

RANCANG BANGUN APLIKASI UJIAN ONLINE REAL TIME DENGAN MENGGUNAKAN ARISTEKTUR MEAN

Ridwan Rismanto¹⁾, Putra Prima Arhandi²⁾, Arief Prasetyo³⁾

1) Teknologi Informasi – Teknik Informatika Politeknik Negeri Malang

2) Teknologi Informasi – Manajemen Informatika Politeknik Negeri Malang
email: putraprima@polinema.ac.id

3) Teknologi Informasi – Manajemen Informatika Politeknik Negeri Malang

Abstract

The development of information technology has developed rapidly into the world of education where more and more campuses and institutions use a variety of web based applications to simplify and accelerate the education services on campus or institution, one of which is the Department of Information Technology State Polytechnic of Malang to the field of web-based applications are frequently used applications is e-learning. One important part of e-learning applications that frequently used are online exams, where student can use the application to take online exam. However, online exam applications is not without drawbacks some weakness that often occurs is the application became slower and less responsive when accessed by many student. To overcome those problems in this research we develop an web based application that uses MEAN (MongoDB, Express Js, Angular Js, Js Node arcitecture).Based on our test with 100 to 150 client that connected simultaniously to our online exam application for one minutes the application have 3938 ms average of response time.

Keywords : Online Examination, e-learning, MEAN

1. PENDAHULUAN

1.1. Latar Belakang

Perkembangan internet dan teknologi informasi di Indonesia pada akhir akhir ini semakin pesat dibuktikan dengan meningkatnya pengguna internet dan semakin banyak teknologi baru yang dikemukakan dalam bidang IT. Salah satu bagian dari teknologi informasi yang berkembang dengan sangat cepat adalah aplikasi berbasis web yang mampu memberikan respon *real-time*.

Perkembangan teknologi informasi ini juga merambah pesat ke dunia pendidikan dimana semakin banyak kampus dan institusi yang menggunakan berbagai aplikasi web untuk mempermudah dan mempercepat layanan pendidikan di kampus atau institusinya, salah satunya adalah Jurusan Teknologi Informasi Politeknik Negeri Malang dengan bidang aplikasi berbasis web yang sering digunakan adalah aplikasi *e-learning*.

Salah satu bagian penting dari aplikasi *e-learning* yang sering digunakan adalah aplikasi ujian *online*, dimana peserta ujian dapat menggunakan aplikasi tersebut untuk mengikuti ujian secara *online*. Aplikasi ujian *online* ini mempermudah pekerjaan baik dari sisi pengajar maupun peserta ujian, karena dapat mempermudah proses ujian dan meningkatkan efisiensi kerja serta mempersingkat proses evaluasi hasil ujian.

Namun demikian sistem ujian *online* ini bukan tanpa kelemahan beberapa kelemahan yang sering terjadi adalah aplikasi menjadi lambat dan kurang responsif ketika diakses oleh banyak peserta, oleh karena itu diperlukan sebuah inovasi baru untuk mengatasi kelemahan tersebut. Untuk mengatasinya pada penelitian ini akan dikembangkan sebuah aplikasi yang menggunakan teknologi berbasis web yang menggunakan aristektur MEAN (MongoDb, Express Js, Angular Js,

Node Js). MEAN adalah sebuah arsitektur aplikasi *realfull-stack* yang berbasis bahasa pemrograman JavaScript dimana semua layer aplikasi baik front-end maupun *back-end* diprogram menggunakan satu jenis bahasa yaitu bahasa pemrograman JavaScript hal ini sangat baik dari sisi pemrograman karena pengembang program cukup mempelajari satu jenis bahasa pemrograman saja untuk mengembangkan sebuah aplikasi. MEAN juga sangat baik untuk pengembangan aplikasi *real-time* karena memiliki sifat *asynchronous IO* yang memungkinkan suatu aplikasi diakses oleh banyak user tanpa mengganggu performa aplikasi. Diharapkan dengan diterapkannya arsitektur MEAN dapat mengatasi permasalahan performa aplikasi pada aplikasi ujian *online* di jurusan Teknologi Informasi.

