



Pertemuan # 3
Rizki Muliono,S.Kom, M.Kom | Teknik Informatika UMA

Aplikasi Berbasis Web II

Web Framework & Technologies



Pendahuluan

- Merupakan framework perangkat lunak yang dirancang untuk pengembangan website dinamis
- Tujuan framework adalah mengurangi overhead yang terkait dengan kegiatan umum yang dilakukan dalam pengembangan web, misalnya menyediakan library untuk mengakses database, framework template, manajemen session dan yang sering di unggulkan adalah adanya **code reuse**



Sejarah

- Tidak adanya inheritance dinamis pada desain world wide web, kode hypertext awalnya di ketik semuanya secara manual yang kemudian dipublish lewat web server
- Setiap ada modifikasi halaman harus dilakukan oleh pembuat halaman itu sendiri
- Untuk menyediakan halaman web dinamis yang mencerminkan input pengguna, Common Gateway Interface (CGI) standar diperkenalkan untuk antarmuka aplikasi eksternal dengan server web. CGI bisa mempengaruhi beban server, karena setiap permintaan harus memulai proses yang terpisah .



Sejarah

- Programmer menginginkan integrasi yang lebih kuat dengan web server untuk memungkinkan aplikasi web dengan lalu lintas tinggi.
- Misalnya Apache HTTP Server, mendukung modul yang dapat memperluas kemampuan web server dengan eksekusi kode arbitrary (seperti mod perl) atau mem-forward permintaan khusus untuk web server yang dapat menangani konten dinamis (seperti mod jk).
- Beberapa web server (seperti Apache Tomcat) secara khusus dirancang untuk menangani konten dinamis dengan menjalankan kode yang ditulis dalam beberapa bahasa, seperti Java.



Jenis Framework

- Kebanyakan aplikasi framework berdasarkan pada pola **model-view-controller(MVC)**



Model-view-controller (MVC)

- Banyak framework menggunakan pola arsitektur **Model-view-controller** untuk memisahkan model data dengan *business rule* dari *user interface*
- Hal ini umumnya dianggap sebuah praktek yang baik karena memodularisasi kode meningkatkan penggunaan kembali kode dan memungkinkan banyak interface untuk diterapkan.



Model-view-controller (MVC)

- Dalam aplikasi Web ini memungkinkan tampilan yang berbeda yang akan disajikan seperti halaman web untuk manusia dan antarmuka web service untuk aplikasi remote.



Push-based vs. pull-based

- Kebanyakan kerangka kerja MVC mengikuti arsitektur push-based juga disebut "action-based". Framework ini menggunakan tindakan yang melakukan pengolahan yang diperlukan dan kemudian "push" data ke Lapisan view untuk merender hasil.
- Struts, Django, Ruby on Rails, Symfony, Yii, Spring MVC, Stripes, CodeIgniter, dan Struts2 adalah contoh yang baik dari arsitektur ini



Push-based vs. pull-based

- Alternatif lain adalah pull-based arsitektur kadang-kadang juga disebut "berbasis komponen". Framework ini mulai dengan Lapisan view yang kemudian dapat "menarik" hasil dari beberapa kontroler yang diperlukan. Dalam arsitektur ini beberapa kontroler dapat terlibat dengan satu tampilan
- Play, Lift, Tapestry, JBoss Seam, dan Wicket adalah contoh *pull-based framework*



Three-tier organization

- Dalam **Three-tier organization**, aplikasi disusun berdasarkan tiga tingkatan fisik: client, application, dan database. D
- **Database** yang digunakan umumnya RDBMS
- **Aplikasi** berisi logika bisnis, berjalan pada server dan berkomunikasi dengan klien menggunakan HTTP
- **Client** pada aplikasi web adalah browser web yang menjalankan HTML yang dihasilkan oleh lapisan aplikasi.
- Istilah ini jangan sampai rancu dengan MVC. Tidak seperti di tiga-tier arsitektur itu dianggap sebuah praktek yang baik untuk menjaga logika bisnis dari controller, "lapisan tengah" di MVC



Content management systems

- Sistem manajemen konten (Inggris: content management System, disingkat CMS), adalah perangkat lunak yang memungkinkan seseorang untuk menambahkan dan/atau memanipulasi (mengubah) isi dari suatu situs Web. Umumnya, sebuah CMS (Content Management System) terdiri dari dua elemen:
 - Aplikasi manajemen isi (Content Management Application, [CMA])
 - Aplikasi pengiriman isi (content delivery application [CDA]).



Fitur Framework

- **Web template system**
- **Caching**
- **Security**
- **Database access, mapping and configuration**
- **Scaffolding**
- **URL mapping**
- **Ajax**
- **Web services**



Web template system

- Halaman web dinamis biasanya terdiri dari bagian statis (HTML) dan bagian yang dinamis, yang merupakan kode yang menghasilkan HTML. Kode yang menghasilkan HTML dapat melakukan hal ini didasarkan pada variabel dalam sebuah template, atau pada kode. Teks yang akan dihasilkan dapat berasal dari database, sehingga sehingga memungkinkan untuk secara dramatis mengurangi jumlah halaman di situs.



Web template system

- Perhatikan contoh seorang agen real estat dengan 500 rumah untuk dijual. Dalam sebuah situs web statis, agen akan harus membuat 500 halaman web untuk membuat informasi yang tersedia. Dalam sebuah website dinamis, agen berpotensi menghubungkan satu halaman web dinamis untuk tabel database dari 500 record.



Web template system

- Dalam template, variabel dari bahasa pemrograman dapat dimasukkan tanpa menggunakan kode, sehingga kehilangan kebutuhan pengetahuan pemrograman untuk membuat update ke halaman dalam situs web. Sebuah sintaks dibuat tersedia untuk membedakan antara HTML dan variabel. Misalnya di JSP tag `<c:out>` digunakan untuk variabel output, dan dalam Smarty, `{$variabel}` yang digunakan.

A close-up photograph of a person's hand typing on a silver laptop keyboard. The laptop is open, and the keyboard is the central focus. The background is a solid orange color.

