.jar** **application.properties**"
 - **applicaton.jar** adalah program utamanya.
 - **application** opsi khusus jika ingin mengubah **application.properties** ke nama yang lain misal: **khusus.properties**
- Maka perintah yang dijalankan "java -jar **applicaton.jar** **khusus.properties**"

```
kaesa@Kaesa_Lyrih MINGW64 /c/xampp/htdocs/_my/kuliah/s1/semester-04/oop-java/praktikum/tugas-12-2/A12.2020.06495/jar (main)
$ java -jar application.jar
File default application.properties

kaesa@Kaesa_Lyrih MINGW64 /c/xampp/htdocs/_my/kuliah/s1/semester-04/oop-java/praktikum/tugas-12-2/A12.2020.06495/jar (main)
$ java -jar application.jar khusus.properties
```



Pengaturan Aplikasi File application.properties

```
# Pengaturan koneksi ke database
database.host=localhost
database.port=3306
database.database=kuliah_pbo_kukuh06495
database.username=root
database.password=

# Pengaturan pemilik program
author.nim=A12.2020.06495
author.fullname=Kukuh Setya Arumansyah
```

Silahkan jika diubah jika pengaturan database Anda berubah, Misalnya:

- Database anda berada di host: **127.0.0.1**
- Port masih default yaitu: **3306**
- Dengan nama database: **program_pbo**
- Tapi usernamenya berbeda misal: **kukuh**
- Passwordnya juga ada misal: **rahasia**

Maka hasilnya:

```
# Pengaturan koneksi ke database
database.host=127.0.0.1
database.port=3306
database.database=program_pbo
database.username=kukuh
database.password=rahasia
```

Atau mungkin anda ingin mengubah authornya atau pemilik programnya hehehe 😊.
Seperti berikut:

```
# Pengaturan pemilik program
author.nim=A12.2020.99999
author.fullname=Kaesa Lyrih (Khusus)
```

Login Aplikasi

Untuk login Anda harus **lihat dulu pada backup.sql** saat memasukan data pegawai atau **insert ke tabel pegawai** pada bagian **nip dan passwordnya**.

Lihat kode sql berikut:

```
INSERT INTO
    pegawai (nip, name, department,gender, `password`)
VALUES
    ( "001", "Admin 001", "Administator", "Laki-Laki", "001" ),
    ( "A12.2020.06495", "Kukuh Setya Arumansyah", "Administator Super", "Laki-Laki", "rahasia" );
```

Dapat dilihat bahwa ada 2 data yang saya masukan ke tabel pegawai yaitu

- **Admin 001** dengan
 - NIP : 001
 - Password : 001
- **Kukuh Setya Arumansyah** dengan
 - NIP : A12.2020.06495
 - Password : rahasia

Maka gunakanlah **kedua nip & password** tersebut untuk **login ke aplikasi**.

Anda juga dapat lihat **nip & password** untuk login pada **phpMyAdmin**.

The screenshot displays two side-by-side windows from a web browser. The left window is the phpMyAdmin interface showing the 'pegawai' table. It includes a status bar at the top indicating 'Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)'. Below this is a SQL query editor with the text 'SELECT * FROM `pegawai`'. A table of data is visible below the editor, with columns 'nip', 'name', 'department', 'gender', and 'password'. The table contains two rows: one for 'Admin 001' with NIP '001' and password '001', and another for 'Kukuh Setya Arumansyah' with NIP 'A12.2020.06495' and password 'rahasia'. The right window is a yellow 'LOGIN' form. It has three input fields: 'NIP' with the value '001', 'Nama' with the value 'Admin 001', and 'Password' with masked characters '...'. At the bottom of the form are two buttons: 'Login' and 'Keluar'.

nip	name	department	gender	password
001	Admin 001	Administator	Laki-Laki	001
A12.2020.06495	Kukuh Setya Arumansyah	Administator Super	Laki-Laki	rahasia

Perbedaan Project Saya dengan Contoh yang Diberikan

Project Saya	Project Contoh Default
<p>Saya menggunakan konsep MVC (Model View Controller) pada Source Code.</p> <p>Dimana kelebihanannya mudah di pelihara (maintenance) kedepannya karena saya memisahkan antara tampilan, query ke database dan logic aplikasi.</p>	<p>Tidak menggunakan konsep MVC, susah dimaintenance.</p> <p>Nama variabel banyak singkatan dan memperlambat dalam proses debugging. Karena harus mencari tahu arti singkatan tersebut.</p>
<p>Saya juga menggunakan konsep Repository Pattern untuk melakukan koneksi dan query ke database.</p> <p>Dimana kelebihanannya, saya tidak perlu tahu perintah query ke database saat membuat logic aplikasi pada controller atau service.</p> <p>Kelebihan lainnya mudah dalam pemeliharaan (maintenance) pada saat ada kesalahan query ke database, saya mudah dalam mencarinya karena source code untuk query ke database ada pada satu pacakge yaitu repository.</p>	<p>Tidak menggunakan konsep Repository Pettern, jika ada kesalahan saat query ke database. Susah mencari kode mana yang salah saat query ke database, karena disemuanya logic aplikasi terdapat kode query ke database.</p> <p>Dan mungkin saat ada pergantian nama tabel, dan kolom pada database harus mencari semua kode yang berkaitan dengan query ke database tersebut dan diubah secara manual.</p> <p>Karena antara controller/service/logic dengan database tidak dipisah.</p>
<p>Menggunakan ".properties" untuk menyimpan konfigurasi aplikasi.</p> <p>Kelebihannya karena konfigurasi aplikasi terpisah dari source code jadi mudah dalam duplikasi aplikasi.</p> <p>Misalnya: aplikasinya itu memiliki database yang berbeda, dapat diubah konfigurasi kedatabasenya pada file ".properties".</p>	<p>Konfigurasi koneksi ke database dihardcode pada source code aplikasi.</p> <p>Membuat sulit untuk diubah-ubah saat ingin aplikasinya berbeda databasenya.</p> <p>Karena dipekerjaan kemungkinan aplikasinya akan disatukan menjadi jar dan hosting pada VPS, kemungkinan host, port, dll yang berkaitan dengan database bisa jadi berbeda-beda setiap ingin deploy aplikasi ke VPS.</p>
<p>Terdapat file ".jar" untuk menjalankan aplikasi.</p> <p>Anda dapat menjalankan aplikasi dengan menggunakan file ".jar" yang sudah saya berikan.</p> <p>Dengan saat mendeploy aplikasi tidak perlu mengirim source code melainkan jarnya, karena dipekerjaan tidak mungkin kita mendeploy source code</p>	<p>Karena tidak jar jadi kemungkinan harus compile dulu supaya bisa di jalankan.</p> <p>Dan ini tidak cocok digunakan saat ingin mendeploy aplikasi ke VPS.</p>

<p>aplikasi dan melakukan compile & run secara bersamaan di VPS.</p> <p>Kemungkinan sudah di compile dulu ke class lalu disatukan dalam bentuk jar.</p> <p>Jadi saat mendeploy aplikasi ke VPS tinggal jalankan saja file jarnya dan konfigurasi file ".properties" jika ada beberapa konfigurasi yang berbeda tiap VPS.</p>	
--	--