

Back-End Development

express

Exploring

#6 Express Middleware

Middleware

Middleware essentially is code that works between request & response. First, we'll use express built-in middleware to serve a static file, for instance a txt file.

- First, create a project directory then

```
$ npm init
```

- Installation

```
$ npm install express --save
```

Middleware (Serving a static file)

Middleware essentially is code that works between request & response. In this section, we'll use express built-in middleware to serve a static file, for instance a txt file.

teks.txt //save on dir *assets* !

```

Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed do eiusmod tempor
incidunt ut labore et dolore magna
aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris nisi
ut aliquip ex ea commodo consequat.
```

satu.js

Middleware (Serving a static file)

```
var express = require('express');
var app = express();
app.set('view engine', 'ejs');

app.use('/assets',
express.static('assets'));
// '/assets' = it's an express route
// 'assets' = it's a ref dir/folder

app.get('/satu', function(req, res){
    res.render('satu');
})

app.listen(3000);
```

<http://localhost:3000/assets/teks.txt>

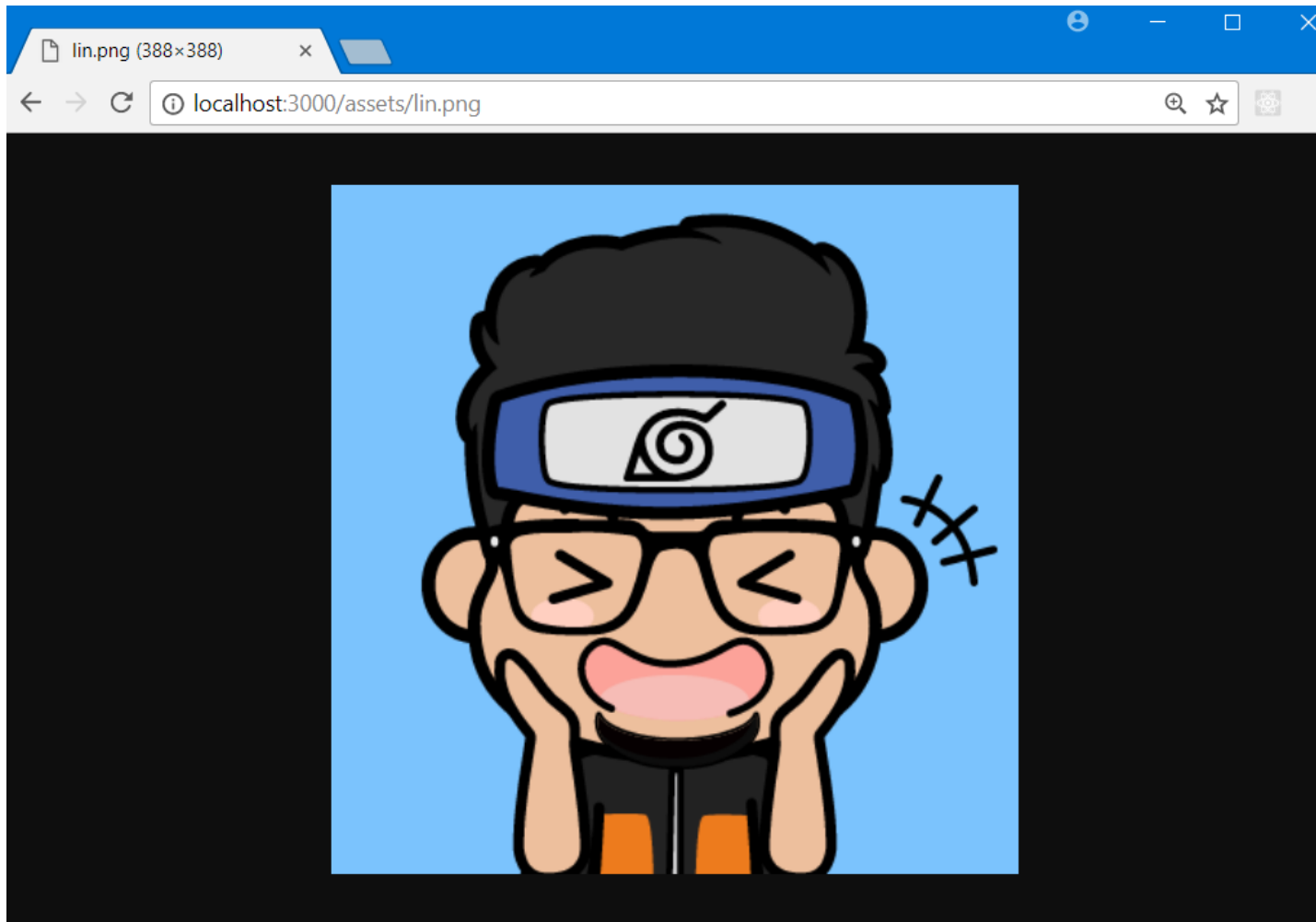


localhost3000/lintang/teks.txt



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

<http://localhost:3000/assets/lin.png>



Insert a pict on */assets* dir!


body-parser

Node.js body parsing middleware. Parse incoming request bodies in a middleware before your handlers, available under the `req.body` property. This module provides the following parsers: JSON body parser, raw body parser, text body parser & URL-encoded form body parser.

■ Installation

```
npm install body-parser --save
```

localhost:3000/kontak



formulir.js - zzz - Visual Studio Code

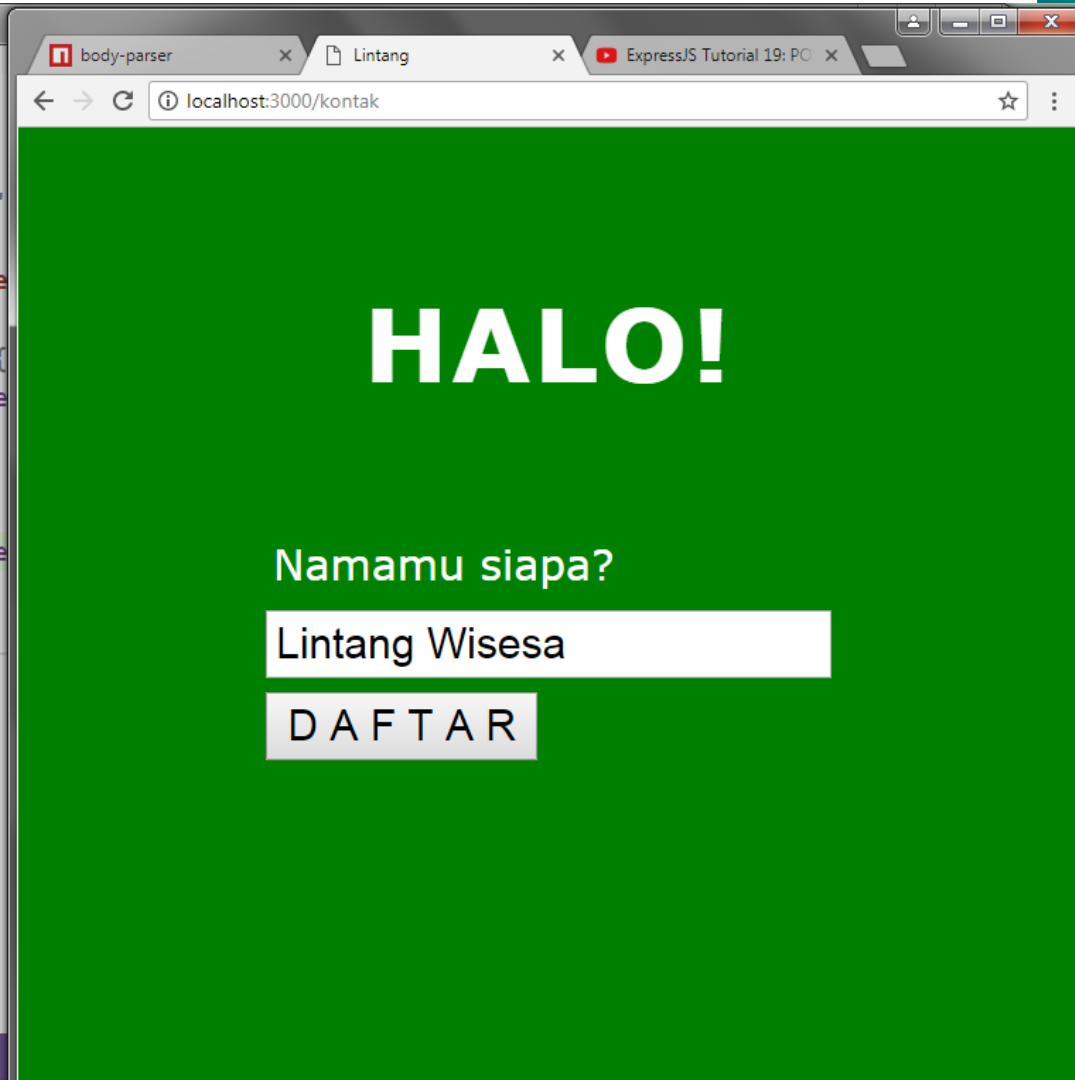
File Edit Selection View Go Debug Tasks Help

JS dua.js <> satu.html JS formulir.js x

```
1 var express = require('express');
2 var bodyParser = require('body-parser');
3 var app = express();
4 var urlencodedParser = bodyParser.urlencoded({ extended: false });
5 app.set('view engine', 'ejs');
6 app.get('/kontak', function(req, res){
7   res.render('formulir', {qs:req.query});
8 })
9 app.post('/kontak', urlencodedParser,
10  console.log(req.body);
11  res.render('formulir', {qs:req.query});
12 })
13 app.listen(3000);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
[nodemon] starting `node satu dua.js`
{ id: 'Andi' }
{ id: 'Susi Susilowati' }
{ id: 'Haider' }
{ id: 'Yuni Shara' }
{ id: 'Roberto Carlos' }
{ id: 'Lintang Wisesa' }
```



formulir.js

```
var express = require('express');
var bodyParser = require('body-parser')
var app = express();
var url = bodyParser.urlencoded({ extended: false })
app.set('view engine', 'ejs');

app.get('/kontak', function(req, res){
    res.render('formulir');
})

app.post('/kontak', url, function(req, res){
    console.log(req.body);
    res.render('formulir');
})

app.listen(3000);
```

formulir.ejs

```
<body>
```

```
<h2>Halo!</h2>
```

```
<form id="daftar" method="POST" action="/kontak">
```

```
<label for="id">Namamu siapa?</label>
```

```
<input type="text" name="id" value=""/>
```

```
<input type="submit" value=" D A F T A R "/>
```

```
</form>
```

```
</body>
```

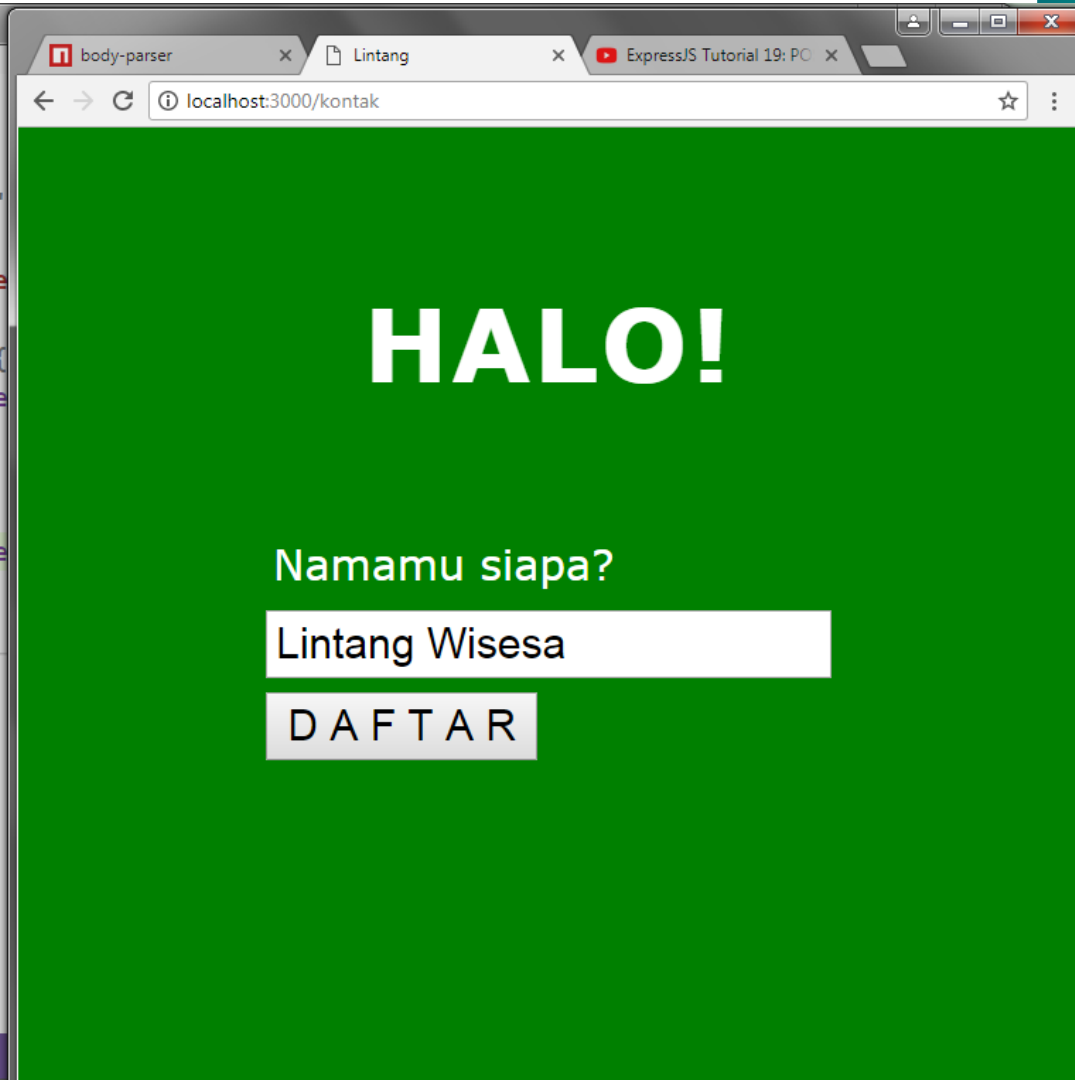
localhost:3000/kontak



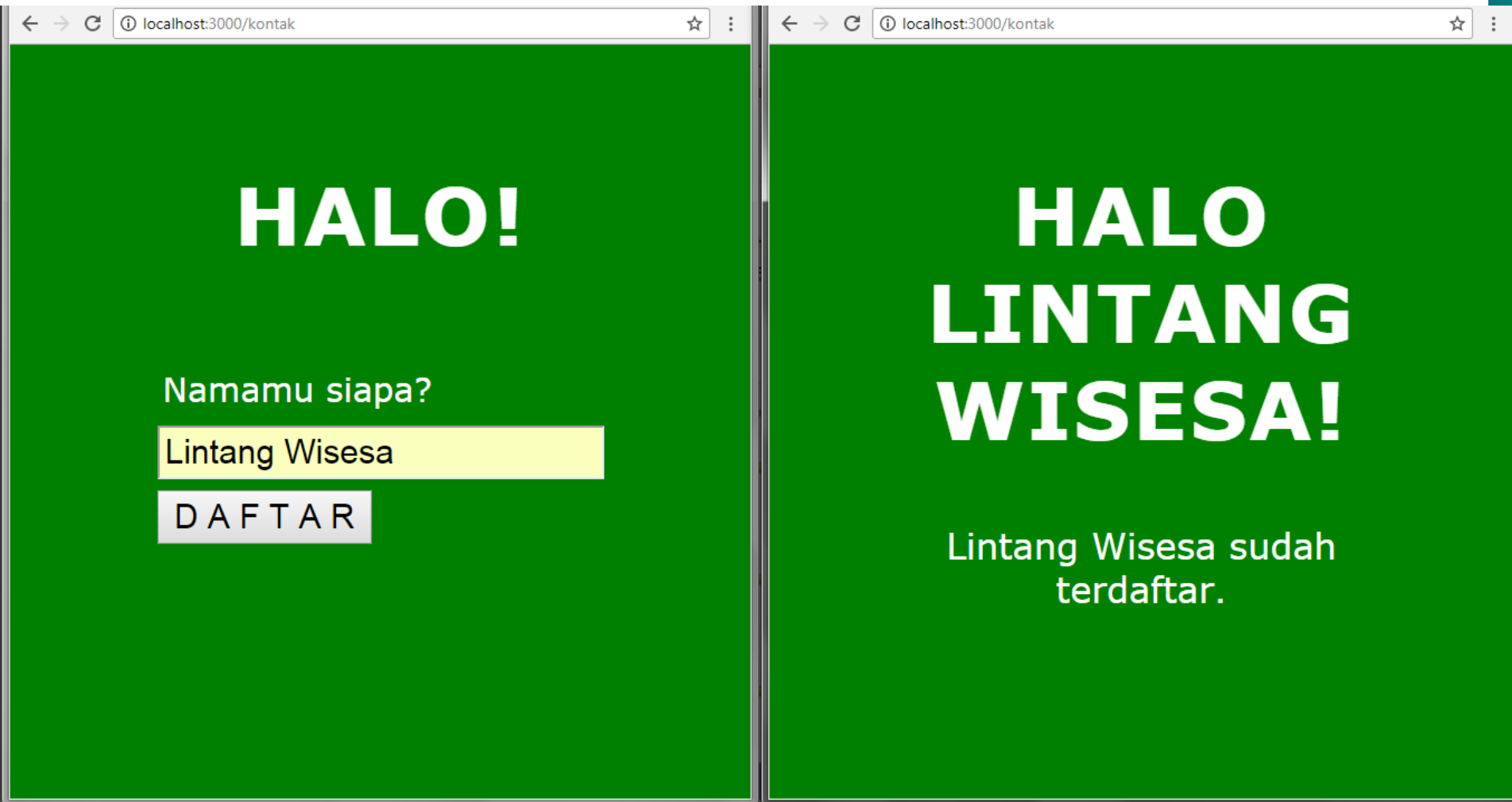
```
1 var express = require('express');
2 var bodyParser = require('body-parser');
3 var app = express();
4 var urlencodedParser = bodyParser.urlencoded({ extended: false });
5 app.set('view engine', 'ejs');
6 app.get('/kontak', function(req, res){
7   res.render('formulir', {qs:req.query});
8 });
9 app.post('/kontak', urlencodedParser,
10   function(req, res){
11     console.log(req.body);
12     res.render('formulir', {qs:req.query});
13   });
14 app.listen(3000);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
[nodemon] starting `node satu dua.js`
{ id: 'Andi' }
{ id: 'Susi Susilowati' }
{ id: 'Haider' }
{ id: 'Yuni Shara' }
{ id: 'Roberto Carlos' }
{ id: 'Lintang Wisesa' }
```



localhost:3000/kontak



formulir.ejs

```
<body>
```

```
<h2>Halo!</h2>
```

```
<form id="daftar" method="POST" action="/kontak">
```

```
<label for="id">Namamu siapa?</label>
```

```
<input type="text" name="id" value=""/>
```

```
<input type="submit" value=" D A F T A R "/>
```

```
</form>
```

```
</body>
```

form_OK.ejs

<body>

<h2>Halo <%= data.id %> !</h2>

<p> <%= data.id %> sudah terdaftar.</p>

</body>

formulir.js

POST to server & Response to client

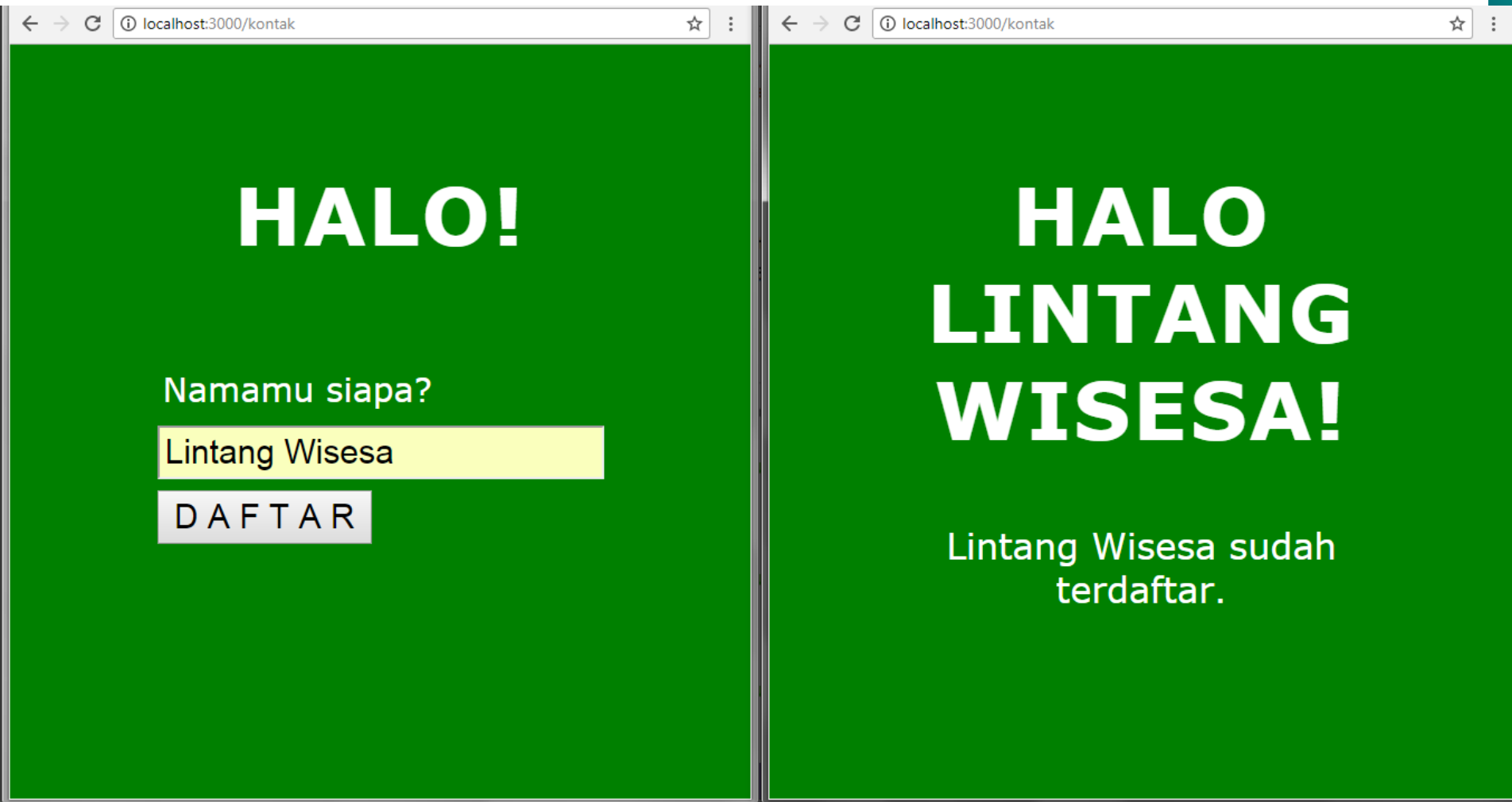
```
var express = require('express');
var bodyParser = require('body-parser')
var app = express();
var url = bodyParser.urlencoded({ extended: false })
app.set('view engine', 'ejs');

app.get('/kontak', function(req, res){
    res.render('formulir');
})

app.post('/kontak', url, function(req, res){
    console.log(req.body);
    res.render('form_OK', {data:req.body});
})

app.listen(3000);
```

localhost:3000/kontak



Back-End Development

express

Exploring

#6 Express Module