Web template system

- Banyak mesin template, mendukung tag logika secara terbatas, seperti IF dan foreach. Hal semacam ini harus digunakan hanya untuk keputusan yang perlu dibuat untuk lapisan presentasi, untuk menjaga pemisahan bersih dari lapisan logika bisnis, atau M (odel) dalam pola MVC.



Caching

- Web Caching adalah caching dokumen web untuk mengurangi penggunaan bandwidth, beban server, dan dianggap "lag". Web Cache Menyimpan salinan dokumen yang melewatinya, permintaan berikutnya bisa terpenuhi dari cache jika kondisi tertentu terpenuhi. Beberapa kerangka aplikasi menyediakan mekanisme untuk caching dokumen dan melewati berbagai tahapan persiapan halaman, seperti akses database atau interpretasi template.



Security

- Beberapa framework aplikasi web, dilengkapi dengan framework otentikasi dan otorisasi, yang memungkinkan web server untuk mengidentifikasi pengguna dari aplikasi, dan membatasi akses ke fungsi didasarkan pada beberapa kriteria yang ditetapkan. Drupal adalah salah satu contoh yang memberikan peran berbasis akses ke halaman, dan menyediakan antarmuka berbasis web untuk membuat pengguna dan memberikan mereka peran.



Database access, mapping and configuration

- Banyak framework aplikasi web, membuat API terpadu untuk backend database, memungkinkan aplikasi web untuk bekerja dengan berbagai database tanpa perubahan kode, dan memungkinkan programmer untuk bekerja dengan tingkat yang lebih tinggi konsep. Untuk kinerja yang lebih tinggi, koneksi database harus dikumpulkan. Selain itu, beberapa berorientasi objek framework berisi alat pemetaan untuk menyediakan Object-Relational Mapping,



Database access, mapping and configuration

- Beberapa framework meminimalkan konfigurasi aplikasi web melalui penggunaan introspeksi dan / atau dikenal sebagai konvensi. Sebagai contoh, banyak framework Java menggunakan Hibernate sebagai lapisan persistensi, yang dapat menghasilkan skema database saat runtime mampu terus terjadi informasi yang diperlukan. Hal ini memungkinkan desainer aplikasi untuk merancang obyek bisnis tanpa perlu secara eksplisit menentukan skema database. Kerangka seperti Ruby on Rails juga dapat bekerja secara terbalik, yaitu, mendefinisikan sifat model objek pada saat runtime berdasarkan skema database.



Scaffolding

- Beberapa model-view-controller framework menghasilkan kode yang dapat digunakan oleh aplikasi untuk membuat, membaca, memperbarui dan menghapus entri database.



URL mapping

- Mapping URL adalah mekanisme pada framework untuk menafsirkan URL.
- Beberapa framework, seperti Drupal dan Django, cocok dengan URL yang diberikan terhadap pola yang telah ditentukan dengan menggunakan ekspresi reguler, sementara beberapa yang lain menggunakan URL Rewriting untuk menerjemahkan URL yang tersedia menjadi satu yang mendasari mesin akan mengenali. Contoh : mapping url pada cms wordpress dengan menggunakan mod rewrite pada modul apache



URL mapping

- Sebuah pemetaan URL sistem yang menggunakan pencocokan pola atau penulisan ulang URL memungkinkan "URL ramah" yang akan digunakan, meningkatkan kesederhanaan situs dan memungkinkan untuk pengindeksan yang lebih baik oleh search engine.
- Sebagai contoh, URL yang berakhir dengan `"/halaman.cgi?cat=Ilmu&topik=fisika"` dapat diubah menjadi hanya `"halaman/sains/fisika"`



URL mapping

- Hal ini membuat URL lebih mudah dibaca dan menyediakan mesin pencari dengan informasi yang lebih baik tentang tata letak struktur dari situs.



Ajax

- Ajax, singkatan untuk "Asynchronous JavaScript and XML", adalah teknik pengembangan web untuk membuat aplikasi web.
- Tujuannya adalah untuk membuat halaman web terasa lebih responsif dengan bertukar sejumlah kecil data dengan server di belakang layar
- sehingga seluruh halaman web tidak harus ulang setiap kali pengguna meminta perubahan. Hal ini dimaksudkan untuk meningkatkan interaktivitas halaman Web, kecepatan, dan kegunaan.



Ajax

- Karena kompleksitas pemrograman Ajax di Javascript, terdapat framework Ajax banyak yang eksklusif berhubungan dengan dukungan Ajax. Beberapa framework Ajax bahkan ditanamkan sebagai bagian dari framework yang lebih besar. Misalnya, Library jQuery JavaScript disertakan dalam wordpress.



Web services

- Beberapa framework menyediakan alat untuk menciptakan dan menyediakan layanan web(web service).

INTRO TO WEB DEVELOPMENT

Tujuan:

- Basic Web App Model
- Web Development Framework dan Bahasa

Sumber:

- Web Framework
- Populer Framework
- 10 Hal yang Perlu Diketahui
- Angular

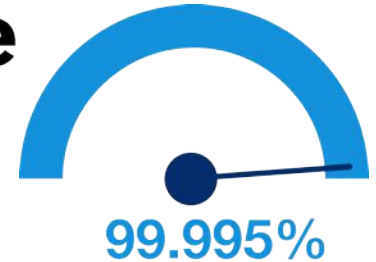
- React
- Knockout
- Rest
- PostMan
- Chrome Dev Tool



