

2022

# PBW

## Praktikum 6 – Express JS



## PETUNJUK Pengerjaan

1. Upload folder (di zip) ke IDE sesuai jam pengumpulan yang tertera.

### Express JS

Pada praktikum ini kita akan memperdalam dan menambah pengalaman dalam menggunakan framework Express JS. Pada praktikum kali ini tidak disediakan template. Silahkan membuat project sesuai dengan yang sudah diajarkan. Selain itu anda, akhirnya pada kesempatan ini anda ditantang untuk mampu membuat website yang fungsional dari kedua sisi (frontend dan backend).

Panduan umum untuk mengerjakan praktikum ini:

1. Menggunakan framework Express JS.
2. Menggunakan view engine EJS.
3. Memanfaatkan nodemon saat development.
4. Data pada tampilan tidak ada yang di-hardcode. Semua data disimpan pada sisi server dan tidak akan hilang ketika berpindah halaman, kecuali server di restart.
5. JS dan CSS di sisi server diletakkan di folder public.
6. Pada sisi server dibebaskan mengerjakan dengan gaya procedural maupun OOP.
7. Mohon fokus pada fungsionalitas website, sebelum anda menambahkan style CSS (CSS dikerjakan paling terakhir).
8. Tidak ada ketentuan penggunaan CSS (yang penting menyerupai contoh).

### Imaginary Friendster

Imaginary Friendster merupakan suatu sosial media terbaru yang fokus bagi pengguna introvert yang kekurangan teman, tetapi tidak mau bergaul dengan yang lain. Oleh karena itu, kami sediakan fitur untuk anda yang ingin ngobrol dengan diri sendiri dan dapat juga dibalas oleh teman khayalannya secara pribadi. Tampilan halaman utama dapat dilihat pada Gambar 1 berikut.

[Home](#) [Add](#) [Filter](#)

#### Imaginary Friend Timeline

Me

Halo dunia, disini saya bersama teman khayalanku

[Tags] : Friend A, Friend B

Friend A

Halo juga bro!! Piye kabare...

[Tags] : Me

Friend A

Woi, kemana aja lu?

[Tags] : Friend B

Gambar 1 – Tampilan utama.

Website ini memiliki 3 buah halaman yaitu dengan url:

1. '/': merupakan halaman utama seperti yang dapat dilihat pada Gambar 1.
2. '/add': merupakan halaman untuk menambahkan postingan.
3. '/filter/:user': merupakan halaman yang mirip dengan halaman pertama, namun hanya menampilkan posting yang dibuat sendiri, maupun posting yang di tag.
4. Selain ketiga halaman html tersebut, website ini juga ditambahkan 1 buah file header.ejs yang berisi menu link yang mengarah ke masing-masing halaman. Header ini akan sama dan dipakai di ketiga halaman tersebut.

## Halaman Utama

Halaman utama seperti dapat dilihat pada Gambar 1, menampilkan seluruh posting yang dibuat oleh semua orang dengan format yang sesuai gambar. Akan tetapi pada saat program dimulai, maka halaman ini akan kosong dan baru akan diisi ketika sudah ditambahkan dari halaman Add.

## Halaman Add

Pada halaman add dibuka, maka akan ditampilkan sesuai dengan Gambar 2 berikut:



Home Add Filter Me ▼

### Add Post

Add

Author: Me ▼

Post: Halo dunia, disini saya bersama teman khayalanku

Tags: Friend A, Friend B

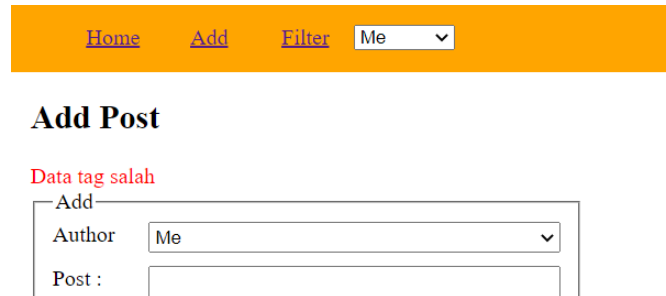
Submit

Gambar 2 – Halaman Add.

Keterangan form adalah sebagai berikut:

1. Form disubmit dengan method POST ke url '/add'.
2. Author dibuat dalam bentuk dropdown, dimana isi dari dropdown ini **diambil dari array users** di sisi server dan diisi dengan menggunakan templating engine. Pada contoh digunakan user ['Me', 'Friend A', 'Friend B']
3. Post merupakan teks yang akan diposting.

- Tags merupakan teman khayalan yang akan di tag. Format penulisan dipisahkan dengan tanda koma jika lebih dari satu.
- Ketika tombol submit ditekan, maka akan diperiksa apakah orang yang di tag tersebut berada pada array **users** yang berada di server. Selain itu kita juga tidak diperbolehkan untuk tag diri sendiri.
- Jika ternyata terdapat kesalahan di nomor 5, maka halaman akan redirect ke halaman add ini dengan ditambah pesan error sesuai dengan Gambar 3. (Hint: gunakan juga EJS untuk menambah pesan error) .



Home Add Filter Me ▾

### Add Post

Data tag salah

Add

Author Me ▾

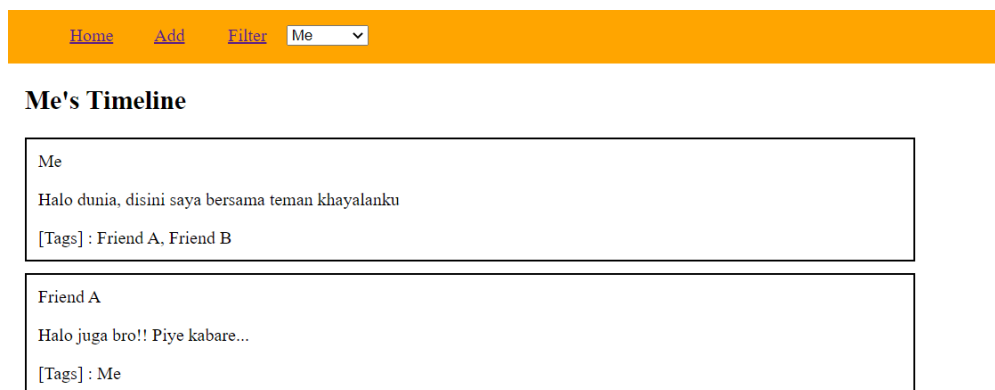
Post :

Gambar 3 – Error input ditandai dengan pesan warna merah.

- Namun apabila, pada pemeriksaan di nomor 5 tidak terjadi kesalahan maka data post ini akan disimpan di sebuah array lain bernama **posts**. Lalu setelahnya halaman akan berpindah ke halaman utama (Tentunya harus sudah muncul post yang baru ditambahkan tersebut).

## Halaman Filter

Seperti dapat dilihat pada gambar-gambar sebelumnya, pada header di sebelah link filter, terdapat dropdown daftar user. Isi daftar user ini menggunakan cara yang sama pada halaman Add. Kegunaan dari dropdown ini, adalah setiap kali isi dari dropdown ini berubah, maka link dari komponen Filter ini akan berubah dan mengarah ke url `/filter/:user`. (Hint: perubahan ini dilakukan dengan event listener di sisi klien.)



Home Add Filter Me ▾

### Me's Timeline

Me

Halo dunia, disini saya bersama teman khayalanku

[Tags] : Friend A, Friend B

Friend A

Halo juga bro!! Piye kabare...

[Tags] : Me

Gambar 4 – Contoh tampilan halaman `/filter/Me`.

Misalkan dropdown sedang menampilkan teks 'Me', lalu ditekan tombol Filter, artinya halaman akan berpindah ke halaman filter seperti pada Gambar 4. Pada halaman ini akan menampilkan halaman yang sama dengan halaman utama. Akan tetapi, akan dilakukan filter agar hanya menampilkan semua post yang diposting oleh 'Me' atau posting lain yang mengandung tag ke user 'Me'. Tentunya jika diarahkan ke halaman lain untuk user lain, maka isi dari tampilan pun akan mengikuti.

Kumpulkan semua pekerjaan dalam bentuk zip dengan nama T06xyyy.zip

**= SELAMAT MENGERJAKAN =**