

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

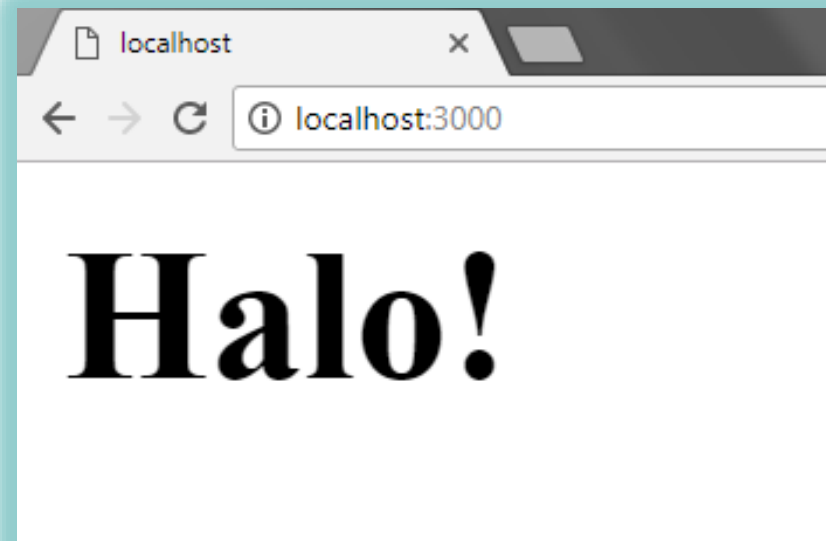
```
var express = require('express');  
var app = express();
```

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

// GET request

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

// Listening port 3000



express

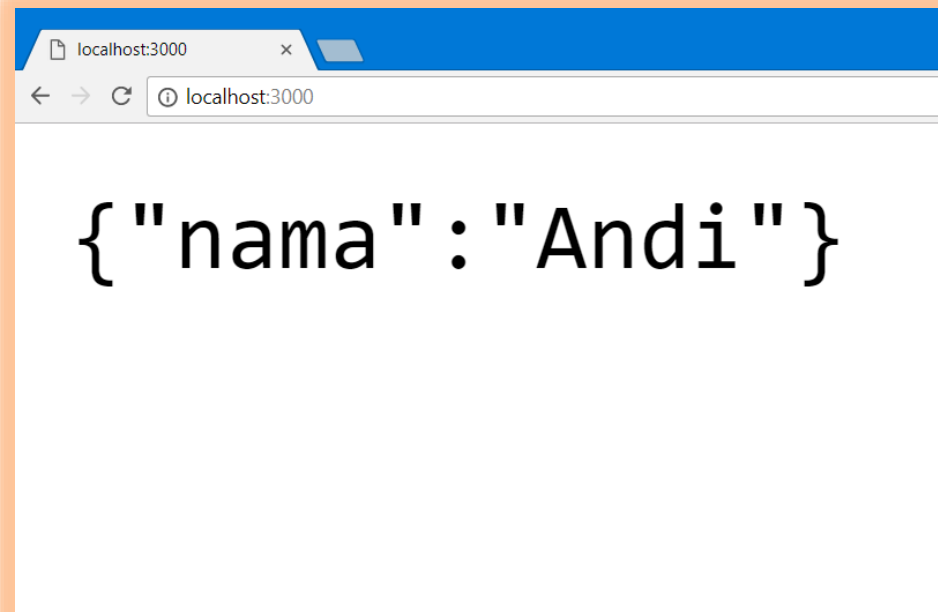
Serving JSON

```
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
```

