

APLIKASI ASISTEN PRAKTIKUM MENGGUNAKAN NODEJS DAN DATABASE MONGODB (STUDI KASUS LAB STMIK AKAKOM)

Ahmad Rohman¹⁾, Indra Yatini B²⁾

Program Studi Teknik Informatika, STMIK AKAKOM Yogyakarta^{1,2)}

Jalan Raya Janti 143 Karang Jambe Yogyakarta 55198

e-mail : ahmadro10084@gmail.com¹⁾ , indrayatini@akakaom.ac.id²⁾

ABSTRAK

Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) AKAKOM Yogyakarta merupakan sekolah tinggi yang bergerak dalam bidang teknologi informasi. Untuk menunjang proses pembelajaran, STMIK AKAKOM memiliki sebuah laboratorium terpadu yang digunakan untuk pelaksanaan perkuliahan praktikum yang mana pelaksanaan praktikum tersebut dibantu dengan beberapa mahasiswa sebagai asisten dan instruktur. Masalah yang kemudian muncul adalah semua kegiatan pendaftaran dan penjadwalan asisten masih manual. Sehingga mengharuskan calon asisten mendatangi satu persatu ruang laboratorium untuk mendapatkan informasi seputar pendaftaran. Hal ini menyebabkan kekosongan atau kekurangan tenaga asisten dalam pelaksanaan praktikum, Oleh karena itu penulis berinisiatif untuk membuat sebuah aplikasi asisten berbasis web.

Aplikasi asisten dalam penelitian ini akan dibuat dengan memanfaatkan platform as a service yaitu nodejs dan menggunakan database NoSQL MongoDB, Serta menggunakan framework expressjs dan menggunakan arsitektur MVC. Pembuatan aplikasi asisten ini mengutamakan pada pembuatan sistem pendaftaran asisten.

Implementasi dan pengujian aplikasi asisten, aplikasi ini berjalan sesuai yang di rencanakan. penulis berharap aplikasi ini dapat memberikan manfaat untuk kepala maupun petugas lab dalam mengelola asisten praktikum, juga memberikan solusi untuk hal pendaftaran asisten yang masih di laksanakan secara manual.

Kata kunci : *asisten, expressjs, Lab, MongoDB, Nodejs, NoSQL*

ABSTRACT

School of Information Management and computer (STMIK) AKAKOM Yogyakarta is a high school that specializes in information technology. To support the learning process, STMIK AKAKOM have an integrated laboratory used for the implementation of practical lectures where practical implementation is assisted by several students as assistants and instructors. The problem that then arises is all registration activities and scheduling assistant is still manual. Thereby necessitating the prospective assistant approached one by one laboratory space to obtain information about registration. This causes a void or a shortage of assistants in practical implementation, authors therefore took the initiative to create a web-based application assistant.

Application assistant in this experiment will be made by utilizing the platform as a service that is nodejs and use MongoDB NoSQL database, as well as using the framework expressjs and using MVC architecture. Application creation assistant is put on making the registration system assistant. Implementation and application testing assistant, the application is progressing as planned. the author hopes this application can provide benefits to the head as well as lab assistant in managing the lab assistant, also provide a solution for the registration of an assistant who still carried manually.

Keywords: *assistant, express js, Lab, MongoDB, Nodejs, NoSQL*

I. PENDAHULUAN

Perkembangan teknologi informasi menjadi pemicu berdirinya sekolah-sekolah tinggi yang bergerak dalam bidang teknologi informasi, salah satunya yaitu AKAKOM. Dengan visi dan misinya, AKAKOM memiliki lima (5) jurusan yaitu Teknik Informatika, Sistem Informasi, Manajemen Informatika, Teknik Komputer dan Komputerisasi Akuntansi. Untuk mewujudkan visi dan misinya, AKAKOM memiliki laboratorium terpadu yang digunakan untuk kegiatan praktikum. Pada kegiatan praktikum setidaknya melibatkan beberapa pihak seperti pengelola lab, dosen pengampu dan asisten. Asisten dalam kegiatan praktikum bertujuan untuk membantu dosen dalam kegiatan praktikum.

Namun dalam pelaksanaannya, terkadang terdapat kegiatan praktikum yang tidak memiliki asisten. Hal ini dikarenakan adanya beberapa masalah yaitu kurangnya informasi bagi para calon asisten tentang mata praktikum apa yang belum memiliki asisten atau kekurangan asisten, pendaftaran asisten yang masih manual harus mendatangi satu persatu ruangan laboratorium untuk mendaftar sebagai asisten, dan masalah lain yang ada yaitu dalam mengelola data asisten, terkadang pengelola lab tidak tahu berapa banyak jumlah asisten praktikum setiap semester, siapa saja asisten yang terdaftar dalam setiap ruangan laboratorium, dan berapa jumlah asisten setiap pelaksanaan praktikum di masing-masing ruangan laboratorium.

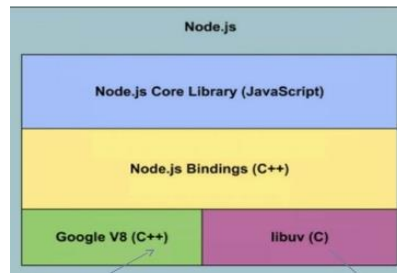
Dari permasalahan di atas, maka akan dibuat sebuah aplikasi asisten berbasis web dengan menggunakan Node.js sebagai peranti lunak yang menggunakan bahasa pemrograman javascript dan menggunakan MongoDB sebagai database untuk mengelola data asisten.

Aplikasi web sendiri merupakan aplikasi yang berjalan dengan menggunakan peranti lunak web browser. Untuk membuat aplikasi berbasis web ada banyak peranti lunak yang dapat digunakan salah satunya yaitu Node.js. Node.js dibuat oleh Ryan Dahl pada tahun 2009, Node.js merupakan peranti pengembang untuk membuat aplikasi web yang menggunakan JavaScript sebagai bahasa pemrograman. Bahasa pemrograman lainnya yang dapat digunakan salah satunya yaitu PHP, yang merupakan bahasa pemrograman server side yang sangat banyak digunakan saat ini, Jika dibandingkan dengan Node.js ada perbedaan mendasar yang membedakan ke duanya yaitu Node.js mendukung model non-blocking I/O (asynchronous) dan event driven, sedangkan PHP belum mendukungnya.

Aplikasi asisten ini juga akan dibuat menggunakan database MongoDB yang merupakan database *Document Store / Document-Oriented Database*, yaitu data disimpan dalam bentuk dokumen, sehingga sangat cocok jika digunakan untuk mengelola data pada aplikasi Asisten karena data tidak mengandung proses transaksi.

II. METODE PENELITIAN

Node.js dibangun dengan menggunakan JavaScript dan C/C++, adapun arsitektur node.js seperti Gambar 1. Google V8 dalam arsitektur node.js berfungsi sebagai mesin compiler yang ditulis dalam C++ dan Library Libuv bertanggung jawab untuk menangani operasi asynchronous I/O dan event loop utama.



Gambar 1. Arsitektur node.js

MongoDB adalah salah satu software NoSQL yang termasuk dalam kategori *Document Store / Document-Oriented Database*, yaitu data disimpan dalam bentuk dokumen. Suatu dokumen bisa diibaratkan seperti suatu record dalam basis data relasional dan isi dari masing-masing dokumen tersebut bisa berbeda-beda dan ada pula yang sama. Hal ini berbeda dengan basis data relasional yang menetapkan keseragaman kolom serta tipe data dengan data yang NULL jika tidak terdapat data. MongoDB menyimpan data dalam bentuk dokumen dengan menggunakan format JSON. Konsep dasar yang harus dipahami dalam MongoDB sebagai document-oriented database adalah documents dan collections. Sama halnya dengan basis data relasional, MongoDB menyimpan data dalam suatu basis data. Di dalam basis data tersebut terdapat collections yang bisa diibaratkan seperti tabel dalam basis data relasional. Collections digunakan untuk menyimpan dokumen (documents). Dalam istilah basis data relasional, documents adalah records.

Pembuatan database pada MongoDB sangat berbeda dengan pembuatan database di SQL. Dalam SQL kita harus membuat database terlebih dahulu sebelum melakukan perintah 'use' dengan menggunakan perintah create database db_name. Dalam MongoDB tidak perlu menggunakan perintah create kita bisa langsung menggunakan perintah use meskipun kita tahu bahwa database yang dimaksud belum ada dalam MongoDB, namun MongoDB mengizinkan kita untuk melakukan hal itu. Tetapi saat keluar dari database tersebut, maka database akan hilang, hal ini dikarenakan kita tidak membuat collections dan mengisi documents dalam collections.

Framework adalah kerangka kerja yang terdiri dari kumpulan kelas dan fungsi yang disusun secara sistematis, sehingga dapat digunakan untuk membantu membuat aplikasi utuh tanpa harus membuat semua kodenya dari awal. Secara umum framework menggunakan struktur MVC. Dalam pembuatan aplikasi menggunakan peranti lunak Node.js ada berbagai framework yang dapat digunakan salah satunya adalah express.js. Express.js adalah kerangka aplikasi web Node.js yang minimal dan fleksibel yang menyediakan seperangkat fitur untuk mengembangkan web dan aplikasi mobile.

Berikut ini adalah beberapa fitur inti dari kerangka Express.js:

1. Memungkinkan untuk mengatur middlewares untuk menangani Permintaan HTTP.
2. Mendefinisikan sebuah tabel routing yang digunakan untuk melakukan tindakan yang berbeda berdasarkan metode HTTP dan URL.
3. Memungkinkan untuk secara dinamis membuat Pages HTML berdasarkan argumen untuk template.

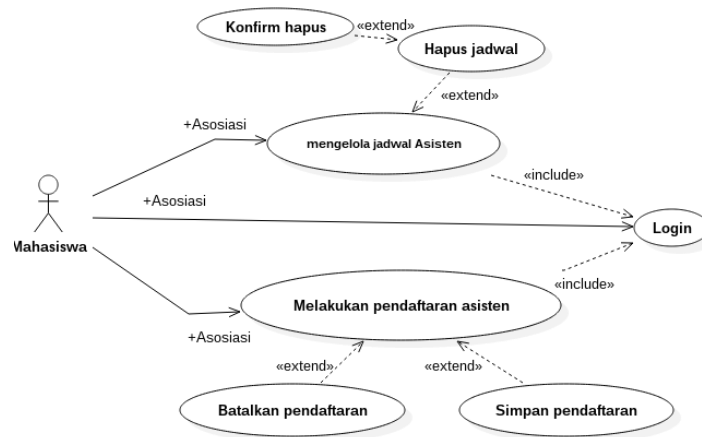
Merupakan sebuah driver mongoose untuk mengkoneksikan antara program JavaScript dengan database MongoDB. Mongoose juga mendukung pustaka ODM (Object-Document Modeler) sehingga lebih handal untuk penanganan pemodelan data. Selain itu mongoose juga mendukung schema type dan validasi data.

Aplikasi asisten ini secara umum merupakan aplikasi untuk pendaftaran asisten, sehingga mahasiswa tidak perlu lagi melakukan pendaftaran secara manual dengan mendatangi satu persatu

ruangan lab. Aplikasi ini nantinya akan menyediakan akun untuk masing-masing mahasiswa, sehingga mahasiswa tidak perlu melakukan registrasi untuk memiliki akun. dalam melakukan pendaftaran tentunya ada dua syarat yang harus dipenuhi bagi calon asisten yaitu, Calon asisten harus sudah mengambil mata praktikum yang bersangkutan, dan nilai dari mata praktikum harus minimal B.

Jika salah satu syarat diatas tidak terpenuhi maka mahasiswa tidak diperkenankan untuk mendaftar sebagai asisten, aplikasi akan secara otomatis menolak pendaftaran tersebut jika salah satu syarat tidak terpenuhi.

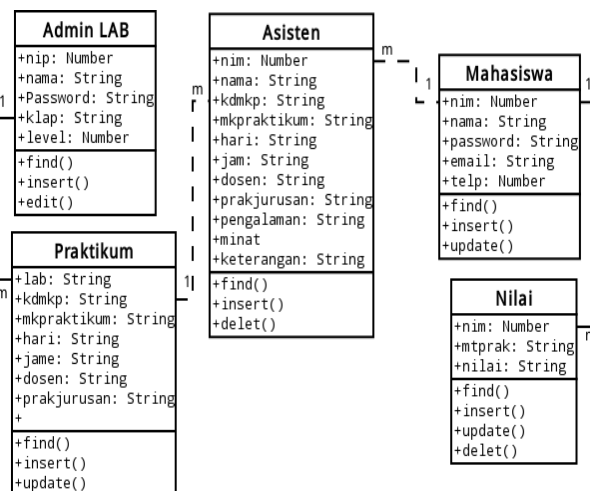
Pemodelan dalam penelitian ini menggunakan UML (Unified Modeling Language) merupakan standard modeling language yang terdiri dari kumpulan-kumpulan diagram. Use-case diagram merupakan model diagram UML yang digunakan untuk menggambarkan requirement fungsional yang diharapkan dari sebuah sistem. Use case diagram mahasiswa pada aplikasi ini adalah sebagai erikut berikut :