1.2. Rumusan Masalah

Rumusan masalah dari penelitian ini adalah:

1. Bagaimana mendesain aplikasi ujian online menggunakan arsitektur MEAN.
2. Bagaimana menerapkan arsitektur MEAN pada aplikasi ujian online.

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk membuat aplikasi ujian online dengan menggunakan arsitektur MEAN.

1.4. Batasan Masalah

Penelitian ini akan dibatasi pada:

1. Data yang digunakan adalah data *e-learning* dan ujian *online* pada Jurusan Teknologi Informasi Politeknik Negeri Malang
2. Arsitektur aplikasi yang dikembangkan menggunakan arsitektur MEAN.

2. KAJIAN LITERATUR

2.1. Ujian Online

Ujian *online* adalah proses untuk mengukur atau mengevaluasi hasil proses belajar mengajar yang dilakukan melalui komputer yang terhubung dengan jaringan baik lokal maupun *online*. Ujian *online* biasanya digunakan untuk mengukur kemampuan kognitif dalam menjelaskan materi yang sudah diajarkan pada satu waktu atau periode tertentu biasanya dilakukan setelah selesai suatu bab atau materi kuliah tertentu. Hasil ujian kemudian digunakan untuk membantu dalam menilai kinerja pembelajaran dan sebagai umpan balik bagi pengajar mengenai perubahan apa yang harus dilakukan dalam pembelajaran [1].

Penggunaan ujian online pada umumnya dilakukan mengikuti berbagai macam bentuk penilaian antara lain *formative assesment*, *summative assesment*, *survey*, dan *practice testing*. *Formative assesment* adalah penilaian yang dilakukan pada saat proses belajar mengajar dilaksanakan biasanya diberikan sebelum atau sesudah materi, berupa pertanyaan pilihan ganda. *Summative Assesment* adalah penilaian yang dilakukan setelah beberapa waktu atau periode pembelajaran biasanya dalam bentuk ujian tengah semester atau ujian akhir semester. Ujian *online* juga bisa digunakan untuk melakukan *survey* kepada peserta didik untuk memberikan *feedback* pada pengajar mengenai materi yang diajarkan.[1]

2.2. MEAN

MEAN (Mongo Db, Express Js, Angular Js, Node Js) adalah sebuah kumpulan perangkat lunak *open source* yang dapat digunakan untuk mengembangkan web site dinamis dan aplikasi web. MEAN terdiri dari empat perangkat lunak pendukung antara lain Mongo Db sebagai database, Express Js sebagai *back end* framework yang berjalan menggunakan Node Js, Angular Js sebagai *front end* framework dan Node Js sebagai *run time environment*. [2]

Kelebihan pengembangan aplikasi dengan menggunakan MEAN adalah sebagai berikut :

1. Menggunakan satu bahasa pemrograman yang sama untuk membangun sebuah aplikasi.
2. Semua bagian perangkat lunak pendukung MEAN mendukung dan dapat menggunakan design pattern MVC.
3. Data yang digunakan sudah dalam bentuk Json sehingga tidak membutuhkan proses serialisasi dan deserialisasi.

2.3. MongoDB

Mongo Db adalah sebuah database yang berorientasi dokumen, mongo db dapat dikategorikan sebagai NoSql database. MongoDB memiliki pendekatan yang berbeda dengan sistem database relasional yang berorientasi tabel, pada MongoDB digunakan berkas dengan format Json yang memiliki skema yang dinamis sehingga memudahkan dalam mengintegrasikan data dengan lebih mudah dan cepat. MongoDB dikembangkan oleh MongoDB Inc dan berlisensi *free and open source* dengan lisensi GNU Affero General Public License dan Apache License, sampai pada bulan Februari 2016 MongoDB merupakan database yang paling populer ke empat di seluruh dunia dan database NoSql yang paling populer [3].

