

PROGRESSIVE WEB APPS UNTUK REKAYASA HYBRID APPLICATION BERBASIS TEKNOLOGI MEAN STACK

Rusdy Agustaf¹, Bernard Renaldy Suteja²

¹Program Studi Teknik Informatika, Universitas Janabadra

²Program Studi Teknik Informatika, Universitas Kristen Maranatha

Email : ¹rustaf52@gmail.com ²bernard.rs@it.maranatha.edu

ABSTRACT

The main obstacle in making a mobile app is the variety of platforms and operating system versions of mobile devices, as a result the application must be adjusted accordingly. In a long period of time web programming techniques have developed into progressive web apps that can be used for hybrid apps. Hybrid application is an application that can run on many platforms. MEAN Stack technology can be used to produce a good hybrid app. The Ionic framework can be used to implement MEAN Stack technology to produce a hybrid app that is easy and stable.

Keywords : *Progressive Web Apps, Hybrid Application, MEAN stack*

1. PENDAHULUAN

Seiring dengan kemajuan teknologi khususnya dalam pemrograman web, membawa dampak aplikasi yang dihasilkan dapat digunakan secara lintas platform. Tidak hanya dapat dikembangkan dan berjalan di platform web saja namun dapat langsung digunakan pada platform *mobile* (berbasis android maupun ios). Aplikasi yang dapat berjalan di kedua platform tersebut sering dikenal dengan istilah *Hybrid Application*.

Dave Chaffey dalam penelitian terkait *mobile marketing* menyampaikan bahwa persentase akses halaman web melalui perangkat *mobile* telah meningkat secara pesat dalam beberapa tahun terakhir [1]. Hal inilah yang menyebabkan makin banyak aplikasi *mobile* (*mobile app*) dikembangkan. Namun demikian dalam membuat dan menghasilkan *mobile app* tersebut terdapat kendala yaitu banyak variasi dari platform dan versi dari perangkat *mobile* dan sistem operasi berakibat aplikasi harus dibuat menyesuaikan dengan platform dan versi. Untuk mengatasi kendala ini *hybrid application* hadir sebagai solusi yang mudah dan murah untuk membangun aplikasi yang cepat dan ringan [2].

Dalam kurun waktu yang cukup lama teknik pemrograman web juga sudah berkembang, termasuk dalam memenuhi

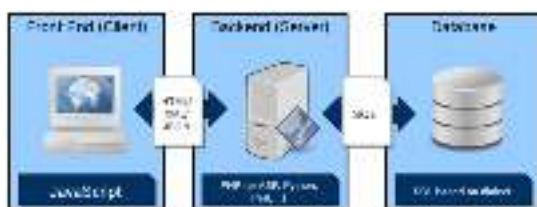
kebutuhan *hybrid app* ini. Sebelumnya sangat dikenal dengan *multitier client server* yang sisi server (*back-end*) berbasis pada teknologi LAMP (Linux Apache, MySQL, dan PHP), yang kemudian untuk meningkatkan produktifitas maka bermunculan *framework-framework* de yang berbasis *Model, View and Controller* (MVC) pada pemrograman PHP. Sedangkan sisi client (*front-end*) untuk representasi digunakan HTML, CSS, Javascript, dan XML. Proses berjalan untuk sebuah aplikasi web dengan teknologi ini membedakan pemrograman web yang digunakan pada sisi client dan server dan mulai dari basis data model ke SQL atau XML ke PHP dan terakhir menjadi dokumen HTML yang dibungkus dengan CSS. Terlalu banyak konversi yang terjadi sehingga dapat berakibat pada performa yang tidak baik.

Teknologi MEAN (Mongo DB, Express, Angular JS dan Node JS) hadir sebagai teknik pemrograman web yang modern. Teknologi ini berbasis pada platform Javascript untuk pemrogramannya. Dengan platform yang sama tersebut maka pengembangan aplikasi akan jauh lebih mudah dilakukan dan performa dapat ditingkatkan. Penelitian ini dilakukan untuk melakukan rekayasa sebuah *hybrid app* dengan berbasis pada teknologi MEAN stack.

2. KAJIAN PUSTAKA

2.1. Linux, Apache, MySQL dan PHP (LAMP)

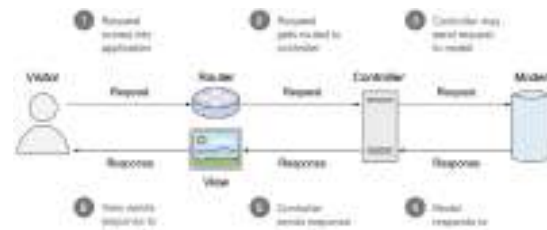
Pembuatan web merupakan bentuk interaksi antara server (*back-end*) dengan client (*front-end*) melalui HTTP (Hypertext Transfer Protocol). Server yang bertanggung jawab menyediakan halaman ataupun servis yang diminta oleh client hingga menampilkannya. Server yang paling sering digunakan adalah Apache [2]. Apache berjalan pada sistem operasi Linux yang memiliki library PHP, Perl atau Python. Melalui PHP ini interaksi *query* ke sebuah basis data MySQL dapat dilakukan. Client yang identik dengan web browser menampilkan hasil *request* dalam bentuk dokumen HTML dan menggunakan javascript dapat dimanipulasi hasilnya. Dalam skala yang lebih besar disebut dengan istilah *full-stack web development* untuk bentuk interaksi client-server dengan memperhatikan berbagai macam jenis hosting server, variasi tipe basis data, user interface dan user experience, quality assurance, dan sebagainya secara lebih detail.



Gambar 1. Layer-layer pada LAMP

2.2. Web Framework

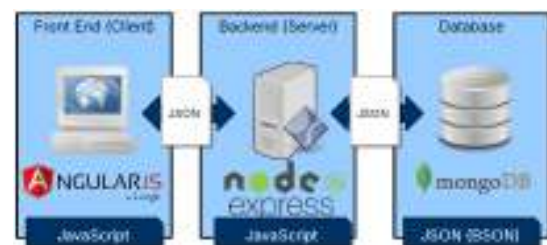
Dalam mengimplementasikan *full-stack web development* dimana client-server berkomunikasi dengan basis data dan fokus pada *reliable*, *scalable* dan *maintainable dynamic web applications* serta *web services* maka digunakan sebuah *web framework*. *Web framework* adalah sebuah kumpulan kode library untuk menjalankan fungsi atau fasilitas tertentu yang dibuat untuk memudahkan pembuatan web. Penggunaan web framework mengakibatkan pembuat web fokus pada logic aplikasi yang berbasis pada pendekatan MVC. Berdasarkan data Github server site framework yang paling populer adalah Meteor, Ruby on Rails, Express, Laravel and Mean [3].



Gambar 2. Alur web framework

2.3. MongoDB, Express, Angular JS dan Node JS (MEAN)

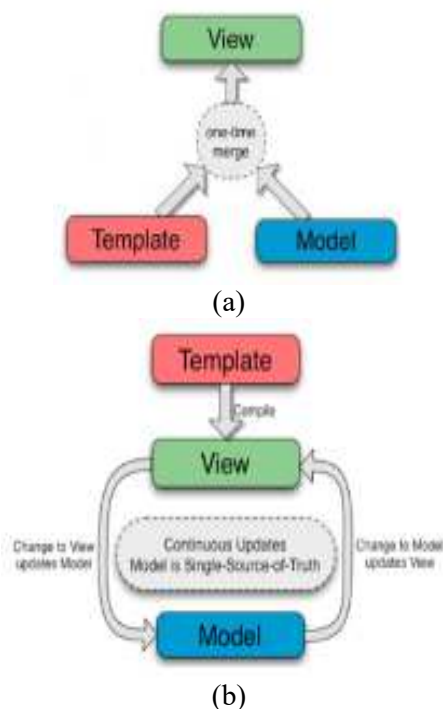
Membangun Web modern maka aspek yang dibutuhkan adalah kemudahan dan kecepatan. Teknologi LAMP yang dibangun dengan lebih dari 5 variasi bahasa pemrograman web sehingga membawa dampak sulitnya pengembangan aplikasi dimasa yang akan datang dan akibatnya terlalu banyak konversi yang terjadi sehingga dapat berakibat pada performa yang tidak baik. Teknologi MEAN sebenarnya analoginya hampir sama dengan LAMP dimana ada sisi back-end yaitu Node JS dan Express yang akan berinteraksi dengan basis data Mongo DB yang berbasis pada non-relational database / not only SQL (NoSQL) dan sisi front-end yaitu Angular JS yang dikembangkan oleh Google.



Gambar 3. Layer-layer pada MEAN

NoSQL telah banyak digunakan untuk *big data* dan aplikasi web yang fokus pada aspek real-time. NoSQL berkembang pesat dalam penggunaannya karena beberapa hal yaitu : kemudahannya dalam melakukan desain data, fleksibilitas dan *scalability* [24]. Penyimpanan dan pengelolaan NoSQL berbasis pada *Javascript Object Notation* (JSON). MongoDB yang menerapkan NoSQL memanfaatkan key-value (konsep JSON) dan menyimpan data dalam format Binary JSON (BSON). MongoDB memiliki *dynamic schemas* yang membolehkan memiliki record tanpa mendefinisikan struktur seperti yang terjadi pada basisdata relasional.

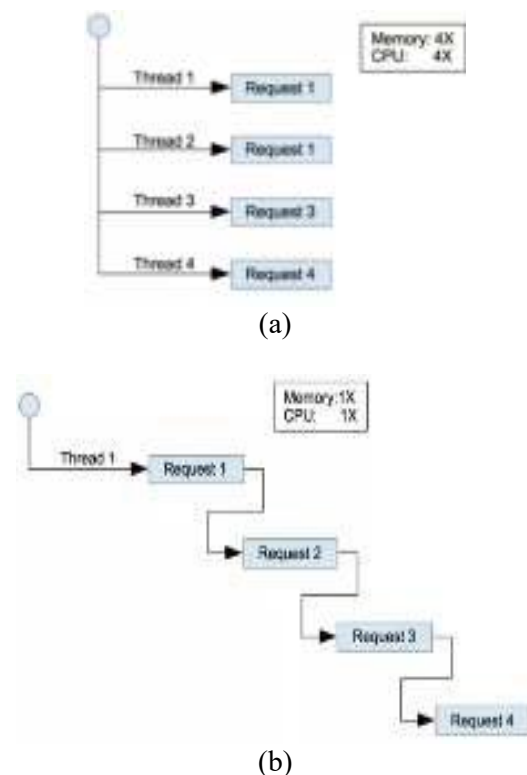
Angular JS merupakan sebuah framework yang berbasis MVC Javascript untuk membuat *front-end* web aplikasi disisi client yang modern dan dinamis. Angular JS dikembangkan Google dan digunakan pada setiap aplikasi yang dibuat Google. Bagian terpenting dalam membuat aplikasi web adalah menerima data, memproses data di server dan menghasilkan dokumen HTML yang berisi data-data (*data binding*) serta menampilkannya pada sisi client. Cara ini adalah yang paling umum dilakukan sehingga disebut *classical template system* atau *one-way data binding*. Template dan Model digabung untuk menghasilkan View. Angular membagi antara *data binding* antara Model dan Template dan browser yang akan memproses Template untuk menghasilkan View yang sesuai dengan Model disebut juga *two-way data binding*[18].



Gambar 4. (a) One-way dan (b) Two-way Data Binding

Node JS berada pada sisi *back-end* dan merupakan platform software yang memungkinkan menjalankan server mandiri dan menggunakan *single threaded-asynchronous event-driven architecture* dan *non-blocking I/O* model sehingga peformanya sangat baik dan jauh jika dibandingkan PHP untuk sebuah aplikasi web yang *realtime* [3].

Konsep *non-blocking I/O* membuat pemrosesan di Node JS lebih efisien karena tidak terkunci ketika sebuah proses sedang berjalan, berbeda dengan PHP yang terjadi adalah sistem antrian. Disisi lain Node JS menggunakan konsep *single threaded* yang artinya 1 proses dijalankan hingga selesai, berbeda dengan PHP yang *multi threaded*. Untuk memperoleh peforma yang baik maka dalam Node JS dapat dipilih sendiri modul yang dibutuhkan dengan memanfaatkan *Node Package Manager* (NPM)



Gambar 5. Analogi Pemrosesan (a) *multi threaded* (b) *single threaded*

Express merupakan sebuah framework yang dapat digunakan pada Node JS. Express hadir dengan modul yang teruji dan terstruktur untuk membuat aplikasi dapat lebih cepat dan lebih mudah termasuk dalam pembuatan *hybrid app* [4]. Express memiliki 3 komponen yaitu : application object, request object, dan response object. Application object merupakan *instance* dari Express, yang dapat dilakukan oleh object ini adalah *routing HTTP request*, konfigurasi *middleware*, *application behavior*, menghasilkan tampilan HTML (*HTML view*), dan integrasi template. Request object dibuat ketika terjadi permintaan client ke server, isinya adalah informasi HTTP request seperti query string,

request body, request header, parameter-parameter, request cookies dan sebagainya. Untuk response object dibuat bersamaan ketika request object dibuat, berisi property dan method sebagai respon dari server. Express membolehkan response dalam bentuk format yang beragam seperti attachment, cookie, json, file atau bahkan status code.

2.4. Hybrid App dengan Progressive Web Apps

Pertumbuhan pengguna *mobile app* makin meningkat pesat sejak 2015 dengan pengguna mobile mencapai 51%, desktop 42%, pengguna dari peralatan lain 7% [1]. Untuk mengadaptasi trend ini maka pembuatan versi mobile app dari versi web app menjadi hal yang penting. Sehingga pembuatan aplikasi cukup dibuat sekali saja dan hasilnya dapat digunakan (akses) pada semua peralatan (device).

Kendala yang dihadapi dalam mengembangkan hybrid app adalah native platform. Native platform merupakan fasilitas yang ada pada peralatan mobile yang bisa diakses dan digunakan seperti kamera, lokasi dan sebagainya. Native platform ini dikembangkan oleh bahasa program yang berbeda-beda tergantung pada pengembang alatnya seperti Xcode dan Objective-C / Swift untuk peralatan berbasis iOS, Android Studio dan Java untuk peralatan berbasis Android, serta Visual Studio dan C# untuk peralatan berbasis *windows phone*. Sehingga hal ini yang tidak dapat dilakukan oleh pemrograman web. Perlu upaya lebih dengan menggunakan pendekatan *Progressive Web Apps* (PWAs), sehingga native platform bisa digunakan dalam pemrograman web (multi platform) [3] [4].

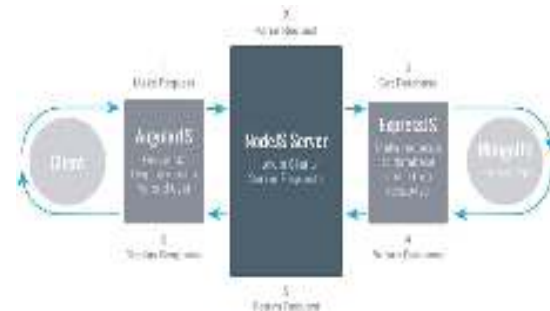
Ionic adalah framework untuk membuat hybrid app dengan mudah dan cepat. Ionic memungkinkan pemanfaatan native platform karena adanya cordova atau capacitor. Melalui cordova atau capacitor ini HTML, CSS dan Javascript dibuat menjadi *native mobile app* yang bisa berjalan disemua platform sistem operasi mobile seperti IOS, Android dan Windows Phone [7].

Ionic hadir dengan berbasis pada *common line interface (CLI)* sehingga hybrid app dapat dibuat dengan mudah dan cepat. Mulai dari membuat project, preview hasil di multi browser, emulasi IOS, Android dan

Windows Phone ke emulator dan menghasilkan (*deploy*) aplikasi ke peralatan sesungguhnya dengan cara yang mudah. Ionic juga dilengkapi web componen yang mampu beradaptasi pada UI dan UX dari platform yang berbeda [7].

3. METODOLOGI

Dalam pemanfaatan PWA dengan Teknologi MEAN untuk rekayasa hybrid app maka dilakukan dengan cara membuat sebuah contoh app dengan arsitektur seperti gambar 6. App yang dibuat berbasis pada MVC yang diimplementasikan baik di *front-end* dan *back-end*. Untuk *back-end* adalah Node JS dan Express, basis data digunakan yang berbasis NoSQL yaitu Mongo DB, dan bagian *front-end* digunakan Angular JS. Contoh app yang dihasilkan dibuat dengan menggunakan Ionic.



Gambar 6. Arsitektur PWA-MEAN

4. IMPLEMENTASI DAN HASIL

Aplikasi yang dibuat memiliki struktur folder yang sesuai dengan MVC. Aplikasi yang dibuat merupakan sebuah prototipe untuk perencanaan sebuah acara konferensi. Aplikasi dibuat dengan menggunakan Ionic versi 4.0.



Gambar 7. Struktur folder contoh hybrid app

Untuk memulai jalankan server.js pada root folder. Seluruh konfigurasi dari app termasuk

Express, MogoDB, dan seting lainnya berada dalam folder config. Seluruh yang terkait dengan back-end ada pada folder backend dan front-end berada pada folder public. Konfigurasi Express terkait middleware, view engine dan session adalah sebagai berikut :

```
var hyapp = express();
var port = config.port || 3000;
hyapp.use(bodyParser.json());
hyapp.set('view engine', 'pug');
hyapp.use(cookieParser());
hyapp.use(flash());
hyapp.use(session(sessionConfigObject));
hyapp.listen(port, function () {
  console.log('Server is running on port',
  port);});
```

MongoDB yang digunakan bersama dengan Express secara langsung dapat memanfaatkan paket yang ada pada NPM yaitu Mongoose. Mongoose dapat mengakses MongoDB secara cepat dan mudah. Untuk menggunakan Mongoose maka gunakan perintah :

```
npm install mongoose
```

Setelah paket terinstall maka untuk terhubung dengan Mongoose digunakan method connect() seperti contoh berikut :

```
var mongoose = require('mongoose');
var db =
mongoose.connect('mongodb://localhost/tiny
-app', options, callback);
```

Ketika Express dan MongoDB berhasil dikonfigurasi dengan benar maka server sudah siap untuk menerima request dan memberikan response. Sebagai salah satu alasan keamanan Express juga memberikan kemudahan untuk routing proses untuk *RESTfull service* (API) dengan membentuk endpoint yang dikonfigurasi sebelumnya seperti contoh berikut :

```
hyapp.get('/api/users/:id',
userCtrl.getUser);
hyapp.post('/api/users/:id',
userCtrl.updateUser);
hyapp.delete('/api/users/:id',
userCtrl.removeUser);
hyapp.put('/api/users/registerDevice',
userCtrl.registerDevice);
```

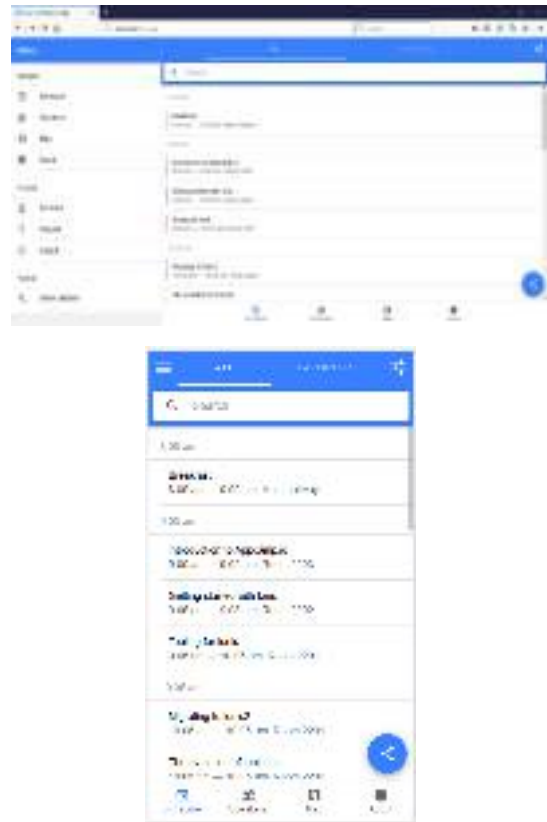
Angular bertanggung jawab terhadap seluruh hal yang ada disisi client. Angular yang akan berinteraksi dengan Express melalui API yang sudah didefinisikan.

```
angular.module('users').factory('User',
function ($resource) {
var userResource =
$resource('/api/users/:id', {id:'@_id'}, {
update: {method: 'PUT', isArray: false}
```

```
});
return userResource;
});
```

```
angular.module('users').controller('UserCo
ntroller', ['$scope', 'User', function
($scope, User) {
var user = User.get({id: 1}, function () {
user.name = 'John Doe';
user.$save();
});
}]);
```

Hasil dari prototipe dapat dihasilkan dengan *user interface* (UI) yang responsive dan sama dengan platform yang berbeda. Pada gambar 8 merupakan UI pada web browser dan mobile APP platform IOS dan Android.



Gambar 8. User Interface Hybrid App yang dihasilkan

5. KESIMPULAN

Berdasarkan pada contoh prototipe yang dibuat maka dapat ditarik kesimpulan bahwa Progressive Web Apps untuk Rekayasa Hybrid Application berbasis Teknologi MEAN Stack telah dapat dibuat dengan menggunakan framework Ionic. Seluruh UI pada setiap platform yang diuji menghasilkan fungsi sama dan dapat berjalan dengan baik dan lancar.

DAFTAR PUSTAKA

<https://docs.angularjs.org/guide>.
[Accessed 20 June 2019].

- [1] D. Chaffey, "Mobile marketing statistics compilation," Smart insights official website, 21 January 2016. [Online]. Available:
<http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics>.
[Accessed 2 July 2019].
- [2] C. Wodehouse, "Client-Side Web Development: How Scripting Languages Work," Upwork official, 5 May 2015. [Online]. Available:
<https://www.upwork.com/hiring/development/how-scripting-languages-work>.
[Accessed 3 July 2019].
- [3] "Web application frameworks," Github official website, [Online]. Available:
<https://github.com/showcases/web-application-frameworks>. [Accessed 25 June 2019].
- [4] S. Holmes, Getting MEAN with Mongo, Express, Angular, and Node, Manning Publications Co, 2015.
- [5] H. Yaapa, Express Web Application Development, Packt Publishing Ltd, 2013.
- [6] M. Korf and E. Oksman, "Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options," Salesforce developer website, April 2015. [Online]. Available:
https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options. [Accessed 5 July 2019].
- [7] P. Saccomani, "Native, Web or Hybrid Apps? What's The Difference?," Mobiloud official website, June 2012. [Online]. Available:
<http://www.mobiloud.com/blog/2012/06/native-web-or-hybrid-apps>. [Accessed 5 July 2019].
- [8] J. Wilken, Ionic in action, Manning Publications Co, 2015.
- [9] "Developer guide," AngularJS official website, [Online]. Available: