Solution : Promise

Apakah Anda sudah berhasil mengubah penerapan callback function menjadi Promise? Jika belum, mari kita lakukan bersama-sama.

Langkah awal, pada class ***DataSource***. Kita hapus constructor yang memuat callback function pada class.

```
class DataSource {
    constructor(onSuccess, onFailed) {
        this.onSuccess = onSuccess;
        this.onFailed = onFailed;
    }

    searchClub(keyword) {
        const filteredClubs = clubs.filter(club => club.name.toUpperCase().includes(keyword.toUpperCase()));

        if (filteredClubs.length) {
            this.onSuccess(filteredClubs);
        } else {
            this.onFailed(`${keyword} is not found"`);
        }
    }
}
```

Sehingga class ***DataSource*** akan tampak seperti ini:

```
class DataSource {
    searchClub(keyword) {
        const filteredClubs = clubs.filter(club => club.name.toUpperCase().includes(keyword.toUpperCase()));

        if (filteredClubs.length) {
            this.onSuccess(filteredClubs);
        } else {
            this.onFailed(`${keyword} is not found"`);
        }
    }
}
```

Karena kita sudah menghapus callback function beserta constructor class-nya, kita tidak perlu membuat instance dari class ***DataSource***. Kita buat method ***searchClub*** dapat diakses secara langsung dari ***DataSource*** tanpa harus membuat *instance*.

Masih ingat bagaimana cara melakukannya? Untuk membuat method dalam class dapat diakses tanpa membuat instance, kita perlu menambahkan keyword ***static*** sebelum membuat method-nya.

```
class DataSource {
    static searchClub(keyword) {
```

```
        const filteredClubs = clubs.filter(club => club.name.toUpperCase().includes(keyword.toUpperCase()));

        if (filteredClubs.length) {
            this.onSuccess(filteredClubs);
        } else {
            this.onFailed(`${keyword} is not found"`);
        }
    }
}
```

Setelah itu, kita buat method *searchClub* mengembalikan nilai promise seperti ini:

```
class DataSource {
    static searchClub(keyword) {
        return new Promise((resolve, reject) => {
            const filteredClubs = clubs.filter(club => club.name.toUpperCase().includes(keyword.toUpperCase()));
            if (filteredClubs.length) {
                this.onSuccess(filteredClubs);
            } else {
                this.onFailed(`${keyword} is not found"`);
            }
        });
    }
}
```

Jangan lupa ubah juga pemanggilan callback *onSuccess()* dan *onFailed()*, menggunakan *resolve()* dan *reject()*. Maka sekarang class *DataSource* akan tampak seperti ini:

```
class DataSource {
    static searchClub(keyword) {
        return new Promise((resolve, reject) => {
            const filteredClubs = clubs.filter(club => club.name.toUpperCase().includes(keyword.toUpperCase()));
            if (filteredClubs.length) {
                resolve(filteredClubs);
            } else {
                reject(`${keyword} is not found"`);
            }
        });
    }
}
```

Namun belum sampai di sini. Dengan mengubah method *searchClub* menjadi *static* dan mengubah *callback* menjadi Promise, tentu kita perlu penyesuaian kembali ketika method itu digunakan.

Pada berkas ***main.js***, ubah fungsi *onButtonSearchClicked* ini:

```
const onButtonSearchClicked = () => {
        const dataSource = new DataSource(renderResult, fallbackResult);
        dataSource.searchClub(searchElement.value);
};
```

Menjadi seperti ini:

```
const onButtonSearchClicked = () => {
        DataSource.searchClub(searchElement.value)
            .then(renderResult)
            .catch(fallbackResult)
};
```

Atau jika Anda lebih suka menggunakan async/await, Anda bisa membuat fungsi *onButtonSearchClicked* berjalan secara asynchronous seperti ini:

```
const onButtonSearchClicked = async () => {
   try{
       const result = await DataSource.searchClub(searchElement.value);
       renderResult(result);
   } catch (message) {
       fallbackResult(message)
   }
};
```

Promise sudah berhasil diterapkan! Untuk memastikan perubahan yang kita lakukan sudah benar, silakan Anda coba buka berkas ***index.html*** pada browser. Jika aplikasi berjalan dengan lancar, maka promise berhasil diterapkan dengan baik.

Namun jika aplikasi tidak berjalan lancar, dan menghasilkan eror pada console browser, jangan malu untuk bertanya pada forum diskusi ya!

Langkah dari solution ini bisa Anda temukan juga pada repository berikut: https://github.com/dicodingacademy/a163-bfwd-labs/tree/107-club-finder-promise-solution

⬅ KEMBALI KE MATERI SEBELUMNYA
LANJUTKAN KE MATERI BERIKUTNYA ➡

dicoding

PERUSAHAAN

Tentang Kami

Blog

Berita Terbaru

PROGRAM

Academy

Challenge

Event

Job

Rewards

SUPPORT

Bantuan

FAQ

Hubungi Kami

Terms Privacy

DIBANTU