Mongodb merupakan database berorientasi berkas (*document-oriented*) berbeda dengan database relasional yang berbasis *table* dan relasi. Database berorientasi berkas mengganti konsep baris (*row*) pada database relasional dengan sebuah model yang lebih lengkap yang diberinama berkas / dokumen, dengan adanya konsep ini pengembang dapat menyimpan data yang memiliki hierarki dan relasi yang kompleks kedalam sebuah berkas. Hal ini sangat cocok dengan cara bekerja dalam program moderen yang berorientasi objek [3].

2.4. Express Js

Express Js adalah sebuah *application server framework*, yang

didesain untuk membuat aplikasi web *single-page*, *multi-page*, dan *hybrid*. Secara *de facto* merupakan standar *server framework* untuk node.js. Framework ini ditulis oleh Tj Holowaychuk yang bertujuan untuk membuat sebuah framework yang minimalis dengan fitur yang dapat ditambahkan melalui plugin. Express js biasanya digunakan untuk back-end pada aplikasi MEAN [4].

2.5. Angular Js

Angular Js adalah sebuah *front-end frameworks* generasi baru yang mampu membantu meningkatkan produktifitas pengembangan aplikasi serta mempermudah perawatan dan testing aplikasi web. Angular Js pertama kali dikembangkan oleh Misko Hevery dan Adam Abrons pada tahun 2009. Angular Js dibangun dengan pemahaman bahwa *declarative programming* lebih cocok untuk membangun *user interface / front-end* sementara *imperative programming* lebih cocok untuk bisnis proses [4].

Angular Js mengadopsi arsitektur *Model View Controller* (MVC) dan *Model View View Model* (MVVM) dimana kedua arsitektur ini digunakan untuk mempermudah maintenance kode program dan proses testing. Angular Js bekerja dengan membaca tag HTML yang diberi *custom* tag milik angular, kemudian framework angular js menterjemahkan tag tersebut sebagai sebuah *directive* yang dapat ditautkan pada input atau output program [5].

2.6. Node Js

Node js adalah sebuah *runtime environment opensource* dan *cross platform* yang biasa digunakan untuk mengembangkan aplikasi web, walaupun node.js bukan merupakan framework javascript namun banyak modul dasarnya yang ditulis menggunakan javascript. *Runtime environment* yang dimiliki oleh node.js menterjemahkan javascript menggunakan google V8 javascript engine [6].

Node.js memiliki ciri khas yang sangat unik yaitu kemudahan dimana

web server dan kode program nya tidak terpisah seperti pada bahasa pemrograman web lainnya. Pada node.js pengembang dapat mengaktifkan web server memodifikasinya dan menampilkan *content* dalam satu file tanpa harus mengubah banyak konfigurasi.

Pada arsitektur MEAN node.js akan bertindak sebagai web server dan runtime environment yang menjalankan Express.js serta mengakses Mongo Db dan menampilkan ke html dengan Angular Js.

3. METODE PENELITIAN

3.1. Metode Pengembangan Aplikasi

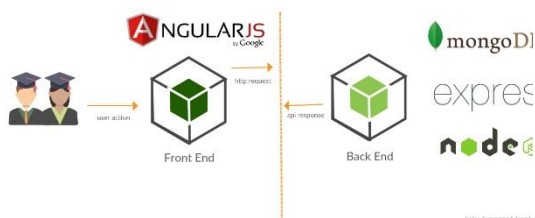
Aplikasi ujian *online* diselesaikan dengan menggunakan metode pengembangan software *back end* dan *front end* dimana *back end* akan dikerjakan dengan menggunakan Node JS dan *front end* akan diselesaikan menggunakan Angular JS dan Mongo Db sebagai *database*.

