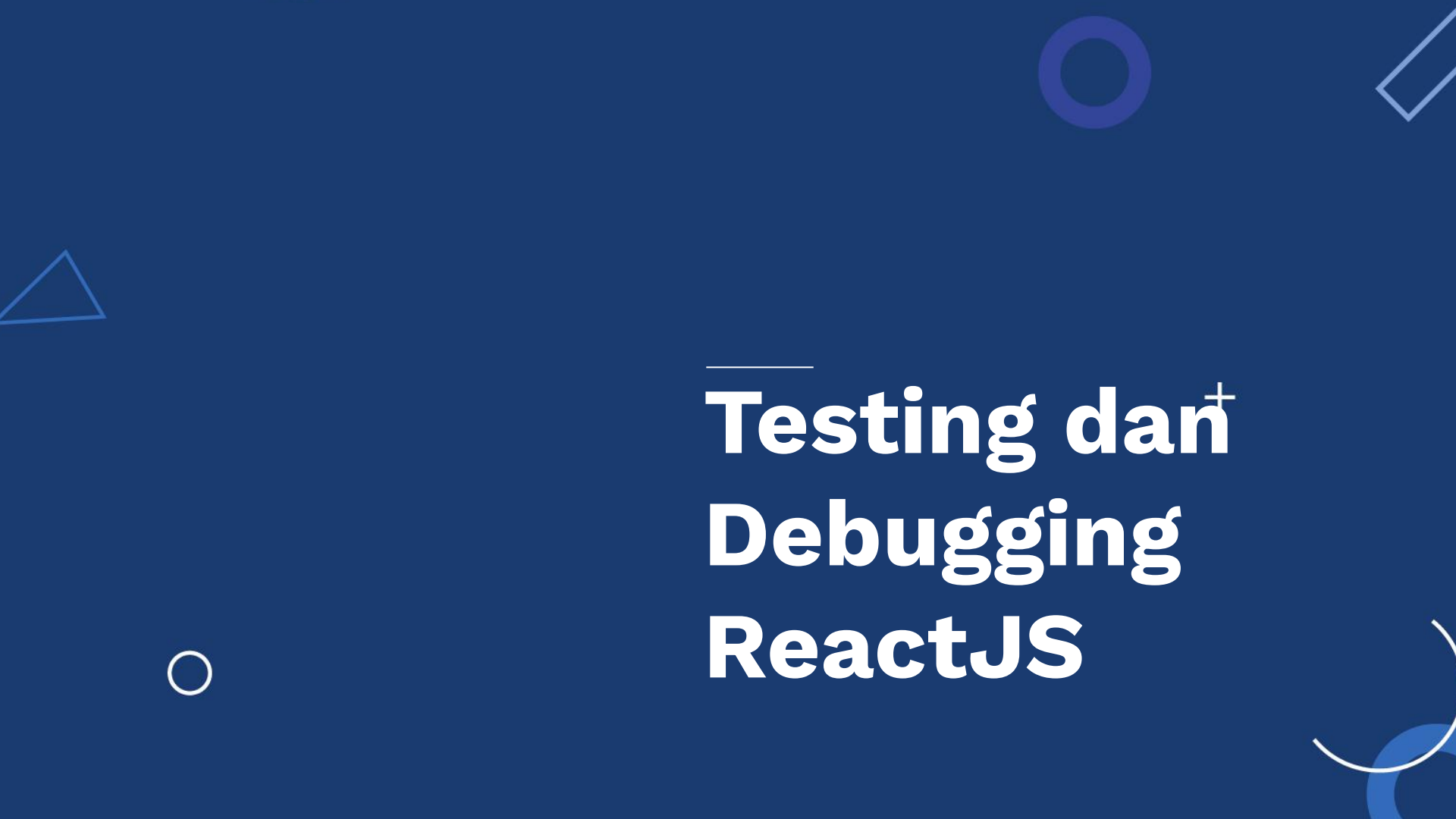




Front End React Development sesi 14



Testing dan⁺ Debugging ReactJS

Sesi 14 | Testing & Debugging ReactJS

INTRO

Dengan mengadaptasi banyak sifat dari *functional programming*, React semakin mudah untuk dites. Dengan bantuan testing library **Jest** dan **Enzyme** saat ini melakukan testing di sisi *front-end* sudah tidak sulit lagi.