Gambar 2. Use case Diagram Mahasiswa

Pada aplikasi ini akan menggunakan basis data dengan nama Asisten yang terdapat beberapa collections yaitu : user (Kepala lap dan pengurus lab),mahasiswa, jadwal praktikum, dan jadwal asisten.

Perancangan collections dapat dilihat pada class diagram berikut :

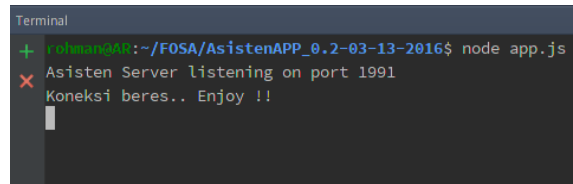


Gambar 3. Collections

III. HASIL

Hasil dari implementasi ini adalah sebuah sistem pendaftaran asisten. Sebelum dapat melakukan pendaftaran, perlu menyiapkan akun untuk mahasiswa dan jadwal praktikum. Dalam aplikasi ini terdapat sebuah fitur untuk mengatur waktu pendaftaran, sehingga jika waktu pendaftaran sudah berakhir maka secara otomatis sistem akan mengalihkan ke halaman informasi.

Aplikasi asisten ini dibuat dengan menggunakan arsitektur MVC, untuk menjalankan aplikasi nodejs dapat menggunakan perintah berikut node name_app_server.js, seperti gambar berikut:



```
Terminal
+ root@kali: ~/FOISA/AsistenAPP_0.2-03-13-2016$ node app.js
Asisten Server listening on port 1991
Koneksi beres.. Enjoy !!
```

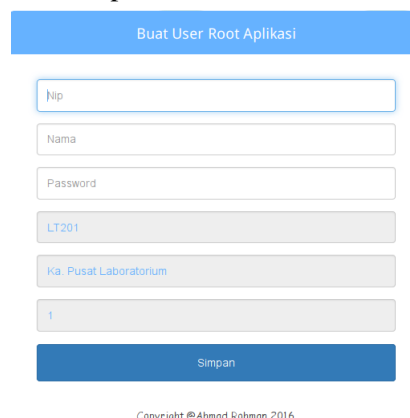
Gambar 4. Server

Dari gambar 4, menjelaskan bahwa aplikasi dapat diakses dengan port 1991. Pada penelitian ini aplikasi memiliki satu halaman login yang dapat digunakan oleh dua user dengan hak akses yang berbeda yaitu admin lab dan mahasiswa.



Gambar 5. Halaman Login

Saat pertama kali aplikasi dijalankan tentunya dalam database AsistenDB belum memiliki document satupun termasuk untuk collectins adminlab. Maka pada aplikasi asisten ini terdapat perintah untuk menghitung admin root menggunakan query count. Jika query count menghasilkan nilai 1 (satu) itu berarti admin root dengan level 1(satu) ditemukan, maka akan dirender ke index.jade untuk menampilkan halaman login, namun jika hasil count memberi nilai 0 atau kosong maka akan dirender ke admin_root.jade untuk menampilkan form buat admin root.

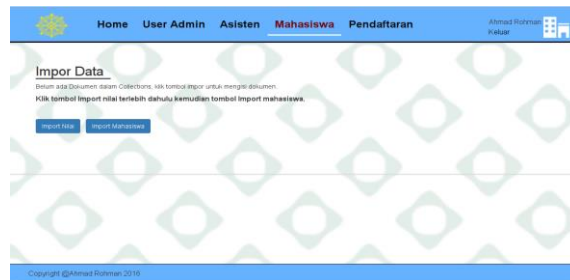


Gambar 6. user root

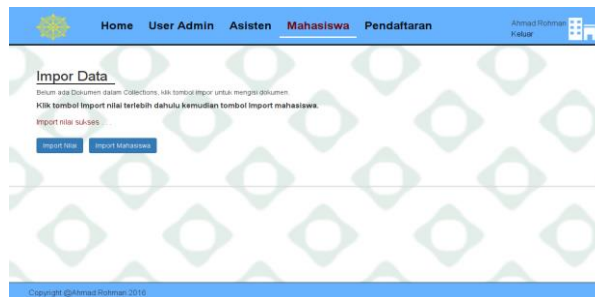
Halaman buat user root pada aplikasi ini hanya akan ditampilkan saat pertama kali aplikasi disiapkan Untuk membedakan halaman dari setiap user baik admin lab maupun mahasiswa, maka pada script login dibuat sebuah perintah untuk menghitung panjang karakter dari username jika username lebih dari 8 karakter maka akan di buat session untuk mahasiswa namu jika kurang dari 8 karakter

akan di buat session untuk admin baik itu admin lab maupun kepala lab.

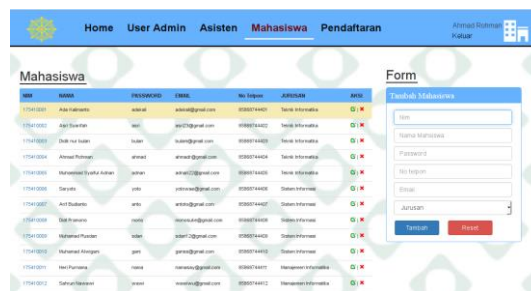
Halaman mahasiswa digunakan untuk mengelola data mahasiswa STMIK AKAKOM Yogyakarta, untuk pertama kali halaman ini diakses maka akan ditampilkan halaman inport. Halaman ini ditampilkan karena data mahasiswa dalam collection belum ada. Pada halaman ini terdapat dua button yaitu import nilai dan import mahasiswa. Jika tombol ini di klik maka aplikasi akan membaca file dengan format xlsx. File ini harus diletakkan dalam folder public/file, aplikasi akan mengkonversi data dari xlsx ke bentuk JSON, kemudian data tersebut di simpan dalam database. Urutan untuk mengimport data pada halaman ini adalah mengklik tombol import nilai, aplikasi akan mengimport data ke database proses ini berjalan beberapa detik. Jika proses import selesai maka akan ditampilkan informasi bahwa data nilai berhasil di import seperti pada gambar 7. Setelah itu kemudian mengklik tombol import mahasiswa, jika proses import berhasil maka akan di render ke halaman data mahasiswa seperti pada gambar :



Gambar 7. Import data

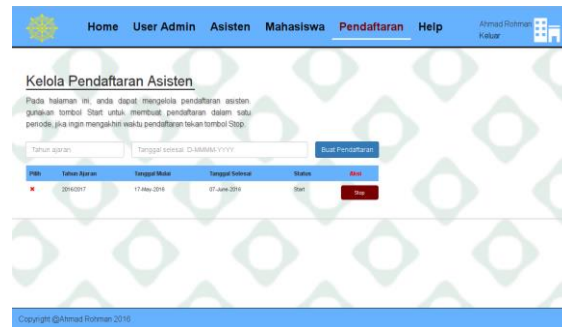


Gambar 8. Import nilai sukses



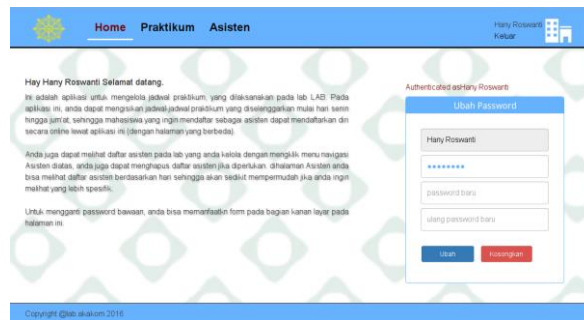
Gambar 9. Data Mahasiswa

Halaman pendaftaran disini digunakan untuk membuat periode pendaftaran asisten, ketika tombol buat pendaftaran di klik maka akan dibuat periode pendaftaran dengan waktu mulai saat periode pendaftaran dibuat dan batas akhir pendaftaran yang telah ditentukan oleh kepala lab dalam form, dan status secara default saat periode pendaftaran dibuat adalah Start. Kemudian untuk mengakhiri periode pendaftaran kepala lab dapat mengklik tombol Stop.



