



ESLint

Silver- Chapter 3 - Topic 3

Selamat datang di **Chapter 3 Topic 3** pada course
React Native dari Binar Academy!





Hai hai hai! 🙌

Kita ketemu lagi nih, kali ini di **Chapter 3 Topic 3 React Native**. Di Topic 2 kemarin kamu sudah belajar tentang Object Synchronous & Asynchronous Javascript. Di topic kali ini kita akan lanjut ke pembahasan selanjutnya, yaitu tentang **ESLint**.

Yuk, langsung saja kita mulai materi kali ini!





Detailnya, kita bakal bahas hal-hal berikut ini:

- Pengenalan ESLint
- Instalasi ESLint





Sebagian besar dari kamu mungkin masih asing banget dengan istilah ini. Jadi kita kenalan dulu yuk sama ESLint.

Apa sih **EsLint** itu?





ESLint berasal dari istilah Lint atau Linter

Lint atau linter adalah alat untuk menganalisa source code untuk **menandai errors, bug, code convention dan memverifikasi kualitas kode.**





Lint ini ada dalam bentuk sebuah tools atau aplikasi

ESLint adalah sebuah **tools/alat untuk melakukan analisa statis terhadap kode-kode** yang telah kamu tulis, agar kamu dapat tahu dengan cepat menemukan masalah atau potensi bug dari kode yang sudah kamu buat.



Biar gampang, mari kita jelaskan ESLint dalam sebuah analogi~

ESLint ibarat metal detector saat kita masuk ke mall, kedutaan besar, atau bandara yang melakukan screening pada tasmu.

Kalau di dalam tas kamu ada yang mencurigakan dan dianggap salah, maka metal detector akan berbunyi dan memberi tahu petugas untuk segera ditindaklanjuti.





Kenapa sih kamu harus menggunakan ESLint?

ESLint merupakan salah satu tools wajib yang harus kamu gunakan ketika mau bikin aplikasi yang menggunakan bahasa pemrograman JavaScript.

Dengan mengintegrasikan project aplikasimu dengan ESLint –dan mengikuti anjuran dari ESLint, kode yang kamu buat akan **jadi lebih konsisten gaya penulisannya dan minim dengan bug.**

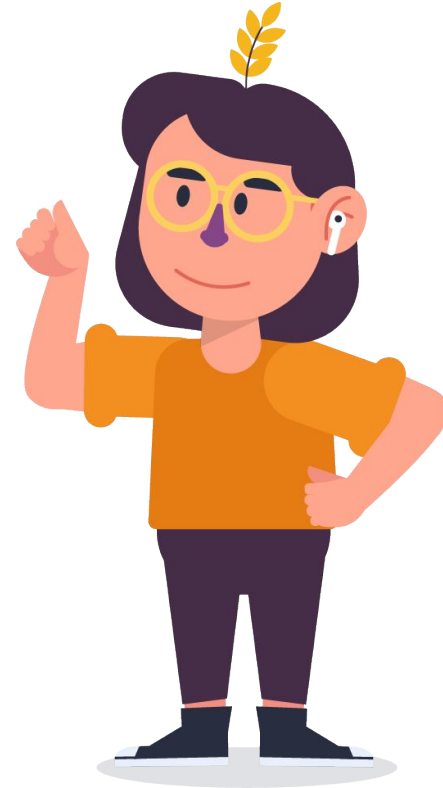




Alasan itu belum cukup? Kalau gitu, ini manfaat lain ESLint yang lain!

Ada tiga manfaat dan fitur lainnya yang akan kamu dapatkan kalau menggunakan ESLint:

- **Mendeteksi masalah**
- **Memperbaiki kode secara otomatis**
- **Menyesuaikan aturan**

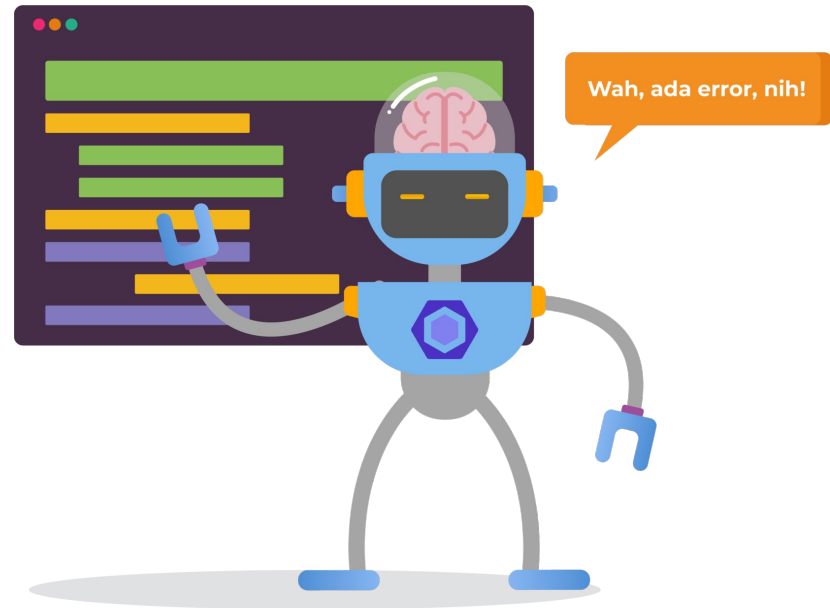




1. Mendeteksi masalah

Ketika kamu menulis JavaScript, ESLint akan **secara otomatis langsung menganalisa kode** yang kamu tulis, apakah punya potensi bug atau error ke depannya.

ESLint akan menganalisa sesuai dengan standar yang diatur pada file konfigurasinya. Keuntungannya adalah kamu akan cepat mendapatkan feedback dari ESLint.





Sebagai contoh adalah kode di samping. **ESLint akan memberi tahu kamu dengan menampilkan pesan error.**

Memang apa masalah pada kode di samping, sih?

Pada kasus kode di samping, kamu menulis sebuah variabel dengan nama foo, tetapi variabel selanjutnya nggak kamu pake.

Nah, ESLint akan langsung ngasih feedback ke kamu, kalo variable foo tersebut nggak berguna karena sudah dibuat dan nggak dipakai.

```
1 'use strict';
2 var foo = "bar";
3
4 fn(function (err) {});
```

The screenshot shows a code editor with four lines of JavaScript code. ESLint has identified several issues:

- Error:** foo is defined but never used (no-unused-vars) at line 2 col 5. This is shown as a tooltip for the variable 'foo' on line 2.
- Error:** foo is defined but never used (no-unused-vars) at line 2 col 5. This is shown in the bottom error list.
- Error:** Strings must use singlequote. (quotes) at line 2 col 11. This is shown in the bottom error list for the double quotes around "bar".
- Error:** "fn" is not defined. (no-undef) at line 4 col 1. This is shown in the bottom error list for the 'fn' function call on line 4.
- Warning:** Expected error to be handled. (handle-callback-err) at line 4 col 4. This is shown in the bottom error list for the 'err' parameter in the callback function on line 4.
- Error:** err is defined but never used (no-unused-vars) at line 4 col 14. This is shown in the bottom error list for the 'err' parameter on line 4.