Testing kode yang sudah kita buat merupakan hal yang bisa dibilang wajib. Tujuannya agar kode yang kita hasilkan selalu berkualitas tinggi dan merupakan sebuah garansi bahwa apa yang sudah kita kerjakan sesuai harapan, dan tentunya jumlah *bugs* atau kesalahan yang sangat minimal.

Saat ini, developer selain *ngoding* juga dituntut untuk juga bisa melakukan testing terhadap *codingan* kita sendiri.

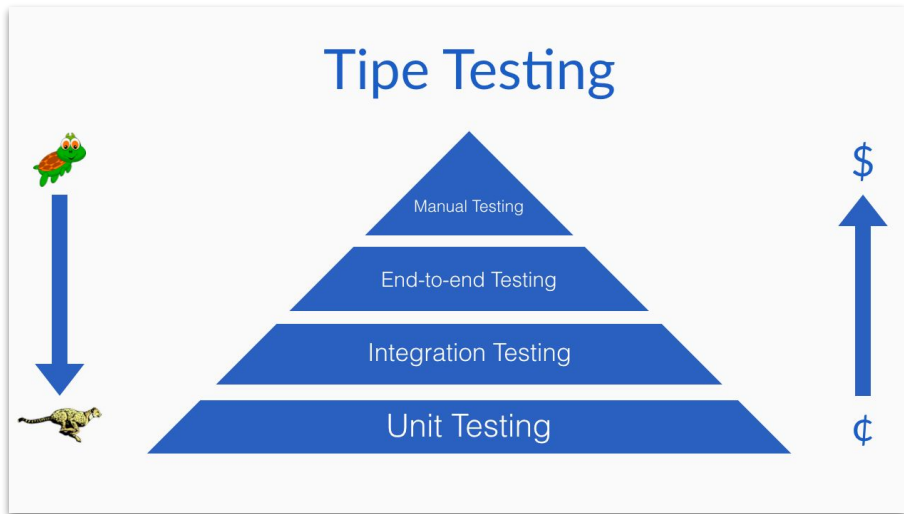


Ini gurita apaan ya guys ? Hmm,,, kita cek penjelasannya di slide selanjutnya



HACKTIV8

INTRO



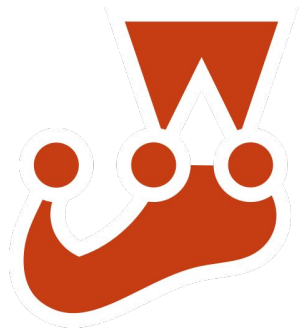
- **Unit:** Testing terhadap fungsi tunggal atau single functionality. Misalnya ada sebuah fungsi yang tugasnya mengambil data dari server. Yang dites Hanya satu fungsi itu saja. Tampilan dan lain sebagainya tidak dites dalam hal ini.
- **Integration:** Beberapa fungsi terkait dites secara bersamaan. Misalnya ada beberapa fungsi yang tugasnya mengambil data dari server dan menampilkan ke layar. Yang dites ya mulai dari ambil data hingga tampilan
- **End to end:** Memastikan seluruh fungsi bekerja dari perspektif user. Jadi ngetest-nya dengan button click, apa yang terjadi dan seterusnya.

Dan seperti yang terlihat di ilustrasi diatas, unit testing merupakan jenis test yang paling cepat dijalankan. Dan semakin keatas proses eksekusi akan semakin lambat. Unit test juga merupakan solusi testing yang paling murah dibandingkan dengan solusi testing lain. Tentu untuk membayar orang melakukan testing seperti QA jauh lebih mahal daripada membuat unit test kan?!