Principles of Web Design

Performance

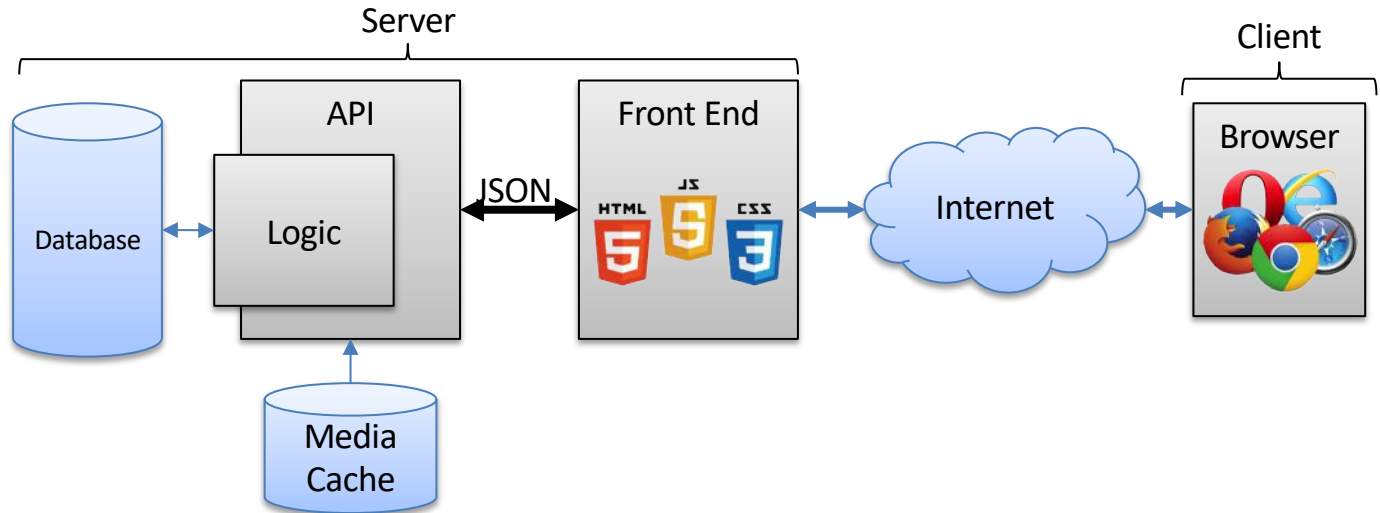
- Availability
- Performance
- Reliability
- Scalability
- Manageability
- Cost



Core Components of Web Applications



- UI (Front End (DOM, Framework))
- Request Layer (Web API)
- Back End (Database, Logic)





FRONTEND DEVELOPMENT

Front End Languages



- HTML/CSS
- Javascript
- Java (applets)

Apa yang paling populer?

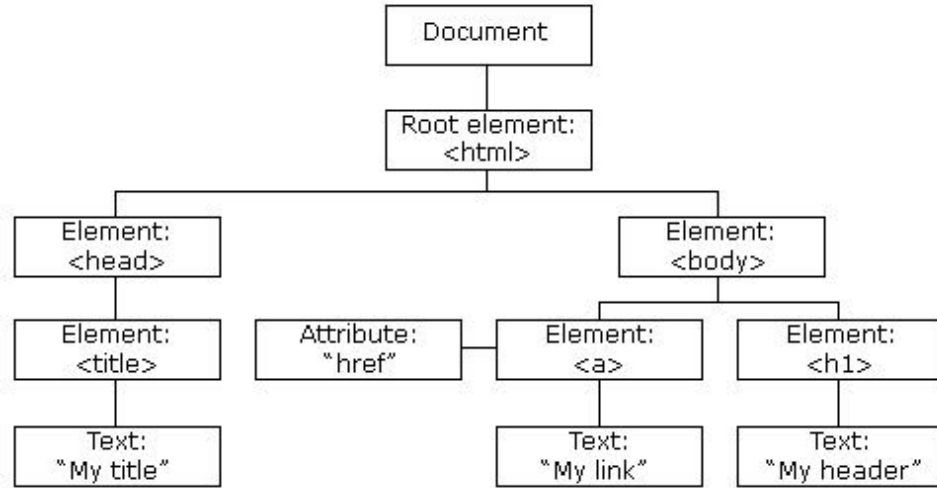
Jawaban: Javascript / HTML / CSS adalah satu-satunya pilihan nyata untuk bahasa asli front-end dan dasar standarnya. Tetapi ada banyak variasi pada JavaScript yang digunakan.



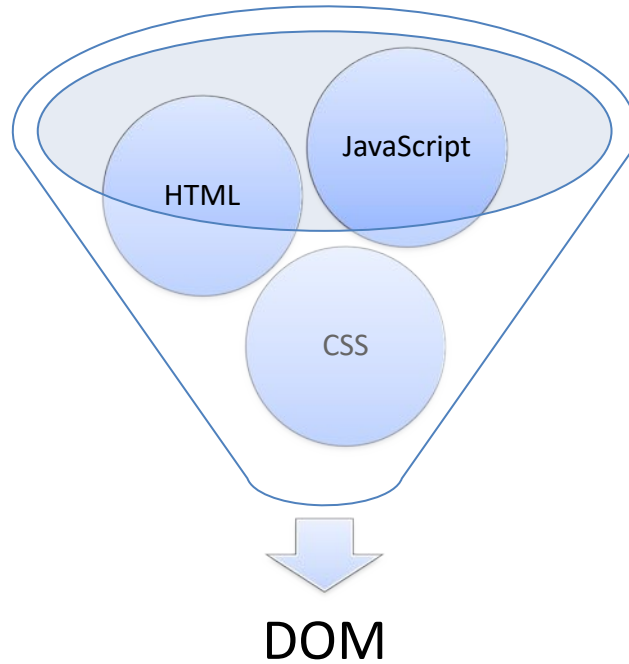
DOM (Document Object Model)



- Model Objek Dokumen menjadikan setiap item yang dapat dialamatkan dalam aplikasi web menjadi Objek yang dapat dimanipulasi untuk warna, transparansi, posisi, suara, dan perilaku.
- Setiap Tag HTML adalah objek DOM



DOM (Document Object Model)

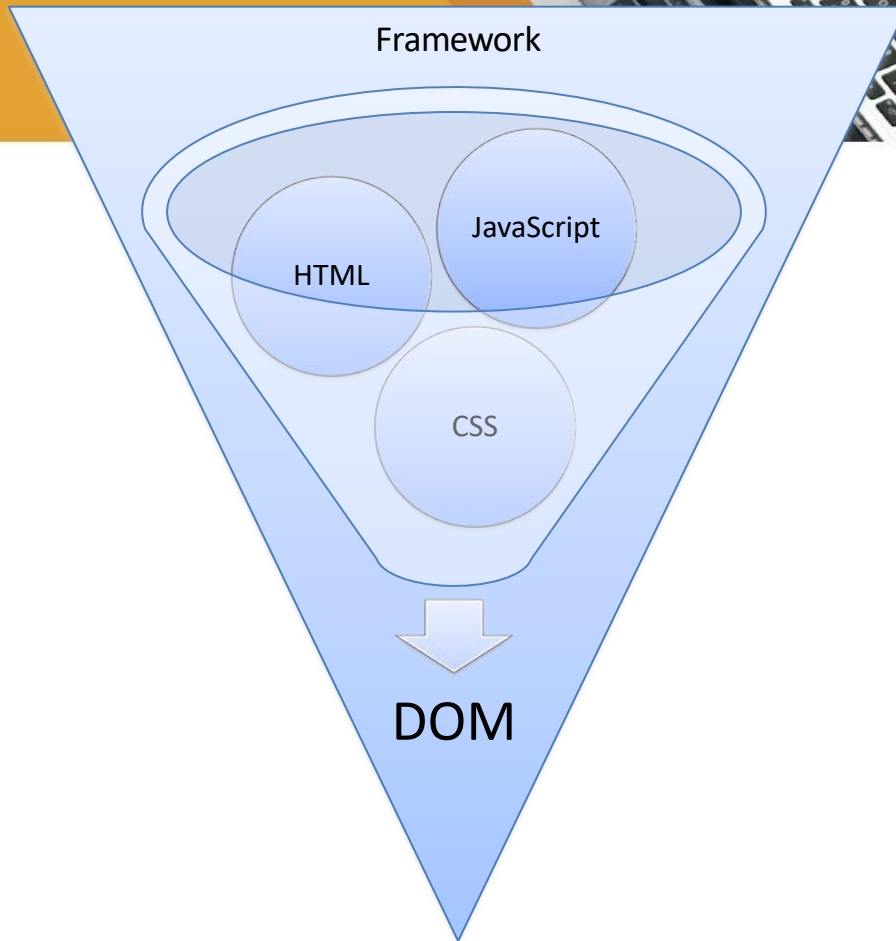


What is a Framework?



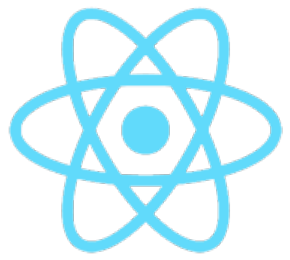
- Kerangka Perangkat Lunak dirancang untuk mengurangi overhead dalam pengembangan web
- Jenis Arsitektur Kerangka
 - Model-View-Controller (MVC)
 - Basis Push vs Pull
 - Sebagian besar arsitektur berbasis push pengguna MVC Frameworks "berbasis tindakan" (Laravel, CI, Django, Ruby on Rails, Symfony, Stripes, Spring)
 - Pull-based or "component based" (Lift, Angular2, React)
 - Organisasi Tiga Tingkat
 - Client (Biasanya browser menjalankan HTML / Javascript / CSS)
 - Aplikasi (Menjalankan Logika Bisnis)
 - Database (Penyimpanan Data)
- Types of Frameworks
 - **Server Side:** Laravel, CodeIgniter, Django, Ruby on Rails, Flask, Spring, etc..
 - **Client Side:** Angular, React, Vue, Svelte

Framework



Javascript Frameworks

- AngularJS/Angular 2
- ASP.net
- React
- Polymer 1.0
- Ember.js
- Vue.js
- Svelte



React



ember

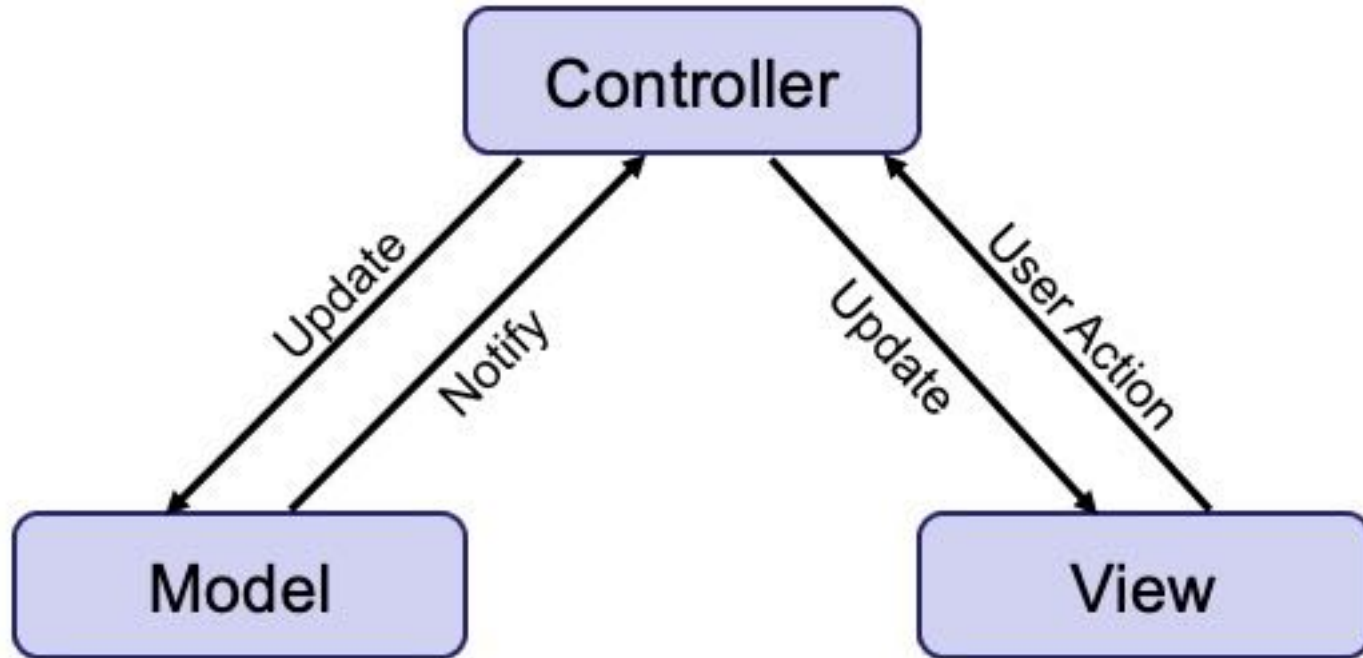


MVC (Model View Controller)



- A Web Application Development Framework
- Model (M):
 - Dimana data untuk DOM disimpan dan ditangani)
 - Di sinilah backend terhubung
- View (V):
 - Seperti Halaman yang merupakan DOM tunggal
 - Di mana perubahan pada halaman dirender dan ditampilkan
- Control (C):
 - Ini menangani masukan dan interaksi pengguna
 - Buttons
 - Forms
 - Antarmuka Umum

MVC Model





BACKEND DEVELOPMENT

What is a Backend?



