



Penanganan Error dan Template Engine

Express Error Handler

- Error Handling refers to how Express catches and processes errors that occur both synchronously and asynchronously. Express comes with a default error handler so you don't need to write your own to get started.

```
app.get('/sync_broken', function (req, res) {  
  /* synchronous error */  
  throw new Error('BROKEN') // Express will  
  catch this on its own.  
})
```

```
app.get('/asyn_broken', function (req, res, next) {  
  /* Asynchronous error */  
  fs.readFile('/file-does-not-exist', function  
  (err, data) {  
    if (err) {  
      next(err) // Pass errors to Express.  
    } else {  
      res.send(data)  
    }  
  })  
})
```

Express Error Handler for Next Processing

```
app.get('/', function (req, res, next) {  
  setTimeout(function () {  
    try {  
      throw new Error('BROKEN')  
    } catch (err) {  
      next(err)  
    }  
  }, 100)  
})
```

Create Default Error Handler

```
app.use(function (err, req, res, next) {  
  console.error(err.stack)  
  res.status(500).send('Something broke!')  
})
```

Template Engine

- Software yang dirancang untuk menggabungkan template dengan model data untuk menghasilkan dokumen hasil
- Express support template engine : Pugjs, Ejs... [selengkapnya di daftar express template engine](#)
-

PugJs Templae Engine

- Install : `npm install pug --save`

- Config :

```
app.set('view engine', 'pug')  
app.set('views', './views')
```

- Lebih lengkap tentang pugjs.org

Membuat Template Engine Sendiri

- Konsep ?

```
app.engine('nt1', function (filePath, options, callback) { //
define the template engine
  fs.readFile(filePath, function (err, content) {
    if (err) return callback(err)

    // this is an extremely simple template engine
    console.log('options')
    console.log(options)
    var rendered = content.toString()
      .replace('#error#', options.error )
      .replace('#title#', '<title>' + options.title +
'</title>')
      .replace('#message#', '<h1>' + options.message +
'</h1>')
    return callback(null, rendered)
  })
})
app.set('view engine', 'nt1') // register the template engine
```

Happy coding