Sesi 14 | Testing & Debugging ReactJS

PACKAGE / LIBRARY



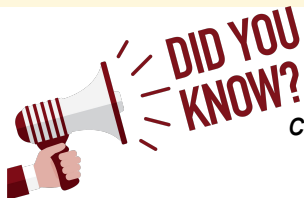
JEST

... adalah sebuah testing framework dalam bahasa JavaScript yang biasanya digunakan untuk melakukan testing di sisi FE. Dibuat dan di design oleh Christoph Nakazawa, dan di maintain oleh Facebook, Inc, dan berfokus pada kemudahan dan dukungan untuk aplikasi berbasis web yang cukup besar. Dapat dipakai untuk aplikasi yang menggunakan Babel, TypeScript, Node.js, React, Angular, Vue.js, dan Svelte



React Testing Library (RTL)

... adalah utility yang bisa kita pakai untuk melakukan testing terhadap aplikasi React. Merupakan sebuah solusi yang sangat ringan (light-weight) untuk melakukan testing terhadap komponen React. Menyediakan utility yang ringan di atas **react-dom** dan **react-dom/test-utils**, dengan cara yang memberikan dorongan / motivasi untuk bisa melakukan praktek testing yang lebih baik.



create-react-app telah memastikan bahwa Jest dan RTL sudah auto terinstall



HACKTIV8

TESTING DI REACT

Dalam konteks React, apa saja aspek yang bisa kita test? Beberapa hal dibawah ini dapat kita konsiderasi untuk dilakukan testing.

1. Test render: memastikan komponen dirender tanpa error
2. Test output: memastikan komponen dirender dengan text, atau output yang diinginkan
3. Test props, state, event, dan lain sebagainya

Dan untuk melakukan testing di React, kita bisa menggunakan library yang namanya Jest. Secara fungsi Jest adalah sebuah tools yang cukup lengkap. Ada test runner, assert library dan mocking library, hingga test covearge sehingga kita cukup menggunakan satu library ini saja untuk dapat melakukan testing yang komprehensif terhadap aplikasi React kita.

RTL berperan dalam berinteraksi dengan aplikasi. Bisa dibilang, RTL menjadi “pengganti” tangan kita dalam melakukan beberapa hal, seperti akses halaman, akses komponen, trigger klik button, hingga menunggu async proses selesai.

Jadi, **RTL** yang melakukan “interaksi”
dan **JEST** yang melakukan pengecekan terhadap hasil “interaksi” nya

Sebagai alternatif, kita juga bisa menggunakan **Mocha** sebagai test runner dan **Chai** sebagai assertion library-nya

Sesi 14 | Testing & Debugging ReactJS

BEBERAPA FUNGSI YANG AKAN DIPAKAI

JEST :

- **test** : untuk memberikan scope / judul dari rangkaian test yang akan dijalankan
- **expect** : untuk binding expektasi terhadap suatu DOM yang kita akan test, dan akan di chain dengan `toBe<something>`
- **toBe<something>** : untuk memeriksa expektasi kita, ada puluhan `toBe` yang bisa digunakan, silakan cek di dokumentasinya

Lebih lengkap bisa dibaca di <https://jestjs.io/>

React Testing Library :

- **render** : untuk render sebuah component
- **debug** : untuk menampilkan seluruh DOM yang ter-render, merupakan shortcut dari `console.log(prettyDOM(baseElement))`
- **fireEvent** : untuk men-trigger sebuah DOM event
- **waitFor** : untuk menunggu sebuah fungsi berjalan hingga selesai, biasanya untuk proses async

Lebih lengkap bisa dibaca di <https://testing-library.com/>



Sesi 14 | Testing & Debugging ReactJS

LATIHAN TESTING

Initiate sebuah aplikasi baru yang memiliki 2 buah halaman : Home (“/”) dan Users (“/users”). Gunakan react-router untuk berpindah antar halaman. Users akan fetch data dari API jsonplaceholder. Buatlah tampilan seperti berikut :

Testing example

Why do we need Test ?

To guarantee our code quality. And to show the people that we also care about it and wish to give something that already proven, at least by ourself

Users List

© 2021

Testing example

Back to Home

ID	Name	Username	Email
1	Leanne Graham	Bret	Sincere@april.biz
2	Ervin Howell	Antonette	Shanna@melissa.tv
3	Clementine Bauch	Samantha	Nathan@yesenia.net
4	Patricia Lebsack	Karianne	Julianne.OConner@kory.org
5	Chelsey Dietrich	Kamren	Lucio_Hettinger@annie.ca
6	Mrs. Dennis Schulist	Leopoldo_Corkery	Karley_Dach@jasper.info
7	Kurtis Weissnat	Elwyn.Skiles	Telly.Hoeger@billy.biz
8	Nicholas Runoldsdottir V	Maxime_Nienow	Sherwood@rosamond.me
9	Glenna Reichert	Delphine	Chaim_McDermott@dana.io
10	Clementina DuBuque	Moriah.Stanton	Rey.Padberg@karina.biz

© 2021

Sesi 14 | Testing & Debugging ReactJS

LATIHAN TESTING

Update file **package.json** menjadi seperti di bawah ini. Lihat pada baris 19

Tujuan dari perubahan ini adalah agar kita dapat melihat berapa banyak kode kita yang sudah kita cover dengan test. Statistik yang didapatkan sangatlah membantu kita. Dan hal ini adalah salah satu best practice jika kita hendak menjadi seorang programmer yang baik

```
16   "scripts": {  
17     "start": "react-scripts start",  
18     "build": "react-scripts build",  
19     "test": "react-scripts test --coverage",  
20     "eject": "react-scripts eject"  
21   },
```



Sesi 14 | Testing & Debugging ReactJS

LATIHAN TESTING

Update file **src/App.test.js** menjadi seperti di bawah ini dan jalankan dengan perintah

```
> npm run test
```

```
1 import { render, screen, fireEvent } from '@testing-library/react';
2 import App from '../App';
3
4 test('Render halaman home dengan benar', () => {
5   render(<App />);
6
7   // mencari text tertentu
8   const title = screen.getByText(/why do we need test ?/i);
9   expect(title).toBeInTheDocument();
10
11   // mencari button tertentu
12   // pastikan button kita memiliki role="button"
13   const btn = screen.getByRole('button');
14   expect(btn).toBeInTheDocument();
15   expect(btn).toHaveTextContent('Users List');
16
17   // klik tombol Users List
18   fireEvent.click(btn)
19   // mencari tombol tertentu
20   // pada halaman yang muncul karena klik tombol
21   // pastikan button kita memiliki role="button"
22   const btn2 = screen.getByRole('button');
23   expect(btn2).toBeInTheDocument();
24   expect(btn2).toHaveTextContent('Back to Home');
25 });
```

PASS src/App.test.js

✓ Render halaman home dengan benar (246 ms)

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	35.29	100	44.44	35.29	
src	11.11	100	33.33	11.11	
App.js	100	100	100	100	
Users.js	0	100	0	0	4-19
src/pages	62.5	100	50	62.5	
Home.js	100	100	100	100	
Users.js	57.14	100	40	57.14	9-10,29

Test Suites: 1 passed, 1 total

Tests: 1 passed, 1 total

Snapshots: 0 total

Time: 6.944 s

Ran all test suites related to changed files.



HACKTIV8

Sesi 14 | Testing & Debugging ReactJS

LATIHAN TESTING

Sekarang kita akan test component Users. Buat file **src/Users.test.js**, isi seperti di bawah ini, dan jalankan test nya

```
1 import { render, screen } from '@testing-library/react';
2 import Users from './pages/Users';
3
4 test('Render halaman Users dengan benar', () => {
5   render(<Users />);
6
7   // mencari tombol tertentu
8   // pastikan button kita memiliki role="button"
9   const btn = screen.getByRole('button');
10  expect(btn).toBeInTheDocument();
11  expect(btn).toHaveTextContent('Back to Home');
12 });
```

PASS

src/App.test.js

PASS

src/Users.test.js

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	66.67	100	57.14	66.67	
src	100	100	100	100	
App.js	100	100	100	100	
src/pages	62.5	100	50	62.5	
Home.js	100	100	100	100	
Users.js	57.14	100	40	57.14	9-10,29

Test Suites:

2 passed, 2 total

Tests:

2 passed, 2 total

Snapshots:

0 total

Time:

6.748 s

Ran all test suites related to changed files.



Sesi 14 | Testing & Debugging ReactJS

LATIHAN TESTING

Wait, kita belum lakukan test untuk async process. Edit file **src/Users.test.js**, isi seperti di bawah ini, dan jalankan test nya

Dari hasil test ini dapat dilihat bahwa coverage test kita terhadap code yang kita kembangkan sudah sangat baik. Dan,,, kolom **Uncovered Line** isinya nya kosong.. HORE !!!

```
1 import { render, screen, waitFor } from '@testing-library/react';
2 import Users from './pages/Users';
3
4 test('Render halaman Users dengan benar', async () => {
5   render(<Users />);
6
7   // mencari tombol tertentu
8   // pastikan button kita memiliki role="button"
9   const btn = screen.getByRole('button');
10  expect(btn).toBeInTheDocument();
11  expect(btn).toHaveTextContent('Back to Home');
12
13  // menunggu proses fetch user dari API
14  // kita kasih waktu sebanyak 5 detik
15  // kemudian mencari text sebuah nama dari API
16  const user = await waitFor(() => {
17    return screen.findByText('Leanne Graham')
18  }, { timeout: 5000 })
19  expect(user).toBeInTheDocument()
20 });
```

```
PASS src/App.test.js
PASS src/Users.test.js (5.001 s)
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	100	100	100	100	
src	100	100	100	100	
App.js	100	100	100	100	
src/pages	100	100	100	100	
Home.js	100	100	100	100	
Users.js	100	100	100	100	

```
Test Suites: 2 passed, 2 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 8.663 s
Ran all test suites related to changed files.
```



LATIHAN TESTING




Yang sudah kita lakukan di sini masih hal-hal dasar ya gaes. Real world practicenya, kamu harus bisa menganalisa core feature dari aplikasi yang kamu kembangkan dan membangun test dari situ.

Sudah banyak cerita penyesalan di luar sana dari rekan-rekan yang lebih senior, terutama karena menyepelekan untuk membuat test. Unpredictable things raised, dan ini bisa berakibat buruk pada performance bahkan bisa berakhir dengan pemecatan.

Banyaklah berlatih untuk menulis test dengan baik. Selain itu, menulis test adalah part of best practice yang dilakukan para programmer kakap di luar sana. Be one of them !





Deployment⁺ ReactJS Online

Sesi 14 | Deployment ReactJS Online

INTRO + NETLIFY

Sekarang saatnya deploy dan push aplikasi kita ke production. Ada banyak sekali alternatif mulai dari AWS S3, Firebase, Surge, Now hingga Netlify. Kita akan coba yang terakhir, yaitu Netlify. Kita akan gunakan aplikasi yang tadi kita bangun beserta test nya untuk kita deploy

1. Build the app

Di sini kita akan melakukan build atau mempersiapkan aplikasi kita agar siap untuk di deploy. FYI, kita tidak akan langsung deploy kode kita secara mentah.

```
> npm run build
```

```
~ npm run build

> new-app@0.1.0 build
> react-scripts build

Creating an optimized production build...
Compiled successfully.

File sizes after gzip:

 49.74 KB  build/static/js/2.d5ddb2d.chunk.js
 22.53 KB  build/static/css/2.4be38407.chunk.css
  1.63 KB  build/static/js/3.9268c8f1.chunk.js
  1.17 KB  build/static/js/runtime-main.76da63f3.js
  1.08 KB  build/static/js/main.4cdd2430.chunk.js
   278 B   build/static/css/main.6dea0f05.chunk.css

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build

Find out more about deployment here:

  https://cra.link/deployment
```



Sesi 14 | Deployment ReactJS Online

```

  build
  static
    css
      # 2.4be38407.chunk.css
      # 2.4be38407.chunk.css.map
      # main.6dea0f05.chunk.css
      # main.6dea0f05.chunk.css.map
    js
      JS 2.d5ddbb2d.chunk.js
      2.d5ddbb2d.chunk.js.LICENSE.txt
      JS 2.d5ddbb2d.chunk.js.map
      JS 3.9268c8f1.chunk.js
      JS 3.9268c8f1.chunk.js.map
      JS main.4cdd2430.chunk.js
      JS main.4cdd2430.chunk.js.map
      JS runtime-main.76da63f3.js
      JS runtime-main.76da63f3.js.map
  {} asset-manifest.json
  ★ favicon.ico
  <> index.html
  🖼 logo192.png
  🖼 logo512.png
  {} manifest.json
  ≡ robots.txt
```

NETLIFY

Berikut struktur folder dan file-file yang terbentuk. Selanjutnya, Sign-up di website Netlify, kemudian kita akan deploy aplikasi kita ke Netlify.

1. Install netlify-cli

```
> npm install -g netlify-cli
```

2. Run deployment

```
> netlify deploy
```

Setelah menjalankan perintah ini, akan ada wizard yang harus kita lalui. Pertama, kita akan diminta login ke Netlify. Kemudian, akan ada pertanyaan seperti di samping ini, pilihlah **Create & configure a new site** dengan key arah dan press Enter

```

~ netlify deploy
Logging into your Netlify account...
Opening https://app.netlify.com/authorize?response_type=...

You are now logged into your Netlify account!

Run netlify status for account details

To see all available commands run: netlify help

This folder isn't linked to a site yet
? What would you like to do?
> Link this directory to an existing site
+ Create & configure a new site
```


NETLIFY

3. Pilih team

Kalau kita masih sendiri, akan muncul user id kita sebagai nama team. Pilih nama tersebut, kemudian press Enter

```
This folder isn't linked to a site yet
? What would you like to do? + Create & configure a new site
? Team:
> [redacted]'s team
```

4. Isi Site name

Bisa di biarkan kosong, atau diisi sesuai keinginan. Jika nama sudah terpakai, akan ada warning

```
Choose a unique site name (e.g. isnt-ffitransyah-awesome.netlify.app) or
leave it blank for a random name. You can update the site name later.
? Site name (optional): app1
> Warning: app1.netlify.app already exists. Please try a different slug.
Choose a unique site name (e.g. isnt-ffitransyah-awesome.netlify.app) or
leave it blank for a random name. You can update the site name later.
? Site name (optional): app1-kampus-merdeka

Site Created

Admin URL: https://app.netlify.com/sites/app1-kampus-merdeka
URL:      https://app1-kampus-merdeka.netlify.app
Site ID:  e623926c-d68f-42de-8790-87c805f211e7
```



NETLIFY

3. Pilih team

Kalau kita masih sendiri, akan muncul user id kita sebagai nama team. Pilih nama tersebut, kemudian press Enter

```
This folder isn't linked to a site yet
? What would you like to do? + Create & configure a new site
? Team:
> [redacted]'s team
```

4. Isi Site name

Bisa di biarkan kosong, atau diisi sesuai keinginan. Jika nama sudah terpakai, akan ada warning

```
Choose a unique site name (e.g. isnt-ffitransyah-awesome.netlify.app) or
leave it blank for a random name. You can update the site name later.
? Site name (optional): app1
> Warning: app1.netlify.app already exists. Please try a different slug.
Choose a unique site name (e.g. isnt-ffitransyah-awesome.netlify.app) or
leave it blank for a random name. You can update the site name later.
? Site name (optional): app1-kampus-merdeka

Site Created

Admin URL: https://app.netlify.com/sites/app1-kampus-merdeka
URL:      https://app1-kampus-merdeka.netlify.app
Site ID:  e623926c-d68f-42de-8790-87c805f211e7
```



Sesi 14 | Deployment ReactJS Online

NETLIFY

5. Pilih directory yang akan di publish

Dalam kasus kita adalah **build**

```
Please provide a publish directory (e.g. "public" or "dist" or "."):
[redacted]/new-app
? Publish directory [redacted] ✓
new-app/build
Deploy path: [redacted]/new-app
/build
Deploying to draft URL...
✓ Finished hashing 20 files
✓ CDN requesting 10 files
✓ Finished uploading 10 assets
✓ Deploy is live!

Logs: https://app.netlify.com/sites/app1-kampus-merdeka/depl
oys/60e815acf6ac3399b4dcacbc
Website Draft URL: https://60e815acf6ac3399b4dcacbc--app1-kampus-merdeka.
netlify.app

If everything looks good on your draft URL, deploy it to your main site U
RL with the --prod flag.
netlify deploy --prod
```

Berikut adalah hasil deployment nya. Sudah bukan localhost:3000. **TAPI**, ini masih draft.

60e815acf6ac3399b4dcacbc--app1-kampus-merdeka.netlify.app

Testing example

Why do we need Test ?

To guarantee our code quality. And to show the people that we also care about it and wish to give something that already proven, at least by ourself

Users List

© 2021



HACKTIV8

Sesi 14 | Deployment ReactJS Online

NETLIFY

6. Full deployment

```
> netlify deploy --prod
```

... dan kembali pilih folder build

```
~ netlify deploy --prod
Please provide a publish directory (e.g. "public" or "dist" or "."):
/[redacted]/new-app
? Publish directory /[redacted]/
new-app/build
Deploy path: /[redacted]/new-app
/build
Deploying to main site URL...
✓ Finished hashing 20 files
✓ CDN requesting 0 files
✓ Finished uploading 0 assets
✓ Deploy is live!

Logs: https://app.netlify.com/sites/app1-kampus-merdeka/depl
oys/60e81b04922616a1c91aee5d
Unique Deploy URL: https://60e81b04922616a1c91aee5d--app1-kampus-merdeka.
netlify.app
Website URL: https://app1-kampus-merdeka.netlify.app
```

Dan boom...
aplikasi pertamamu sudah mengudara (online)

app1-kampus-merdeka.netlify.app

Testing example

Why do we need Test ?

To guarantee our code quality. And to show the people that we also care about it and wish to give something that already proven, at least by ourself

Users List

© 2021



HACKTIV8

Sesi 14 | Deployment ReactJS Online

WHAT'S NEXT ?

Ada beberapa tempat lain yang bisa temen-temen explorasi untuk deployment. Salah satunya HEROKU

Heroku menggunakan technology git untuk deployment nya. Secara singkat, berikut cara kita deploy ke Heroku :



Via CLI :

Register Heroku account (<https://signup.heroku.com/>)

Install depend CLI Heroku first! Follow this link
<https://devcenter.heroku.com/articles/heroku-cli>

```
> heroku login
> git init
> git add .
> git commit -m "message_commit"
> heroku create nama_deploy
> git push heroku master
```

Yang lain lagi ? Silakan pilih di bawah ini



Firebase



Amazon
EC2

