

## Tutorial 09 – Deployment (Heroku)

Tutorial ini mengasumsikan penggunaan pada *Windows*, tanpa melarang penggunaan OS lain, silakan menyesuaikan.

### 1. Pendahuluan

- Unduh Maven Project (Tutorial-06) yang ada di Scele lalu ekstrak (tip: pada *workspace*).
  - Agar berbeda dengan *project* sebelumnya ubah nama folder menjadi tutorial-09.
  - Jalankan STS, lalu *import project* tersebut.
  - Ubah nama package dari \*.tu08 menjadi \*.tu09, klik kanan package >> Refactor >> Rename
  - Buat DB dengan nama tu09, lalu import SQL yang ada di Scele (Tutorial-06 schema dan data).
  - Install Git (*skip* jika sudah).
  - Lakukan pendaftaran akun Heroku pada link berikut: <https://www.heroku.com/>
  - Install Heroku CLI. Unduh pada link berikut: <https://devcenter.heroku.com/articles/heroku-cli#download-and-install>
  - Install DBConvert for MySQL & PostgreSQL. Unduh pada link berikut: <https://dbconvert.com/mysql/postgresql/>
  - Install PostgreSQL lalu tambahkan directory bin pada path (*skip* jika sudah memiliki psql). Unduh pada link berikut: <https://www.postgresql.org/download/>
- \*anda dapat melanjutkan tutorial sembari menunggu proses download

### 2. Mengubah DB MySQL ke PostgreSQL

- Pada Heroku, penggunaan MySQL harus terlebih dahulu melakukan *verify account* dengan melakukan *input credit card*, sehingga pada *environment* Heroku anda akan menggunakan DB PostgreSQL.
- Lakukan perubahan pada pom.xml dengan menghapus bagian:

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <scope>runtime</scope>
</dependency>
```

ubah menjadi:

```
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
</dependency>
```
- Hapus isi pada application.properties dan ubah menjadi sbb.:

```
spring.datasource.platform=postgresql

spring.datasource.url=jdbc:postgresql://localhost:5432/tu09?useSSL=false&serverTimezone=UTC

spring.datasource.username=[USERNAME]
spring.datasource.password=[PASSWORD]

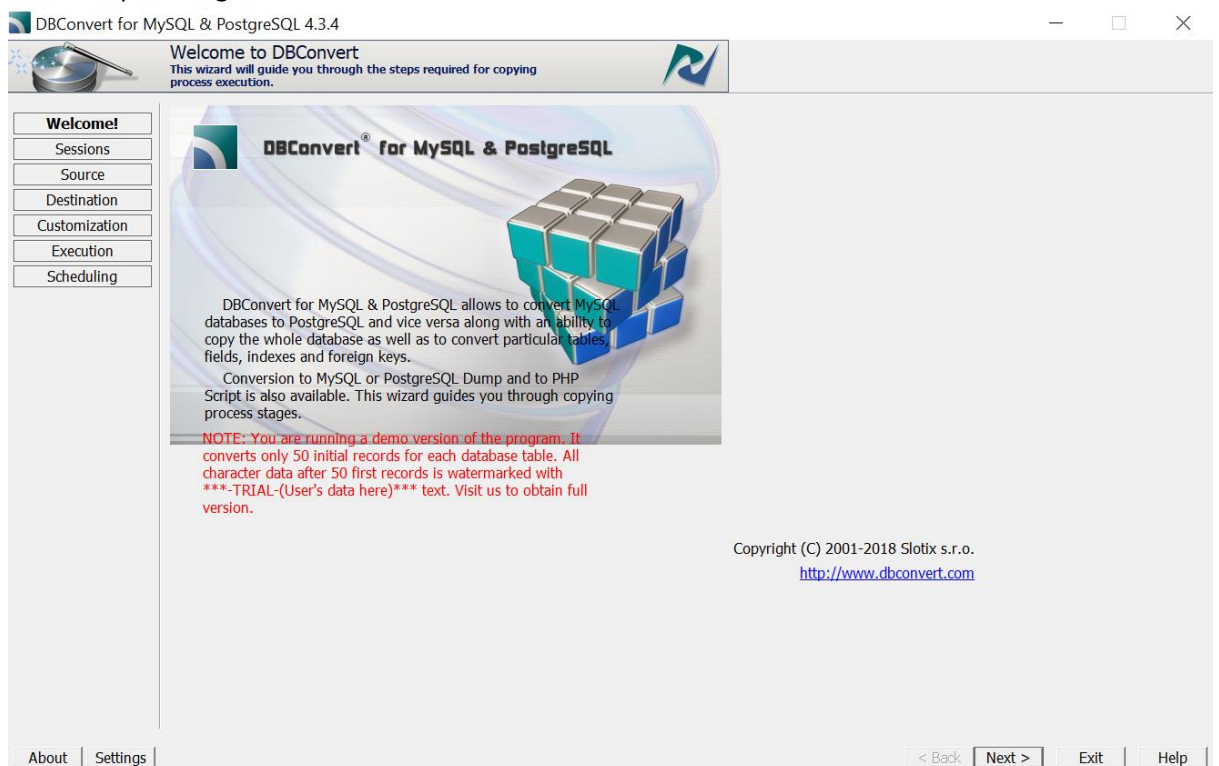
spring.jpa.database-platform = org.hibernate.dialect.PostgreSQLDialect
spring.jpa.properties.hibernate.jdbc.lob.non_contextual_creation = true

