

# RESUME EKSPLORASI I EXPRESS.JS

Garmastewira

13514068

express





# PHILOSOPHY: INTRODUCTION

---

- **JavaScript** merupakan bagian dari **trio** penyokong sebuah web page, yaitu **HTML** (structure), **CSS** (presentation), dan **JS** (behavior).
- Pada tahun 2009, **Ryan Dahl** mengenalkan sebuah cross-platform runtime environment agar JS dapat digunakan diluar web browser, yaitu untuk membuat aplikasi web. Node.js akan berperan sebagai **server-side code** yang biasanya ditulis dengan bahasa **PHP, Ruby, Python**, dst.<sup>[1]</sup>
- Problem: Node.js memiliki modul I/O **low level** untuk modul HTTP. Perlu sebuah **framework** agar memudahkan dalam memanfaatkan Node.js

# PHILOSOPHY: WHAT IS EXPRESS.JS?

---

- Express.js merupakan sebuah web-application framework yang berjalan di platform Node.js untuk mengorganisasikan web-application di **server side**.<sup>[2]</sup>
- Express.js membantu pengguna untuk mengorganisasi **routing** aplikasi, **payload/body HTTP parsing**, **cookies parsing**, dan lain-lain dengan mudah.
- Sebagai perbandingan, tanpa Express.js, untuk membuat server REST API sederhana menggunakan Node.js, jumlah baris kode bisa mencapai **10x lipat** dari jumlah baris kode menggunakan Express.js.



# USAGE

---

Salah satu Solution Stack web app yang paling terkenal adalah **XAMPP**:

X : Operating System

A : Apache (Web server)

M : MySQL (Database)

P : PHP (Programming language)

P : Perl (Programming language)

Beberapa Solution Stack terkenal lainnya adalah **LAMP** (Linux) dan **WAMP** (Windows)



# USAGE

---

**Now, everything can be written in only one language: JavaScript!**

Solution Stack yang berkembang pesat saat ini: **MEAN**.

M : MongoDB

E : **Express.js**

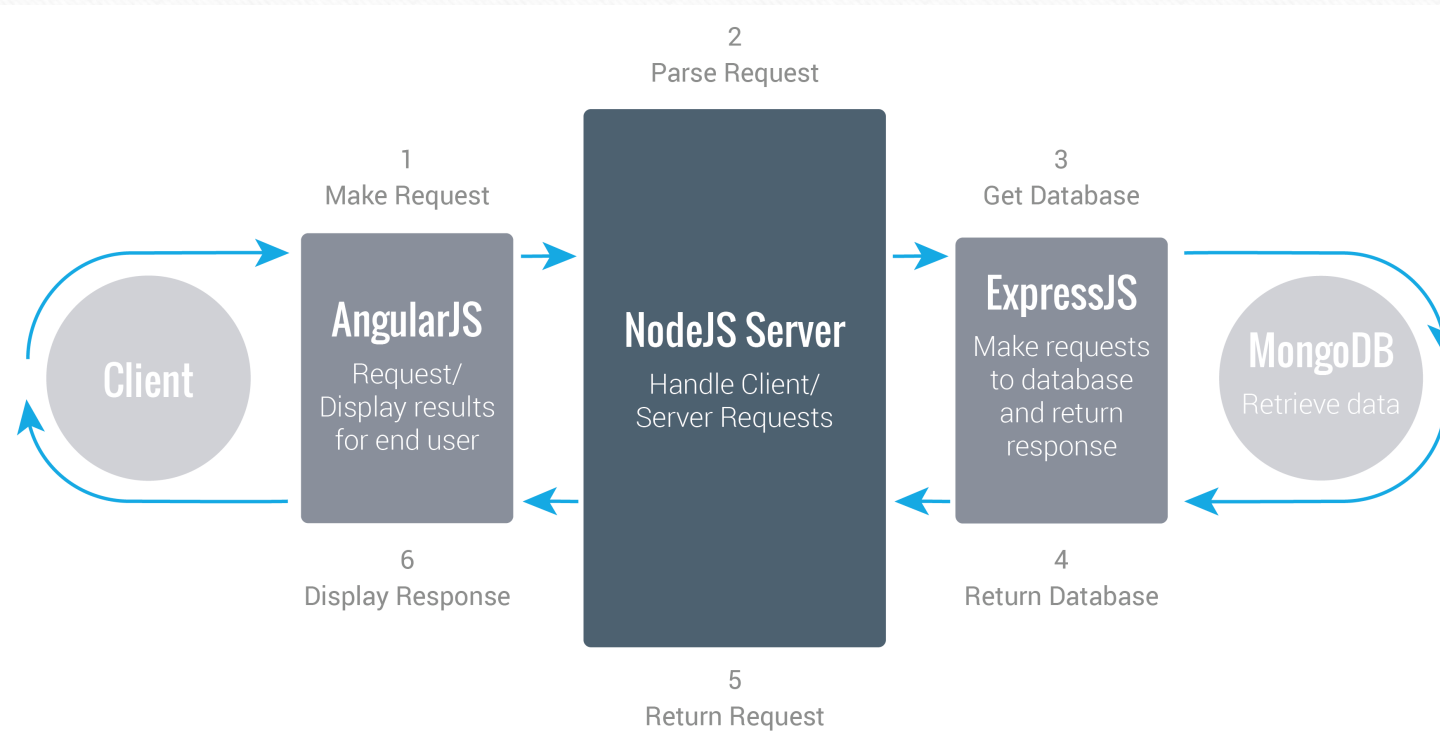
A : AngularJS

N : Node.js





# USAGE: HOW MEAN STACK WORKS<sup>[3]</sup>



# USAGE

---

Framework Express.js juga telah digunakan oleh website-website berikut:

- MySpace
- LearnBoost
- Storify
- GeekList
- Countly



# PROS

---

1. Menggunakan JavaScript yang sangat banyak diketahui orang dan mudah dipelajari.
2. Seseorang dapat mengembangkan front-end dan back-end dengan **satu bahasa** (JavaScript).
3. Streaming file dengan ukuran besar cepat.
4. Request handling secara konkuren sangat mudah (karena nature JavaScript yang I/O-nya asynchronous).
5. Komunitas yang sangat aktif.



# CONS

---

1. Agar fitur asynchronous event-driven I/O JS dapat bekerja optimal, maka butuh multiple-core CPU yang banyak (hardware server yang sangat bagus).
2. Sangat buruk dalam berkerja dengan **relational database**.
3. Tidak cocok untuk CPU-bound task. Hanya cocok untuk **I/O bound task** (seperti web server).

# STRUCTURE

---

- **Struktur Dasar:**

1. **app.js/index.js** – menginisialisasi aplikasi dan “merekatkan” seluruh komponen project.
2. **package.json** – mengingat package-package apa saja yang diperlukan oleh aplikasi.



# STRUCTURE

---

- Dipadukan dengan *AngularJS*, maka struktur harus disesuaikan dengan arsitektur **MVC**, sehingga selain item-item pada struktur dasar, berikut adalah struktur *express.js* yang baik<sup>[4]</sup>:
3. **controllers/** – mendefinisikan routing dan logic aplikasi.
  4. **helpers/** – fungsionalitas yang dapat digunakan berkali-kali di project.
  5. **middlewares/** – middleware *express* yang memproses request sebelum melakukan routing request tersebut.

# STRUCTURE

---

- 6. **models/** – struktur data, *business logic*, mengatur storage.
- 7. **public/** – file statis seperti gambar, css, dsb.
- 8. **views/** – template response yang diberikan ke pengguna.
- 9. **db/** – lokasi file-file yang merupakan basis data aplikasi.
- 10. **tests/** – file yang digunakan sebagai tester app.



project/  
  controllers/  
  helpers/  
  middlewares/  
  models/  
  public/  
  views/  
  db/  
  tests/  
  app.js  
  package.json

project/  
  controllers/  
    index.js  
    users.js  
  middlewares/  
    auth.js  
  db/  
    users.bson  
  views/  
    index.jade  
  ...

## STRUCTURE: SAMPLE

# EXPRESS.JS BASICS

---

**PRASYARAT:** Telah menginstall Node.js dan npm

- Membuat directory project
- Di directory project, lakukan command

```
$ npm init
```

- Lakukan instalasi express pada directory

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

Express.js siap digunakan di project tersebut 😊



# REFERENCES

---

- [1] Laurent Orsini. (2013-11-07). “What You Need To Know About Node.js”
- [2] Gesoulis, Aggelos. “What is Express.js and Angular.js?”, *Quora*. November, 22, 2014. <<https://www.quora.com>>
- [3] Brantley, Richard. “What Developers Mean When They Build a MEAN Stack”. August 25 2015. <<https://www.newspindigital.com>>
- [4] Fidanov, Stefan. “Best practices for Express app structure”. August 25, 2014. <<https://www.terlici.com>>