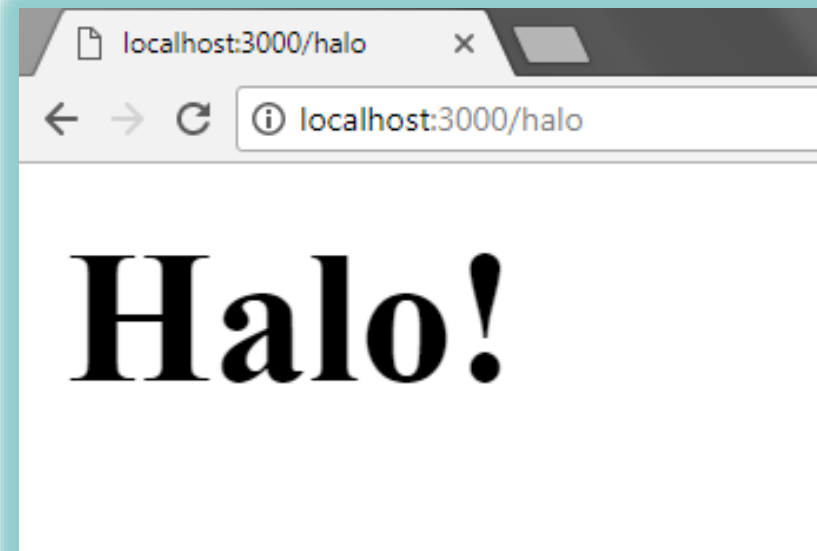
```
app.listen(3000);  
// Listening port 3000
```



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
```

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!*

EJS

<% Embedded JavaScript %>

Embedded JavaScript

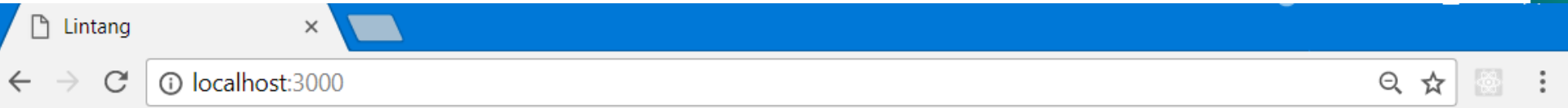
EJS is a package of client side templating language (***view engine***) that was originally part of JavaScriptMVC, which has now been replaced by DoneJS. EJS combines data & a template to produce HTML. JavaScript between **<% %>** will be executed. JavaScript between **<%= %>** adds HTML to the result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

■ 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



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



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

EJS

<% Embedded JavaScript %>

Inject JS code

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

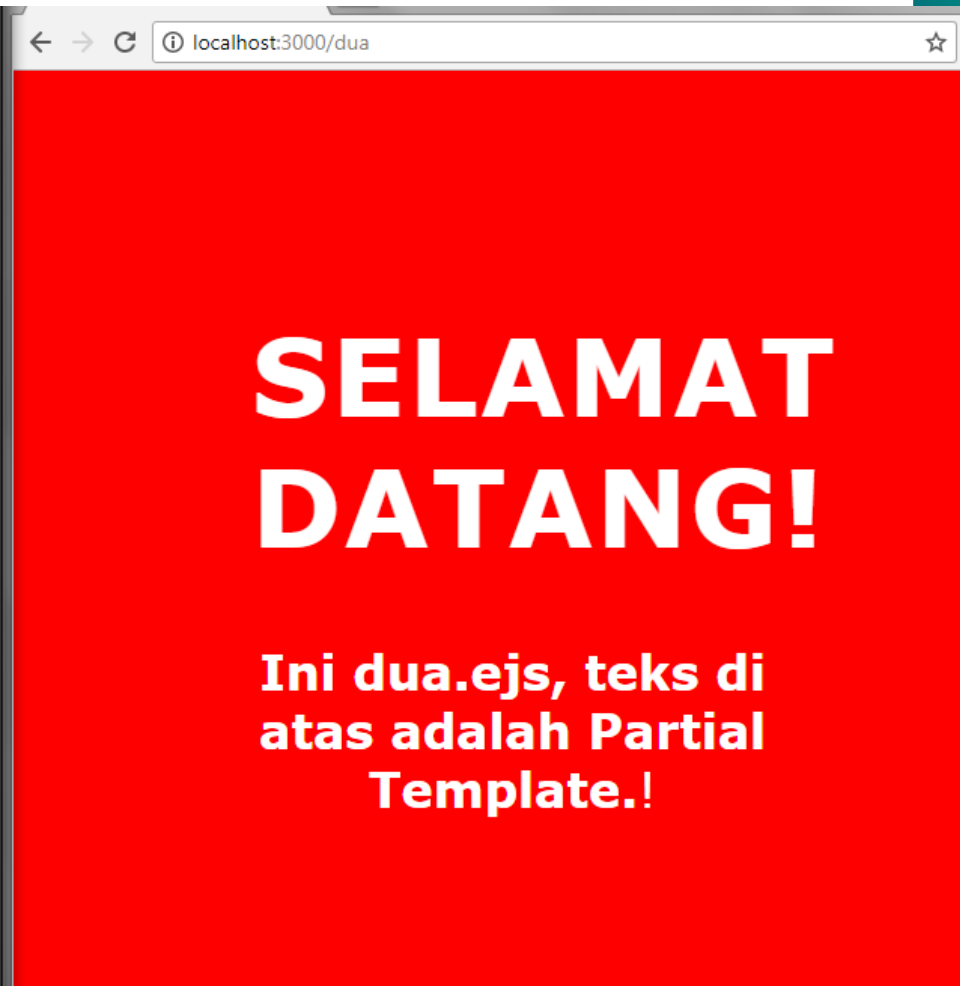
WELCOME!

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 %>`

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

`</body>`

dua.ejs

`<body>`

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

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

`</body>`

satu.js

Partial Template

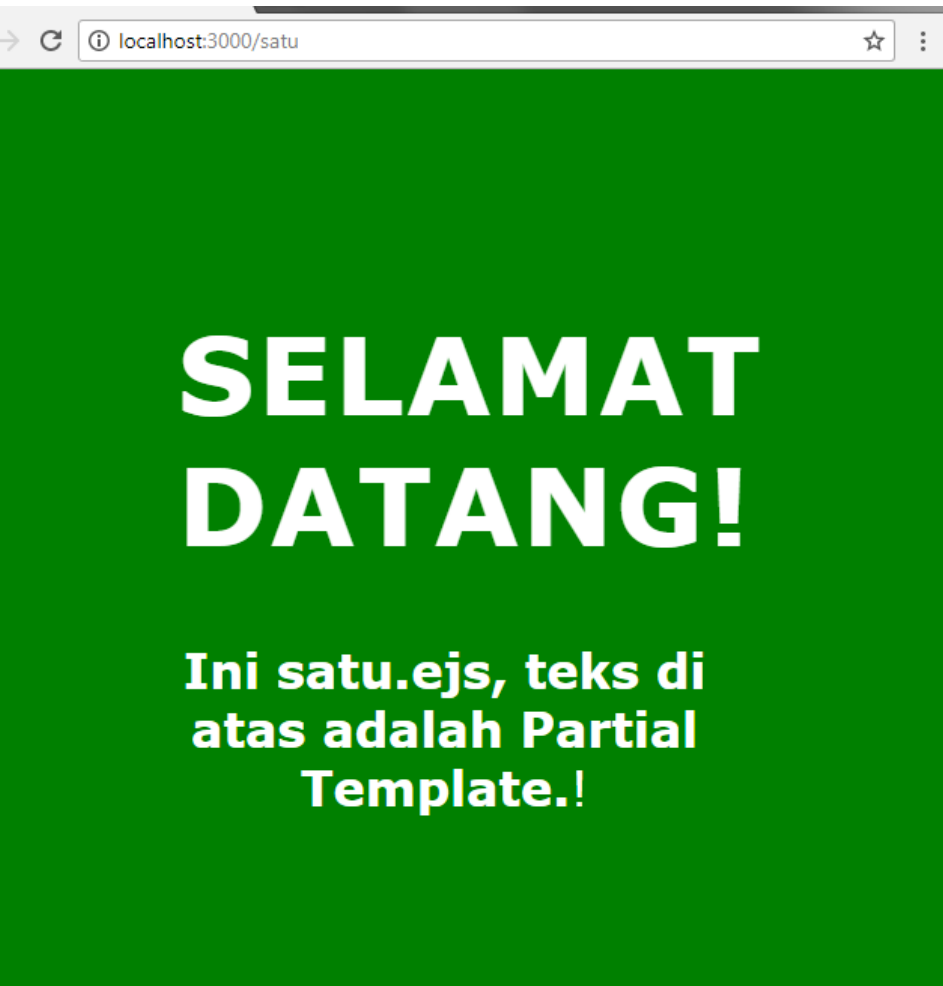
```
var express = require('express');
var app = express();
app.set('view engine', 'ejs');

