


Nama : Izhar Mayranda
Program : Intro To NodeJs

Summary Sesi 4

- ExpressJs
ExpressJs adalah framework dari NodeJs yang digunakan untuk membantu tahap pengembangan *back-end* secara fleksibel dan sederhana.
- 4 Bagian Penting Dalam ExpressJs
 1. Require statement
Require statement adalah statement untuk import ExpressJs lalu mendeklarasikan variabel untuk invoke express.
 2. Middleware
Middleware didalam ExpressJs adalah fungsi yang menentukan perputaran request, dan respon. Fungsi ini bisa mengakses request, response, dan next. Fungsi ini selalu terpanggil setelah ada request yang datang, dan menentukan bagaimana request ini direspon. Tujuan dari middleware adalah untuk memodifikasi objek request dan response untuk mengerjakan sesuatu seperti parsing request body, menambahkan response header, memanggil fungsi middleware selanjutnya, dan sebagainya.
 3. Routing
Routing ini mengacu pada bagaimana server merespon request ke endpoint tertentu. Endpoint ini mengandung URL (contohnya / atau /user) dan sebuah HTTP method yakni get, post, put ,delete dan lain-lain.
 4. Statement `app.listen(port, fungsiCallback)`
Statement ini digunakan untuk menjalankan dan mendengarkan server di port yang telah ditentukan.
- JSON adalah format yang digunakan untuk menyimpan dan mentransfer data. JSON bisa berbentuk objek maupun array.
- Request Method
Ada beberapa request method yang digunakan untuk routing yakni:
 1. GET untuk mengambil/menampilkan data dari server.



```
app.get('/users', (request,response) => {  
  console.log('GET /users');  
  response.send(data);  
});
```

2. POST untuk menambahkan data ke server.

```
app.post('/users', (request, response) => {
  console.log('POST /users', request.body);
  const {email, first_name, last_name,
  avatar} = request.body;
  const newUser = {
    id: data.length + 1,
    email,
    first_name,
    last_name,
    avatar
  };
  data.push(newUser);
  fs.writeFile('./data.json',
  JSON.stringify(data, null, 2), () => {
    console.log('Data berhasil
    ditambahkan');
  });
})
```

3. PATCH untuk mengedit data server

```
app.patch('/users/:id', (req, res) => {
  const {body} = req;
  const prevData = data[req.params.id - 1];
  data[req.params.id - 1] = {
    ...prevData,
    email: body.email ?? prevData.email,
    first_name: body.first_name ??
    prevData.first_name,
    last_name: body.last_name ??
    prevData.last_name,
    avatar: body.avatar ?? prevData.avatar
  };
  fs.writeFile('./data.json',
  JSON.stringify(data, null, 2), () => {
    console.log('Data berhasil diupdate di
    json') });
})
```

4. DELETE untuk menghapus data dari server

```
app.delete('/users/:id', (req, res, next) => {
  {
    if (!req.params.id || req.params.id >
    data.length) {
      res.send('Data tidak ada');
    };
    next();
  }, (req, res) => {
    data.splice(req.params.id - 1, 1);
    fs.writeFile('./data.json',
    JSON.stringify(data, null, 2), () => {
      console.log('Data berhasil dihapus');
    });
    res.send('Data berhasil dihapus')
  })
})
```

