

## URL

Contoh penggunaan **nodemon**:

...

Bagian yang terpenting dalam sebuah website adalah URL, agar bisa tampil di halaman website. Dari web URL diatur disini (Routing). Dan disinilah fungsi dari HTTP Request yang telah kita pelajari sebelumnya (GET, POST, PUT, DELETE)

-> Tulis code ini diatasnya **app.listen**

```
app.get('/', function(req, res, next){  
  console.log('Home page')  
});
```

-> Arti code tsb diatas pada dasarnya adalah melakukan **Get Request!** ke **Server!**.

-> '/' adalah URL root !

-> Buka localhost:3000 hanya halaman kosong dan cek pada terminal as console.log.

-> Kemudian ganti dengan:

```
res.json('Halaman Beranda')
```

-> Membuat halaman baru:

```
app.get('/products', function(req, rest, next)){
```

```
    res.json("Supercar", "Superman", "Super web dev")
  }
```

Kemudian akses dengan localhost:3000/products.

## Dinamis URL

```
app.get('/:name', function(req, res, next)){
  res.json(req.params.name)
};
```

-> :name disebut placeholder, placeholder memuat **parameter** Kita bisa menggantikan placeholder dengan apapun, dan bila kita response dengan **req.params.placeholder** maka akan tampilkan placeholder tsb.

```
app.get('/:iwanfalse', function(req, res, next)){
  res.json(req.params.name);
  res.json(req.params.iwanfalse);
};
```

## Middleware

Middleware adalah library yang digunakan untuk membuat sebuah kondisi yang harus dipenuhi sebelum melanjutkan tujuan yang ingin dicapai, apabila syarat tidak terpenuhi maka akan dialihkan seperti dengan "else".

Misalnya anda dari Jakarta ingin ke Bandung, jika ingin masuk tol ternyata tidak memiliki uang untuk membayar tol maka anda tidak akan bisa sampai ke bandung (melalui tol).

Dalam code kita tuliskan seperti ini :

```
app.get('/bandung', function(req, res, next){
    res.json("Silahkan anda menuju ke Bandung")
});
```

Nah middleware nya apa??? -> "Tol" sbb:

```
app.use(function(req, res, next){
    if (0 < 50){
        next();
    } else {
        res.json("Maaf anda tidak bisa melanjutkan ke
Bandung via Tol")
    }
});
```

Apabila menggunakan middleware **app.use** tempatkan middleware diatas code tujuan (URL Target), seperti dibawah ini:

```
app.use(function(req, res, next){
    if (0 < 50){
        next();
    } else {
        res.json("Maaf anda tidak bisa melanjutkan ke
Bandung via Tol")
    }
});

app.get('/bandung', function(req, res, next){
    res.json("Silahkan anda lanjutkan perjalanan anda
via Tol ke Bandung")
});
```

Test di browser localhost:3000/bandung.

-> Ganti kondisi nya menjadi false.

-> Coba dengan menghilangkan next(), akan loops forever.

-> Untuk menghentikan loop ctrl c server dan start lagi dengan nodemon server.

Dalam menggunakan middleware anda gunakan app.use dan next(). next() adalah "callback", jika dalam suatu kondisi terpenuhi go to next routes.

## Cara ke dua menuliskan middleware

-> Menggunakan function :

```
function travelingKeBandung(req, res, next){
  if (req.params.uang === "oke") {
    next();
  } else {
    res.json("Anda tidak cukup punya uang untuk
traveling ke Bandung")
  }
}

app.get('/bandung/:uang', travelingKeBandung,
function(req, res, next){
  res.json("Silahkan anda traveling ke Bandung")
});
```

-> Test di browser localhost:3000/bandung/oke dan no

Jadi anda punya 2 cara dalam menuliskan middleware, menggunakan app.use atau dengan membuat function tersendiri.

Copyright ©GandrungStudio 2018 Dilarang memperbanyak dan mendistribusikan Materi Training ini tanpa izin dari PT.Gandrung Media Corporation. Pelanggaran akan hak cipta ini akan kami ajukan ke Institusi Hukum yang berlaku.