

Back-End Development

express

# Exploring

**#6** Express Module

# express

# Express

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web & mobile applications.  
(See <https://expressjs.com>).

- First, create a project directory then

```
$ npm init
```

- Installation

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

# express

## Response to a Create Server

***satu.js***

```
var express = require('express');  
var app = express();
```

```
app.listen(3000);  
// Listening port 3000
```

***Run node satu***  
***Open localhost:3000***



# Nodemon

Nodemon is a tool/utility that will monitor for any changes in your source and automatically restart your server. Just use **nodemon** instead of **node** to run your code, and now your process will automatically restart when your code changes. See <https://nodemon.io>.

## ■ Installation

```
$ npm install -g nodemon
```

## ■ Open & run source with Nodemon

```
$ nodemon namaFile.js
```

## ■ Restart

```
$ rs
```

# express

## Response to a GET Request

```
var express = require('express');  
var app = express();
```

```
app.get('/', function(req, res){  
    res.send('Selamat datang!');  
})
```

*// GET request*

```
app.listen(3000);
```

*// Listening port 3000*



# express

## Serving HTML

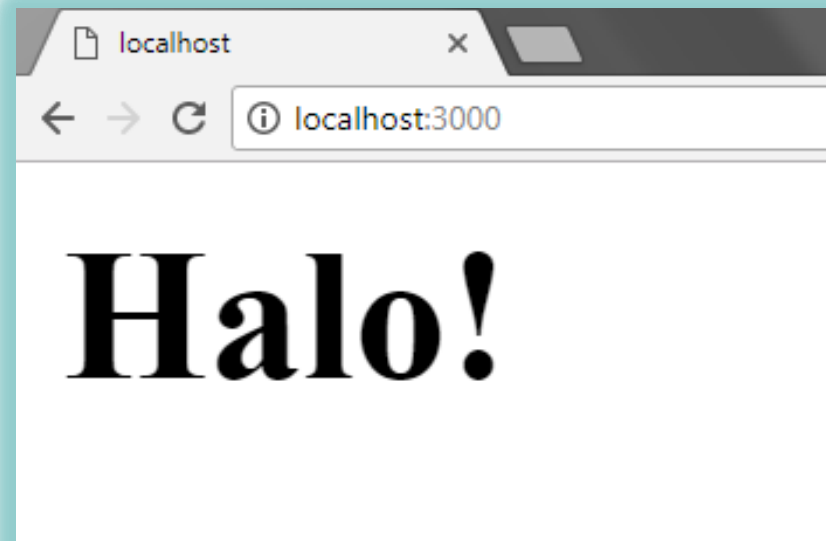
```
var express = require('express');  
var app = express();
```

```
app.get('/', function(req, res){  
    res.send('<h1>Halo!</h1>');  
})
```

*// GET request*

```
app.listen(3000);
```

*// Listening port 3000*



# express

## Serving JSON

```
var express = require('express');  
var app = express();
```

```
app.get('/', function(req, res){  
    res.json({"nama": "Andi"});  
})
```

*// GET request*

```
app.listen(3000);  
// Listening port 3000
```



# Routing



**/beranda**



**/profil**



**/berita**



**/galeri**



**/tentang**



**/kontak**



# express

## Response to a GET Request

```
var express = require('express');  
var app = express();
```

```
app.get('/', function(req, res){  
    res.send('Selamat datang!');  
})
```

*// GET request*

```
app.listen(3000);  
// Listening port 3000
```



# express

## Response to a Static Request

```
var express = require('express');
var app = express();

app.get('/halo', function(req, res){
  res.send('<h1>Halo!</h1>');
})
// GET static request

app.listen(3000);
// Listening port 3000
```



# express

## Response to a Dynamic Request

```
var express = require('express');
var app = express();

app.get('/id/:nama', function(req, res){
    res.send('Request ID '+req.params.nama);
})
// GET dynamic request w/ parameter nama

app.listen(3000);

// coba buka:
// localhost:3000/id/Lintang
// localhost:3000/id/Andi
```

*satu.html*

Serving HTML File  
#Static Req & Res

# SELAMAT DATANG!

~ Ini webpage paling unyu sedunia ~

```
<!DOCTYPE html>
<html>
  <head><title>Lintang</title>
    <style>
      body{background:green; font-family:verdana;
color:white; padding:150px;}
      h1{font-size:72px; text-transform:uppercase;
letter-spacing:2px; text-align:center;}
      p{font-size:32px; text-align:center;}
    </style>
  </head>
  <body>
    <h1>Selamat Datang!</h1>
    <p>~ Ini webpage paling unyu sedunia ~</p>
  </body>
</html>
```

# express

## Serving HTML File #Static Req & Res

### *satu.js*

```
var express = require('express');  
var app = express();  
  
app.get('/', function(req, res){  
    res.sendFile(__dirname + '/satu.html');  
})  
  
app.listen(3000);
```

*\* satu.html & satu.js ada dalam 1 folder!*



<% Embedded JavaScript %>

# Embedded JavaScript

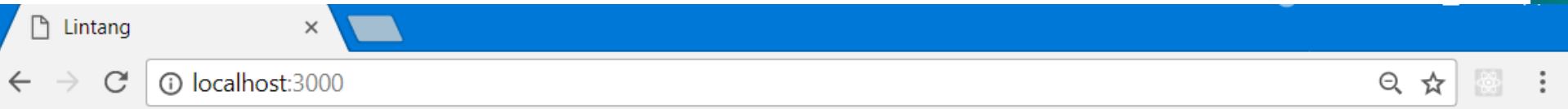
**EJS** is a package of client side templating language (*view engine*) that was originally part of JavaScriptMVC, which has now been replaced by DoneJS. EJS combines data & a template to produce HTML. JavaScript between **<% %>** will be executed. JavaScript between **<%= %>** adds HTML to the result (see [embeddedjs.com](http://embeddedjs.com)).

## ■ Installation

**npm install ejs --save**

localhost:3000/

Serving EJS File  
#Static Req & Res



# SELAMAT DATANG!

~ Ini webpage paling unyu sedunia ~



// Buat 'satu.ejs' dalam folder 'views'

*satu.ejs*

Serving EJS File  
#Static Req & Res

```
<!DOCTYPE html>
<html>
  <head><title>Lintang</title>
    <style>
      body{background:green; font-family:verdana;
        color:white; padding:150px;}
      h1{font-size:72px; text-transform:uppercase;
        letter-spacing:2px; text-align:center;}
      p{font-size:32px; text-align:center;}
    </style>
  </head>
  <body>
    <h1>Selamat Datang!</h1>
    <p>~ Ini website paling unyu sedunia! ~</p>
  </body>
</html>
```

### *satu.js*

```
var express = require('express');  
var app = express();
```

```
app.set('view engine' , 'ejs');  
// buat folder 'views'!
```

```
app.get('/', function(req, res){  
    res.render('satu');  
})  
// merujuk ke file 'satu' dalam views.
```

```
app.listen(3000);
```

localhost:3000/id/Lintang

Serving HTML File  
#Dynamic Req & Res

← → ↻ ⓘ localhost:3000/id/Lintang ☆ ⋮

# WELCOME LINTANG

# Ini halaman profil **Lintang** #

// Buat 'satu.ejs' dalam folder 'views'

## *satu.ejs*

Serving HTML File  
#Dynamic Req & Res

```
<!DOCTYPE html>
<html>
  <head><title>Lintang</title>
    <style>
      body{background:green; font-family:verdana;
        color:white; padding:150px;}
      h1{font-size:72px; text-transform:uppercase;
        letter-spacing:2px; text-align:center;}
      p{font-size:32px; text-align:center;}
    </style>
  </head>
  <body>
    <h1>Welcome <%= orang %></h1>
    <p># Ini halaman profil <b><%= orang %></b> #</p>
  </body>
</html>
```

*satu.js*

```
var express = require('express');  
var app = express();
```

```
app.set('view engine' , 'ejs');  
// buat folder 'views'!
```

```
app.get('/id/:nama', function(req, res){  
    res.render('satu', {orang:req.params.nama});  
})  
// merujuk ke file 'satu' dalam views.
```

```
app.listen(3000);
```

localhost:3000/id/Lintang

Serving HTML File  
#Dynamic Req & Res

← → ↻ ⓘ localhost:3000/id/Lintang ☆ ⋮

# WELCOME LINTANG

# Ini halaman profil **Lintang** #

### *satu.js*

```
var express = require('express');  
var app = express();  
  
app.set('view engine' , 'ejs');  
  
app.get('/', function(req, res){  
    var orangjs = {nama:'Andi', usia:24};  
    res.render('satu', {orangejs: orangjs});  
})  
  
app.listen(3000);
```

*satu.ejs*

## Inject an Object

```
<body>
```

```
  <h1>Welcome
```

```
  <%= orangejs.nama %></h1>
```

```
  <p># Usia Anda <b>
```

```
  <%= orangejs.usia %>
```

```
  </b> th #</p>
```

```
</body>
```



localhost:3000

Inject an Object



**WELCOME ANDI**

# Usia Anda sudah **24** tahun #

### *satu.js*

```
var express = require('express');
var app = express();

app.set('view engine' , 'ejs');

app.get('/', function(req, res){
  var orangjs = ['Adi', 'Budi', 'Caca'];
  res.render('satu', {orang: orangjs});
})

app.listen(3000);
```

*satu.ejs*

```
<body>
```

```
<h1>Welcome!</h1>
```

```
<% for(i=0;i<3;i++){ %>
```

```
<p> <%= orang[i]; %> </p>
```

```
<% } %>
```

```
</body>
```

localhost:3000

Inject JS code

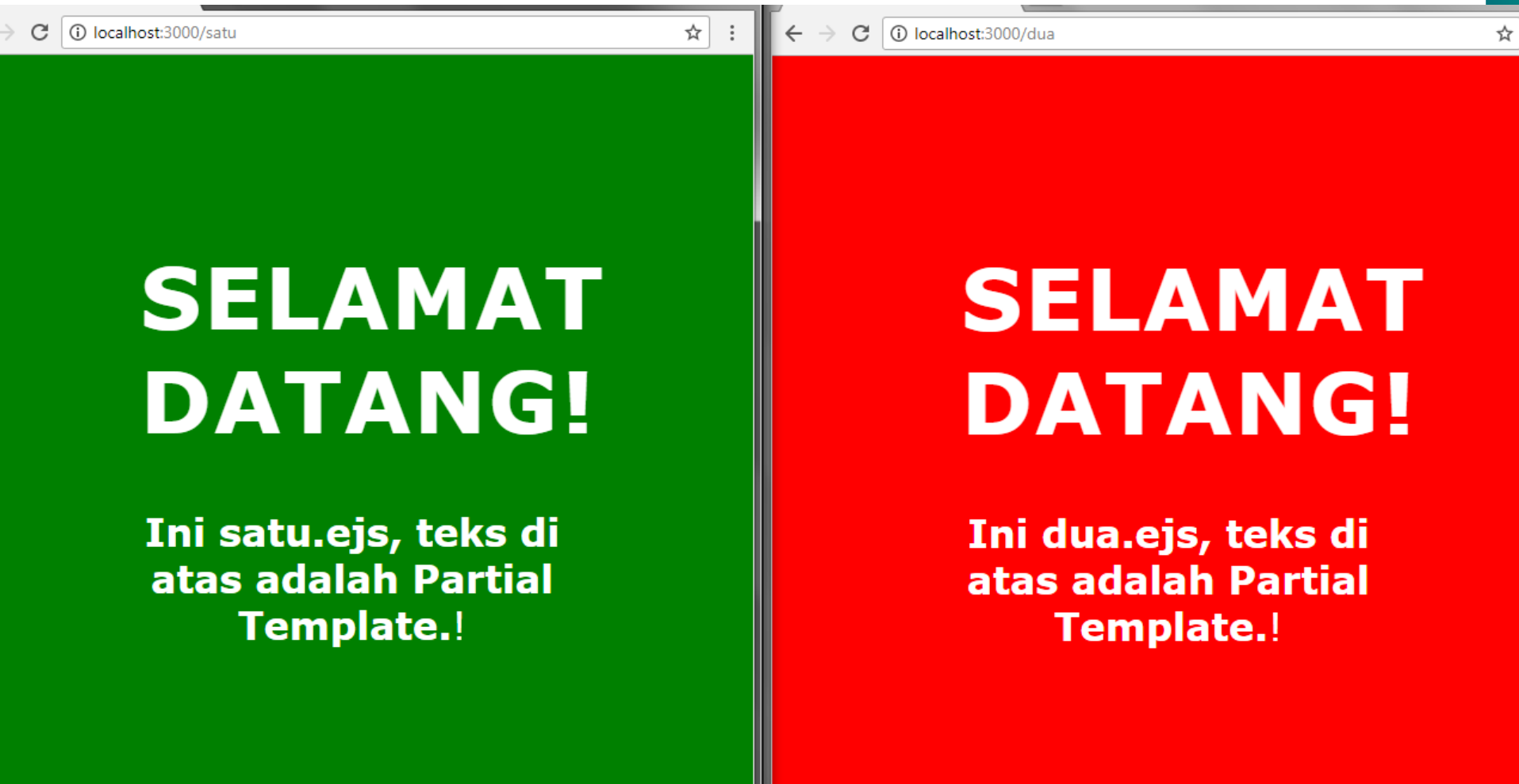
# WELCOME!

Adi

Budi

Caca

localhost:3000/satu & localhost:3000/dua



**Partial Template**

*judul.ejs*

<h1>Selamat Datang!</h1>

// create **judul.ejs** on dir **views/parsial**

*satu.ejs*

`<body>`

`<% include parsial/judul.ejs %>`

`<b><p>`Ini **satu.ejs**, teks di atas adalah Partial Template`</b>`!

`</body>`

*dua.ejs*

`<body>`

`<% include parsial/judul.ejs %>`

`<b><p>`Ini dua.ejs, teks di atas  
adalah Partial Template`</b>`!

`</body>`



# *satu.js*

## Partial Template

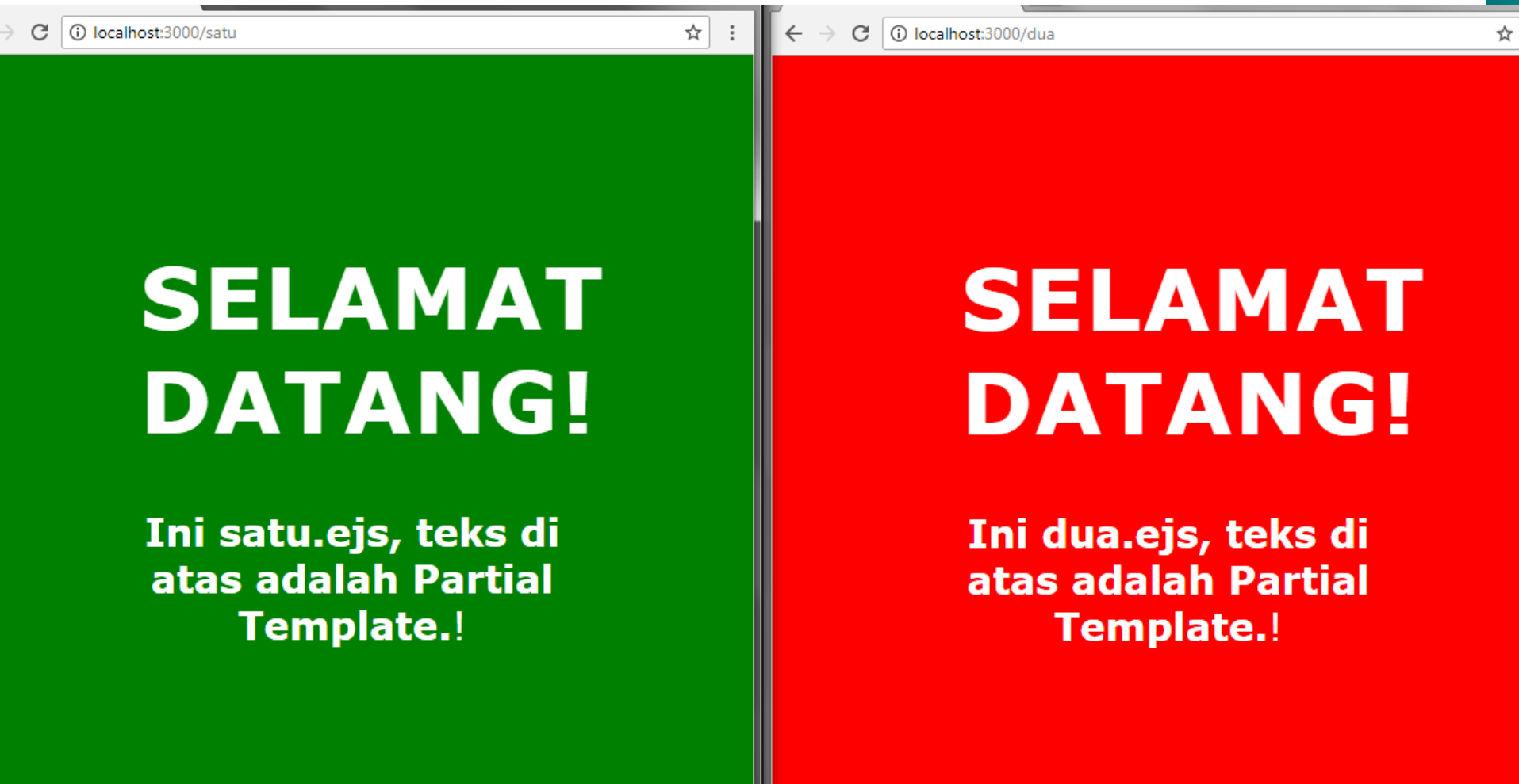
```
var express = require('express');
var app = express();
app.set('view engine', 'ejs');

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

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

app.listen(3000);
```

localhost:3000/satu & localhost:3000/dua



**Partial Template**

# 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 a route
//      'assets' = it's a ref dir

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

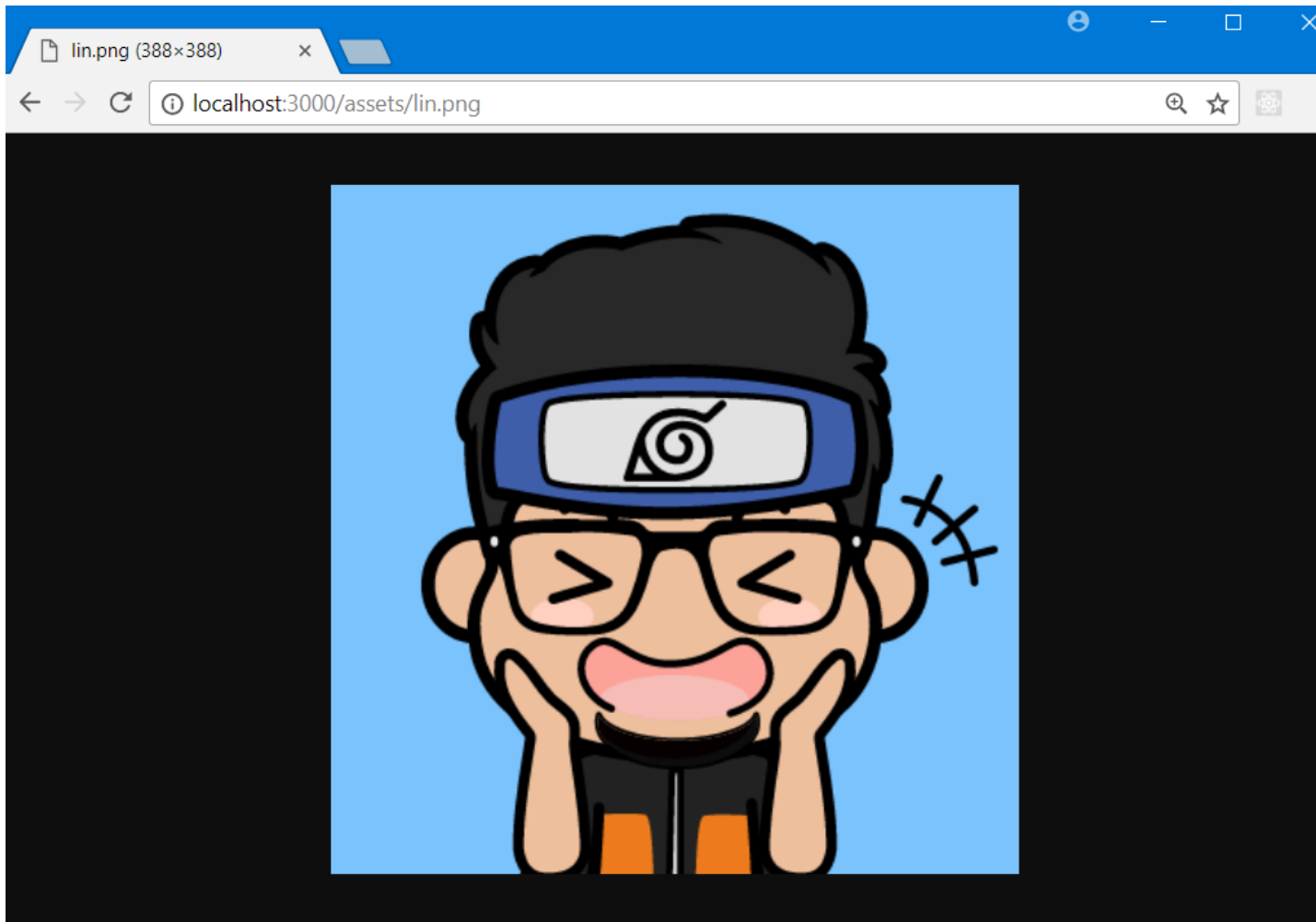
app.listen(3000);
```

# 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!

# Query String

*Query string* is the part of a uniform resource locator (URL) containing data that does not fit conveniently into a hierarchical path structure.

- `http://123.com/news?page=2`
- `http://xyz.com/a?x=1234&y=5678`

*satu.js*

```
var express = require('express');  
var app = express();
```

```
app.set('view engine', 'ejs');
```

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

```
app.listen(3000);
```



*satu.ejs*

```
<body>  
  <h1>Welcome! </h1>  
  <p>Selamat datang ~</p>  
</body>
```

# localhost:3000/?nama=Iwa%20K&usia=52

zzz - Visual Studio Code

Selection View Go Debug Tasks Help

JS dua.js

<> satu.html

JS formulir.js

```
1 var express = require('express');
2 var app = express();
3
4 app.set('view engine', 'ejs');
5
6 app.get('/', function(req, res){
7   console.log(req.query)
8   res.render('satu');
9 })
10
11 app.listen(3000);
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

```
[nodemon] starting `node satu dua.js`
{ nama: 'Andi', usia: '23' }
{ nama: 'Sari', usia: '13' }
{ nama: 'Wildan', usia: '26' }
{ nama: 'Ashanti Pasaribu', usia: '32' }
{ nama: 'Hanni Nuzula', usia: '9' }
{ nama: 'Raisa', usia: '29' }
{ nama: 'Tito', usia: '49' }
{ nama: 'Lintang Wisesa', usia: '24' }
```

body-parser

Lintang

Query string - Wikipedia

localhost:3000/?nama=Lintang%20Wisesa&usia=24

# WELCOME!

Selamat datang ~

# Query to Server & Res HTML page

*satu.js*

```
var express = require('express');  
var app = express();  
  
app.set('view engine', 'ejs');  
  
app.get('/', function(req, res){  
    console.log(req.query)  
    res.render('satu', {qs: req.query});  
})  
  
app.listen(3000);
```

*satu.ejs*

## Query to Server & Res HTML page

<body>

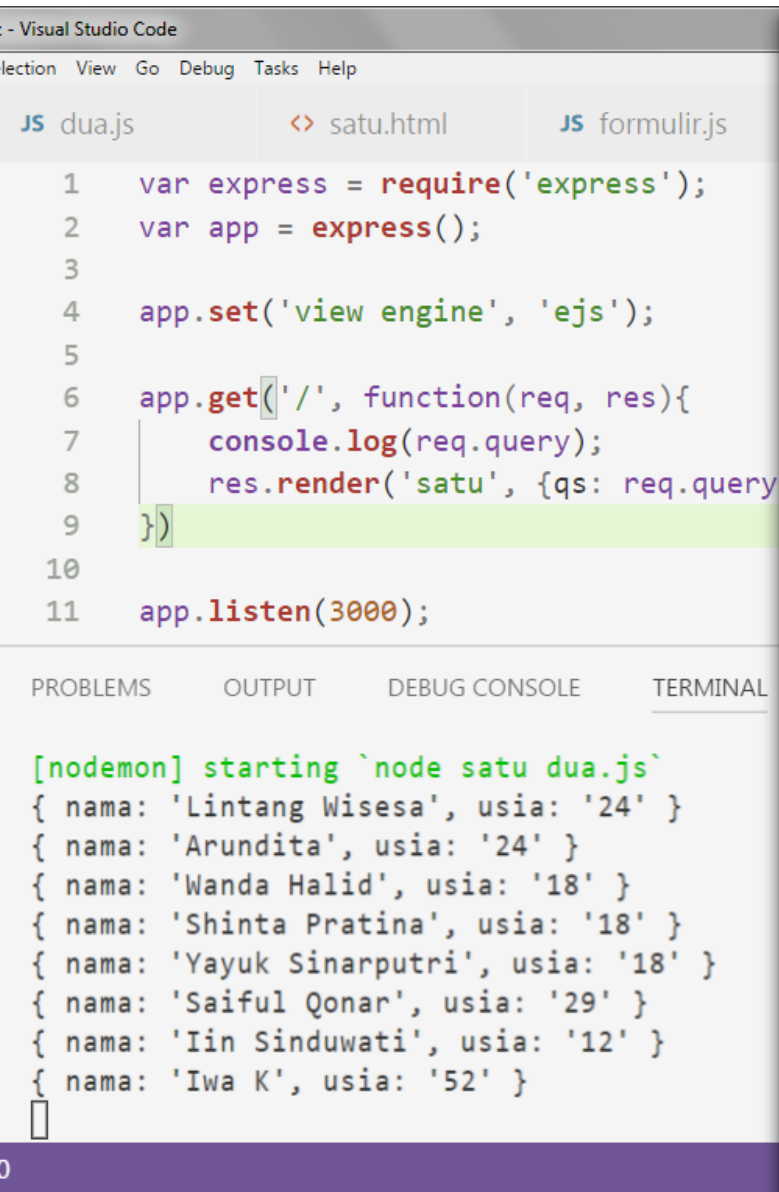
<h1>Welcome!</h1>

<p>Hai kak <b><%= qs.nama %></b>!

<p>Usiamu <b><%= qs.usia %></b> th!

</body>

# localhost:3000/?nama=Iwa%20K&usia=52

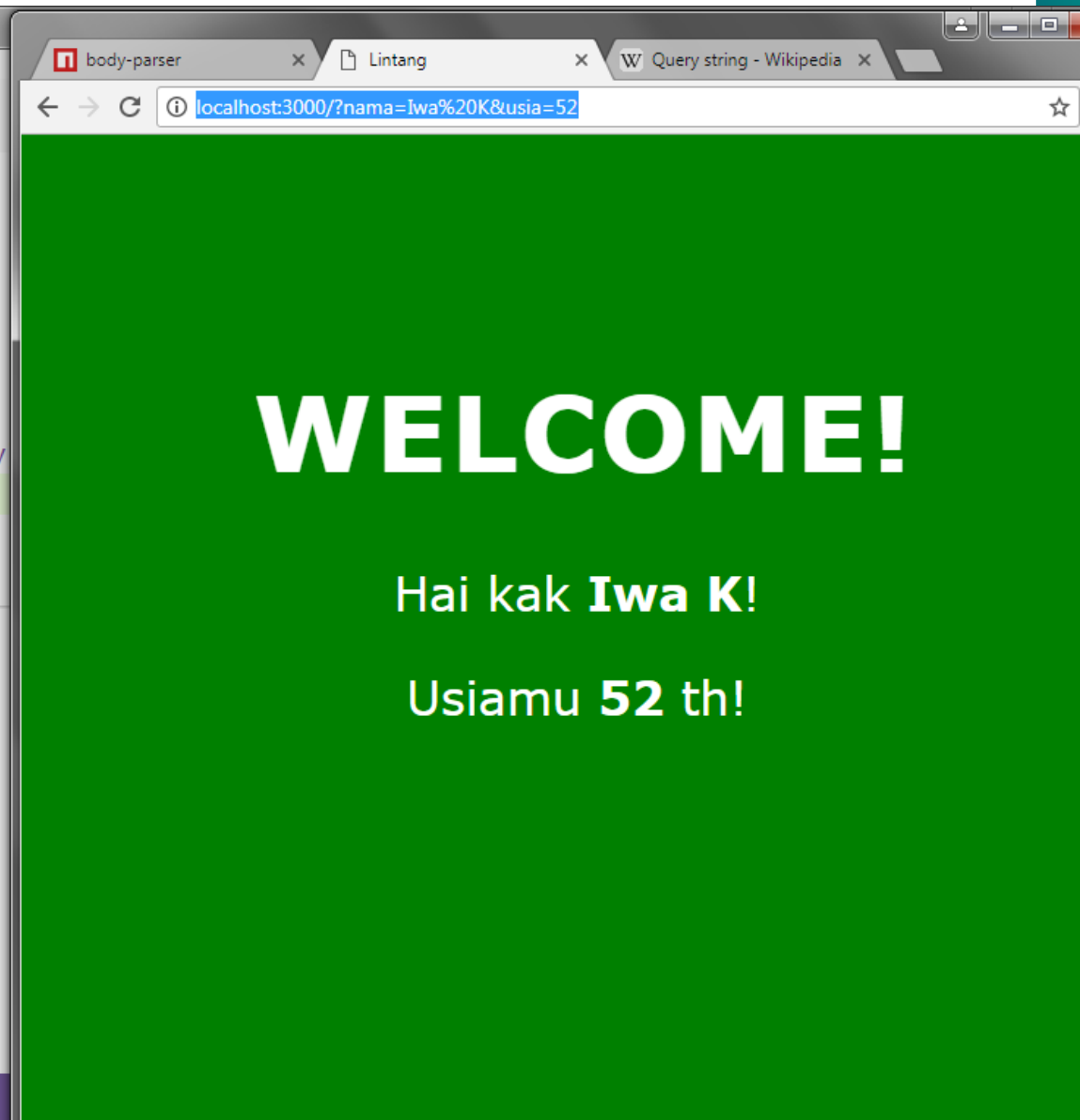


The image shows a Visual Studio Code editor with three files open: `dua.js`, `satu.html`, and `formulir.js`. The `dua.js` file contains the following JavaScript code:

```
1 var express = require('express');
2 var app = express();
3
4 app.set('view engine', 'ejs');
5
6 app.get('/', function(req, res){
7   console.log(req.query);
8   res.render('satu', {qs: req.query});
9 })
10
11 app.listen(3000);
```

The terminal at the bottom shows the output of the application:

```
[nodemon] starting `node satu dua.js`
{ nama: 'Lintang Wisesa', usia: '24' }
{ nama: 'Arundita', usia: '24' }
{ nama: 'Wanda Halid', usia: '18' }
{ nama: 'Shinta Pratina', usia: '18' }
{ nama: 'Yayuk Sinarputri', usia: '18' }
{ nama: 'Saiful Qonar', usia: '29' }
{ nama: 'Iin Sinduwati', usia: '12' }
{ nama: 'Iwa K', usia: '52' }
```




# 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
```

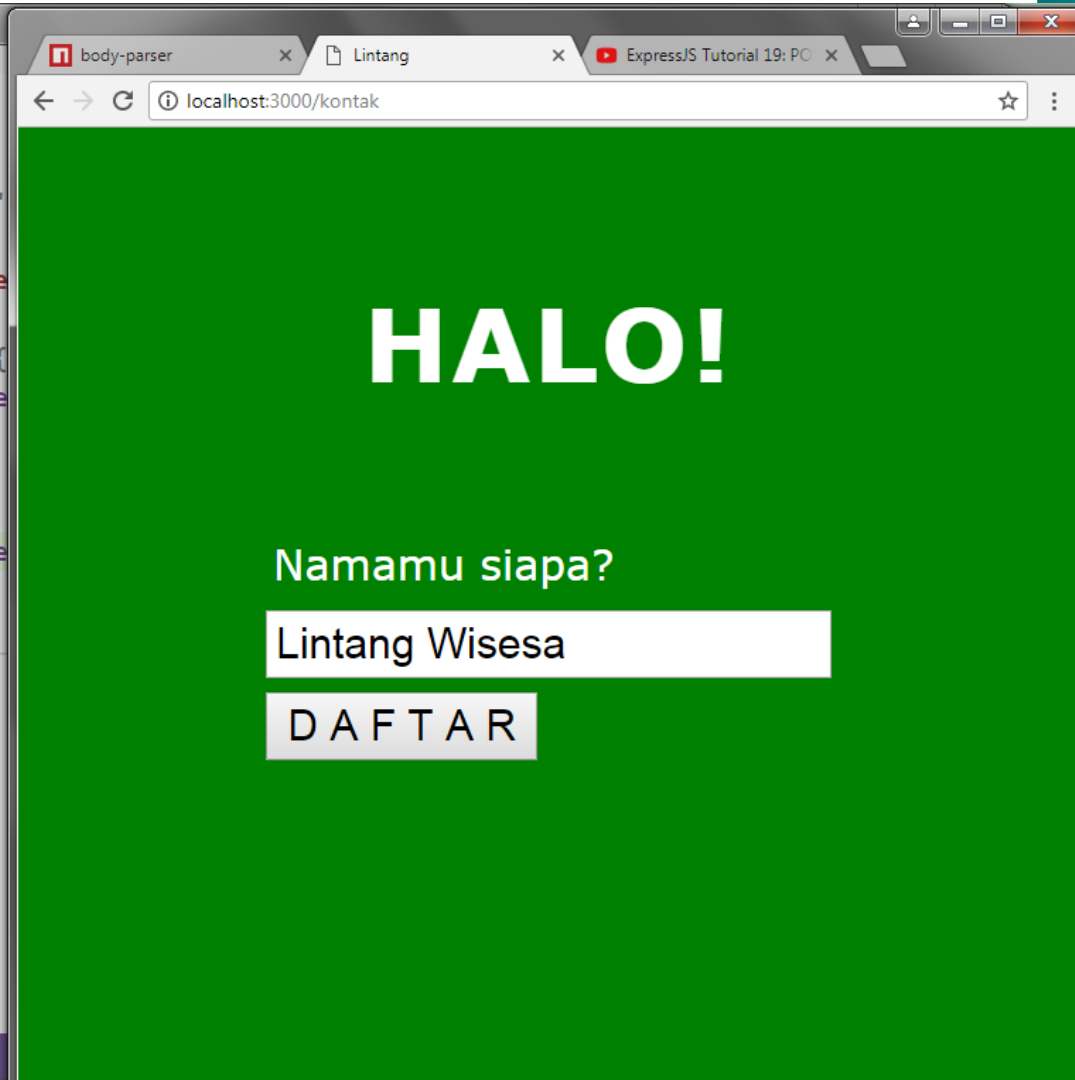
# localhost:3000/kontak



```
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' }
```



# *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

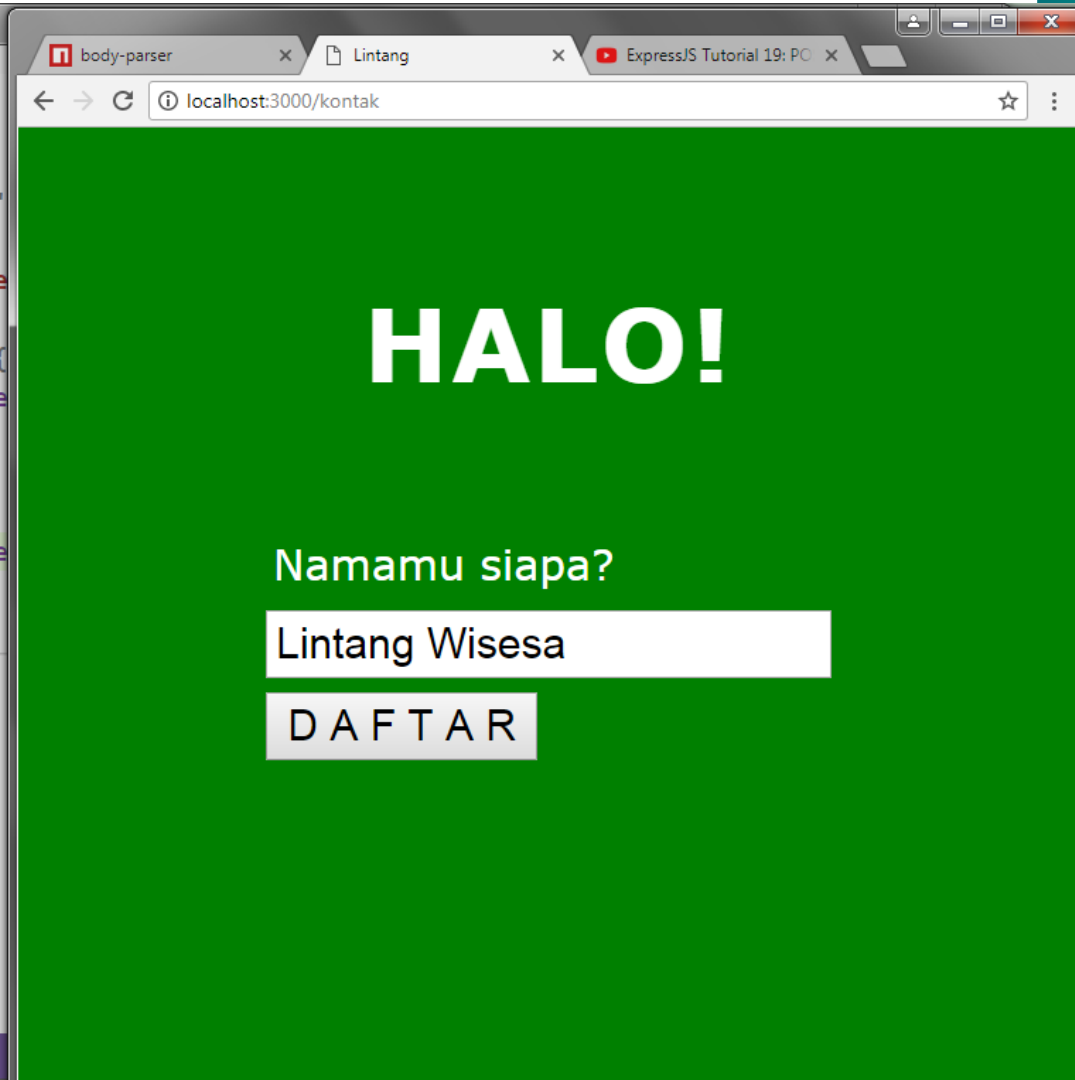


The screenshot shows the Visual Studio Code editor with the file `formulir.js` open. The code is a Node.js application using Express and body-parser. It defines a GET route for `/kontak` that renders a view named `formulir` with a query object. The query object is populated with an array of contact data. The application listens on port 3000. The terminal output shows the application starting successfully and displaying the contact data.

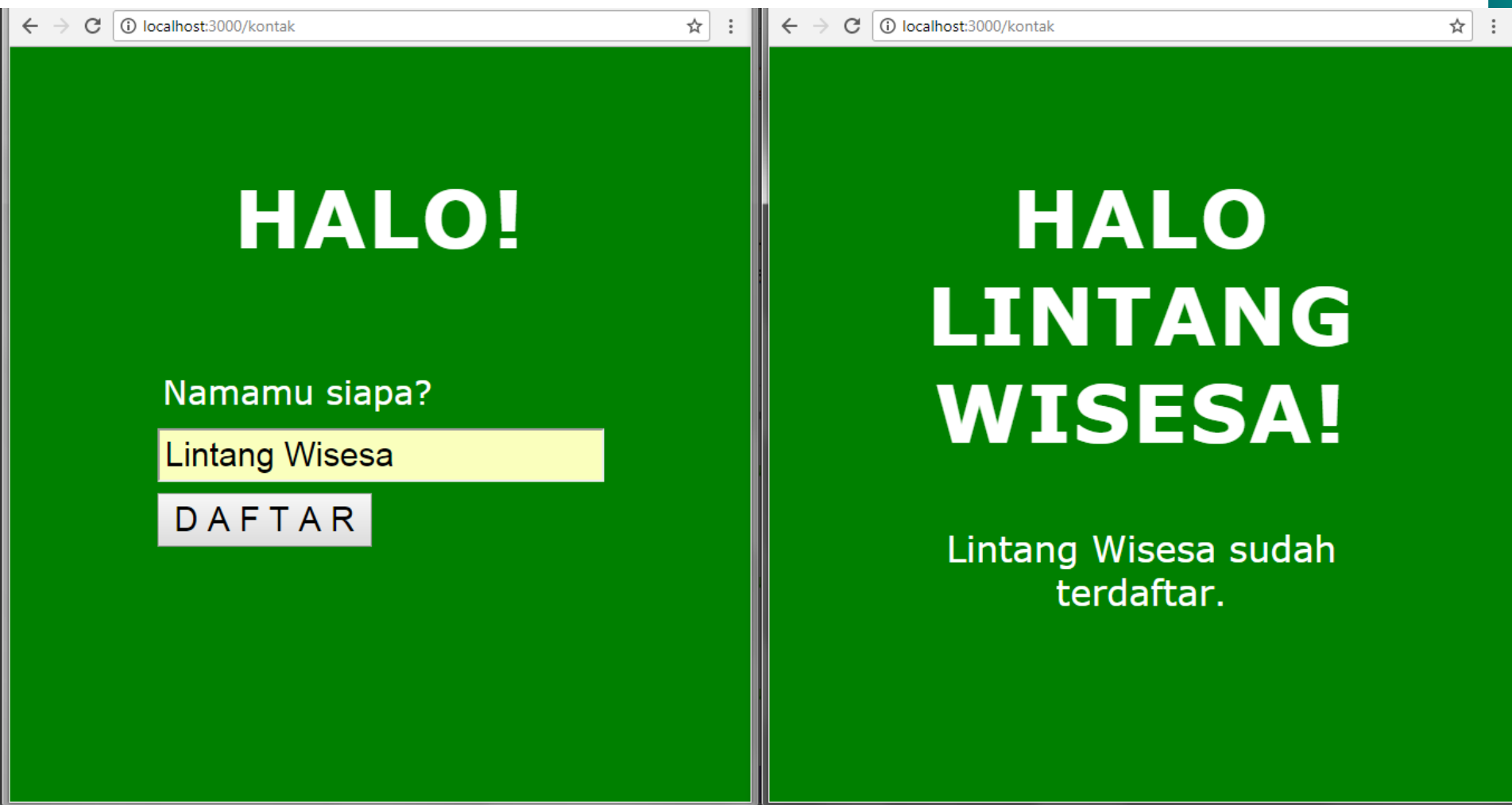
```
1 var express = require('express');
2 var bodyParser = require('body-parser');
3 var app = express();
4 var urlencodedParser = bodyParser.urlencoded({ extended: false });
5 app.set('view engine', 'ejs');
6 app.get('/kontak', function(req, res){
7   res.render('formulir', {qs:req.query});
8 })
9 app.post('/kontak',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



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

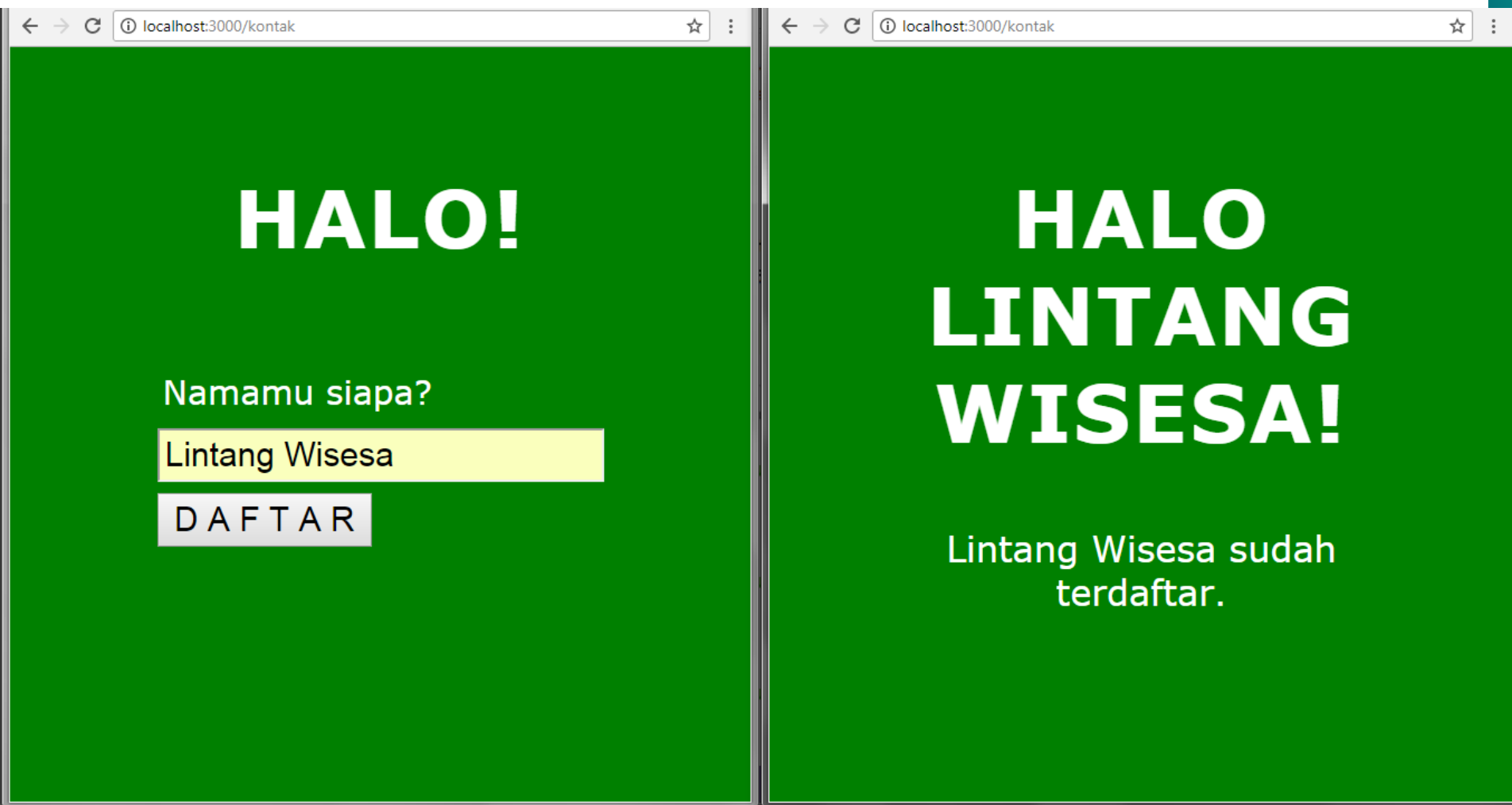
```
var express = require('express');
var bodyParser = require('body-parser')
var app = express();
var url = bodyParser.urlencoded({ extended: false })
app.set('view engine', 'ejs');

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