



# BUKU PANDUAN

Otentikasi Dua Faktor pada Aplikasi  
Web Berbasis Blockchain

## Tentang Aplikasi:

Otentikasi dua faktor dibuat untuk mengatasi masalah keamanan otentikasi satu faktor dimana pengguna cenderung menggunakan password yang sama pada berbagai macam platform dan cenderung menggunakan password yang mudah diingat. Kedua hal tersebut dapat dikategorikan sebagai password dengan tingkat keamanan yang rendah sehingga rentan untuk diretas. Tetapi otentikasi dua faktor juga memiliki kelemahan, yaitu data pengguna masih disimpan dalam sebuah database yang terpusat sehingga memiliki resiko keamanan dimana database memungkinkan untuk diretas yang menyebabkan kerusakan bahkan kehilangan data dalam jumlah yang besar. Oleh karena itu dibuatlah aplikasi **“Otentikasi Dua Faktor pada Aplikasi Web Berbasis Blockchain”** yang mengkombinasikan teknologi otentikasi dua faktor dengan teknologi blockchain, dimana blockchain dapat menyimpan data tidak secara terpusat (desentralisasi) dengan sistem hashing yang menyebabkan seorang *attacker* sulit dan hampir tidak mungkin untuk melakukan perubahan data yang ada pada suatu block pada arsitektur blockchain.

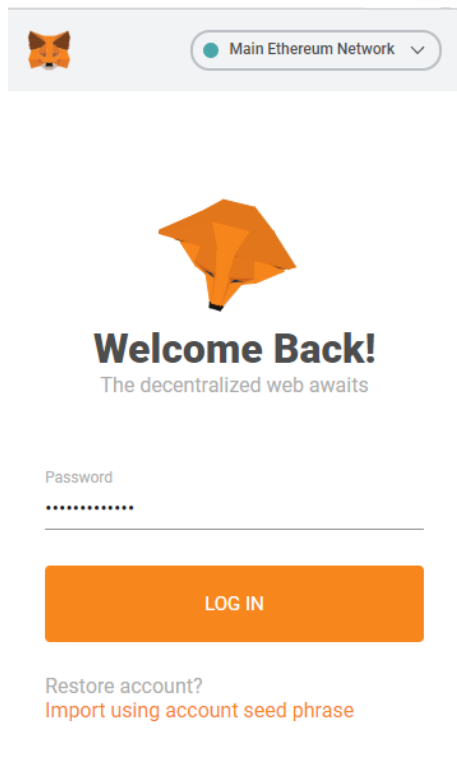
**Untuk menjalankan aplikasi ini, ada beberapa hal yang harus dipersiapkan yaitu:**

### 1. Download dan install aplikasi pendukung:

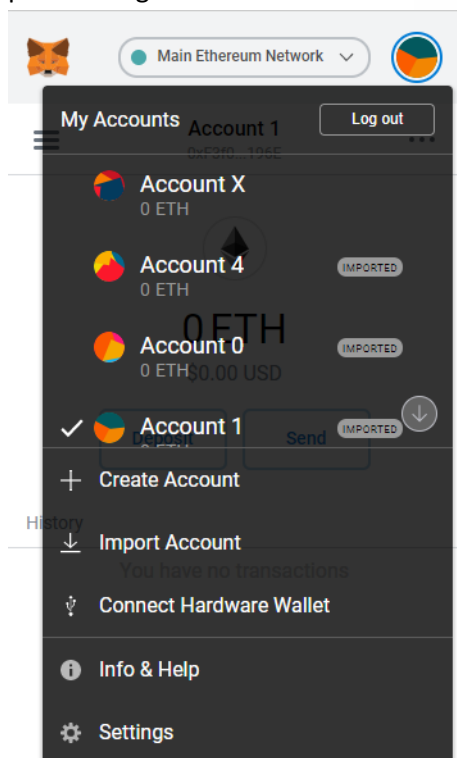
- Download personal Ethereum Blockchain yaitu Ganache pada link berikut <https://www.trufflesuite.com/ganache> kemudian install.
- Download visual studio code dari link berikut <https://code.visualstudio.com/download> kemudian install.
- Download google chrome extension bernama Metamask pada chrome web store kemudian install.

