# Dokumentasi Tugas Akhir Praktek

Kukuh Setya Arumansyah

A12.2020.06495

# Daftar Isi

[Dokumentasi Tugas Akhir Praktek 1](#_Toc107479061)

[Daftar Isi 1](#_Toc107479062)

[Struktur Folder 2](#_Toc107479063)

[Impor backup.sql 3](#_Toc107479064)

[Menjalankan File application.jar 4](#_Toc107479065)

[Pengaturan Aplikasi File application.properties 5](#_Toc107479066)

[Login Aplikasi 6](#_Toc107479067)

[Perbedaan Project Saya dengan Contoh yang Diberikan 7](#_Toc107479068)

# 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 arcive 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 soruce code jangan dihapus

┗ 📜readme.txt // catatan sederhana

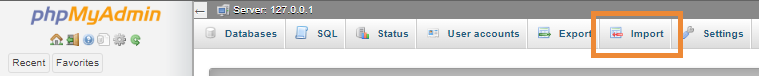
# Impor backup.sql

|  |  |
| --- | --- |
|  |  |

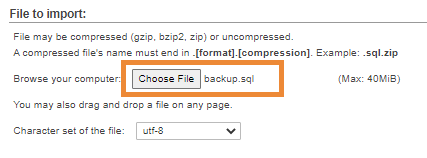
Pastikan anda **tidak memiliki database yang namanya sama** dengan yang ada di backup.sql karena pada backup.sql ada perintah:

**“CREATE DATABASE kuliah\_pbo\_kukuh06495”**

Dimana perintah tersebut akan membuat database dengan nama **kuliah\_pbo\_kukuh06495**.



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



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

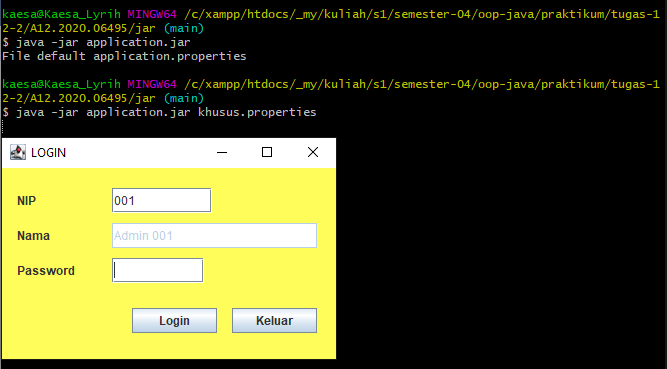
# Menjalankan File application.jar

|  |  |
| --- | --- |
|  | Klik kanan pada mouse lalu git bash here jika anda menggunakan git bash sebagai terminal.  Jika anda menggunakan cmd silahkan sampai teriminal berada pada folder “…/A12.2020.06495/jar”.  Jika sudah coba ketik perintah “ls” untuk melihat daftar file dalam folder tersebut.    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”



# 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**.

|  |  |
| --- | --- |
|  |  |

# Perbedaan Project Saya dengan Contoh yang Diberikan

|  |  |
| --- | --- |
| Project Saya | Project Contoh Default |
| Saya menggunakan konsep MVC (Model View Controller) pada Source Code.  Dimana kelebihannya mudah di pelihara (maintenance) kedepannya karena saya memisahkan antara tampilan, query ke database dan logic aplikasi. | Tidak menggunakan konsep MVC, susah dimaintenance.  Nama variabel banyak singkatan dan memperlambat dalam proses debuging. Karena harus mencari tahu arti singkatan tersebut. |
| Saya juga menggunakan konsep Repository Pattern untuk melakukan koneksi dan query ke database.  Dimana kelebihannya, saya tidak perlu tahu perintah query ke database saat membuat logic aplikasi pada controller atau service.  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. | 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.  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.  Karena antara controller/service/logic dengan database tidak dipisah. |
| Menggunakan “.properties” untuk menyimpan konfigurasi aplikasi.  Kelebihananya karena konfigurasi aplikasi terpisah dari source code jadi mudah dalam duplikasi aplikasi.  Misalnya: aplikasinya itu memiliki database yang berbeda, dapat diubah konfigurasi kedatabasenya pada file “.properties”. | Konfigurasi koneksi ke database dihardcode pada source code aplikasi.  Membuat sulit untuk diubah-ubah saat ingin aplikasinya berbeda databasenya.  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. |
| Terdapat file “.jar” untuk menjalankan aplikasi.  Anda dapat menjalankan aplikasi dengan menggunakan file “.jar” yang sudah saya berikan.  Dengan saat mendeploy aplikasi tidak perlu mengirim source code melainkan jarnya, karena dipekerjaan tidak mungkin kita mendeploy source code aplikasi dan melakukan compille & run secara bersamaan di VPS.  Kemungkinan sudah di compile dulu ke class lalu disatukan dalam bentuk jar.  Jadi saat mendeploy aplikasi ke VPS tinggal jalankan saja file jarnya dan konfigurasi file “.properties” jika ada beberapa konfigurasi yang berbeda tiap VPS. | Karena tidak jar jadi kemungkinan harus compile dulu supaya bisa di jalankan.  Dan ini tidak cocok digunakan saat ingin mendeploy aplikasi ke VPS. |