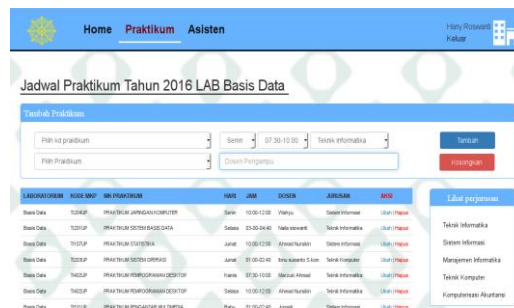
Gambar 10. Periode Pendaftaran

Petugas lab dalam penelitian ini hanya memiliki beberapa halaman untuk dikelola yaitu ; Halaman home merupakan halaman awal setelah petugas lab melakukan login. Dalam halaman ini terdapat sebuah form yang dapat digunakan untuk mengganti password.



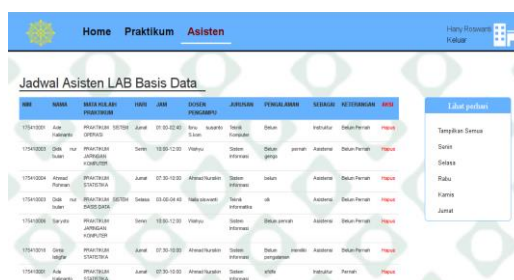
Gambar 11. Home

Pada halaman ini petugas lab dapat menambah, mengubah dan menghapus jadwal praktikum. Dalam halaman ini petugas lab juga dapat melihat jadwal praktikum yang di laksanakan perhari. Jika penambahan jadwal terdapat jam yang sama pada suatu hari maka sistem akan menolak penambahan jadwal praktikum.



Gambar 12. Praktikum

Petugas lab juga memiliki hak untuk melihat data asisten yang ada pada lab yang dikelola. Daftar asisten pada halaman ini ditampilkan berdasarkan lab yang dikelola petugas lab. Dalam halaman ini juga terdapat menu untuk melihat daftar asisten berdasarkan hari. Petugas lab memiliki hak untuk menghapus data asisten jika ada kondisi tertentu.



Gambar 13. Buat user root

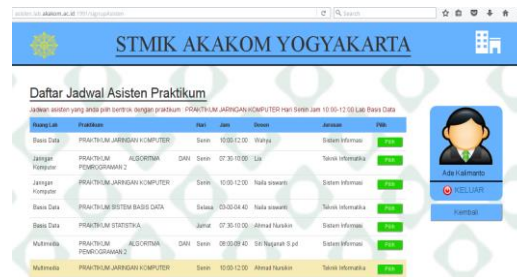
Halaman ini akan ditampilkan jika mahasiswa meng-klik tombol kelola jadwal pada halaman home. Halaman ini hanya akan ditampilkan jika masa pendaftaran asisten masih dibuka.



Gambar 14. Kelola Jadwal asisten

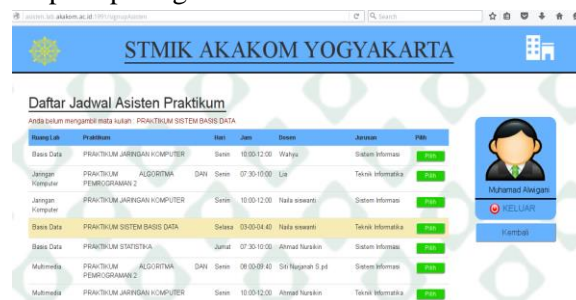
Jika pendaftaran asisten telah ditutup maka sistem akan secara otomatis menampilkan halaman informasi bahwa pendaftaran asisten telah ditutup atau informasi bukan periode pendaftaran.

Memastikan jadwal tidak bentrok : Pada aplikasi ini terdapat perintah untuk memastikan bahwa jadwal asisten praktikum yang akan diambil tidak bentrok dengan jadwal asisten yang sudah diambil, berikut adalah pemberitahuan jika terjadi bentrok.



Gambar 15. Informasi Jadwal Bentrok

Praktikum belum diambil : Mahasiswa harus sudah mengambil praktikum yang akan di asistensi. Pada pada saat mahasiswa mengklik tombol pilih maka akan ada suatu kondisi didalam program untuk mencari apakah mahasiswa sudah mengambil mata praktikum, jika belum diambil maka akan ditampilkan pemberitahuan seperti pada gambar berikut.



Gambar 16. Informasi Praktikum Belum Diambil

Nilai harus A atau B : Kemudian setelah mata praktikum yang akan diasistensi ditemukan maka kemudian aplikasi akan melihat nilai dari praktikum yang diambil, jikan nilai C maka tidak diperkenankan untuk menjadi asisten.

IV. PEMBAHASAN

Pada aplikasi ini dibuat menggunakan platform as a service yaitu Nodejs, serta menggunakan database MongoDB. Untuk mempermudah dalam pembuatan aplikasi ini, digunakan framework expressjs dan aplikasi ini dibuat dengan memanfaatkan arsitektur MVC.

Aplikasi ini dibuat tanpa memerlukan web server yang terpisah seperti apache2, httpd atau Nginx.

Web server pada aplikasi ini sudah tersedia dalam Nodejs sehingga kita hanya perlu memanggil modul http atau https yang tersedia. Jika menggunakan framework express, otomatis akan dibuat sebuah file dengan nama www dalam folder bin dalam projek sebagai web server.

Aplikasi ini difokuskan pada pendaftaran asisten. Proses pendaftaran asisten berjalan seperti yang direncanakan, seperti yang telah di bahas hasil penelitian, semua kondisi pada program berjalan sesuai prosedur. Namun pada halaman jadwal asisten yang ditawarkan tidak terdapat kolom kuota, sehingga sedikit mempersulit mahasiswa jika ingin melihat praktikum mana yang kuotanya penuh. Untuk mengetahui kuota dari suatu jadwal asisten penuh, maka calon asisten harus mengklik tombol pilih satu-persatu. Hal ini dikarenakan query mongodb untuk agregat terbilang rumit.

Aplikasi asisten ini memungkinkan bagi mahasiswa untuk membuat jadwal asisten praktikum sendiri, dengan begitu pengurus lab tidak perlu meminta jadwal kuliah mahasiswa kemudian baru membuatkan jadwal asisten, seperti yang ada pada penelitian sebelumnya.

V. SIMPULAN DAN SARAN

Kesimpulan dari implementasi ini adalah bahwa, node.js merupakan platform yang sangat memudahkan programmer dalam membangun aplikasi web. Dengan menggunakan node.js kita hanya perlu memahami satu bahasa pemrograman yaitu JavaScript, yang dapat dijadikan bahasa disisi server sekaligus disisi client. Hampir semua yang dibutuhkan programmer dalam membuat aplikasi telah disediakan oleh node.js, sehingga programmer tidak perlu membuat sebuah fungsi dari awal, contoh : dalam bahasa pemrograman lain yang pernah dipelajari, untuk menangkap data yang dikirim dari form maka kita perlu membuat sebuah file atau fungsi untuk menangkap data yang dikirim dari form tersebut, di node.js ada sebuah modul yaitu body-parser, jika ada data yang dikirim dari form dengan method post maka data akan disimpan dalam body-parser, selanjutnya kita hanya perlu memanggilnya dengan perintah req.body.

Selain itu pemanfaatan mongodb sebagai database juga sangat memudahkan dalam pembuatan aplikasi ini, karena kita tidak perlu meng-create database atau collections melalui console mongodb, kita dapat langsung membuat file koneksi ke mongodb meskipun database yang kita tuju belum dibuat, dan collections(dalam SQL disebut tabel) dapat didefinisikan dalam script node.js. Sehingga data yang akan disimpan ke database akan mengikuti model dari collections yang sudah didefinisikan. Namun dalam mengatasi hubungan antar collections mongodb cukup sulit untuk diterapkan karena dasar dari mongodb adalah schema-free tidak mendefinisikan primary key ataupun foreign key. Juga dokumentasi untuk penyajian collections dalam bentuk diagram sangat minim, sehingga dalam implementasi ini peneliti membuat schema collections berdasarkan kebutuhan peneliti.