spring.jpa.hibernate.ddl-auto=update
```

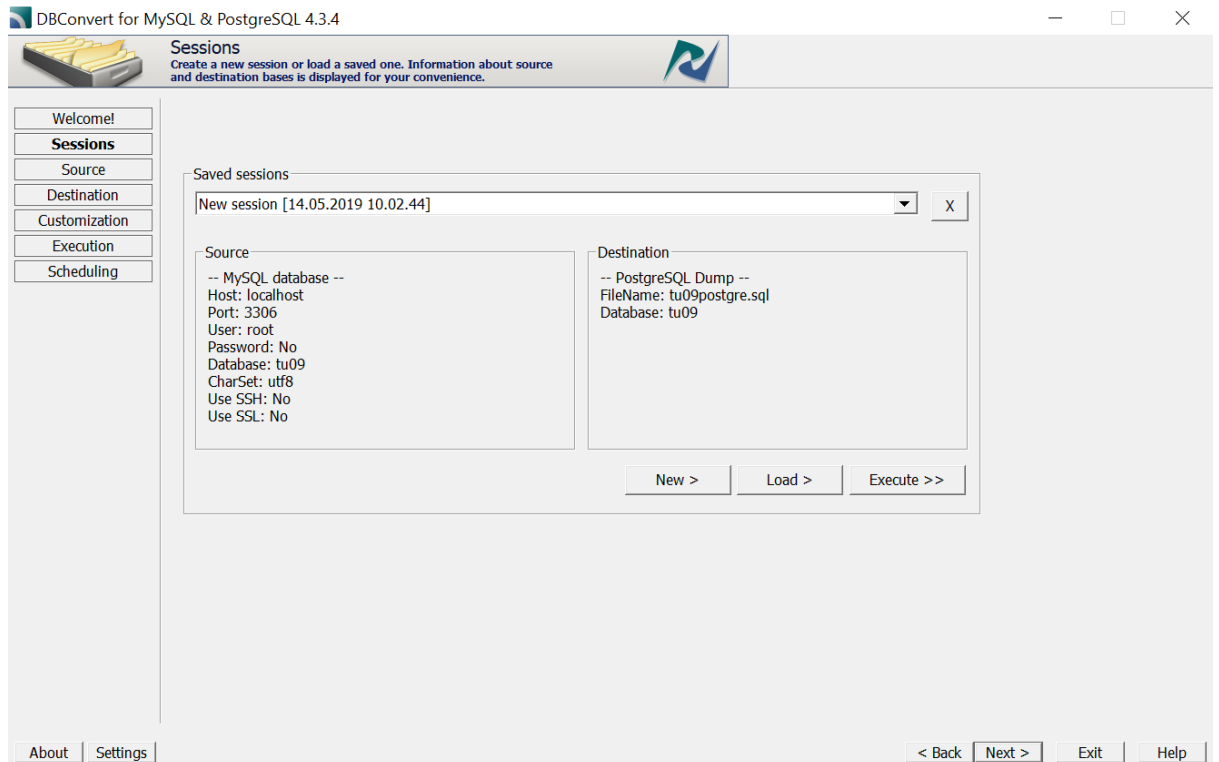
- Jalankan DBConvert for MySQL & PostgreSQL untuk meng-convert DB dari MySQL ke PostgreSQL lalu Klik Try.



- Klik Next pada bagian Welcome.

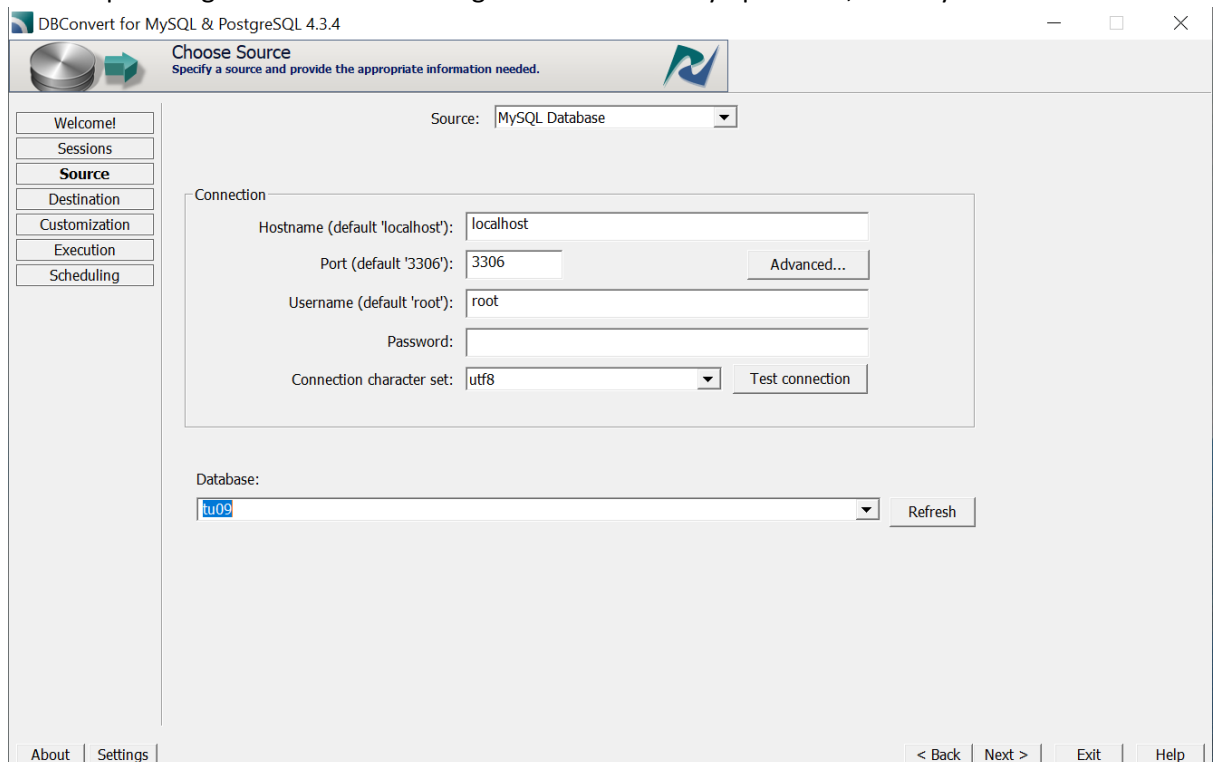


- Klik *Next* pada bagian Sessions.



The screenshot shows the 'Sessions' window of DBConvert for MySQL & PostgreSQL 4.3.4. The window has a sidebar on the left with buttons: Welcome!, Sessions (highlighted), Source, Destination, Customization, Execution, and Scheduling. The main area is titled 'Sessions' and contains a 'Saved sessions' section. It features a dropdown menu showing 'New session [14.05.2019 10.02.44]' and a close button 'X'. Below this are two text areas: 'Source' and 'Destination'. The 'Source' area contains the following text: '-- MySQL database --', 'Host: localhost', 'Port: 3306', 'User: root', 'Password: No', 'Database: tu09', 'CharSet: utf8', 'Use SSH: No', and 'Use SSL: No'. The 'Destination' area contains: '-- PostgreSQL Dump --', 'FileName: tu09postgre.sql', and 'Database: tu09'. At the bottom of the main area are three buttons: 'New >', 'Load >', and 'Execute >>'. The bottom of the window has a status bar with 'About' and 'Settings' buttons on the left, and '< Back', 'Next >', 'Exit', and 'Help' buttons on the right.

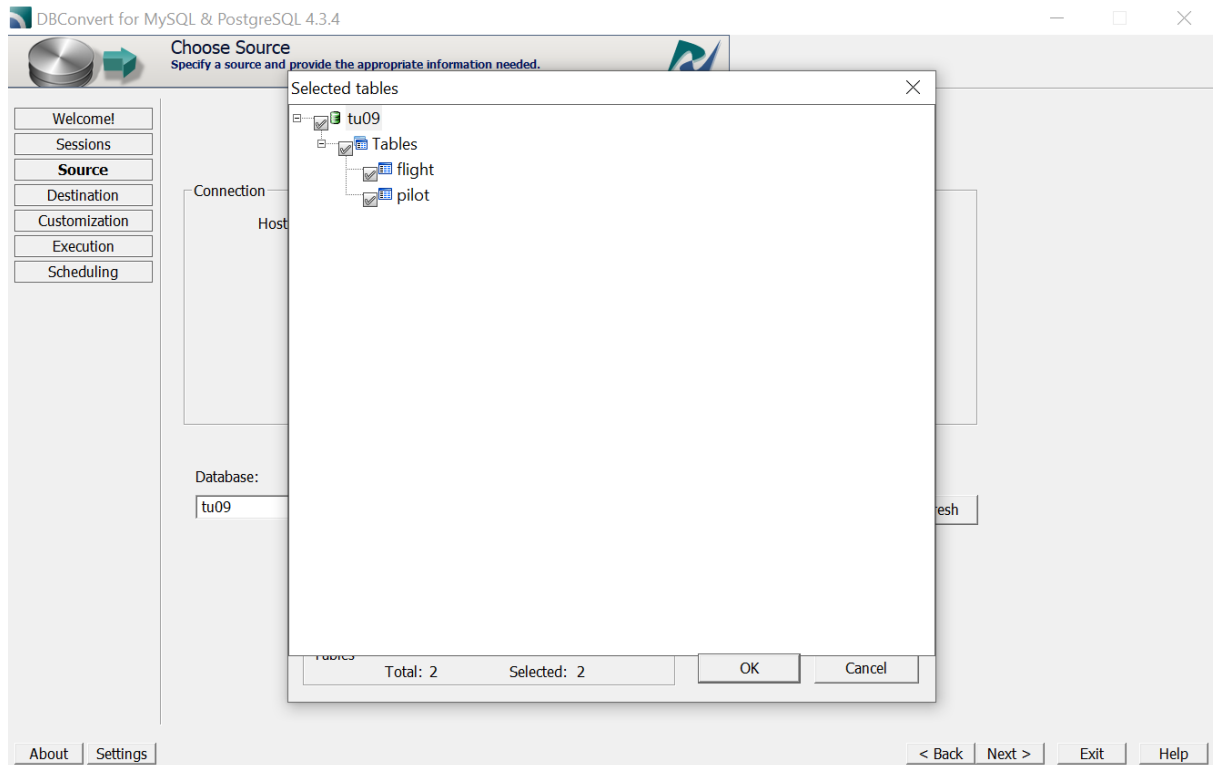
- Isi form pada bagian Source sesuai dengan informasi DB MySQL saat ini, misalnya sbb.:



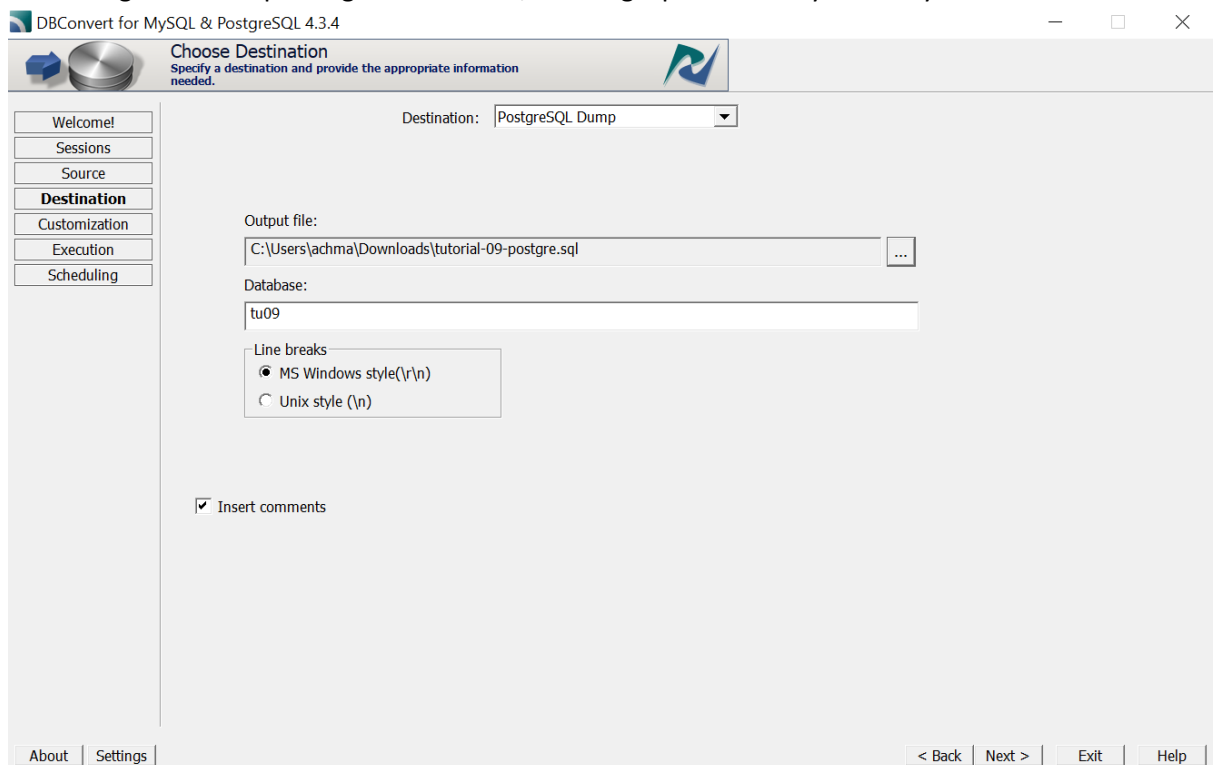
The screenshot shows the 'Choose Source' window of DBConvert for MySQL & PostgreSQL 4.3.4. The window has a sidebar on the left with buttons: Welcome!, Sessions, Source (highlighted), Destination, Customization, Execution, and Scheduling. The main area is titled 'Choose Source' and contains a 'Source' dropdown menu set to 'MySQL Database'. Below this is a 'Connection' section with several input fields: 'Hostname (default \'localhost\')' set to 'localhost', 'Port (default \'3306\')' set to '3306', 'Username (default \'root\')' set to 'root', 'Password' (empty), and 'Connection character set' set to 'utf8'. There are 'Advanced...' and 'Test connection' buttons. Below the 'Connection' section is a 'Database' dropdown menu set to 'tu09' and a 'Refresh' button. The bottom of the window has a status bar with 'About' and 'Settings' buttons on the left, and '< Back', 'Next >', 'Exit', and 'Help' buttons on the right.

lalu klik *Next*.

- Pilih tabel yang akan di *convert*, lalu klik OK.

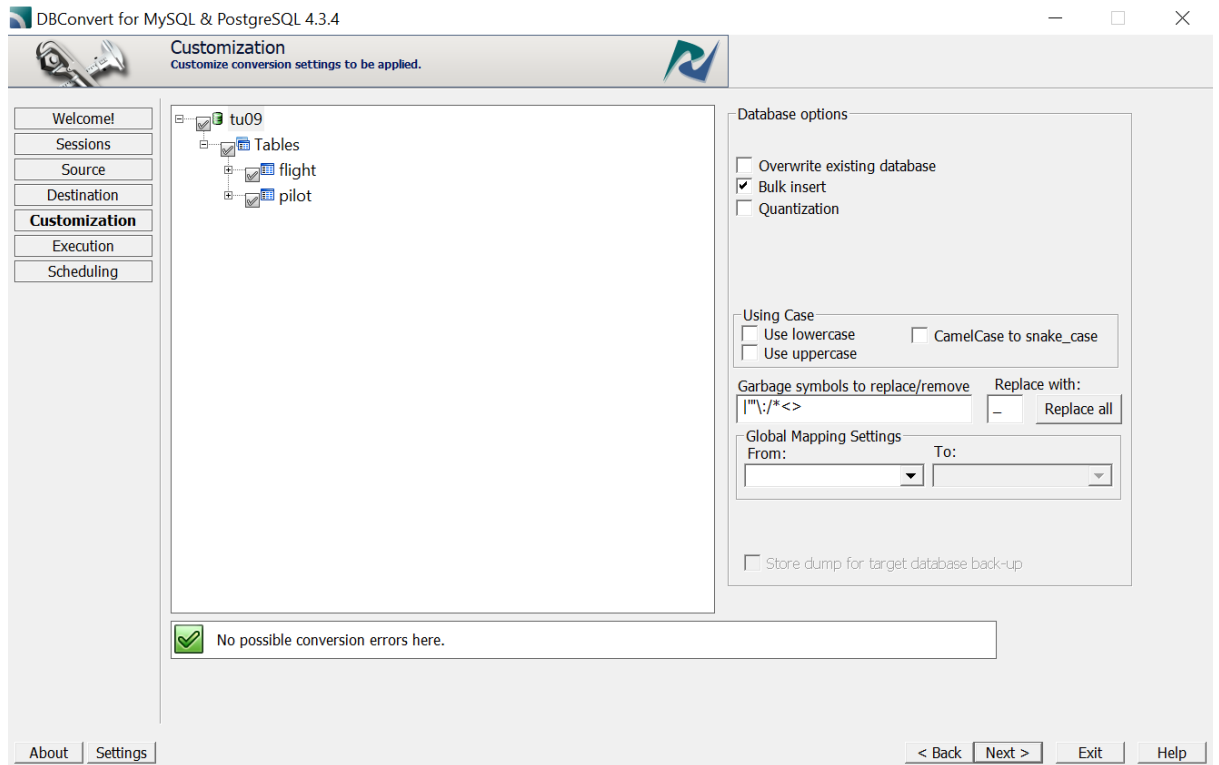


- Pilih PostgreSQL dump sebagai Destination, lalu lengkapi isian lainnya misalnya sbb.:

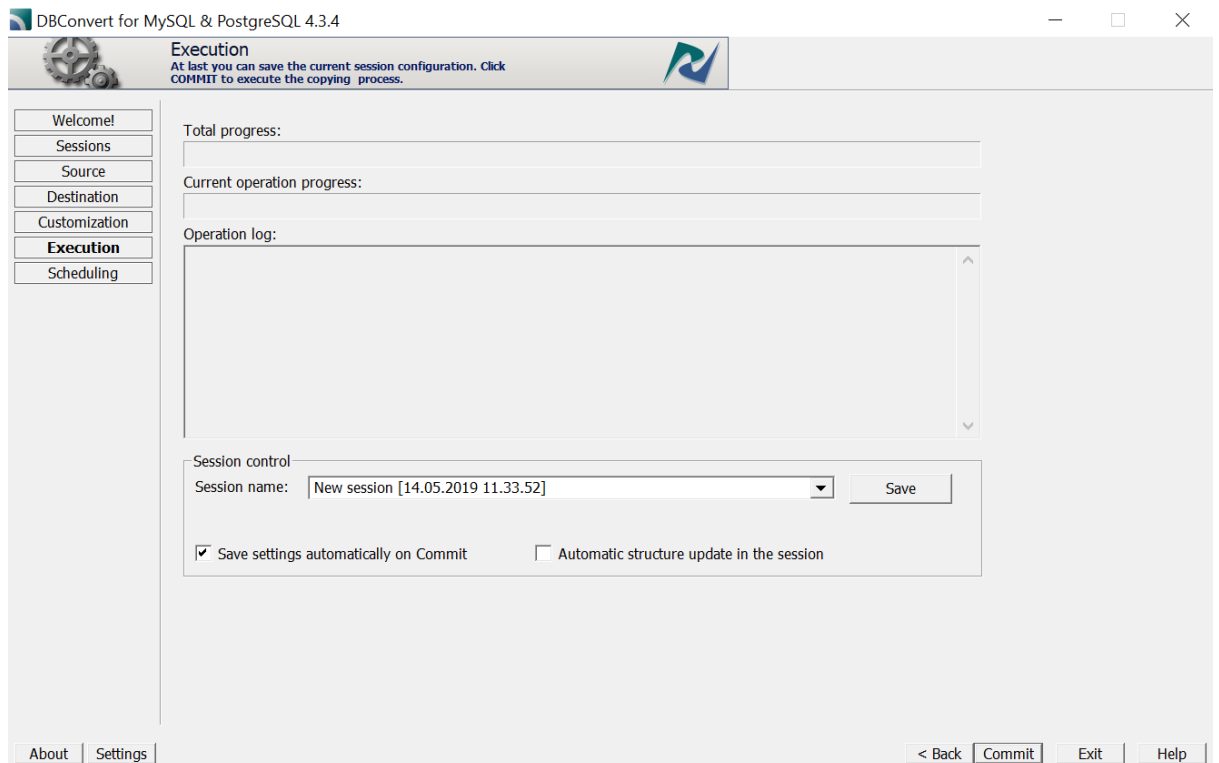


lalu klik *Next*.

- Klik *Next* pada bagian Customization.



- Klik Commit pada bagian Execution.



- Akan muncul *warning* seperti gambar di bawah karena menggunakan *demo version*, Klik OK.

DBConvert for MySQL & PostgreSQL 4.3.4



You are running a demo version of the program. When Trigger-based sync option is activated, it synchronizes 5 first records, the rest are not affected but shows the actual time of the process. If not, the tool synchronizes only 50 initial records for each database table. When copying data without syncing options checked, 50 first records are converted for each database table. All character data after 50 first records is watermarked with \*\*\*-TRIAL-(User's data here)\*\*\* text. Visit us to obtain full version.

OK

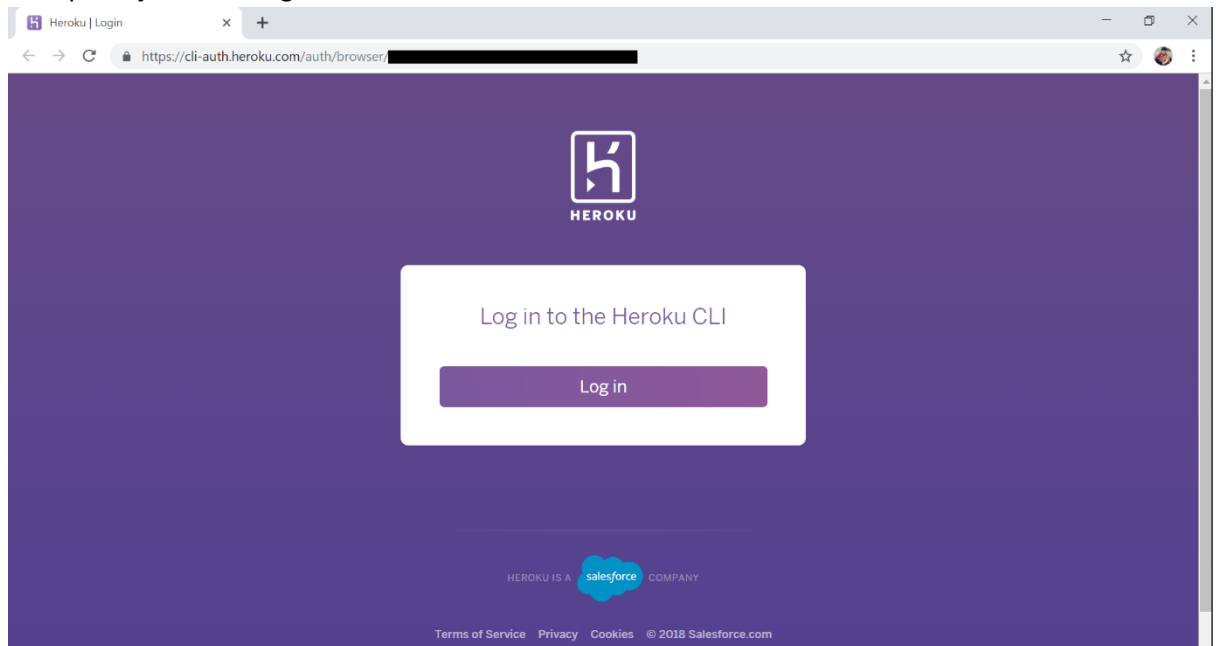
- Sebuah *file* akan terbuat pada *directory output file* yang telah ditentukan sebelumnya.

### 3. Deployment Aplikasi

- Hapus file *.gitignore* dan folder *.git* *directory project* yang akan di-*deploy* (jika ada).
- Buka *cmd/terminal* lalu masuk ke *directory project*, kemudian login ke Heroku dengan menjalankan perintah sbb.:

**heroku login**

- Ikuti petunjuk untuk login melalui browser.



- Jika berhasil login akan muncul pesan sbb.:

```
Logging in... done
Logged in as achmad.f.abka@gmail.com
```

- Kemudian jalankan perintah sbb.:

```
git init
git add .
git commit -m "Initial commit"
```

- Buat aplikasi pada Heroku menggunakan perintah sbb.:

```
heroku create [nama_app]
```

Ganti [nama\_app] dengan nama aplikasi yang anda inginkan.

- Jika berhasil akan muncul pesan misalnya sbb.:

```
Creating tutorial-09... done
https://tutorial-09.herokuapp.com/ | https://git.heroku.com/tutorial-09.git
```

\*tutorial-09 merupakan contoh nama aplikasi yang digunakan

- Ubah *remote access* git dengan menggunakan perintah sbb.:

```
heroku git:remote -a [nama_app]
```

Ganti [nama\_app] dengan nama aplikasi anda.

- Lalu push *directory project* dengan menggunakan perintah sbb.:

```
git push heroku master
```

Tunggu hingga proses *deploy* selesai.

- Jika berhasil akan muncul pesan misalnya, sbb.:

```
remote: [INFO] -----
remote: [INFO] BUILD SUCCESS
remote: [INFO] -----
remote: [INFO] Total time: 01:08 min
remote: [INFO] Finished at: 2019-05-14T05:31:57Z
remote: [INFO] -----
remote: -----> Discovering process types
remote: Procfile declares types -> (none)
remote: Default types for buildpack -> web
remote: -----> Compressing...
remote: Done: 83.8M
remote: -----> Launching...
remote: Released v5
remote: https://tutorial-09.herokuapp.com/ deployed to Heroku
remote: Verifying deploy... done.
To https://git.heroku.com/tutorial-09.git
* [new branch] master -> master
```

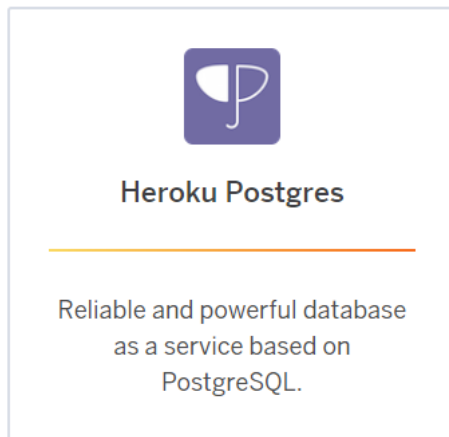
#### 4. Deployment Database

- Akses Heroku site menggunakan browser lalu login menggunakan akun Anda.
- Buka aplikasi dengan nama yang anda buat pada proses *deployment* aplikasi.
- Lalu pilih tab Resources.

- Klik tombol Find more add-ons.

Find more add-ons

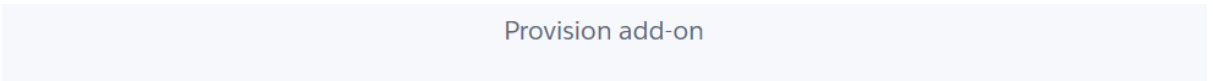
- Carilah dan pilih Heroku Postgres.



- Klik Install Heroku Postgres.



- Cari dan pilih aplikasi Anda untuk dihubungkan dengan DB.



Provision this add-on to an app  
Heroku Postgres

 [View on the Elements Marketplace](#)

#### Add-on plan

Hobby Dev - Free



#### App to provision to


 Search for an app to provision...

By provisioning this add-on, I agree to the [Terms of Service](#)

Provision add-on




- Lalu klik Provision add-on pada bagian bawah.




Provision this add-on to an app

Heroku Postgres

 [View on the Elements Marketplace](#)


#### Add-on plan

Hobby Dev - Free



Heroku Postgres  
Hobby Dev - Free

→




tutorial-09  
Personal apps

[Choose a different app](#)


By provisioning this add-on, I agree to the [Terms of Service](#)

**Provision add-on**

- Pilih DB yang telah Anda buat.

 **HEROKU**

Jump to Favorites, Apps, Pipelines, Spaces...



Personal > tutorial-09

★ Open app More

Overview Resources Deploy Metrics Activity Access Settings



Free Dynos [Change Dyno Type](#)

web java -Dserver.port=\$PORT \$JAVA\_OPTS -jar target/tutorial16-0.0.1-SNAPSHOT.jar \$0.00

Add-ons [Find more add-ons](#)

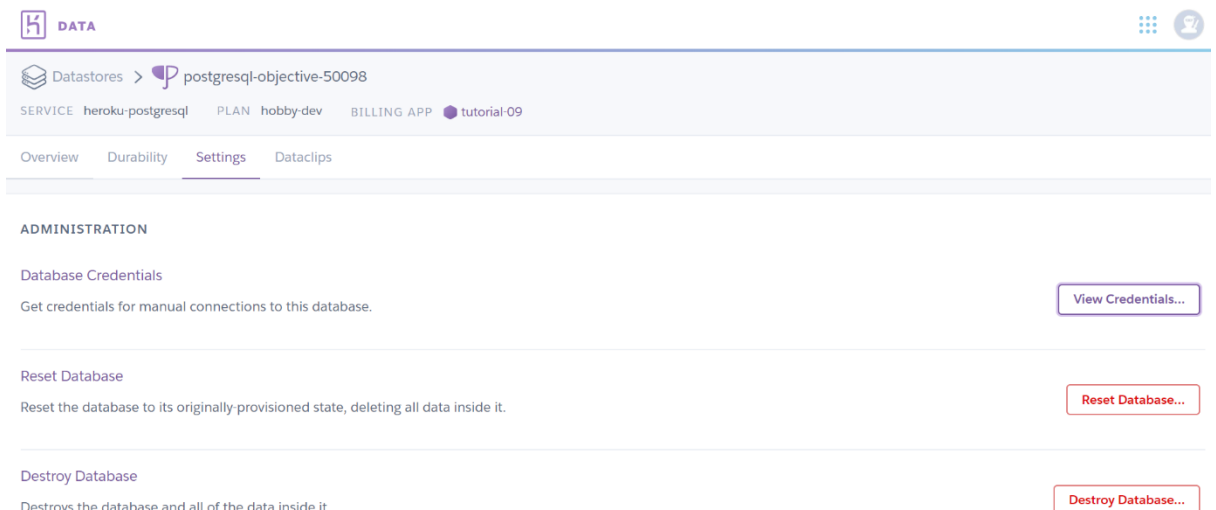
The add-on heroku-postgresql has been installed. Check out the documentation in its [Dev Center](#) article to get started.

Quickly add add-ons from Elements

 Heroku Postgres	Attached as DATABASE	Hobby Dev	Free	
 Heroku Postgres	Attached as HEROKU_POSTGRES_IVORY	Hobby Dev	Free	

Estimated Monthly Cost \$0.00

- Lalu pilih tab Setting.



- Klik View Credentials.



- Berikut contoh tampilan credentials.



Perintah pada bagian Heroku CLI digunakan untuk mengakses database melalui Heroku CLI (cmd/terminal).

- Copy perintah tersebut lalu jalankan pada cmd/terminal, misalnya sbb.:

```
heroku pg:psql postgresql-objective-50098 --app tutorial-09
```

- Jika berhasil maka akan terkoneksi dengan DB sbb.:

```
--> Connecting to postgresql-objective-50098
psql (11.3, server 11.2 (Ubuntu 11.2-1.pgdg16.04+1))
WARNING: Console code page (437) differs from Windows code page (1252)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

tutorial-09:~$ HEROKU_POSTGRES_SQL_IVORY=>
```

- Bukalah dengan editor *file dump database* yang sudah dibuat sebelumnya menggunakan DBConvert for MySQL & PostgreSQL misalnya sbb.:

```

tutorial-09-postgre.sql x
1  -- File name: C:\Users\achma\Downloads\tutorial-09-postgre.sql
2  -- Created by DBConvert http://www.dbconvert.com
3
4
5  --
6  -- Table structure for table `pilot`
7  --
8
9  CREATE TABLE "pilot" ( "id" BIGSERIAL NOT NULL ,
10 "fly_hour" INTEGER NOT NULL ,
11 "license_number" VARCHAR(50) NOT NULL ,
12 "name" VARCHAR(50) NOT NULL ,
13 PRIMARY KEY ("id"),
14 UNIQUE ("license_number")
15 );
16
17
18 --
19 -- Table structure for table `flight`
20 --
21
22 CREATE TABLE "flight" ( "id" BIGSERIAL NOT NULL ,
23 "destination" VARCHAR(50) NOT NULL ,
24 "flight_number" VARCHAR(50) NOT NULL ,
25 "origin" VARCHAR(50) NOT NULL ,
26 "time" DATE NOT NULL ,
27 "pilot_license_number" VARCHAR(255) NOT NULL ,
28 PRIMARY KEY ("id"),FOREIGN KEY ("pilot_license_number") REFERENCES "pilot" ( "license_number" ) ON
29 );
30 CREATE INDEX "flight_FK3la8bbsoy9ck6byoqa6bntwly" ON "flight" ("pilot_license_number");
31
32

```

- Cek pada bagian akhir sql tersebut, lalu hapus baris dibawah ini (jika ada):

```

CREATE DATABASE `tu09`;
USE `tu09`;

```

- Lalu *Copy* seluruh *code database* tersebut dan *paste* pada cmd/terminal (Heroku CLI) yang sebelumnya sudah terkoneksi dengan DB. Tunggu hingga proses selesai.
- Setelahnya, untuk memeriksa tabel pada DB jalankan perintah sbb.:

```
\d
```

- Contoh hasilnya sbb.:

List of relations			
Schema	Name	Type	Owner
public	flight	table	eovdshaecycrrb
public	flight_id_seq	sequence	eovdshaecycrrb
public	pilot	table	eovdshaecycrrb
public	pilot_id_seq	sequence	eovdshaecycrrb
(4 rows)			

- Untuk memeriksa jumlah data pada tabel dapat menggunakan perintah sbb.:

```
select count(*) from flight
```

\*flight merupakan nama tabel

- Contoh hasilnya sbb.:

```

count
-----
    560
(1 row)

```

- Keluar dari Heroku CLI (tutorial-09::HEROKU\_POSTGRESQL\_IVORY), gunakan perintah:

```
exit
```

- Kembali ke STS, lalu ubah **application.properties** sesuai dengan *database credentials* yang ada pada Heroku.

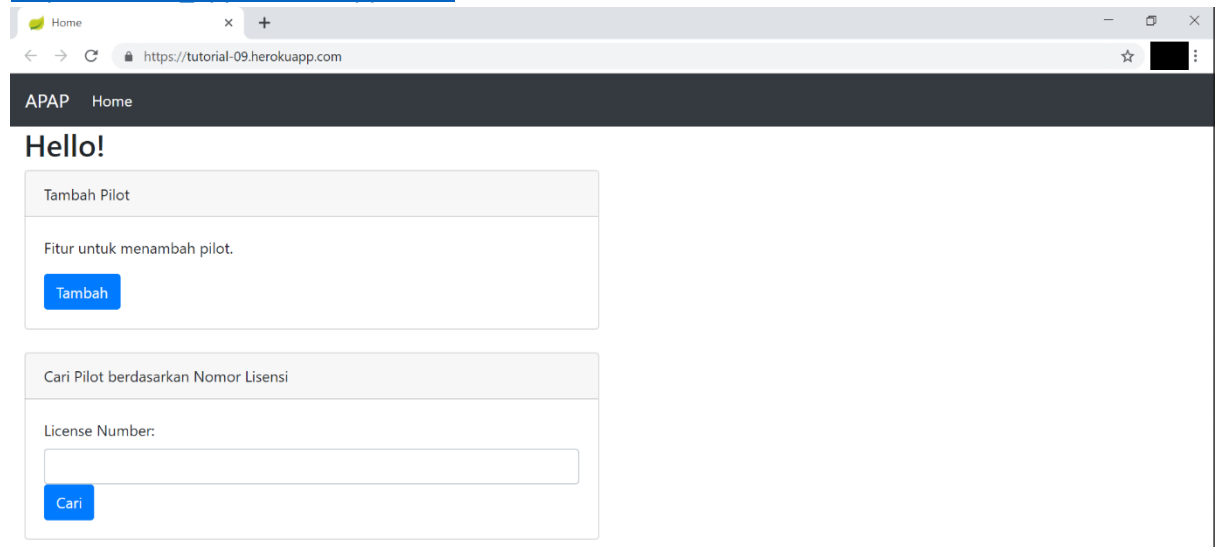
```
spring.datasource.url=jdbc:postgresql://{Host}
spring.datasource.username={User}
spring.datasource.password={Password}
```

- Perubahan tersebut baru dilakukan pada *local machine* maka selanjutnya lakukan push ke Heroku, pada cmd/terminal gunakan perintah sbb.:

```
git add .
git commit -m "Update credentials"
```

- Jika berhasil maka aplikasi dan database telah berhasil di-*deploy*.
- Aplikasi dapat diakses melalui browser dengan mengakses link:

[https://\[nama\\_app\].herokuapp.com/](https://[nama_app].herokuapp.com/)



## Pengumpulan

1. Penjelasan pengerjaan tutorial mengenai apa yang Anda pelajari. Tambahkan *screen capture* (*output* pada *browser*) yang menunjukkan aplikasi berhasil di-deploy. Pengumpulan dituliskan dalam satu file dengan format **npm\_nama-lengkap-anda.pdf** dan unggah ke submission slot yang disediakan di Scele.
2. Folder tutorial-09 dikumpulkan dengan cara push ke GitHub (<https://github.com/achmad-f-abka/apap>).