Aplikasi ujian *online* pada penelitian ini dianalisa dengan metode pendekatan berorientasi objek dimana akan dilakukan analisa sistem sesuai dengan metode OOP.

3.2. Ruang Lingkup

Objek Penelitian yang digunakan adalah aplikasi ujian *online* pada Jurusan Teknologi Informasi Politeknik Negeri Malang.

3.3. Arsitektur Sistem



Gambar. 1 Arsitektur Sistem

Aplikasi ujian *online* yang dibuat pada penelitian kali ini memiliki arsitektur sistem yang berbeda dengan aplikasi ujian *online* pada umumnya dimana pada penelitian ini dilakukan

pemisahan *layer* antara *front end* dan *back end*.

Pemisahan *layer* ini dilakukan untuk mempermudah pengerjaan aplikasi karena *front end* dan *back end* dapat berdiri sendiri, dengan konsep seperti ini sebuah *back end* dapat memiliki beberapa *front end* baik dalam bentuk *web* ataupun aplikasi *mobile*.

Jika sebuah *back end* ujian online selesai dibuat untuk *front end* nya dapat dibuat pada beberapa *platform* yang berbeda contohnya pada *platform web* dengan menggunakan *angular js* atau pada *platform mobile* dengan menggunakan *android*.

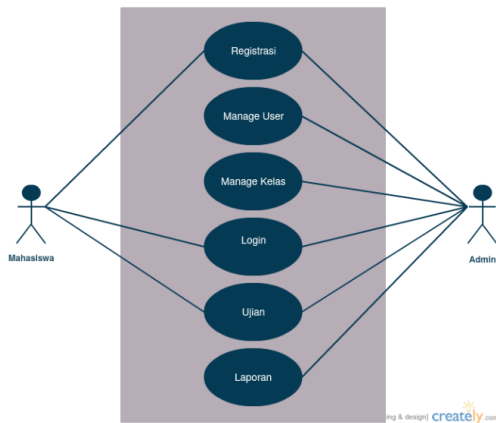
Arsitektur sistem penelitian ini dapat dilihat pada Gambar. 1.

3.4. Analisis Sistem

Analisa sistem dilakukan dengan mulai menganalisa kebutuhan yang diperlukan untuk membuat sebuah Aplikasi Ujian *Online*, untuk mempermudah analisa kebutuhan dilakukan dengan bantuan *use case* untuk menganalisa aktor dan *class diagram* untuk menganalisa atribut dan method masing masing aktor.

Aplikasi ujian *online* pada penelitian ini membutuhkan dua aktor yaitu aktor mahasiswa dan aktor admin dimana masing masing aktor memiliki tugas atau fitur masing masing, contohnya admin dapat melakukan registrasi, *manage* data user, *manage* data kelas, *manage* data ujian dan melihat laporan, sedangkan aktor mahasiswa dapat melakukan login, mengikuti ujian dan melihat hasil ujian.

Dari data *use case* yang dianalisa sebelumnya akan dilanjutkan dengan membuat sebuah *class diagram* yang dapat mengakomodasi aktor dan fitur atau tugas yang dimilikinya. Potongan *use case* pada penelitian ini dapat dilihat pada Gambar 2.



Gambar.2 Use case Diagram

Setelah selesai menganalisa *use case*, dilanjutkan dengan membuat class diagram dimana pada *class diagram* ini *use case* sudah diterjemahkan menjadi kelas lengkap dengan atribut dan method yang dimilikinya. Potongan *Class diagram* pada penelitian ini dapat dilihat pada Gambar 3.



Gambar.3 Potongan Class Diagram

3.4.1 Analisis Front end

Analisa pada *front end* fokuskan pada pembagian menu per masing masing user yang akan berbentuk sebuah *routes* yang berisi url dan file file yang diperlukan untuk membentuk *user interface*. Beberapa menu untuk masing masing user dapat dilihat pada Tabel 1.