### 2. Menghubungkan Ganache dengan Metamask:

- Buka Ganache yang sudah diinstall. Kemudian pilih “NEW WORKSPACE”.
  - a. Pada Tab Workspace, isi workspace name dengan “2fablockchain”. Kemudian klik tombol “add project” untuk menambahkan file truffle.js yang ada dalam folder 2fablockchain/solidity/truffle.js
  - b. Kemudian pada Tab Server, ubah port number menjadi 8545. Setelah itu klik tombol “Save Workspace”
- Login ke metamask dengan password yang sudah didaftarkan

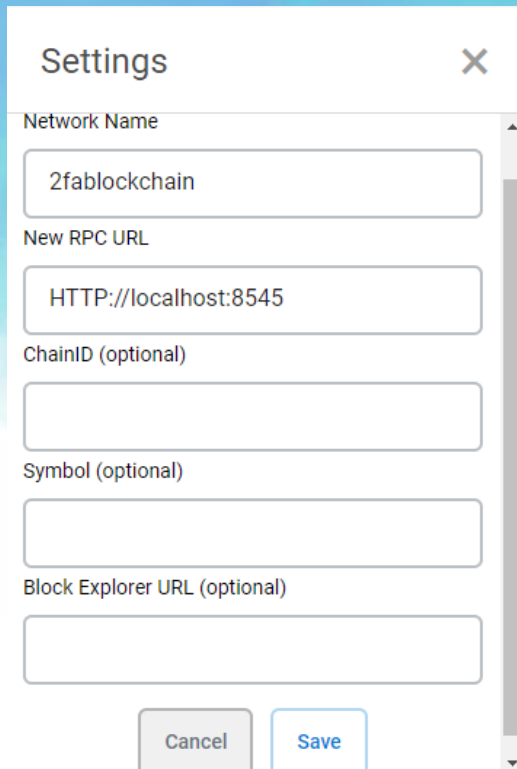


- Setelah berhasil masuk ke Metamask, klik gambar akun di sebelah kanan atas, kemudian pilih Settings



- Pilih Networks → Add network

- Masukkan network name kemudian isi bagian “New RPC URL” sesuai dengan RPC Server yang tertulis di Ganache seperti gambar dibawah ini. Kemudian klik save.



Settings

Network Name

2fablockchain

New RPC URL

HTTP://localhost:8545

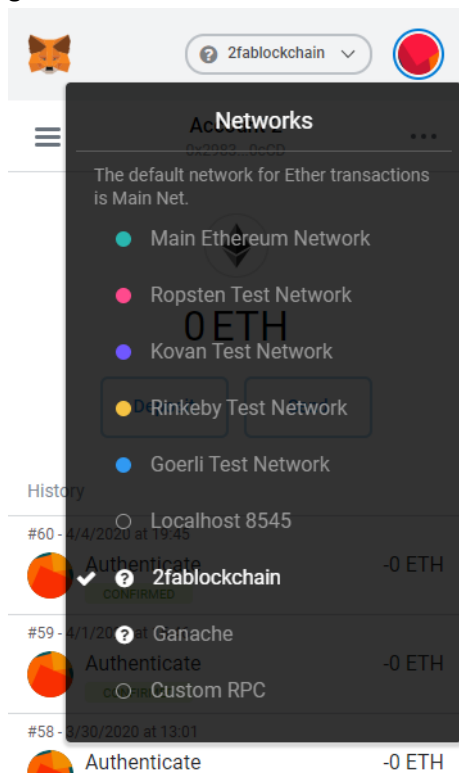
ChainID (optional)

Symbol (optional)

Block Explorer URL (optional)

Cancel Save

- Kemudian ganti network dari Main Ethereum Network menjadi “2fablockchain” seperti gambar dibawah



### 3. Import akun dengan private key Ganache ke Metamask

- Buka Ganache, pilih salah address dengan index 1 yang akan di import ke metamask kemudian klik icon kunci di sebelah kanan

ADDRESS	BALANCE	TX COUNT	INDEX	
0x26E4255c3b46a3deD54DA50BC3d4213e91Bc238e	100.00 ETH	0	1	

- Kemudian akan muncul account information, yang harus anda lakukan adalah copy private key yang muncul