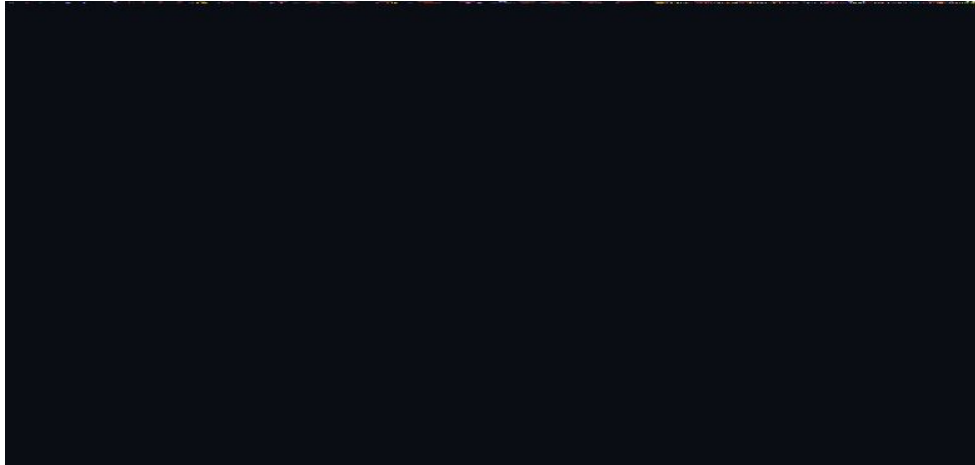


Contoh lain adalah saat **kamu nggak sengaja bikin nama variabel yang sama lebih dari satu kali**. ESLint akan menyadarkanmu dengan menampilkan pesan error bahwa variabel dengan nama tersebut sudah pernah dibuat. ESLint sudah ngasih tahu duluan tentang bug di kodemu, sehingga saat nanti saat kamu menjalankan programmu, aplikasimu nggak bakal error.

```
1  const dataPeserta = {
2    nama: ''
3    emai Parsing error: Identifier 'dataPeserta' has already been declared.
4    no_h
5  };
6
7  functi > 11 | const dataPeserta = 'Oke';
8    cons    |           ^
9  }          | 12 | eslint
10
11 | const dataPeserta = 'Oke';
12
```

View Problem No quick fixes available

You, seconds ago • Uncommitted changes



2. Memperbaiki kode secara otomatis

Selain menganalisa dan memverifikasi kodemu, apakah sudah sesuai dengan standard yang dibuat atau belum, fitur ini nggak kalah keren. Kenapa? Karena dengan fitur ini, ESLint bisa memperbaiki kode secara otomatis.

Pada VSCode ada icon lampu 💡, jika kamu klik maka akan muncul saran untuk perbaikan kodemu yang salah.



3. Menyesuaikan aturan

Dengan menggunakan ESLint, kamu juga bisa **meng-custom alias mengganti konfigurasi** parser (pemecah data) dan rules. Nah, parser dan rules ini nantinya akan digabungkan dengan rules built-in dari ESLint sesuai kebutuhan projectmu.

```
.eslintrc > [ ] plugins > 0
You, 4 months ago | 1 author (You)
1  {
2    "parser": "@typescript-eslint/parser",
3    "parserOptions": {
4      "ecmaVersion": 2020,
5      "sourceType": "module"
6    },
7    "plugins": ["@typescript-eslint/eslint-plugin"],
8    "extends": [
9      "eslint:recommended",
10     "plugin:@typescript-eslint/eslint-recommended",
11     "plugin:@typescript-eslint/recommended",
12     "eslint-config-prettier",
13     "eslint-config-prettier/@typescript-eslint"
14   ],
15   "env": {
16     "node": true
17   }
18 }
```



Kamu bisa **mengatur semua konfigurasi terkait ESLint** pada file **.eslintrc** yang ada di root folder project kamu.

Tapiiii.. kalau kamu belum install ESLint, kamu belum punya file .eslintrc ini ya. Jangan khawatir, kita akan bahas tentang instalasi juga pada subtopik selanjutnya.

```
.eslintrc > [ ] plugins > 0
You, 4 months ago | 1 author (You)
1 {
2   "parser": "@typescript-eslint/parser",
3   "parserOptions": {
4     "ecmaVersion": 2020,
5     "sourceType": "module"
6   },
7   "plugins": ["@typescript-eslint/eslint-plugin"],
8   "extends": [
9     "eslint:recommended",
10    "plugin:@typescript-eslint/eslint-recommended",
11    "plugin:@typescript-eslint/recommended",
12    "eslint-config-prettier",
13    "eslint-config-prettier/@typescript-eslint"
14  ],
15  "env": {
16    "node": true
17  }
18 }
```




Begini contoh dari penyesuaian aturan menggunakan ESLint

Misalnya kamu mau ESLint menampilkan error ketika ada `console.log()` pada kode-kode di project-mu dan menampilkan warning bila ada variabel yang nggak digunakan.

Untuk nama rules yang ingin digunakan, kamu bisa lihat daftar rules apa aja yang tersedia beserta contohnya di [sini](#).

```
1 {
2   "extends": ["eslint-config-airbnb"],
3   "env": {
4     "browser": true,
5     "node": true
6   },
7   "parser": "babel-eslint",
8   "rules": {
9     "no-unused-vars": "warn",
10    "no-console": "error",
11  },
12  "plugins": ["react", "import", "react-hooks"],
13  "settings": {
14    "import/parser": "babel-eslint",
15    "import/resolve": {
16      "moduleDirectory": ["node_modules", "src"]
17    }
18  },
19 }
20
```



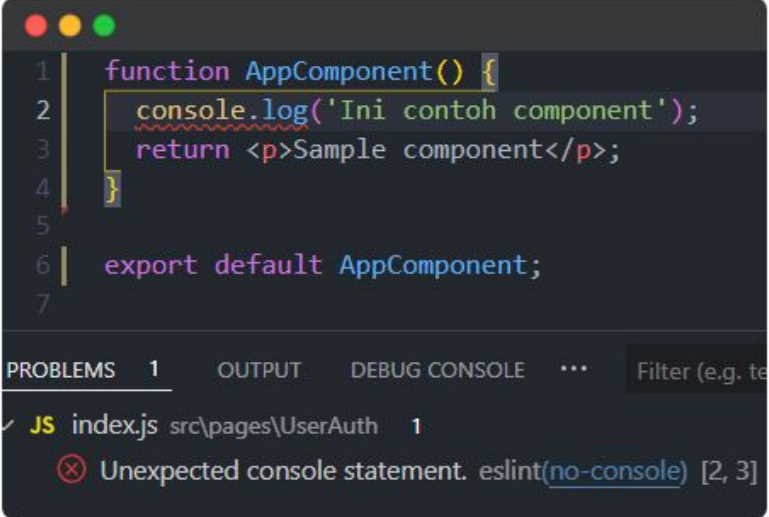
Ada 3 pilihan status dari rules tersebut, yaitu **“off”** untuk mematikan rules, **“warn”** untuk agar ESLint menampilkan pesan warning, **“error”** agar ESLint menampilkan sebagai pesan error.

```
1 {
2   "extends": ["eslint-config-airbnb"],
3   "env": {
4     "browser": true,
5     "node": true
6   },
7   "parser": "babel-eslint",
8   "rules": {
9     "no-unused-vars": "warn",
10    "no-console": "error"
11  },
12  "plugins": ["react", "import", "react-hooks"],
13  "settings": {
14    "import/parser": "babel-eslint",
15    "import/resolve": {
16      "moduleDirectory": ["node_modules", "src"]
17    }
18  },
19 }
20
```



Begini nih hasilnya~

Kalau kamu set status rules 'no-console' menjadi 'error' maka saat kamu tulis `console.log` pada kode, ESLint akan menampilkan pesan Error, sehingga kode nggak bisa dijalankan, kecuali `console.log` dihapus.



```
1 function AppComponent() {  
2   console.log('Ini contoh component');  
3   return <p>Sample component</p>;  
4 }  
5  
6 export default AppComponent;  
7
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE ... Filter (e.g. te

✓ JS index.js src\pages\UserAuth 1

✗ Unexpected console statement. eslint([no-console](#)) [2, 3]



Kalau udah tahu manfaat ESLint,
maka yang nggak kalah penting
adalah tahu cara **Instalasi ESLint**.

Cusss, cari tahu caranya!

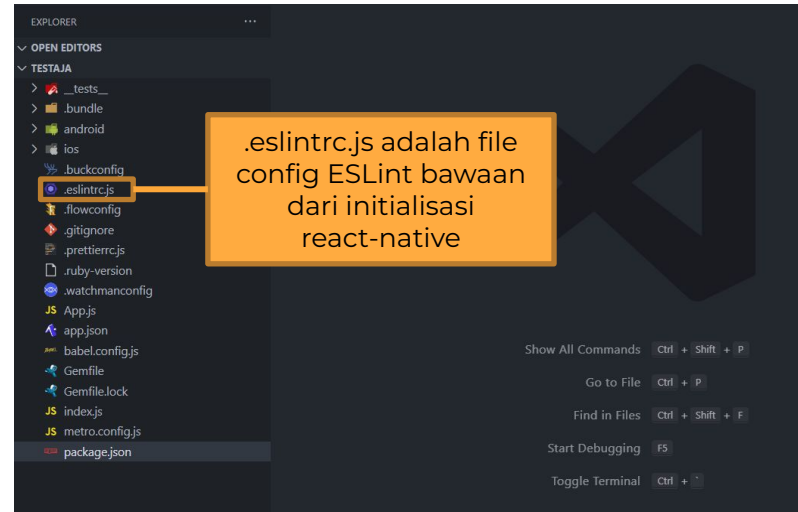




Cara instalasi dan integrasi ESLint dengan project React Native

Sekarang coba lakukan instalasi dan integrasi ESLint dengan project React Native kamu yuks.

Semenjak versi react native 0.64, ESLint memang telah terinstall secara otomatis, tapi kamu akan coba bikin konfigurasi sendiri sehingga nantinya kamu bisa lebih fleksibel menambahkan atau mengurangi rules-rules baru.





Mari kita mulai proses **Instalasi ESLint**.
Perhatikan baik-baik cara yaa~





1. Uninstall konfigurasi eslint versi react native community.



```
1 $ npm uninstall @react-native-community/eslint-config
```



2. Buat configuration file untuk ESLint. Ketika perintah berikut dijalankan akan muncul beberapa pertanyaan via CLI.



```
1 $ npx eslint --init
```




Pertanyaan 1



```
1 # pertanyaan 1:  
2 ? How would you like to use ESLint? ...  
3   To check syntax only  
4   To check syntax and find problems  
5 › To check syntax, find problems, and enforce code style
```

Note : Jawaban dari pertanyaan ditandai dengan kotak warna orange



Pertanyaan 2



```
1 # Pertanyaan 2:  
2 ? What type of modules does your project use? ...  
3 › JavaScript modules (import/export)  
4   CommonJS (require/exports)  
5   None of these
```

Note : Jawaban dari pertanyaan ditandai dengan kotak warna orange



Pertanyaan 3



```
1 # pertanyaan 3, pilih 'React'  
2 ? Which framework does your project use? ...  
3 > React  
4   Vue.js  
5   None of these
```

Note : Jawaban dari pertanyaan ditandai dengan kotak warna orange



Pertanyaan 4

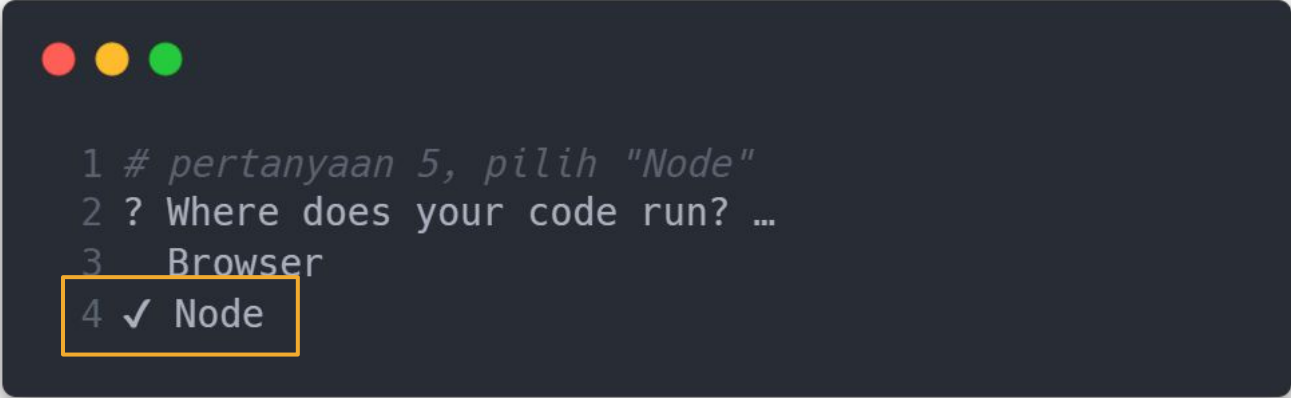


```
1 # pertanyaan 4 (pilih "No"):  
2 ? Does your project use TypeScript? > No / Yes
```

Note : Jawaban dari pertanyaan ditandai dengan kotak warna orange



Pertanyaan 5 (Instalasi dan Integrasi ESLint dengan project React Native)



```
1 # pertanyaan 5, pilih "Node"  
2 ? Where does your code run? ...  
3   Browser  
4 ✓ Node
```

Note : Jawaban dari pertanyaan ditandai dengan kotak warna orange



Pertanyaan 6



```
1 # pertanyaan 6, pilih "Use a popular style guide"
2 ? How would you like to define a style for your project? ...
3 › Use a popular style guide
4 Answer questions about your style
5 Inspect your JavaScript file(s)
6
```

Note : Jawaban dari pertanyaan ditandai dengan kotak warna orange



Pertanyaan 7



```
1 # pertanyaan 7 (pilih "Airbnb"):  
2 ? Which style guide do you want to follow? ...  
3 > Airbnb: https://github.com/airbnb/javascript  
4   Standard: https://github.com/standard/standard  
5   Google: https://github.com/google/eslint-config-google
```

Note : Jawaban dari pertanyaan ditandai dengan kotak warna orange



Pertanyaan 8



```
1 # pertanyaan 8 (pilih 'JSON')
2 ? What format do you want your config file to be in? ...
3   JavaScript
4   YAML
5 > JSON
6
```

Note : Jawaban dari pertanyaan ditandai dengan kotak warna orange



Pertanyaan 9

```
1 # pertanyaan 9 (pilih 'YES')
2
3 Checking peerDependencies of eslint-config-airbnb@latest
4 The config that you have selected requires the following dependencies:
5
6 eslint-plugin-react@^7.21.5 eslint-config-airbnb@latest eslint@^5.16.0 ||
  ^6.8.0 || ^7.2.0 eslint-plugin-import@^2.22.1 eslint-plugin-jsx-a11y@^6.4.1
  eslint-plugin-react-hooks@^4 || ^3 || ^2.3.0 || ^1.7.0
7
8 ? Would you like to install them now with npm? > No / Yes
```

Note : Jawaban dari pertanyaan ditandai dengan kotak warna orange



- Setelah proses instalasi selesai, akan ada sebuah file baru dengan nama `.eslintrc.json` pada folder project. **Isi file tersebut akan terlihat persis seperti gambar di samping**

```
.eslintrc.json > ...
1  {
2    "env": {
3      "browser": true,
4      "es2021": true
5    },
6    "extends": [
7      "eslint:recommended",
8      "plugin:react/recommended"
9    ],
10   "parserOptions": {
11     "ecmaFeatures": {
12       "jsx": true
13     },
14     "ecmaVersion": "latest",
15     "sourceType": "module"
16   },
17   "plugins": [
18     "react"
19   ],
20   "rules": {
21   }
22 }
23
```



4. Nah, sekarang kamu punya 2 file eslint, **yaitu .eslintrc.js** (bawaan react native) dan **.eslintrc.json** (hasil instalasi yang kamu lakukan).

Selanjutnya, kamu bisa hapus file .eslintrc.js yang merupakan file bawaan react-native.

```
.eslintrc.json > ...
1  {
2    "env": {
3      "browser": true,
4      "es2021": true
5    },
6    "extends": [
7      "eslint:recommended",
8      "plugin:react/recommended"
9    ],
10   "parserOptions": {
11     "ecmaFeatures": {
12       "jsx": true
13     },
14     "ecmaVersion": "latest",
15     "sourceType": "module"
16   },
17   "plugins": [
18     "react"
19   ],
20   "rules": {
21   }
22 }
23
```



```
1 $ npm install --save-dev eslint-plugin-react-native
```

5. Tambahkan eslint-plugin untuk platform react-native. **Ketik command/perintah di bawah ini pada terminal, lalu tekan enter.**




6. Setelah proses instalasi `eslint-plugin-react-native` selesai, tambahkan **'react-native'**, pada bagian **plugins**, dan **'react-native/react-native'** pada bagian **env** di file `.eslintrc.json`

```
.eslintrc.json > {} parserOptions
1  {
2    "env": {
3      "browser": true,
4      "es2021": true,
5      "react-native/react-native": true
6    },
7    "extends": [
8      "eslint:recommended",
9      "plugin:react/recommended"
10   ],
11   "parserOptions": {
12     "ecmaFeatures": {
13       "jsx": true
14     },
15     "ecmaVersion": "latest",
16     "sourceType": "module"
17   },
18   "plugins": [
19     "react",
20     "react-native"
21   ],
22   "rules": {
23   }
24 }
```



Setelah semua tahap instalasi selesai, berarti kamu udah berhasil mengintegrasikan ESLint dengan project react native-mu.

Kamu bisa menambahkan rules ESLint yang menurutmu diperlukan untuk project kamu. Kamu juga bisa menambahkan rules tersebut pada bagian rules di file `.eslintrc.json`



```
"rules": {  
  "no-unused-vars": "error",  
  "no-console": "error",  
}
```



Yeaayy!! Selamat yaa, guys! Akhirnya kamu berhasil menuntaskan Topik 3 tentang ESLint ini. Biar makin jago, Kamu coba praktekan lagi ya, guys!

Semangaaattsss!!!



Terima Kasih!



Next Topic

loading...