User level	Menu
Public	Register
Mahasiswa	Login, Ujian, Lihat Nilai, Logout
Admin	Login, Manage Kelas, Manage Ujian, Manage Matakuliah, Manage Soal,

User level	Menu
	Logout, Laporan

Tabel 1 Menu per User Level

Setelah analisa menu dilakukan dilanjutkan dengan menganalisa *routes* dimana *routes* berisi konfigurasi url, template, dan file yang bertugas mengatur logika pada sebuah url. Cuplikan daftar *routes* untuk aplikasi ujian online dapat dilihat pada Tabel 2.

Menu	Routes
Manage Kelas	/kelas, /kelas/add, kelas/:id/edit, kelas/:id/register
Manage Ujian	/ujian, /ujian/add, /ujian/:id/edit
Manage Soal	/matakuliah/:id/banksoal, /matakuliah/:id/banksoal/add, /matakuliah/:id/banksoal/edit

Tabel 2 Routes per menu aplikasi

3.4.2 Analisis Back end

Analisa *back end* difokuskan pada aliran data yang masuk melalui request REST dari *front end*.

Untuk menampung request ini diperlukan sebuah endpoint dimana endpoint ini akan menerima request dengan verb http yang sesuai baik get, post, delete, put atau patch. Daftar endpoint yang dibuat pada backend dapat dilihat pada Tabel 3.

Routes	Method
/api/kelas	Post
/api/kelas	Get
/api/kelas/:id	Get
/api/kelas/:id	Put
/api/kelas/:id	Patch
/api/kelas/:id/update	Put
/api/kelas/:id/regis	Put
/api/kelas/:id/unregis	Put

Tabel 3 Routes api untuk menu kelas pada backend

3.4.3 Desain Database

Desain database mongodb akan disesuaikan dengan desain *class diagram* pada analisa sebelumnya, hal ini dilakukan karena mongo db secara natural mampu menampung format / skema database yang berorientasi objek.

4. PENGUJIAN

Setelah dilakukan proses analisis dan desain aplikasi ujian online dengan arsitektur MEAN, selanjutnya adalah proses implementasi dan pengujian program. Sebelum implementasi dan pengujian diperlukan spesifikasi perangkat keras dan lunak.

4.1 Spesifikasi Perangkat Keras

- a. Processor Intel Pentium
- b. Memory 1 GB
- c. Harddisk 30 GB SSD

4.2 Spesifikasi Perangkat Lunak

1. Sistem Operasi Linux Ubuntu 16.04
2. Nodejs 4.2.6
3. Angular Js 2.0
4. MongoDB 3.2.9-rc0

4.3 Implementasi dan Pengujian Sistem

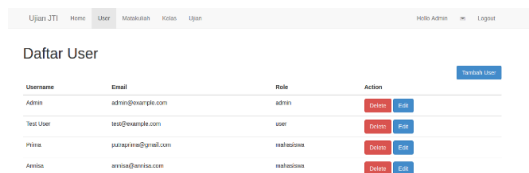
Tahap selanjutnya dari penelitian ini adalah melakukan implementasi sesuai dengan analisa pada bab sebelumnya dan melakukan pengujian untuk mengukur performa sistem yang dibuat.

4.3.1 Implementasi *Front end*

Sesuai dengan analisa pada bab 3 *front end* diimplementasikan dengan menggunakan framework Angular 2.0. Hasil tampilan *front end* untuk menu home page, dan admin dapat dilihat pada Gambar 4 dan Gambar 5.



Gambar.4 Implementasi halaman Homepage



Gambar.5 Implementasi halaman admin Pada *front end* selain implementasi user interface dilakukan juga implementasi routes untuk menghubungkan antara data dari *back end* ke *front end*. Data request dari *front end* dan respon dari *back end* dapat dilihat pada Tabel 4.