Dalam membuat sebuah aplikasi tentunya, masih jauh dari kata sempurna atau bahkan tidak bisa dikatakan aplikasi sudah jadi. Aplikasi yang dibuat pada penelitian ini hanya berupa pendaftaran asisten, sehingga masih banyak yang perlu ditambahkan dalam aplikasi ini. Berikut adalah beberapa saran yang mungkin dapat melengkapi kekurangan dari aplikasi ini.

Menambahkan fitur untuk upload laporan berformat pdf, dan sekaligus membuka dokument pdf pada aplikasi asisten, jadi mahasiswa praktikum tidak perlu mengumpul laporan dalam bentuk hard copy. Dan asisten atau dosen dapat membuka laporan mahasiswa tanpa harus mengunduhnya terlebih dahulu.

Menambahkan fitur untuk presensi kehadiran asisten dan fitur perhitungan insentif untuk asisten praktikum. Dimana fitur ini nantinya dikelola oleh petugas lab dan kepala lab atau siapapun yang berwenang dalam masalah pemberian insentif.

Menambahkan fitur untuk mengexport data dari database SQL ke database NoSQL, dimana proses

expor dilakukan secara otomatis oleh sistem setiap kali ada perubahan atau penambahan data dari database SQL, sehingga pihak admin tidak perlu melakukan input data secara manual dari ratusan data yang ada.

REFERENSI

- [1] Agus Kurniawan, 2014, Node.js Succintly,[Online],Aerial Center Parkway Suite 200 Morrisville, NC : 27560 USA : Syncfusion Inc. Link <https://www.syncfusion.com/resources/techportal/details/ebooks/nodejs> (Diakses pada 19 Agustus 2015 pukul 02:01)
- [2] Agus Kurniawan, 2014, MongoDB Succintly,[Online],Aerial Center Parkway Suite 200 Morrisville, NC : 27560 USA : Syncfusion Inc. Link <https://www.syncfusion.com/resources/techportal/details/ebooks/mongodb> (Diakses pada 19 Agustus 2015 pukul 02:15)
- [3] Bambang Purnomosidi D. P, 2015, Express-mongoose-mvc, <https://github.com/bpdp/express-mongoose-mvc> (Diakses pada 13 Februari 2016 pukul 04:05)
- [4] Dinny Wahyu Widarti, Dwi Safiroh Utsalina, dan Sri Esti Trisno Sami, 2011, DESAIN DAN IMPLEMENTASI SISTEM PENYUSUNAN JADWAL ASISTEN PRAKTIKUM PADA LABORATORIUM KOMPUTER SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER (STMIK) PRADNYA PARAMITA MALANG MENGGUNAKAN ALGORITMA GENETIKA <http://ejurnal.pradnya-paramita.ac.id/index.php/TI/article/view/128> (11:46 10-04-2016)
- [5] Muhammad Rizki Samsul Ariefin, Cucu Suhery, dan Yulrio Brianorman, 2014, SISTEM REAL-TIME UNTUK MANAJEMEN MOBIL ANTARKOTA MENGGUNAKAN NODE JS BERBASIS TCP/IP, <http://jurnal.untan.ac.id/index.php/jcskommipa/article/view/7616> (Diakses pada 23 Oktober 2015 pukul 13:39)
- [6] Putri Dyah Apsari, 2015, SISTEM PENDUKUNG KEPUTUSAN DALAM PEMILIHAN ASISTEN PRAKTIKUM UNIVERSITAS MUHAMMADIYAH SURAKARTA MENGGUNAKAN METODE TOPSIS (Studi Kasus Laboratorium Informatika) <http://eprints.ums.ac.id/36043/41/NASKAH%20PUBLIKASI%20NEW.pdf> (05:35 11-04-2016)
- [7] Suprihatin, APLIKASI JADWAL ASISTENSI FASILKOM UNIVERSITAS MERCU BUANA BERBASIS WEB. Skripsi UNIVERSITAS MERCU BUANA, Jakarta Barat. 2013
- [8] Zuniar Rizqi Prastyo, 2015, PEMANFAATAN GOOGLE MAPS API UNTUK Mencari Lokasi SPBU TERDEKAT DI KOTA JEPARA DAN KUDUS DENGAN TEKNOLOGI NODE-JS <http://eprints.umk.ac.id/5007/>