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

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

zzy - Visual Studio Code

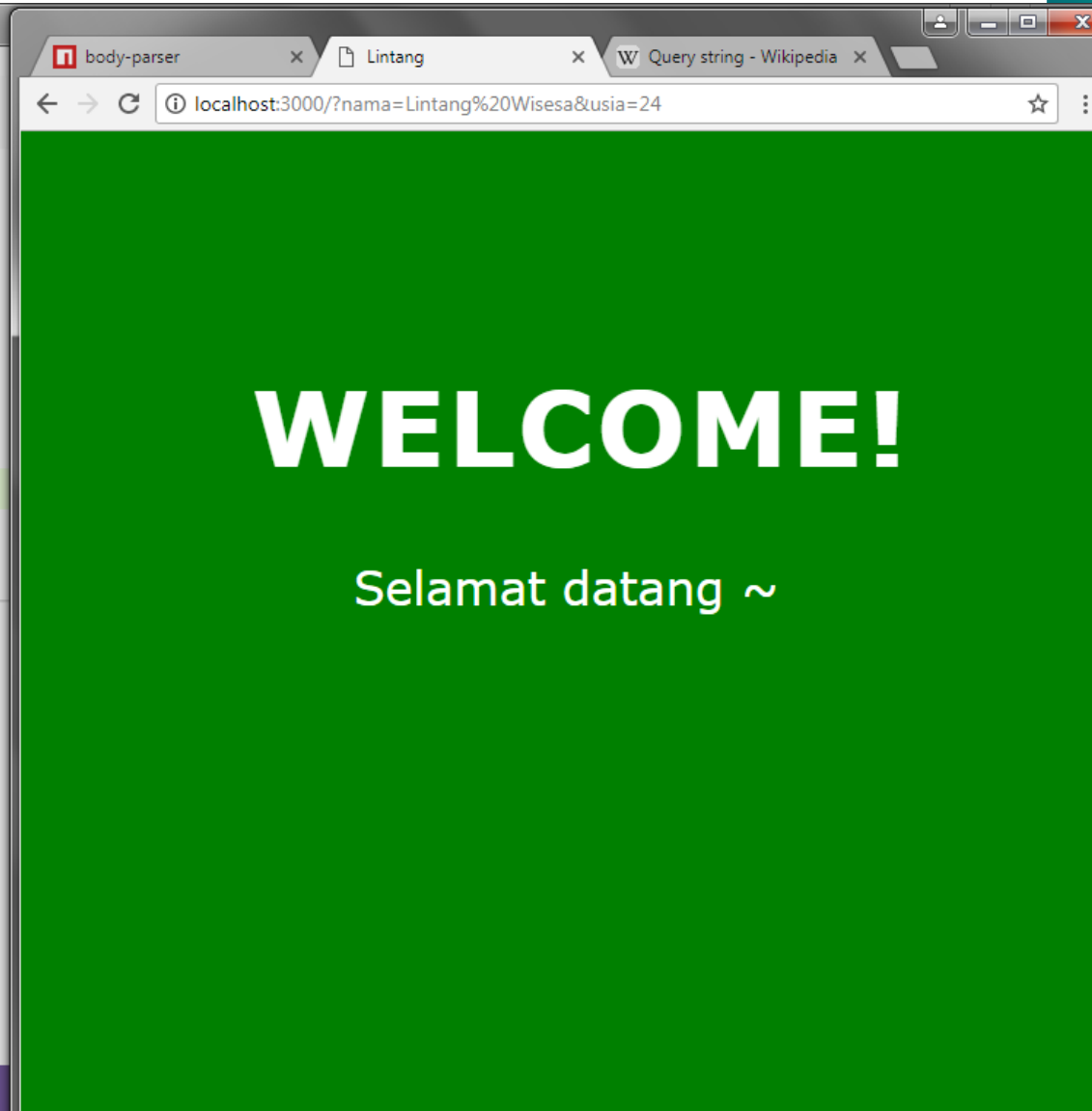
Selection View Go Debug Tasks Help

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' }
```



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>

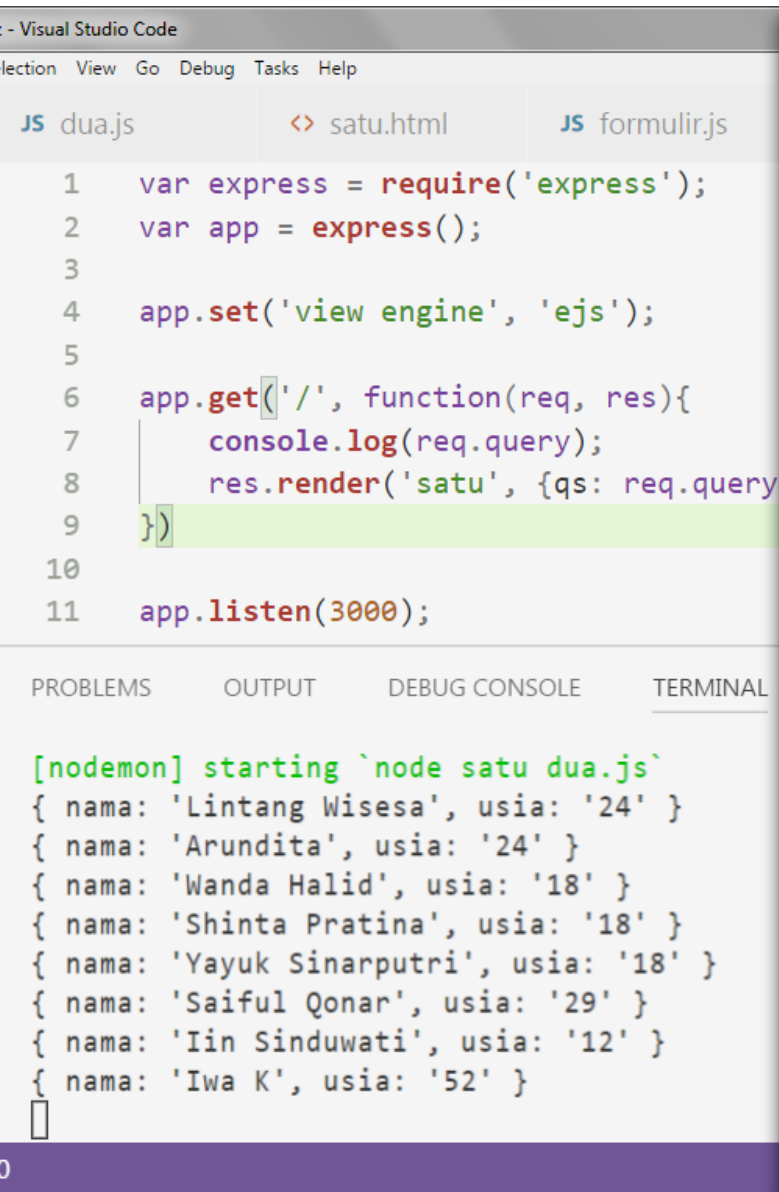
<h1>Welcome!</h1>

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

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

</body>

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

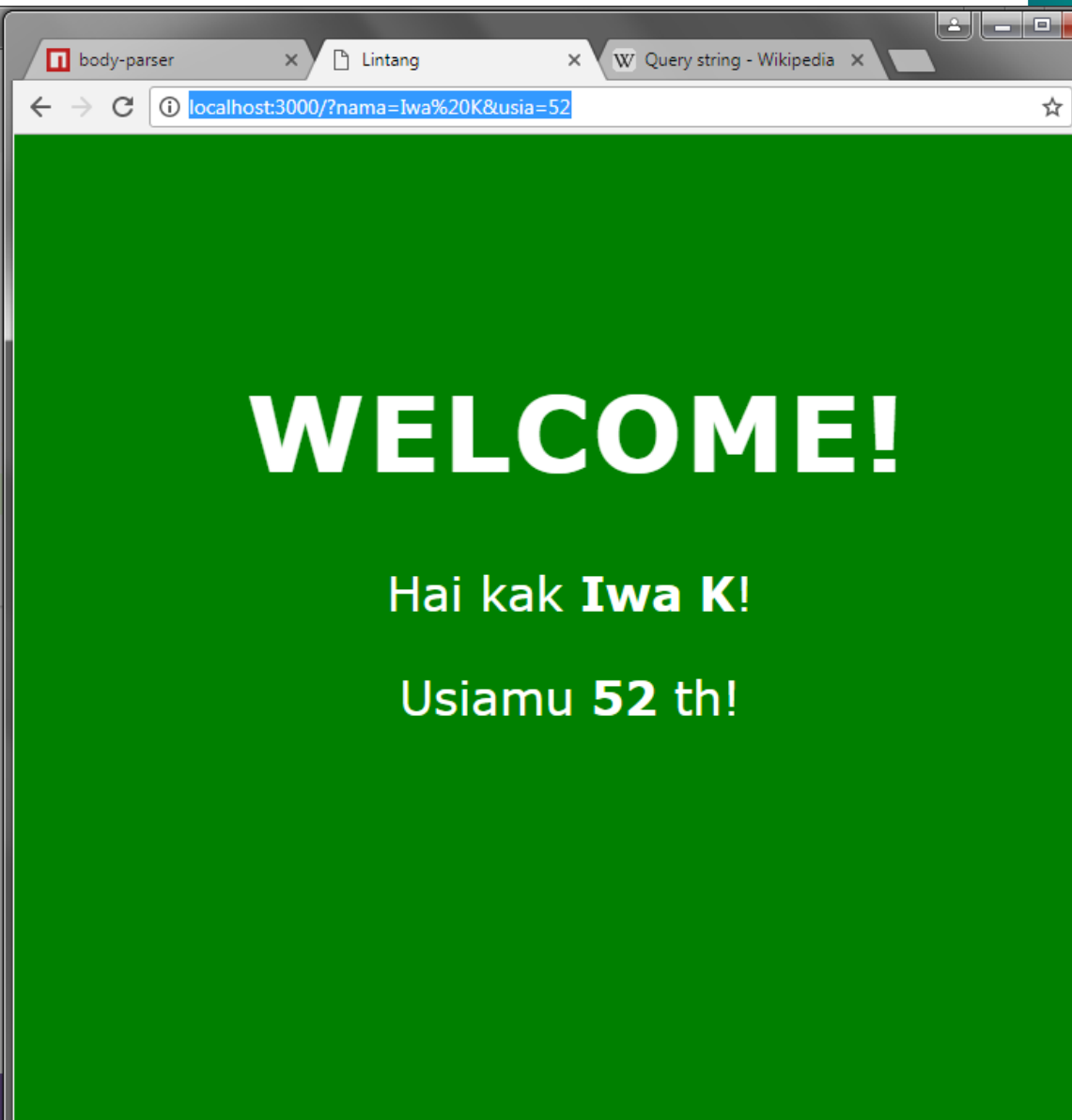


The screenshot shows the Visual Studio Code editor with a file named `dua.js` open. The code is a simple Express.js application that listens on port 3000 and renders a view named 'satu' based on the query string. The terminal output shows the application starting and logging the query parameters for several requests, including the one for 'Iwa K' with age 52.

```
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);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
[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' }
```



Back-End Development

express

Exploring

#6 Express Module