Request <i>Front end</i>	Respond <i>Back end</i>
/kelas	[{ "_id":"57fcee3baec866289f6b872", "name":"Pemrograman GUI", "__v":55, "daftarSoal":["5804368b407c2b3098e077e1", "580439a5407c2b3098e077e6", "58043a28407c2b3098e077eb", "58043ad0407c2b3098e077f0", "58043c42407c2b3098e077f5",] }]
/kelas/add	{ "__v":0, "name":"TIIA", "_id":"580cc751ff708e1c94901131", "pesertaKelas":[] }

Tabel 4 Hasil Request *front end* dan respon dari *back end*

Setelah implementasi front end dan *back end* selesai dilakukan selanjutnya dilakukan pengujian dimana aplikasi akan diuji kecepatan response nya dengan menggunakan tools independent loader.io. Pengujian dilakukan dengan memberikan url yang akan di test.

Test dilakukan dengan konfigurasi *client* yang terkoneksi ke aplikasi sebanyak 100 sampai dengan 150 *client* dan melakukan *request* yang sama terhadap salah satu *router* yang ada pada aplikasi selama 1 menit.

Selama test berjalan akan dicatat rata rata response time, kecepatan respon maksimal dan minimal dan jumlah keberhasilan respon. Pada tabel 5 dapat dilihat route / url mana yang dilakukan pengujian dan pada tabel 6 dapat dilihat hasil pengujiannya.

Routes	Method
/kelas	Get
/ujian	Get
/kelas	Post
/ujian	Post

Tabel 5 Daftar routes yang di uji

Routes	Hasil
/kelas	Rata Rata Respon Time : 3966 ms
Get	Respon Sukses :1839
	Kecepatan Maksimal : 5314 ms
	Kecepatan Minimal : 518 ms
/ujian	Rata Rata Respon Time : 3756 ms
Get	Respon Sukses : 1917
	Kecepatan Maksimal : 5574 ms
	Kecepatan Minimal : 552 ms
/kelas	Rata Rata Respon Time : 4177 ms
Post	Respon Sukses : 1889
	Kecepatan Maksimal : 5244 ms
	Kecepatan Minimal : 558 ms
/ujian	Rata Rata Respon Time : 3856 ms
Post	Respon Sukses : 1897
	Kecepatan Maksimal : 5615
	Kecepatan Minimal : 531

Tabel 6 Daftar hasil pengujian

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil analisis, pengujian dan hasil didapatkan kesimpulan bahwa:

- Aplikasi ujian *online* yang dibuat berhasil menerapkan arsitektur MEAN.
- Aplikasi ujian *online* mampu melayani request user dengan rata rata respon time sebesar 3938 ms.

5.2 Saran

Aplikasi ujian *online* yang dibuat sudah mendukung dapat mendukung terlaksananya ujian online yang bersifat objektif, namun pada kenyataannya di Jurusan Teknologi Informasi Politeknik Negeri Malang ada model ujian lain yang perlu di layani salah satunya adalah ujian pemrograman, dimana mahasiswa membuat kode program dan dilakukan penilaian oleh dosen. Proses tersebut dapat dipermudah dengan adanya ujian online yang mempunyai fitur “online judge” yang mampu menilai hasil program mahasiswa secara otomatis.

6. REFERENSI

- [1] Alfred Rovai, Michael Ponton, Jason Baker, 2008, "Distance Learning in Higher Education: A programmatic Approach to Planning, Design, Instruction, Evaluation and Accreditation. Teacher College Press.
- [2] Amos Q. Haviv, 2014, "MEAN Web Development". Packt Publishing Ltd, Birmingham B2 2PB UK.
- [3] Chodorow, Kristina, 2013, "MongoDB- The Definitive Guide, 2nd Edition", O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
- [4] Yaapa, Hage, 2013, "Express Web Application Development". Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK.

- [5] Branas, Rodrigo, 2014, "Angular Js Essentials".Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK.
- [6] Mark, David Clement, 2012,"Node Js Cookbook",Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK.