

**LAPORAN PRAKTIKUM**  
**TEKNOLOGI CLOUD**  
**PERTEMUAN KE – 10**



**Disusun Oleh :**

<b>NAMA</b>	<b>: TARISA DWI SEPTIA</b>
<b>NIM</b>	<b>: 205410126</b>
<b>JURUSAN</b>	<b>: TEKNIK INFORMATIKA</b>
<b>JENJANG</b>	<b>: S1</b>

**UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA**  
**YOGYAKARTA**  
**2020**

## Heroku

### A. Tujuan

- Mahasiswa dapat melakukan Implementasi Deploy Apps ke Heroku dengan menggunakan database PostgreSQL

### B. Dasar Teori

Heroku adalah sebuah cloud platform yang menjalankan bahasa pemrograman tertentu, Heroku mendukung bahasa pemrograman seperti Ruby, Node.js, Python, Java, PHP, dan lain-lain. Heroku termasuk ke dalam kriteria *Platform As A Service* (PaaS), sehingga bagi anda yang ingin melakukan deploy aplikasi ke heroku cukup hanya dengan melakukan konfigurasi aplikasi yang ingin di deploy dan menyediakan platform yang memungkinkan pelanggan untuk mengembangkan, menjalankan, dan mengelola aplikasi tanpa kompleksitas membangun dan memelihara infrastruktur yang biasanya terkait dengan pengembangan dan peluncuran aplikasi.

Manfaat menggunakan Heroku adalah layanannya yaitu menjalankan script app langsung tanpa memerlukan setting yang sangat rumit, memungkinkan pengembang aplikasi lebih fokus pada kode aplikasi mereka, tanpa terlalu dipusingkan dengan arsitektur dan server.

### C. Praktik

#### a. Persiapan

Disini, kita akan melakukan deploy aplikasi ke Heroku. Namun sebelum melakukan deploy ke Heroku, ada beberapa yang harus kita siapkan. Diantaranya yaitu :

- Aplikasi itu sendiri, bisa menggunakan bahasa pemrograman apapun. Namun disini kami mengambil random project dari Github yang beralamat di <https://github.com/haidarafif0809/express-crud>
- Akun heroku
- Alat untuk deploy ke Heroku menggunakan tool **heroku-cli & git**

#### b. Membuat akun heroku

- Buka Browser anda kemudian kunjungi alamat <https://www.heroku.com>
- Setelah anda sudah berada pada tampilan website heroku, kemudian klik **Sign Up** untuk melakukan pendaftaran

SPOTLIGHT

## Data on Heroku

Build data-driven apps with fully managed data services.

**SIGN UP FOR FREE**

Explore Managed Data Services

- Setelah di klik akan masuk ke form pendaftaran seperti gambar dibawah

**@ Free account**

Create apps, connect databases and add-on services, and collaborate on your apps, for free.

**@ Your app platform**

A platform for apps, with app management & instant scaling, for development and production.

**Deploy now**

Go from code to running app in minutes. Deploy, scale, and deliver your app to the world.

First name \*

Last name \*

Email address \*

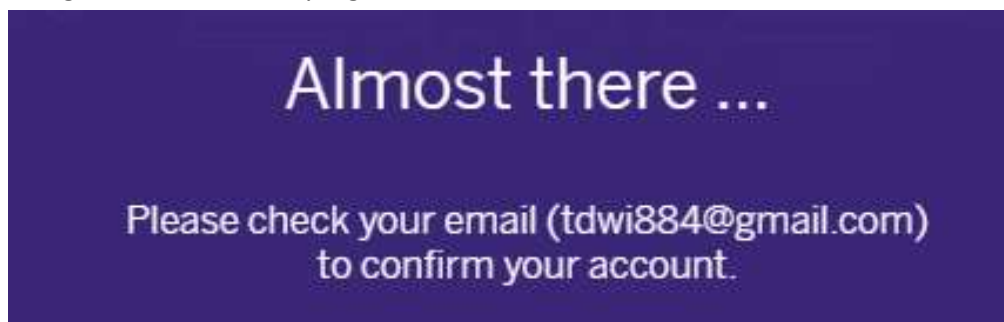
Company name

Role \*

Country \*

Primary development language \*

- Setelah Anda mengklik “CREATE FREE ACCOUNT”, Anda akan dibawa ke halaman baru, dimana terdapat keterangan bahwa: “**Anda harus mengecek email Anda** dan mengklik link konfirmasi yang dikirim ke alamat email Anda”.



- Setelah Anda membuka email Anda, Anda akan mendapatkan email dari Heroku untuk melakukan konfirmasi. **Silahkan buka email** tersebut.



Thanks for signing up with Heroku! You must follow this link within 30 days of registration to activate your account:

<https://id.heroku.com/account/accept/10924756/6806d270f6e464c233285f4bd6f3d162>

- Jika Anda telah membuka pesan email tersebut, Anda akan mendapatkan link konfirmasi yang ada di dalam pesan email tersebut. Untuk melanjutkan, **klik link konfirmasi** tersebut.
- Setelah Anda mengklik link konfirmasi, Anda akan dibawa ke halaman “Set your password”. Untuk melanjutkan, silahkan Anda masukkan *password* Anda pada **kotak Password** (“1”), setelah itu ketikkan ulang *password* Anda pada **kotak “Password confirmation”** (“2”), kemudian pilih dan **klik “SET PASSWORD AND LOG IN”** (“3”).

## Set your password

Create your password and log in to your Heroku account.

Create a new password \*

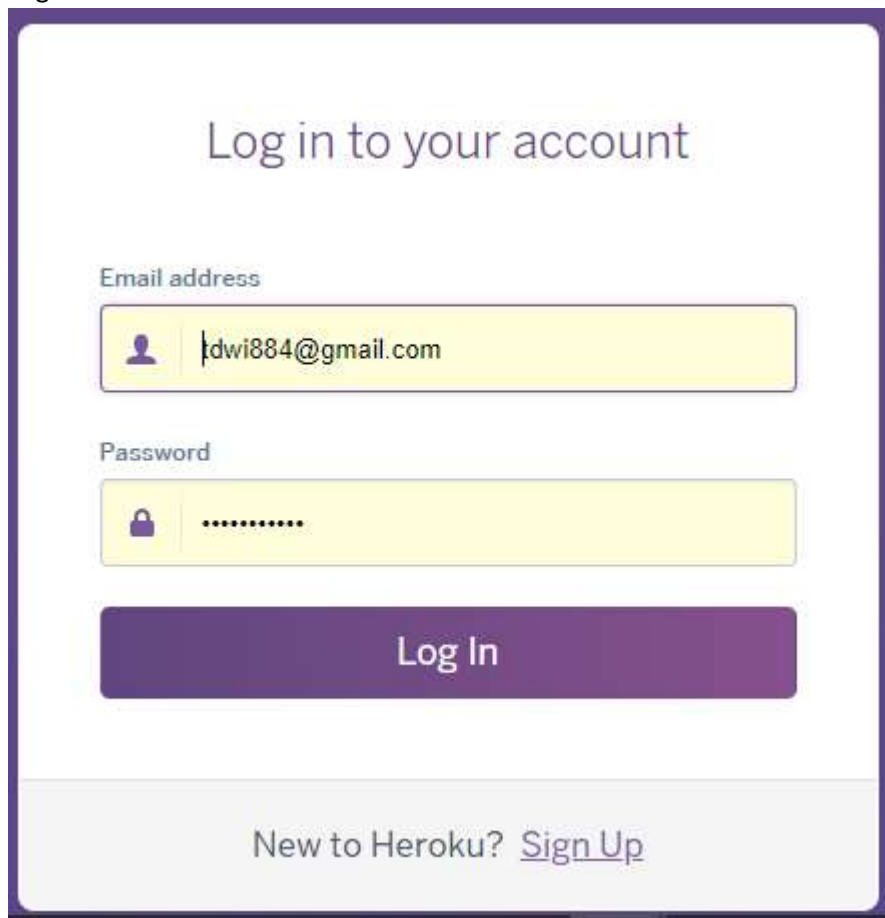
Password confirmation \*

Password requirements

- Must be a minimum of 8 characters.
- Must contain letters, numbers, and symbols.
- Passwords must match.

**SET PASSWORD AND LOG IN**

- Kemudian masukan Username & Password yang telah anda buat tadi, jika sudah klik Log in

A screenshot of the Heroku login page. The page has a white background with a purple border. At the top, it says "Log in to your account". Below this, there are two input fields: "Email address" and "Password". The email field contains "dwi884@gmail.com" and the password field contains a series of dots. Below the password field is a purple "Log In" button. At the bottom, there is a link that says "New to Heroku? Sign Up".

- Setelah Anda melakukannya, Anda akan langsung dibawa ke halaman Heroku Anda.



### c. Instalasi Heroku CLI (Command Line Interface)

Setelah melakukan registrasi di Heroku, kemudian install **heroku-cli** di komputer. Berbeda sistem operasi tentu berbeda cara installnya, untuk itu dibawah ini sudah kami sediakan link untuk berbagai macam sistem operasi

Kami menggunakan sistem operasi linux, jadi kami mengambil cara instalasi langsung melalui Terminal atau jika anda tidak memiliki linux di laptop atau VM, anda dapat menggunakan [playground ubuntu katacoda](#). Berikut kami sediakan gambar instalasinya

```
$ sudo snap install --classic heroku
2021-12-06T03:40:49Z INFO Waiting for automatic snapd restart...
heroku v7.59.1 from Heroku✓ installed
```

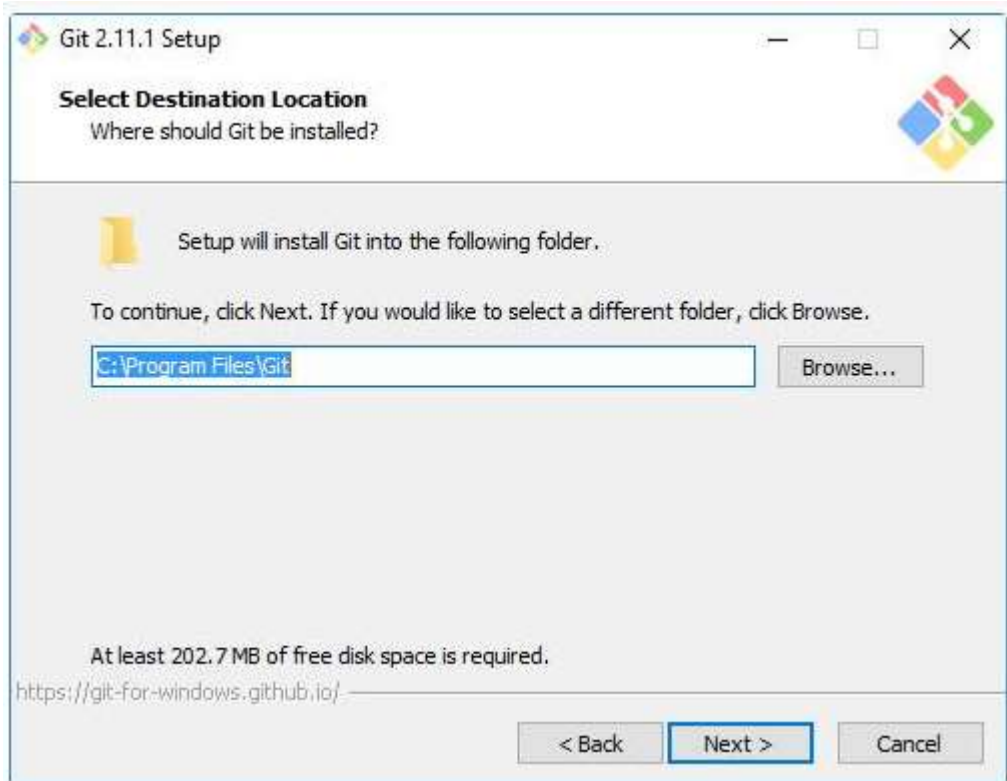
Jika sudah download, silahkan lakukan instalasi terlebih dahulu. Kemudian apabila instalasi sudah selesai, kita akan masuk ke tahap berikutnya, yaitu persiapan untuk deploy project kita kedalam repository Heroku.

#### d. Instalasi Git

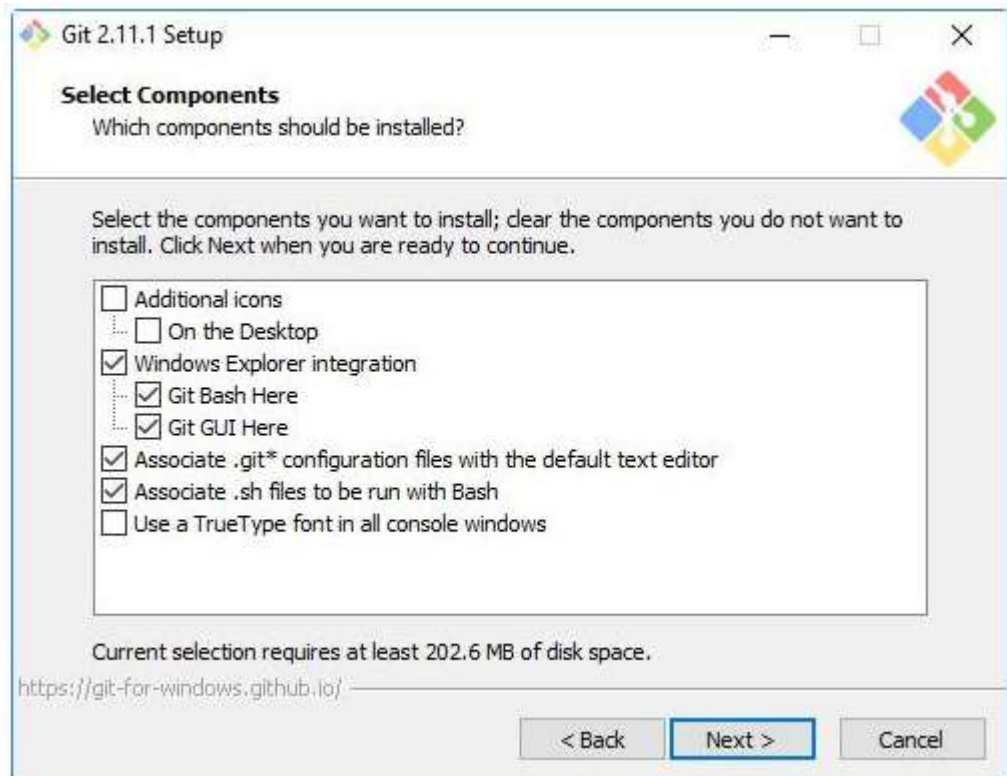
- Download git
- Klik 2x file instaler *Git* yang sudah diunduh. Maka akan muncul informasi lisensi Git, klik *Next >* untuk melanjutkan.



- Selanjutnya menentukan lokasi instalasi. Biarkan saja apa adanya, kemudian klik *Next >*.

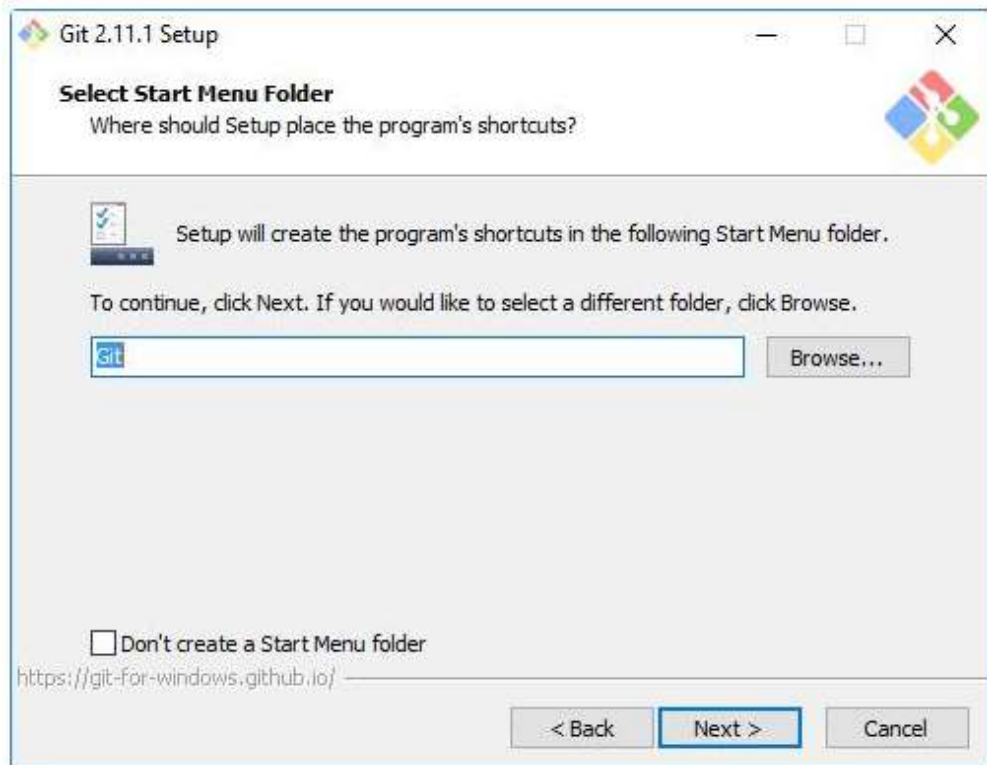


- Selanjutnya pemilihan komponen, biarkan saja seperti ini kemudian klik *Next >*.

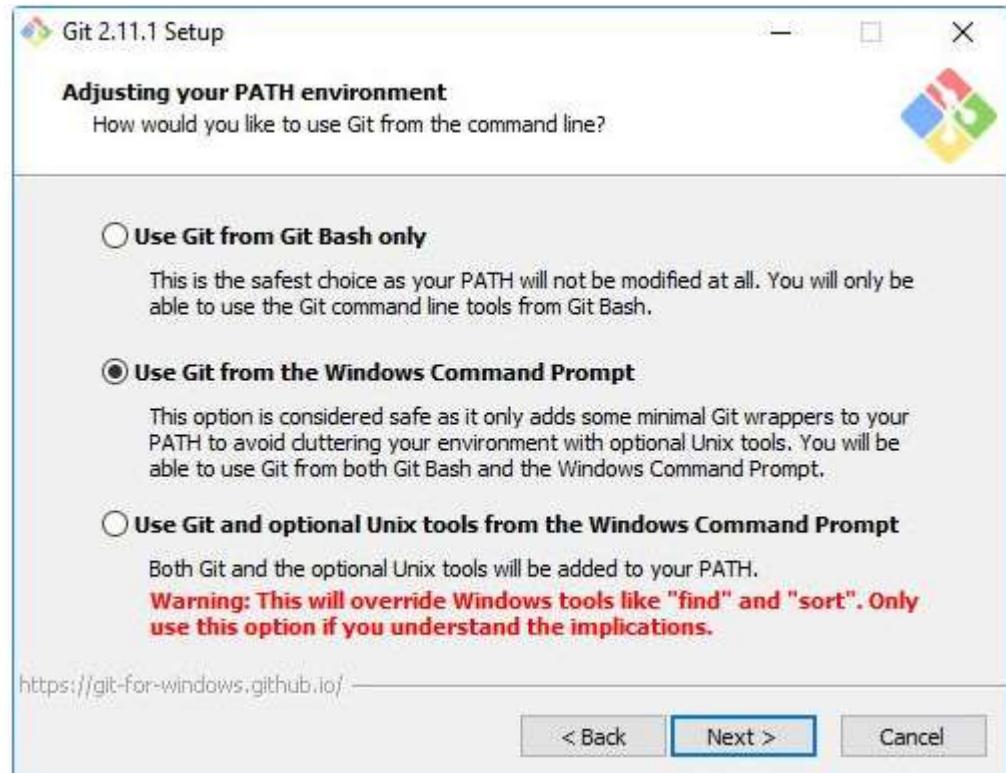




- Selanjutnya pemilihan direktori start menu, klik *Next >*.

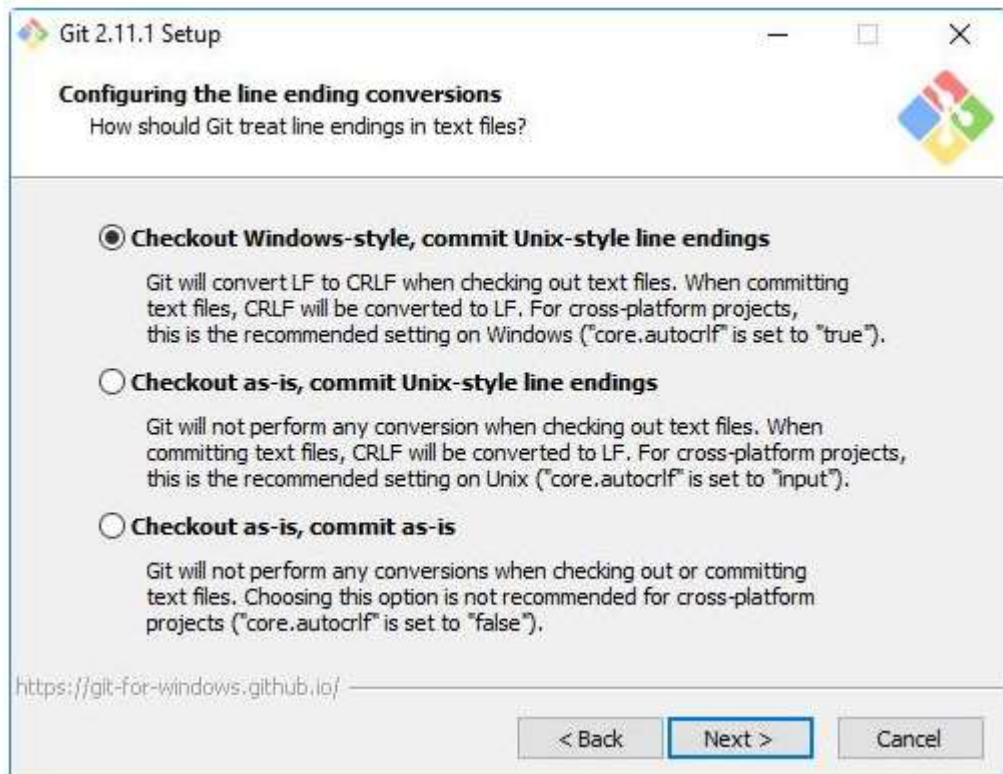


- Selanjutnya pengaturan *PATH Environment*. Pilih yang tengah agar perintah git dapat di kenali di *Command Prompt* (CMD). Setelah itu klik *Next >*.

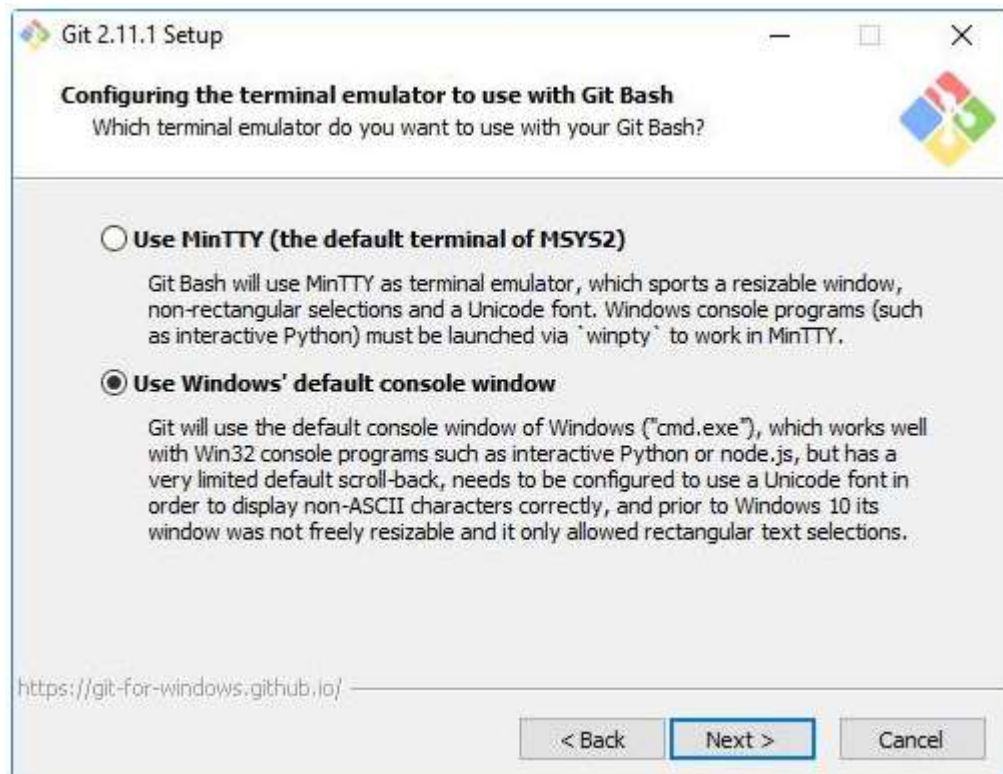




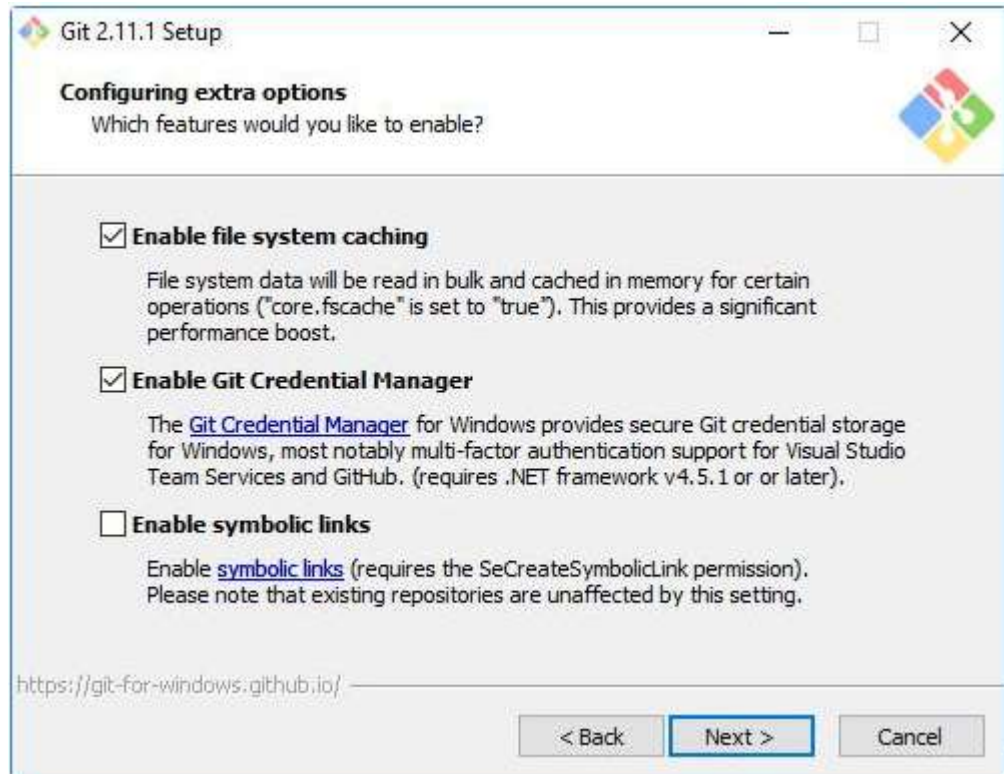
- Selanjutnya konversi *line ending*. Biarkan saja seperti ini, kemudian klik *Next >*.



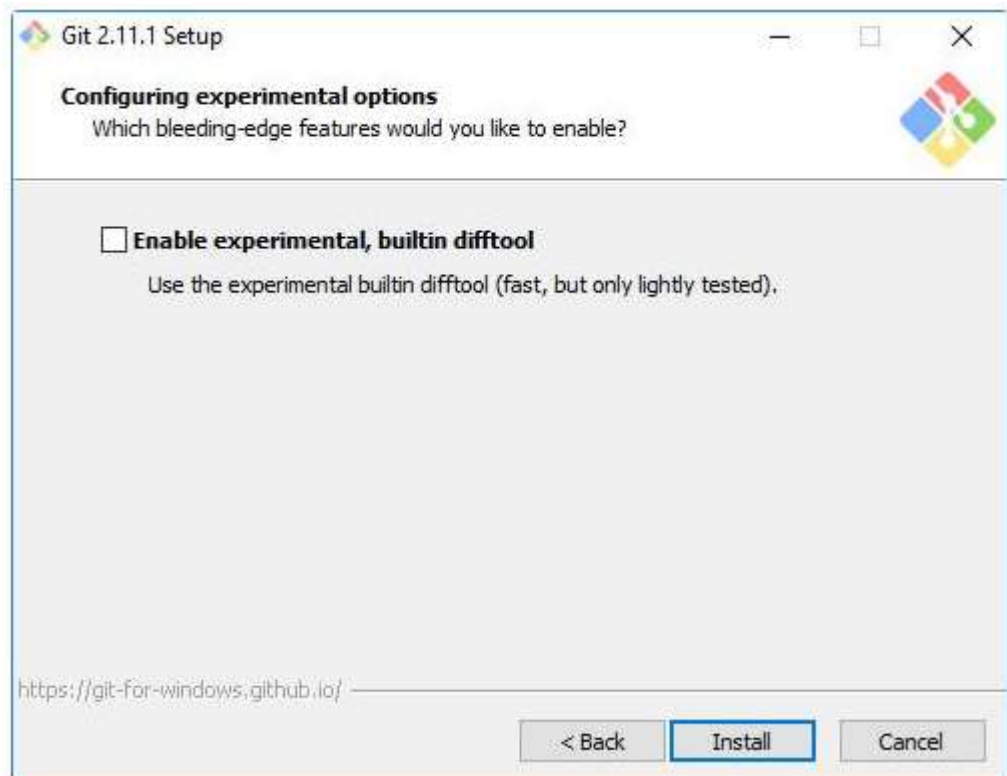
- Selanjutnya pemilihan emulator terminal. Pilih saja yang bawah, kemudian klik *Next>*.



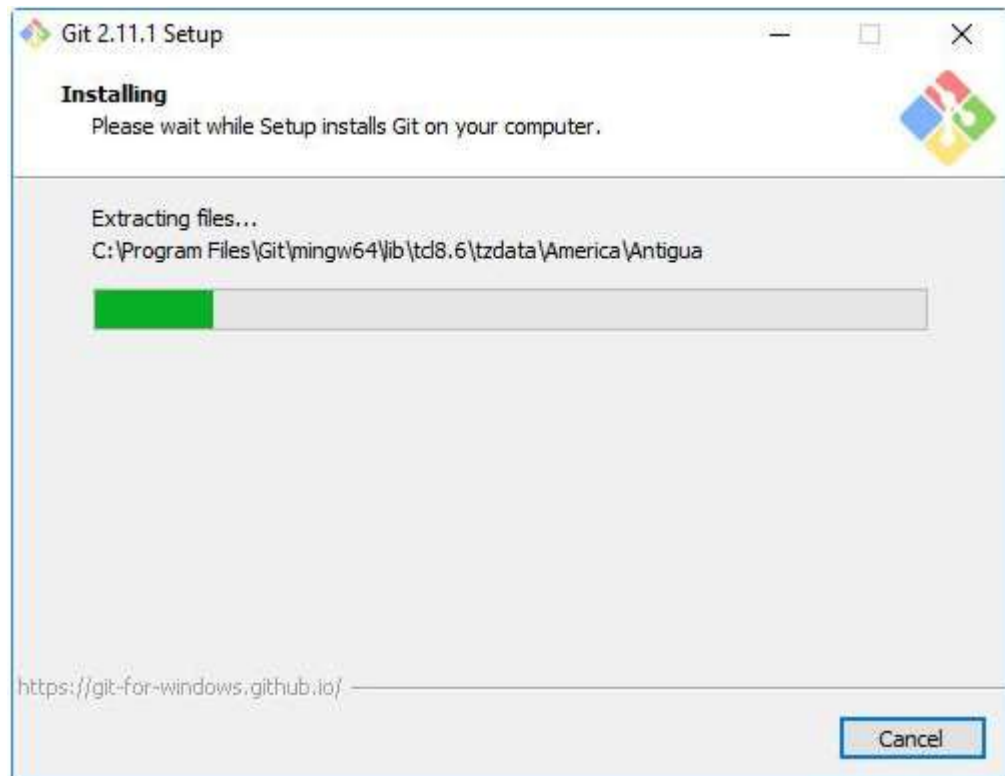
- Selanjutnya pemilihan opsi ekstra. Klik saja *Next >*.



- Selanjutnya pemilihan opsi eksperimental, langsung saja klik *Install* untuk memulai instalasi.



- Tunggu beberapa saat, instalasi sedang dilakukan.



- Setelah selesai, kita bisa langsung klik *Finish*. Git sudah terinstal di Windows. Untuk mencobanya, silahkan buka CMD atau PowerShell, kemudian ketik perintah

```
hp@DESKTOP-U98GI9G MINGW64 ~  
$ git --version  
git version 2.33.0.windows.2
```

#### e. Konfigurasi Awal

Ada beberapa konfigurasi yang harus dipersiapkan sebelum mulai menggunakan Git, seperti *name* dan *email*. Silahkan lakukan konfigurasi dengan perintah berikut ini:

```
hp@DESKTOP-U98GI9G MINGW64 ~
$ git config --global user.name "Tarisa-source"

hp@DESKTOP-U98GI9G MINGW64 ~
$ git config --global user.email "tdwi884@gmail.com"

hp@DESKTOP-U98GI9G MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
core.usebuiltinfsmonitor=true
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Tarisa-source
user.email=tdwi884@gmail.com
usre.name=Tarisa-05
usre.email=tarisadwi94@gmail.com
```

Jika memiliki akun Github, Gitlab, Bitbucket atau yang lainnya maka *username* dan *email* harus mengikuti akun tersebut agar mudah diintegrasikan.

#### f. Cloning Apps dari Github

Sebelum melakukan cloning, pertama kita harus pilih repositorynya dulu. Contoh disini, kita akan menggunakan apps CRUD menggunakan Express dengan database PostgreSQL. Langsung saja menuju linknya : <https://github.com/haidarafif0809/express-crud>

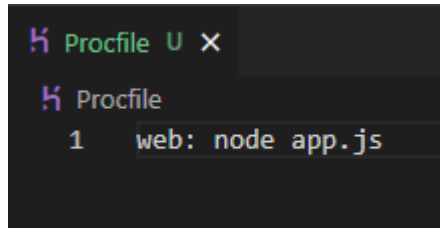
- Jika sudah mendapatkan target repo yang akan kita clone, maka kemudian klik tombol **Clone or download**
- Lalu copy url yang tertera
- Kemudian paste di Terminal dengan menggunakan command CTRL + SHIFT + V

```
$ git clone https://github.com/haidarafif0809/express-crud.git
Cloning into 'express-crud'...
remote: Enumerating objects: 140, done.
remote: Counting objects: 100% (1/1), done.
Rremote: Total 140 (delta 0), reused 0 (delta 0), pack-reused 139
Receiving objects: 100% (140/140), 26.33 KiB | 313.00 KiB/s, done.
Resolving deltas: 100% (65/65), done.
```

- Setelah itu masuk ke dalam project dengan command:

```
$ cd express-crud
```

- Kemudian pastikan terdapat file bernama **procfile** dengan tampilan seperti berikut:



#### g. Deploy Apps ke Heroku

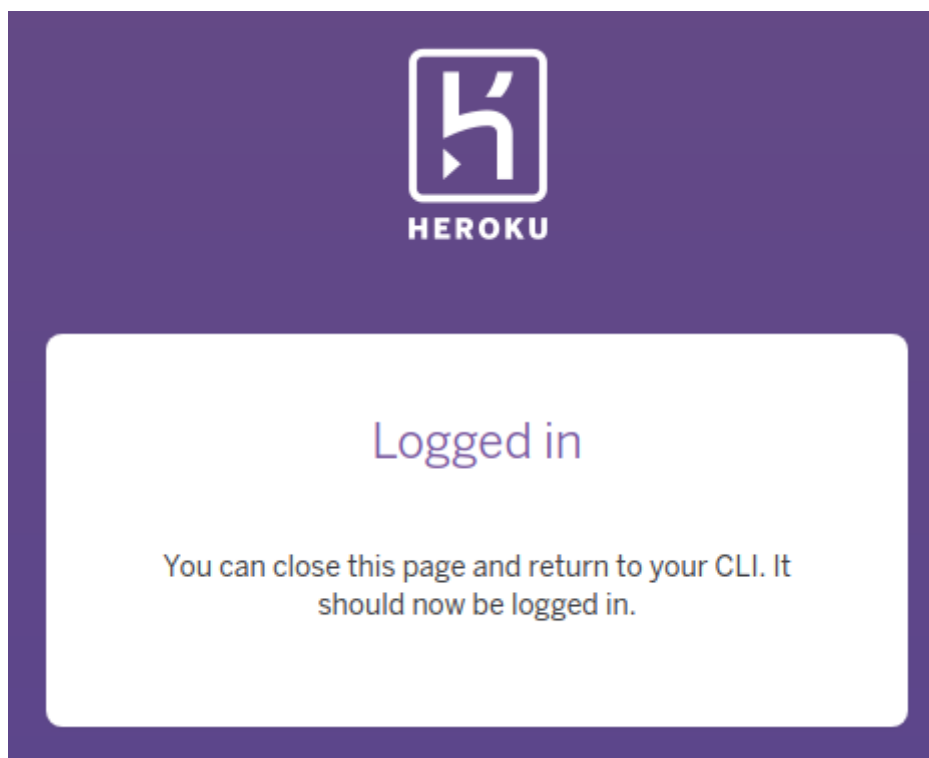
Agar apps bisa di akses oleh publik, maka kita harus menyimpan apps tersebut di Heroku. Caranya adalah sebagai berikut :

Otorisasi Akun Heroku di Terminal

- Sebelum melakukan deploy, terlebih dahulu kita login ke Heroku melalui browser seperti biasa [disini](#)
- Buka Terminal, lalu ketik perintah:

```
hp@DESKTOP-U98GI9G MINGW64 ~/express-crud (master)
$ heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/17e06593-7d99-4d
fd-9ad7-566414bbed08?requestor=SFMyNTY.g2gDbQAAAA4xMTQuMTQyLjE3MC4yMW4GALhw0JJ9A
WIAAVGA.sc93bDoz-918gYAnaGbDLELPddGD0Mtp0Cqp8PZYipc
heroku: Waiting for login...
```

- Setelah menggunakan perintah tersebut, bkan muncul tab di browser **Log in to the Heroku CLI**, maka klik **login**



- Lalu cek kembali di Terminal, jika login berhasil maka akan terdeteksi email akun Heroku anda, terdapat informasi : **Logged in as** [username@emailanda.com](#)

```
Logging in... done
Logged in as tdwi884@gmail.com
```

#### h. Membuat Repository dan Deploy Apps ke Heroku

Kemudian kita buat repository dulu di Heroku, sebenarnya ada 2 cara yaitu dengan membuat dahulu melalui browser atau melalui Terminal, letak perbedaannya hanya pada awal saja yaitu nama repository jika melalui Terminal akan secara random, berbeda dengan membuat repository secara manual yang bisa kita isi sendiri namanya. Tapi itu bukan masalah, karena nanti bisa diganti. Kita disini menggunakan cara melalui Terminal, langsung saja :

- Ketik command **heroku create** untuk membuat random name repo

```
$ heroku create
Creating app... done, peaceful-ravine-26983
https://peaceful-ravine-26983.herokuapp.com/ | https://git.heroku.com/peaceful-ravine-26983.git
```

- Sebelum melakukan deploy kita cek dahulu remote url git sudah bertambah ke alamat repository Heroku, caranya ketik **git remote -v**

```
$ git remote -v
heroku https://git.heroku.com/peaceful-ravine-26983.git (fetch)
heroku https://git.heroku.com/peaceful-ravine-26983.git (push)
origin https://github.com/haidarafif0809/express-crud.git (fetch)
origin https://github.com/haidarafif0809/express-crud.git (push)
```

- Jika sudah ada, maka lanjut untuk deploy apps. Ketik **git push heroku master** untuk melakukan deploy ke repository Heroku akun kita

```
hp@DESKTOP-U98GI9G MINGW64 ~/express-crud (master)
$ git push heroku master
Enumerating objects: 140, done.
Counting objects: 100% (140/140), done.
Delta compression using up to 2 threads
Compressing objects: 100% (70/70), done.
Writing objects: 100% (140/140), 26.34 KiB | 539.00 KiB/s, done.
Total 140 (delta 65), reused 140 (delta 65), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-20 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:       NPM_CONFIG_LOGLEVEL=error
remote:       NODE_VERBOSE=false
remote:       NODE_ENV=production
remote:       NODE_MODULES_CACHE=true
remote:
remote: -----> Installing binaries
remote:       engines.node (package.json): unspecified
```

- Maka Heroku akan mendeteksi jenis bahasa pemrograman yang kita gunakan dalam pembuatan apps, dan kebutuhan lainnya yang sudah tertulis di *package.json*
- Kemudian kita tunggu sampai proses deploy selesai, jika sudah maka akan tertulis di Terminal seperti gambar dibawah ini

```
remote: Building source:
remote:
remote: -----> Building on the Heroku-20 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:       NPM_CONFIG_LOGLEVEL=error
remote:       NODE_VERBOSE=false
remote:       NODE_ENV=production
remote:       NODE_MODULES_CACHE=true
remote:
remote: -----> Installing binaries
remote:       engines.node (package.json):  unspecified
remote:       engines.npm (package.json):   unspecified (use default)
remote:
remote:       Resolving node version 14.x...
remote:       Downloading and installing node 14.18.2...
remote:       Using default npm version: 6.14.15
remote:
remote: -----> Installing dependencies
remote:       Installing node modules
remote:       added 96 packages in 2.198s
remote:
remote: -----> Build
remote:
remote: -----> Caching build
remote:       - node_modules
remote:
remote: -----> Pruning devDependencies
remote:       audited 96 packages in 0.689s
remote:       found 9 vulnerabilities (3 moderate, 3 high, 3 critical)
remote:         run `npm audit fix` to fix them, or `npm audit` for details
remote:
remote: -----> Build succeeded!
remote: -----> Discovering process types
remote:       Procfile declares types      -> (none)
remote:       Default types for buildpack -> web
remote:
remote: -----> Compressing...
remote:       Done: 36.6M
remote: -----> Launching...
remote:       Released v3
remote:       https://peaceful-ravine-26983.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/peaceful-ravine-26983.git
* [new branch]      master -> master
```



- Lalu kita coba buka apps kita di <https://peaceful-ravine-26983.herokuapp.com/>

```
2021-12-07T03:04:24.197894+00:00 app[web.1]: /app/node_modules/sequelize/lib/dialects/mysql/connection-manager.js:38
2021-12-07T03:04:24.197906+00:00 app[web.1]: throw new Error('Please install mysql2 package manually');
```

## Application error

An error occurred in the application and your page could not be served. If you are the application owner, [check your logs for details](#).

You can do this from the Heroku CLI with the command

```
heroku logs --tail
```

- Ternyata error, maka kita akan cek errornya seperti apa dengan command **heroku logs --tail**
- Katanya di suruh install package mysql2, padahal kita gak pakai mysql , tapi pakai postgre. Oke berarti cek file config.json kita yang mengatur koneksi database.

```
"production": {
  "username": "root",
  "password": null,
  "database": "database_production",
  "host": "127.0.0.1",
  "dialect": "mysql"
}
```

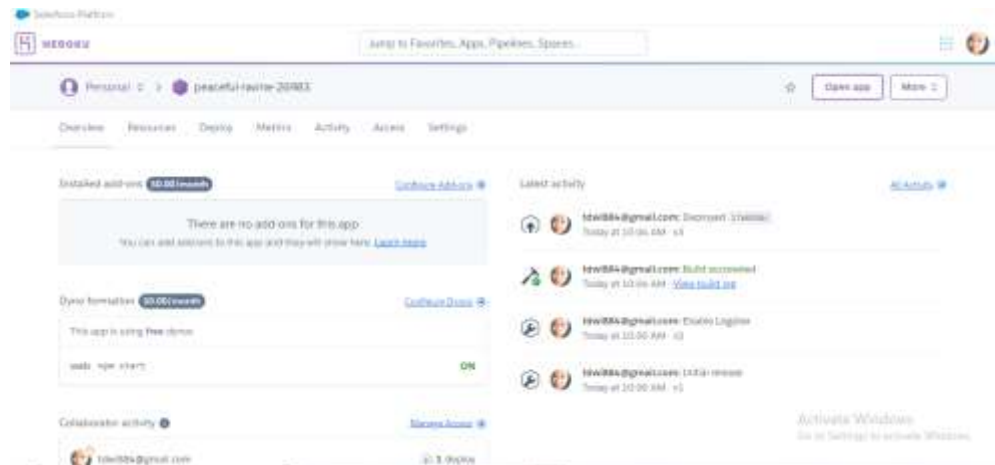
Ternyata config db production nya masih default. Ketika app nya di deploy ke heroku. maka sequelize akan menggunakan config production. Maka kita harus setting config production nya, tapi sebelum itu kita harus menambah database driver yang akan digunakan yaitu PostgreSQL di repository Heroku terlebih dahulu, caranya ada di langkah berikutnya

### i. Konfigurasi Database di Heroku (PostgreSQL)

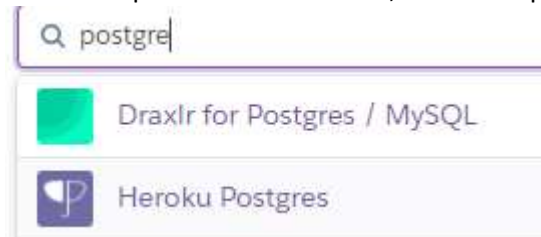
Untuk menambahkan database driver yang akan kita gunakan yaitu PostgreSQL di repository Heroku yang telah kita buat tadi caranya adalah sebagai berikut :

Inisialisasi Database Driver dan Konfigurasi

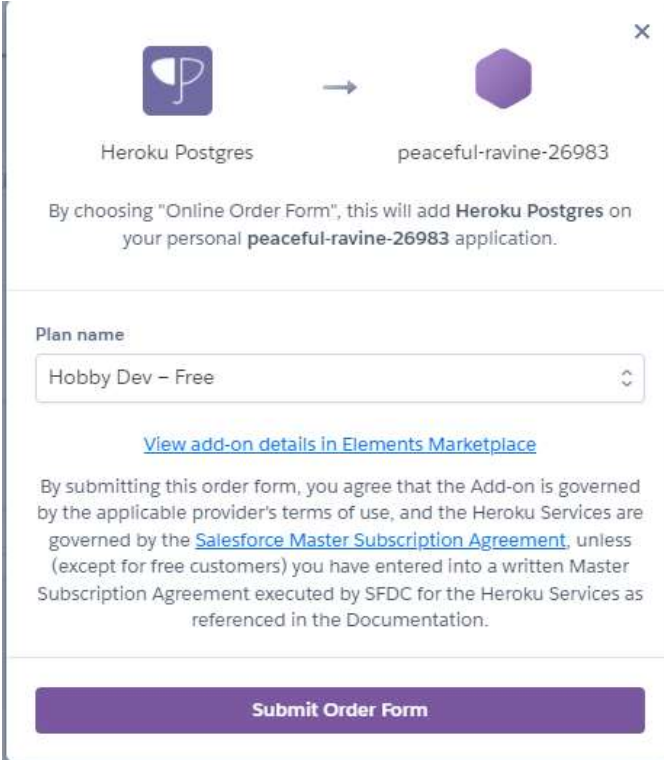
- Masuk ke halaman dashboard Heroku, kemudian pilih letak repository yang sudah kita buat



- Kemudian pilih menu **Resources**, lalu ketik “postgres”



- Akan muncul pop-up seperti ini, kita klik **Provision**



Heroku Postgres → peaceful-ravine-26983

By choosing "Online Order Form", this will add Heroku Postgres on your personal peaceful-ravine-26983 application.

Plan name  
Hobby Dev – Free

[View add-on details in Elements Marketplace](#)

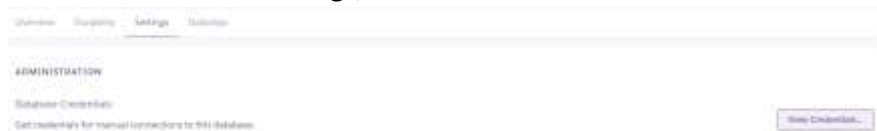
By submitting this order form, you agree that the Add-on is governed by the applicable provider's terms of use, and the Heroku Services are governed by the [Salesforce Master Subscription Agreement](#), unless (except for free customers) you have entered into a written Master Subscription Agreement executed by SFDC for the Heroku Services as referenced in the Documentation.

**Submit Order Form**

- Kita klik add-on yang sudah kita pilih baru saja



- Lalu masuk ke menu **Settings**, dan klik **View Credentials**



- Muncul detail konfigurasi database-nya



- Lalu ubah konfigurasi di file **config.json** yang berada di folder *config* hingga menjadi seperti di bawah ini

```
"production": {
  "username": "dlqqsrucltubibl",
  "password": "3c56d8255ae4397185fbae334045e4b3a01014d18aee339e7b37653635303665",
  "database": "d2ochf84ugi538",
  "host": "ec2-34-195-69-118.compute-1.amazonaws.com",
  "dialect": "postgres"
}
```

Tapi pasti nya akan masih error, karena table kita belum di migration. Untuk melakukan command migration sequelize berarti kita harus masuk ke heroku nya lewat ssh. Ada di langkah selanjutnya

#### j. Migrasi Database Menggunakan Sequelize di SSH Heroku

- Ketik command **heroku run bash**

```
heroku run bash
$ heroku run bash
Running bash on peaceful-ravine-26983... starting, run.8722 (Free)
Running bash on peaceful-ravine-26983... connecting, run.8722 (Free)Running bash on peaceful-ravine-26983... up, run.8722 (Free)
```

- Lalu install *sequelize-cli* melalui SSH Heroku dengan command **npm install -g sequelize-cli**

```
$ npm install -g sequelize-cli
npm install -g sequelize-cli
/app/.heroku/node/bin/sequelize -> /app/.heroku/node/lib/node_modules/sequelize-cli/lib/sequelize
/app/.heroku/node/bin/sequelize-cli -> /app/.heroku/node/lib/node_modules/sequelize-cli/lib/sequelize
+ sequelize-cli@6.3.0
added 80 packages from 49 contributors in 9.049s
```

- Kemudian ketik command **sequelize db:migrate** dan **sequelize db:seed:all** untuk migrasi table dan seeder-nya

```
~ $ sequelize db:seed:all
sequelize db:seed:all

Sequelize CLI [Node: 14.18.2, CLI: 6.3.0, ORM: 4.34.0]

Loaded configuration file "config/config.json".
Using environment "production".
sequelize deprecated String based operators are now deprecated.
erators node_modules/sequelize/lib/sequelize.js:242:13
```

- Selesai, kemudian ketik command **exit** untuk keluar dari SSH Heroku

```
~ $ exit
exit
exit
```

**D. Kesimpulan**

Setelah melakukan praktik diatas dapat disimpulkan bahwa mahasiswa dapat melakukan Implementasi Deploy Apps ke Heroku dengan menggunakan database PostgreSQL.