- Semua hal luar biasa yang menjalankan aplikasi Anda.
- Web API
 - Lapisan koneksi antara frontend dan backend
 - Terhubung melalui panggilan API (POST, GET, PUT, dll.)
 - Mengirimkan Konten dari Backend ke Frontend biasanya di JSON Blobs
- Arsitektur Layanan yang menggerakkan segalanya (Di mana semua logikanya)

What is a WebAPI?



- Lapisan perantara antara sistem front-end dan back-end
- "Harus dimiliki" jika API Anda akan dikonsumsi oleh layanan pihak ketiga

Yang Harus diperhatikan secara detail:

- Seberapa dapat dikonsumsi API (tanda tangan, negosiasi konten)?
- Apakah itu sesuai dengan standar (kode respon, dll.)?
- Apakah itu aman?
- Bagaimana Anda menangani banyak versi?
- Apakah ini benar-benar RESTful?

Representational State Transfer (REST)



- Client-server
- Stateless
- Resource-based (vs. remote *procedure call*)
- HTTP methods (GET, POST, PUT, DELETE)
- Side Effects
- It's a style, not a standard
- Don't hate on HATEOAS

WebAPI Terms

- GET – “read”
- POST – “insert” (collection)
- PUT – “replace”
- DELETE – “remove”
- PATCH – “update”
- Custom (proceed with caution)



Web Status Codes



- 200 – OK – things are great (return the item)
- 201 Created – after POST (HATEOAS – return location)
- 204 No Content (i.e. successful DELETE)
- 400 – Bad Request (validation error, missing parms, etc.)
- 401 – Unauthorized – Who are you?
- 403 – Forbidden – No soup for you
- 404 – Not Found

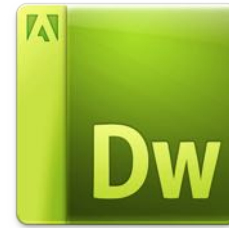
Popular Tools

Development Tools:

- Chrome/Firefox Developer Tools
- Postman (API)
- Dreamweaver
- VS Code
- SublimeText
- Git / SourceTree

Analytics Tools:

- Google/Adobe Analytics



Adobe
Analytics



Google Analytics

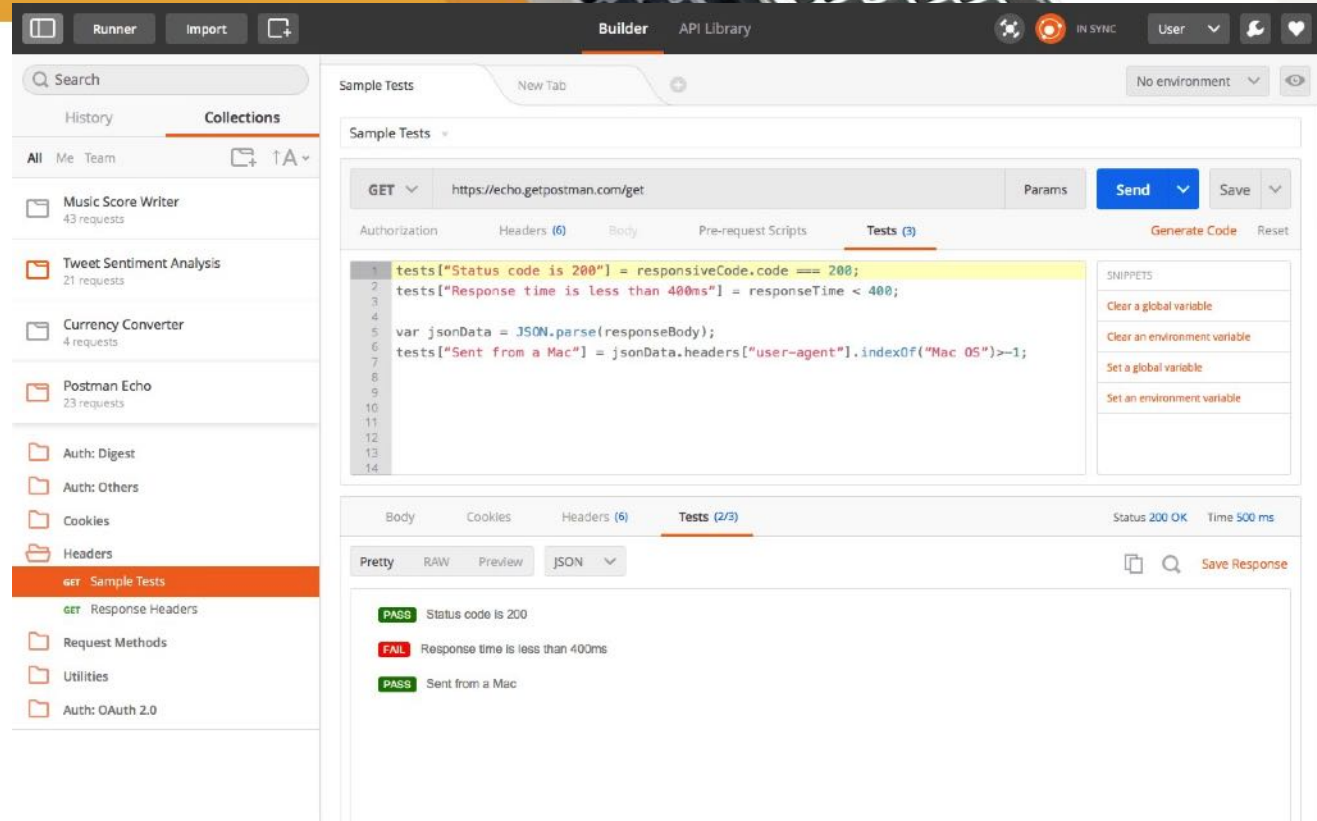
Tools for Testing WebAPI

- Postman Chrome extension

<http://bit.ly/postmanext>

- Fiddler by Telerik

<http://www.Telerik.com/fiddler>



The screenshot displays the Postman Chrome extension interface. The top navigation bar includes 'Runner', 'Import', and 'Builder' tabs. The main workspace shows a REST client configuration for a GET request to `https://echo.getpostman.com/get`. The 'Tests' tab is active, showing a JavaScript test script:

```
1 tests["Status code is 200"] = responseCode.code === 200;
2 tests["Response time is less than 400ms"] = responseTime < 400;
3
4
5 var jsonData = JSON.parse(responseBody);
6 tests["Sent from a Mac"] = jsonData.headers["user-agent"].indexOf("Mac OS")>=1;
7
8
9
10
11
12
13
14
```

The test results are displayed below the script:

- PASS** Status code is 200
- FAIL** Response time is less than 400ms
- PASS** Sent from a Mac

The interface also shows a sidebar with a 'Collections' view containing various API collections like 'Music Score Writer', 'Tweet Sentiment Analysis', and 'Postman Echo'. The bottom right corner shows the status 'Status 200 OK' and 'Time 500 ms'.



APPENDIX

Hypermedia as the Engine of Application State (HATEOAS)



- Hypermedia adalah kuncinya
- Semuanya dimulai dari URL
- Sumber daya dikembalikan
- Jenis dan lokasi media disertakan
- Referensi berdasarkan negara bagian

What is Angular

- MVC Structure
- Framework
- Single Page Application (SPA)
- Client Side Template
- Testing



Why Angular?

New Developers

- Popularity
- Demand
- Support and Resources
- Front End

Seasoned Developers

- Structured and Opinionated Framework
- Productivity
- Consistency

Team Leads

- Efficiency
- Longevity

Compare Search terms ▾

AngularJS
Software

React
Software

Knockout
Software

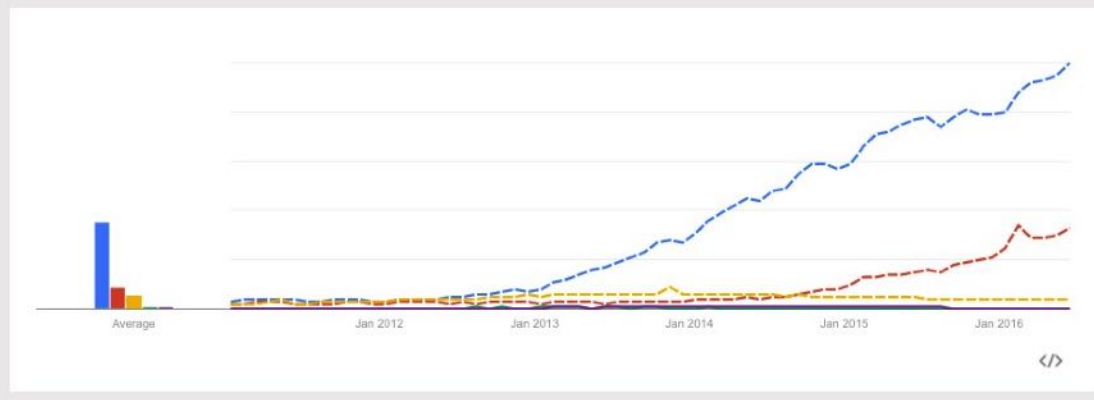
Backbone...
Search term

EmberJS
Search term

Beta: Measuring search interest in *topics* is a beta feature which quickly provides accurate measurements of overall search interest. To measure search interest for a specific *query*, select the "search term" option. [?](#)

Interest over time [?](#)

News headlines [?](#) Forecast [?](#)



Angular vs. Angular 2



- Angular 1
 - Structured MVC Framework
 - Separation of HTML and Logic
 - Client Side Templating
- Angular 2
 - Component Based UI
 - More Modular Design
 - TypeScript
 - Backwards Compatible
 - Faster

Angular vs. Angular2



```
angular.module('myModule')
  .controller('myController',function(){
  })

<body>
  <div ng-controller="myController">
  </div>
</body>
```

```
import { Component } from '@angular/core'

@Component({
  selector: 'my-app',
  template: ``
})
export class MyAppComponent {
}

<my-app></my-app>
```

Typescript



JavaScript

```
var num = 5;
var name = "Speros";
var something = 123;
var list = [1,2,3];

function square(num) {
    return num * num;
}
```

TypeScript

```
var num: number = 5;
var name: string = "Speros"
var something: any = 123;
var list: Array<number> =
[1,2,3];

function square(num: number):
number {
    return num * num;
}
```

Typescript



JavaScript

```
var Person = (function () {  
    function Person(name) {  
        this.name = name;  
    }  
    return Person;  
})();  
  
var aPerson = new Person("Ada");
```

TypeScript

```
class Person {  
    constructor(public name: string){  
    }  
}  
  
var aPerson = new Person("Ada  
Lovelace");
```

Building Blocks



- Directives
 - **Component** – *Templates (HTML), Styles (CSS), & Logic (JavaScript)*
 - **Attribute** – *Styling HTML*
 - **Structural** – *Manipulating HTML*
- Data Flow
 - **Interpolation** – *Variable Printing in Templates*
 - **Event Binding** – *Trigger Events*
 - **2-Way Binding** – *Variables updated in real time*
- Providers
 - **Services**
 - Reusable Logic
 - Data Storing and Manipulation
 - **Libraries**

Component Directives



" ... Blok penyusun yang dapat digunakan kembali untuk suatu aplikasi "

Komponen memiliki :

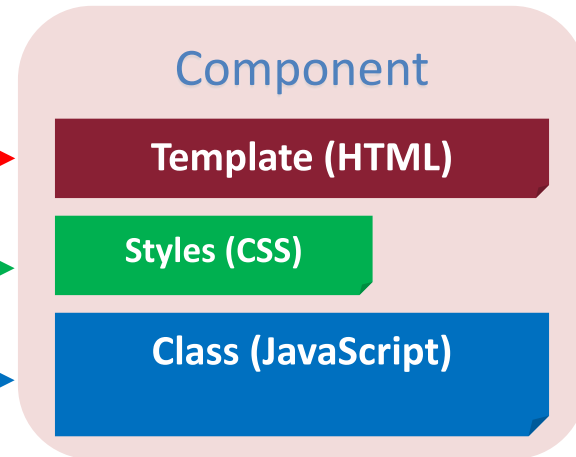
– **HTML**



– **CSS**



– **JavaScript**



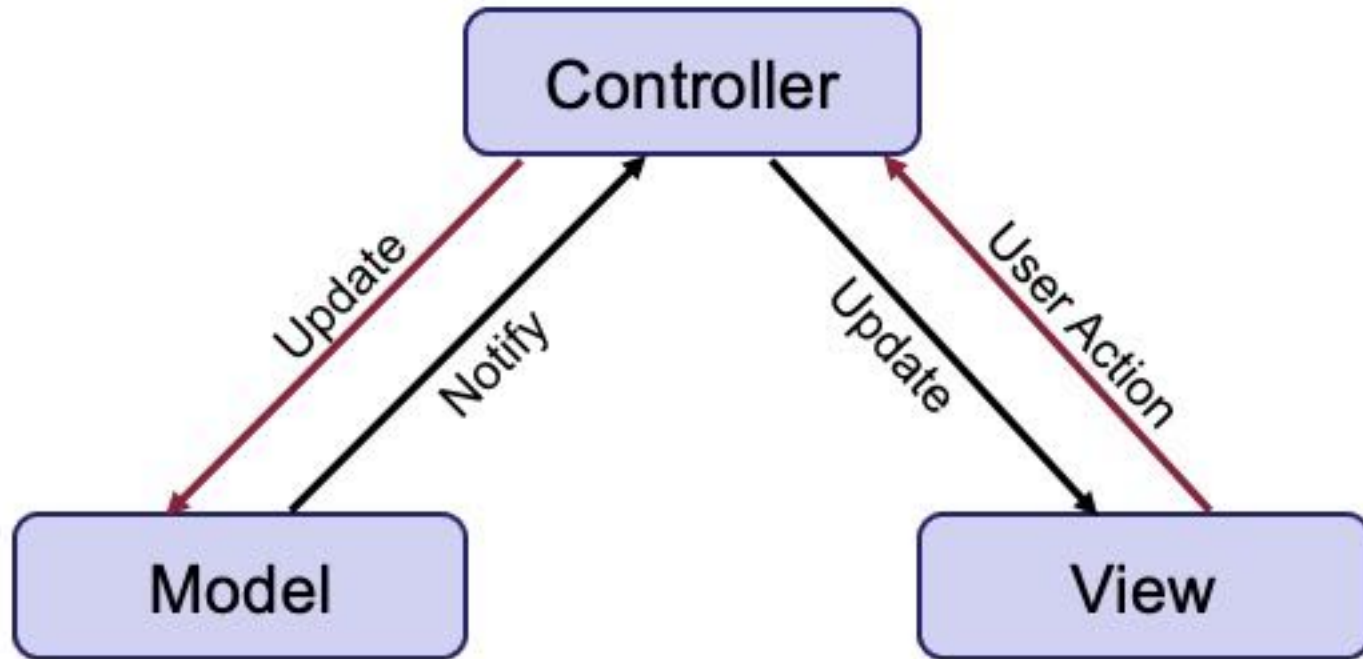
Learn Angular/Angular2



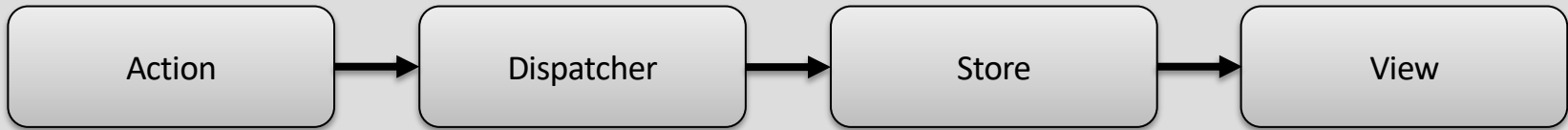
<https://angular.io/>

<https://www.tutorialspoint.com/angular2/index.htm>

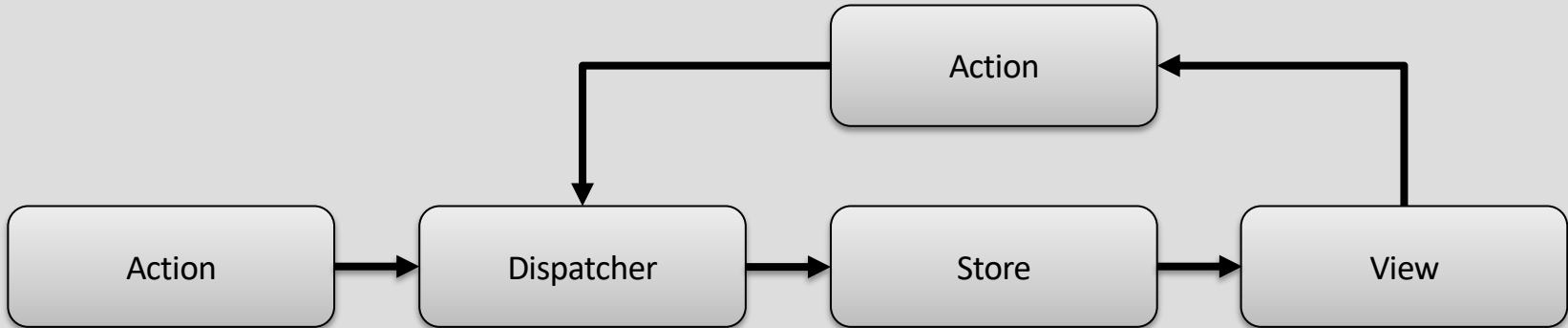
How does React fit MVC?



Flux Model



Data Flow



Action Flow

React Components



```
// Create a component name MessageComponent
var MessageComponent = React.createClass({
  render: function() {
    return (
      <div>{this.props.message}</div>
    );
  }
});
```

```
// Render an instance of MessageComponent into document body
ReactDOM.render(
  <MessageComponent message="Hello!" />
  document.body
);
```

React Components

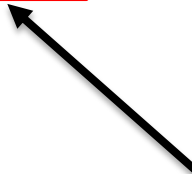


```
// Create a component name MessageComponent
var MessageComponent = React.createClass({
  render: function() {
    return (
      <div>{this.props.message}</div>
    );
  }
});

// Render an instance of MessageComponent into document body
ReactDOM.render(
  <MessageComponent message="Hello!" />
  document.body
);
```

React Components

```
// Create a component name MessageComponent
var MessageComponent = React.createClass({
  render: function() {
    return (
      <div>{this.props.message}</div>
    );
  }
});
```



```
// Render an instance of MessageComponent into document body
ReactDOM.render(
  <MessageComponent message="Hello!" />
  document.body
);
```

React



First Render

getDefaultProps

getInitialState

componentWillMount

render

componentDidMount

Props Change

componentWillReceiveProps

shouldComponentUpdate

componentWillUpdate

render

componentDidUpdate

State Change

shouldComponentUpdate

componentWillUpdate

render

componentDidUpdate

Unmount

componentWillUnmount

Learn React



<https://reactjs.org/tutorial/tutorial.html>

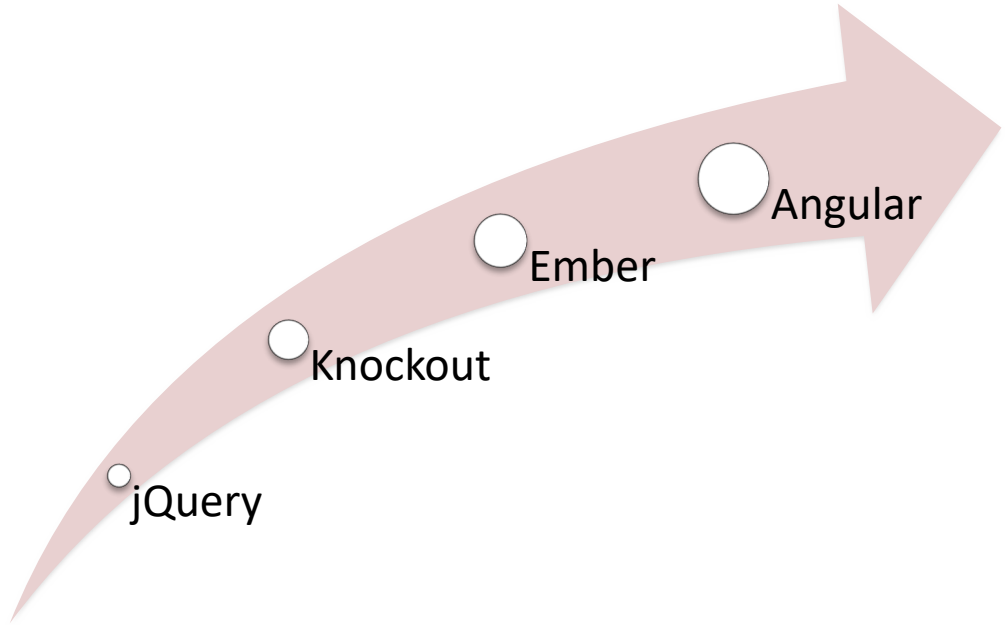
<https://www.codecademy.com/learn/react-101>

<https://css-tricks.com/learning-react-redux/>

Intro to Knockout



- Perpustakaan MVVM
- Pembaruan dan pembaruan UI otomatis
- Template yang dapat digunakan kembali
- Dapat digunakan dengan hampir semua kerangka kerja
- Berfokus pada pengikatan data
- Ukuran Library kecil



MVVM (Model, View, View-Model)



View

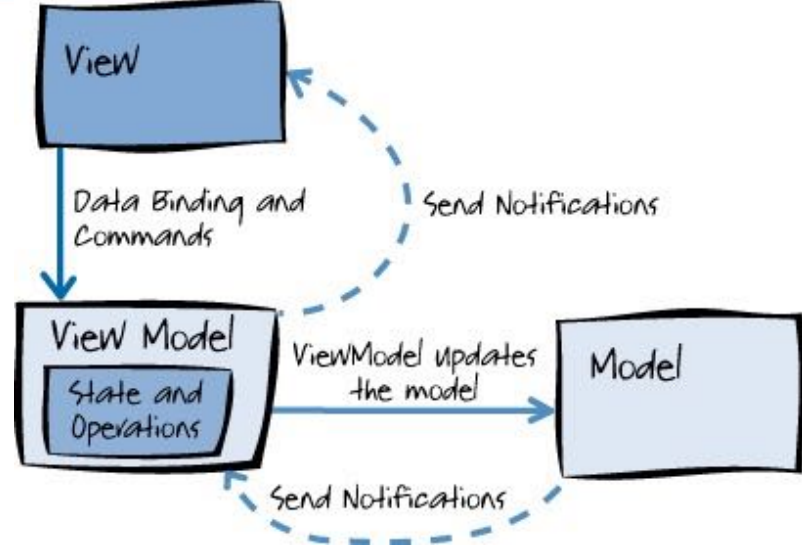
- Mendefinisikan struktur dan tata letak UI

Model

- Domain Model
- Data Model
- Business logic

View Model

- Perantara antara M / V
- Menangani Logika Tampilan
- Berurusan dengan Perubahan State



Learn Knockout



<http://learn.knockoutjs.com/#/?tutorial=intro>



TERIMA KASIH