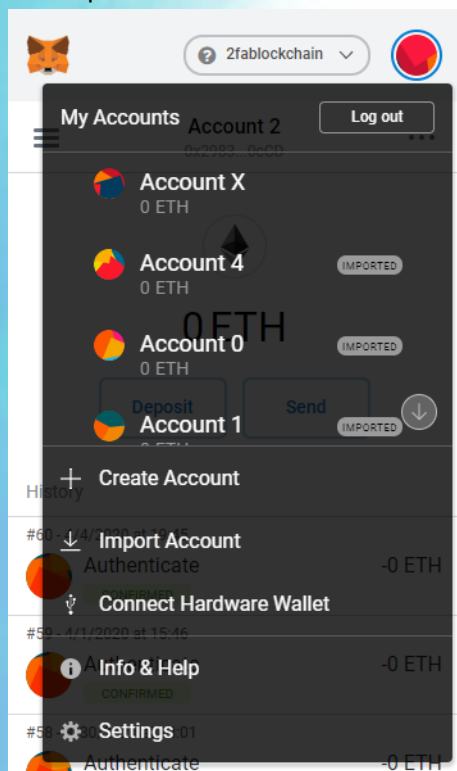
**ACCOUNT INFORMATION**

**ACCOUNT ADDRESS**  
0x26E4255c3b46a3deD54DA50BC3d4213e91Bc238e

**PRIVATE KEY**  
5248504ed43c4c4229c5e54fd8c9910bc7fadc674c93072dac745f71c6e7d3f1  
Do not use this private key on a public blockchain; use it for development purposes only!

DONE

- Kemudian buka metamask dan klik gambar akun di sebelah kanan atas, kemudian pilih Import Account



- Kemudian pilih type private key, dan paste private key pada kolom yang disediakan. Kemudian klik import

Create **Import** Connect

Imported accounts will not be associated with your originally created MetaMask account seedphrase. Learn more about imported accounts [here](#)

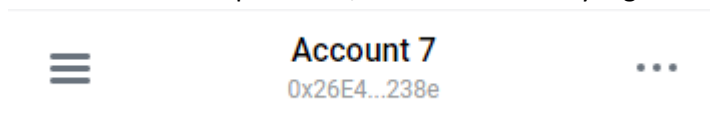
Select Type Private Key ▼

Paste your private key string here:

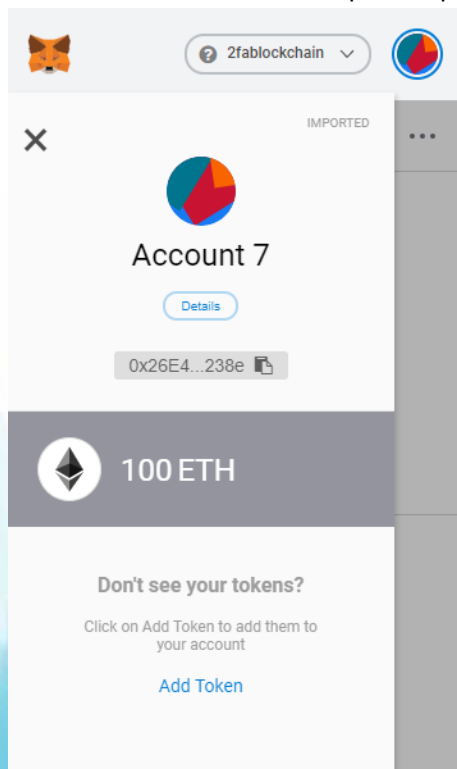
.....

Cancel Import

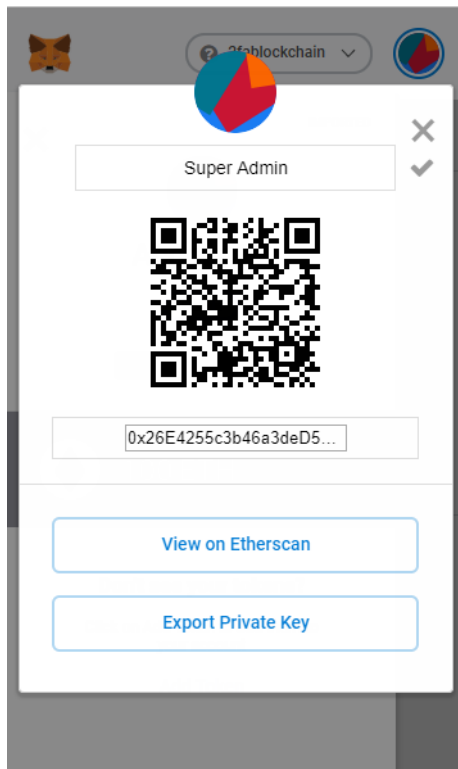
- Setelah berhasil import akun, klik tombol menu yang berada di sebelah kiri



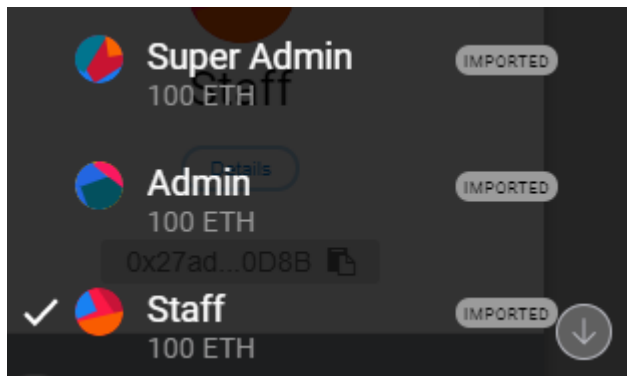
- Setelah itu akan muncul tampilan seperti dibawah ini. Kemudian pilih “Details”



- Kemudian ubah nama akun menjadi “Super Admin”



- Kemudian ulangi langkah-langkah diatas untuk mengimport address index 2 sebagai akun admin, dan address index 3 sebagai akun staff seperti gambar dibawah ini



#### 4. Install library yang dibutuhkan untuk menjalankan aplikasi ini:

Beberapa library yang harus di install sebelum menggunakan aplikasi ini yaitu express, body-parser, cookie-session, web3, truffle, nodemon dan crypto dengan cara:

- Buka 2fablockchain dengan menggunakan visual studio code dengan cara klik Tab File → Open Folder → 2fablockchain
- Ketik “npm install <nama\_library>” pada console visual studio code  
Contoh: npm install body-parser

#### 5. Deploy smart contract dengan menggunakan truffle



- Dari folder 2fablockchain, masuk ke folder solidity dengan mengetikkan “cd solidity” pada console visual studio




```
C:\2fablockchain>cd solidity
```

- Kemudian ketik “truffle deploy --network development” atau “truffle.cmd deploy --network development” pada visual studio console

```
D:\2fablockchain\solidity>truffle.cmd deploy --network development
Compiling .\contracts\Migrations.sol...
Compiling .\contracts\Registration.sol...
Compiling .\contracts\Tufa.sol...
```

- Setelah berhasil deploy, buka file solidity/build/Tufa.json kemudian copy abi dan transaction address (berada di paling bawah file) ke file appweb/public/contract.js pada “const contractAddress” dan “const contractABI”
- Buka file solidity/build/Registration.json kemudian copy abi dan transaction address (berada di paling bawah file) ke file appweb/public/contract.js dan appweb/api/controllers/controller.js pada “const AddressRegistration” dan “const ABIRegistration”
- Kemudian buka file appweb/api/controllers/controller.js dan ganti address gambar dibawah ini sesuai dengan address pada index 1, 2, dan 3 pada Ganache

```
const addressSuperAdmin = "0x26E4255c3b46a3deD54DA50BC3d4213e91Bc238e";
const addressAdmin = "0x8Dab0Bd9E82975474544893a9362e246ae8625EC";
const addressStaff = "0x27ad543DaEC63a4822ADf4B49c0cA321670A0D8B";
```

ADDRESS 0x26E4255c3b46a3deD54DA50BC3d4213e91Bc238e	BALANCE 100.00 ETH	TX COUNT 0	INDEX 1	
ADDRESS 0x8Dab0Bd9E82975474544893a9362e246ae8625EC	BALANCE 100.00 ETH	TX COUNT 0	INDEX 2	
ADDRESS 0x27ad543DaEC63a4822ADf4B49c0cA321670A0D8B	BALANCE 100.00 ETH	TX COUNT 0	INDEX 3	

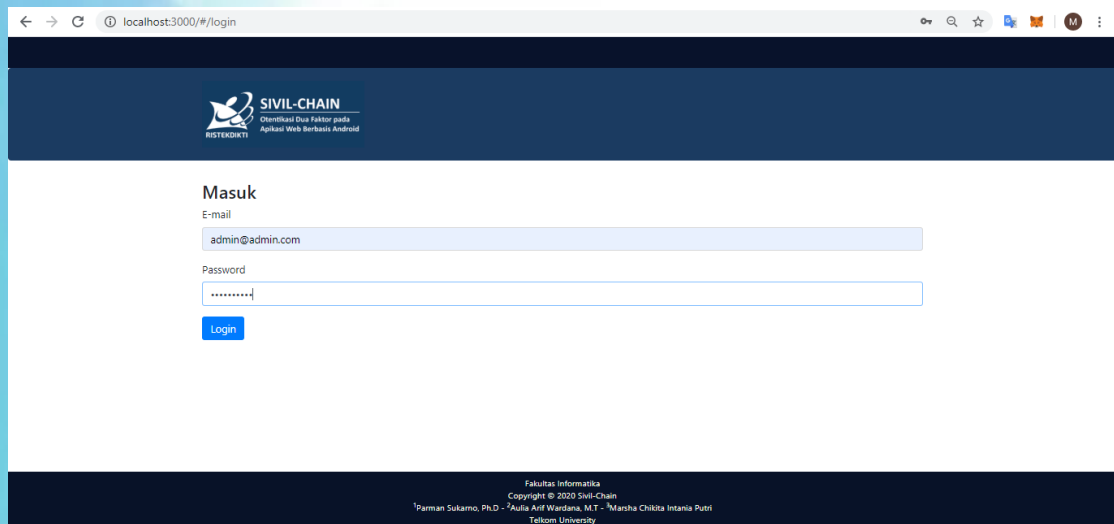
Setelah semua tahap persiapan selesai dilakukan, maka aplikasi sudah dapat dijalankan dengan cara:

- Masuk ke folder appweb dengan cara ketik “cd appweb” pada visual studio console
- Untuk menjalankan server, ketik “node server.js” atau “nodemon server.js” pada visual studio console

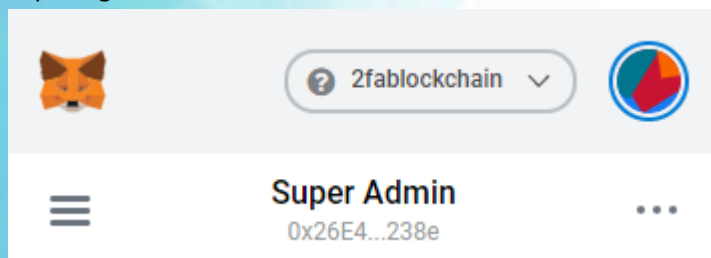
```
[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
Port Server: 3000
```

- Setelah server berjalan, buka browser kemudian ketik “localhost:3000” kemudian akan muncul tampilan aplikasi web otentikasi dua faktor berbasis blockchain

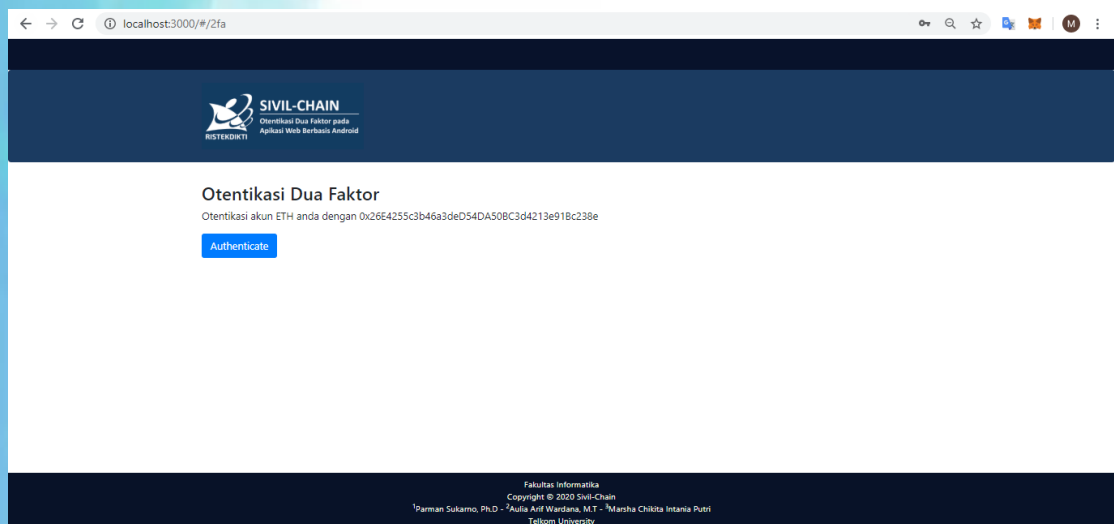




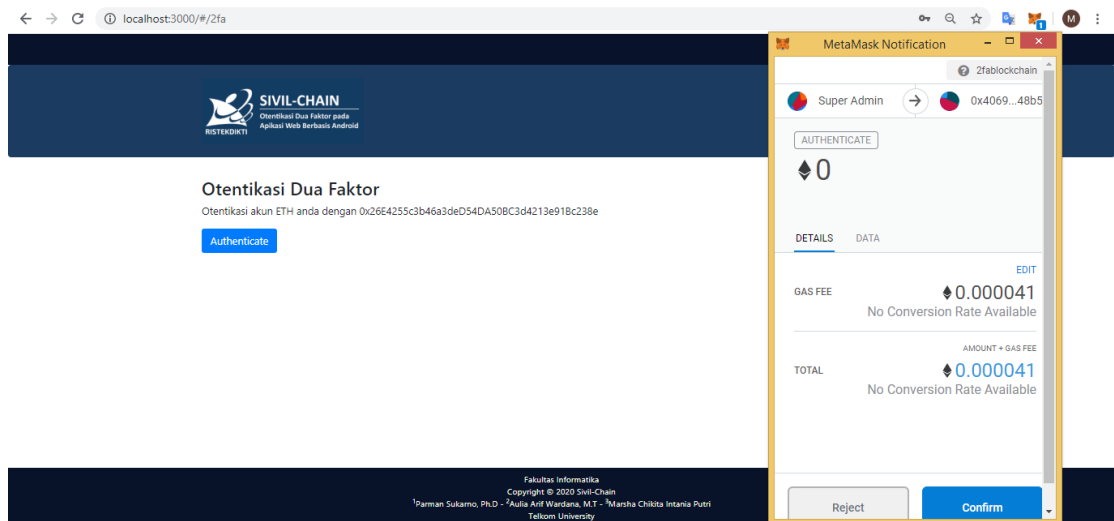
- d. Untuk masuk pertama kali, ketikkan “[admin@admin.com](mailto:admin@admin.com)” pada kolom input email dan “superAdm1n” sebagai password. Pastikan akun metamask berada pada akun Super Admin seperti gambar dibawah



- e. Setelah memasukkan email dan password, user akan diarahkan pada laman verifikasi dua faktor. Klik tombol “Authenticate” untuk melakukan verifikasi akun



- f. Setelah itu akan muncul popup window dari metamask seperti gambar dibawah ini. Kemudian klik tombol “Confirm”



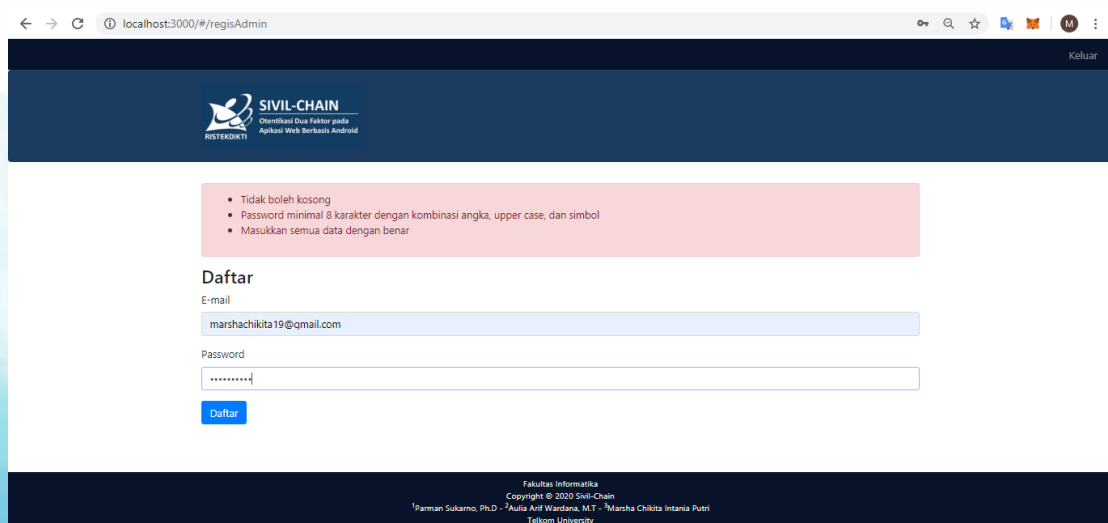
- g. Jika verifikasi berhasil, maka user akan diarahkan menuju laman super admin. Super admin memiliki akses untuk dapat mendaftarkan akun super admin yang lain sekaligus juga dapat mendaftarkan akun admin biasa



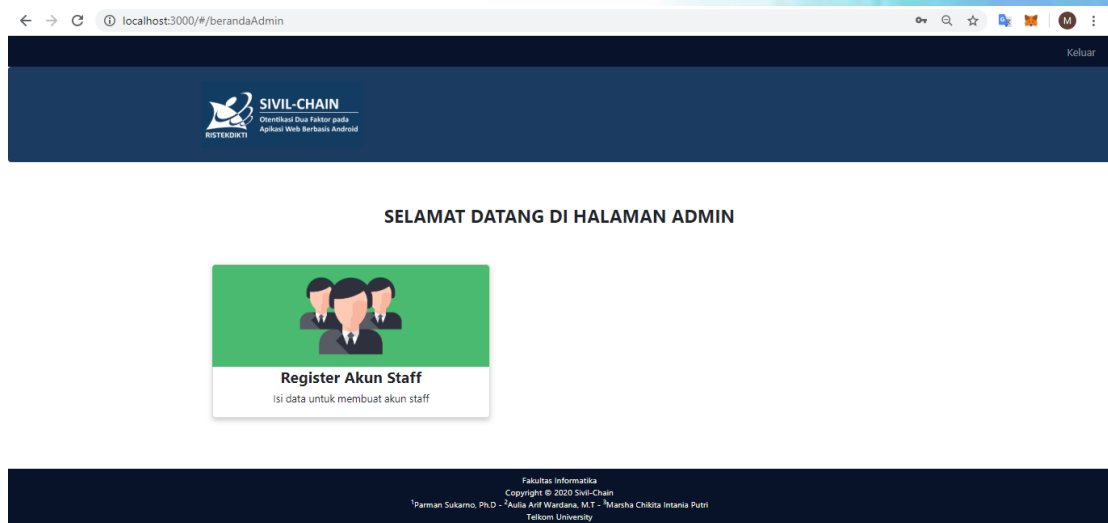
#### SELAMAT DATANG DI HALAMAN SUPER ADMIN



- h. Menu register admin dapat dilihat pada gambar dibawah ini. Klik tombol “Daftar” untuk membuat akun admin



- i. Kemudian cobalah login kembali dengan menggunakan akun admin. Pada halaman admin terdapat menu untuk mendaftarkan akun staff seperti gambar dibawah ini



## Pseudocode Otentikasi Dua Faktor pada Aplikasi Web Berbasis Blockchain

### 1. Registrasi Akun

```
If(email == null){send error message}
else if(!email.match(email_format)){send error message}

If(password == null){send error message}
else if(password.length < 8){send error message}
else if(!password.match(password_pattern)){send error message}

email_hash = "0x" + createHash(email)
pass_hash = "0x" + createHash(password)
role
eth_addr

web3.eth.getAccounts{
  akun = result[0]

  regisContract.getData.call(email_hash){

    if(result[0].toString() == email_hash.toString()){alert("Email sudah terdaftar")}
    else{
      regisContract.addUserData(email_hash, pass_hash, eth_addr,role){
        from: akun,
        gas: 300000
      }
    }
  }
}
```

```
}
```

## 2. Login

```
email_hash = "0x" + createHash(email)
pass_hash = "0x" + createHash(password)

if(email == "admin@admin.com" && password == "superAdm1n"){
  const user = {
    "email" : email,
    "password" : password,
    "address" : addressSuperAdmin,
    "role" : 1
  }
  user.token = nextToken(user.token)
  session.user = new SessionUser(user, tufa.account)
}else{
  regisContract.getData.call(email_hash){
    if(result[0].toString() == email_hash.toString() && result[1].toString() ==
pass_hash.toString()){
      const user = {
        "email" : result[0],
        "password" : result[1],
        "address" : result[2],
        "role" : parseInt(result[3])
      }
      user.token = nextToken(user.token)
      session.user = new SessionUser(user, tufa.account)

    }else{ send error message}

  }
}
```

## 3. 2FA

```
const user = session.user
tufa.getAuthenticationToken(user.address){
  if(token.toString() == user.token.toString()){
    session.user.tokenVerification = true;
  }else{send(error: Token tidak cocok!)}
}
```