

RANCANGAN ARSITEKTUR APLIKASI PENGUMPULAN TUGAS DENGAN *PUSH NOTIFICATION REAL-TIME* MENGGUNAKAN NODE.JS

Muhammad Agung Rizkyana¹⁾, R Sandhika Galih Amalga²⁾

^{1,2)}Program Studi Teknik Informatika Universitas Pasundan Bandung

Jl. Dr. Setiabudhi no. 193 Bandung

e-mail : agung.rizkyana@mail.unpas.ac.id¹⁾, sandhikagalih@unpas.ac.id²⁾

Abstrak

Ketepatan penilaian serta kedisiplinan dalam pengumpulan tugas merupakan kondisi ideal yang harus ada khususnya di lingkungan laboratorium teknik informatika UNPAS. Namun, cara yang dilakukan masih belum bisa untuk mencapai kondisi ideal tersebut. Penelitian ini bertujuan untuk memberikan rancangan aplikasi pengelolaan pengumpulan tugas praktikan dengan melakukan pengumpulan data melalui studi literatur, wawancara dan observasi, analisis kebutuhan dan perancangan aplikasi. Aplikasi ini berbasis web dan menerapkan teknologi javascript server side serta push notification real-time. Melalui pendekatan real-time event triggered, diharapkan mampu memberikan efisiensi sumber daya dan efektifitas notifikasi agar kondisi ideal tersebut dapat terjadi.

Kata Kunci : notifikasi, real-time, javascript, penilaian, rancangan.

1. PENDAHULUAN

Ketepatan penilaian serta kedisiplinan dalam pengumpulan tugas merupakan dua hal yang harus bisa di peroleh oleh asisten laboratorium dan praktikan itu sendiri. Ketepatan penilaian akan memberikan pengaruh positif bagi praktikan karena hasil kerja mereka di apresiasi dan asisten menjadi tidak bingung dalam hal menilai tugas praktikan. Kedisiplinan dalam pengumpulan dan penilaian tugas akan menjadi suatu budaya yang baik bagi praktikan dan asisten untuk menghadapi dunia kerja maupun kegiatan mereka lainnya.

Deskripsi pada paragraf diatas merupakan kondisi ideal yang minimal harus ada dalam kegiatan praktikum di laboratorium teknik informatika UNPAS. Kondisi saat ini adalah pengumpulan tugas pada saat pertemuan berlangsung atau melalui email. Pengumpulan tugas saat pertemuan praktikum akan mengurangi waktu praktek serta penumpukan tugas yang harus dinilai. Pengumpulan tugas melalui email masih dirasa kurang efektif karena belum ada notifikasi secara langsung bahwa ada email dengan format subjek yang telah di sepakati masuk ke *inbox* asisten. Keadaan tersebut mengakibatkan pengelolaan tugas praktikan menjadi sulit dan penilaian akan berlangsung tanpa adanya standar penilaian yang tepat meskipun aturan estimasi waktu pengumpulan sudah disepakati.

Perlu adanya sistem dalam bentuk aplikasi dengan kemampuan notifikasi *real – time* serta pengelolaan estimasi waktu pengumpulan tugas. Aplikasi tersebut harus berbasis web dan menggunakan *javascript server side*. Penghematan sumber daya dan kemampuan *real – time* yang dimiliki javascript akan sangat mendukung pengelolaan tugas praktikan laboratorium teknik informatika UNPAS. Teknologi *javascript server side* menggunakan node.js. Node.js memberikan efisiensi sumber daya, mendukung implementasi *real-time system* melalui modul socket.io untuk *push notification*. Manfaat yang ada pada node.js menjadi gagasan penulis untuk melakukan penelitian dan menulis makalah dengan judul "Rancangan arsitektur aplikasi pengumpulan tugas dengan *push notification real-time* menggunakan node.js".

2. TINJAUAN PUSTAKA

2.1 Real-time system

Real-time system merupakan sistem yang harus menghasilkan tanggapan yang tepat dalam suatu batas waktu yang telah ditentukan. *Real-time system* juga merupakan sistem yang kebenarannya secara logis didasarkan pada kebenaran komputasi dari hasil – hasil keluaran sistem dan ketepatan waktu kapan hasil – hasil tersebut di keluarkan. Ada dua pendekatan *real-time system*, yaitu *event triggered* dan *time triggered*. (Sridadi, 2010).

2.2 Paradigma Object Oriented

Object oriented merupakan konsep untuk mendapatkan solusi melalui pengembangan serta rekayasa perangkat lunak dengan membuat analogi terhadap objek – objek di dunia nyata. Solusi tersebut dihasilkan melalui analisis dan identifikasi objek – objek. Pada tahapan analisis, menentukan objek – objek apa saja yang harus ada sesuai dengan *requirement*. Menentukan atribut serta operasi yang dilakukan oleh objek tersebut. Membuat skenario

aliran pesan antar objek. Menentukan aturan *encapsulation*, *polymorph* serta *inheritance* pada objek – objek tersebut.(Pressman, 2001)

2.3 Javascript

Javascript merupakan bahasa pemrograman yang rilis pada tahun 1995. Meskipun memiliki nama *javascript*, bahasa ini berbeda dengan bahasa pemrograman java. Ada beberapa catatan mengenai javascript, antara lain (Suehring, 2013) :

1. Javascript adalah bahasa *scripting*. Javascript adalah program *script* yang dibaca dan dieksekusi menggunakan *interpreter*. Berbeda dengan bahasa pemrograman yang di-*compiled*, dimana setiap program dibaca oleh *compiler* dan di translasikan kedalam bentuk *executable file*.
2. Seperti bahasa C. Dasar dari *syntax* dan struktur javascript mengacu pada bahasa pemrogramana C.
3. Bahasa berorientasi objek.

Inti dari javascript mengandung kumpulan inti objek – objek, seperti *array*, *date*, dan *math* serta kumpulan inti dari bahasa pemrograman seperti operator, struktur kendali dan *statements*. Inti dari javascript dapat diturunkan untuk memenuhi beberapa tujuan dengan menambahkan beberapa objek – objek tambahan pada inti javascript.

2.4 Javascript server side

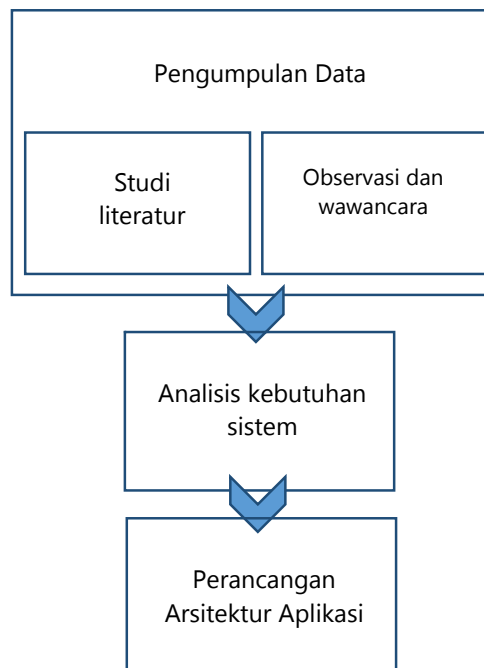
Server side javascript diturunkan dari bahasa inti javascript dengan menambah objek – objek yang relevan agar javascript dapat berjalan di server. Sebagai contoh, javascript berkomunikasi dengan basis data berelasi, manipulasi berkas pada server dan menjaga siklus informasi dari sebuah permintaan ke bagian lain dari aplikasi. (Netscape,1998).

2.5 Node.js

Node.js merupakan platform server yang dibangun menggunakan javascript dan berjalan di dalam interpreter *Chrome javascript runtime*. Dibuat untuk pengembangan perangkat lunak berbasis web dengan cepat, aplikasi jaringan yang *scalable* . Node.js menggunakan *event-driven*, model *non-blocking I/O* yang membuatnya menjadi ringan dan efisien. Sangat baik digunakan untuk aplikasi waktu-nyata yang digunakan diberbagai perangkat. sumber : <http://nodejs.org/>, di akses pada 1 Mei 2014, pukul 8.00 WIB

3. METODE PENELITIAN

Penelitian ini melakukan pengumpulan data melalui studi literatur, observasi dan wawancara. Setelah data diperoleh, kemudian melakukan analisis kebutuhan sistem. Analisis kebutuhan sistem mengidentifikasi interaksi fungsional, informasi dan objek yang terlibat. Perancangan arsitektur aplikasi memuat deskripsi implementasi kebutuhan sistem kedalam aplikasi. Pada gambar 1 (satu) mendeskripsikan metode penelitian untuk paper ini.



Gambar 1. Metode Penelitian

4. HASIL DAN PEMBAHASAN

Ada dua cara yang paling sering digunakan saat pengumpulan tugas yaitu tugas dikumpulkan saat pertemuan dan tugas kirim melalui email. Pengumpulan tugas saat pertemuan praktikum akan mengurangi waktu pelaksanaan praktikum yang mengakibatkan penyampaian materi menjadi berkurang serta sesi diskusi menjadi berkurang pula. Pengumpulan tugas melalui email masih terjadi kesalahan subjek yang mengakibatkan tugas praktikan tidak sampai ke email asisten dan tugas menjadi tidak di nilai.

Muncul kebutuhan akan suatu sistem yang mampu mengelola pengumpulan tugas. Kebutuhan sistem tersebut antara lain, mengelola estimasi waktu pengumpulan, membuat subjek atau nama file dengan format yang telah disepakati, memberikan notifikasi *real-time*. Pengelolaan estimasi waktu diperlukan agar pelaksanaan pengumpulan tugas serta penilaian tugas menjadi tepat waktu. Praktikan menjadi disiplin dan rekapitulasi tugas yang di lakukan oleh asisten menjadi mudah. Format nama subjek atau file yang di kelola secara otomatis membuat tugas menjadi mudah untuk di identifikasi dan di lakukan penilaian terhadap tugas tersebut. Notifikasi *real-time* pada perangkat yang dimiliki asisten (*Smartphone, laptop, feature phone*) dalam bentuk notifikasi dari aplikasi ataupun aplikasi memberikan sms dari server. Asisten akan segera mengenali tugas tersebut dan melakukan penilaian.

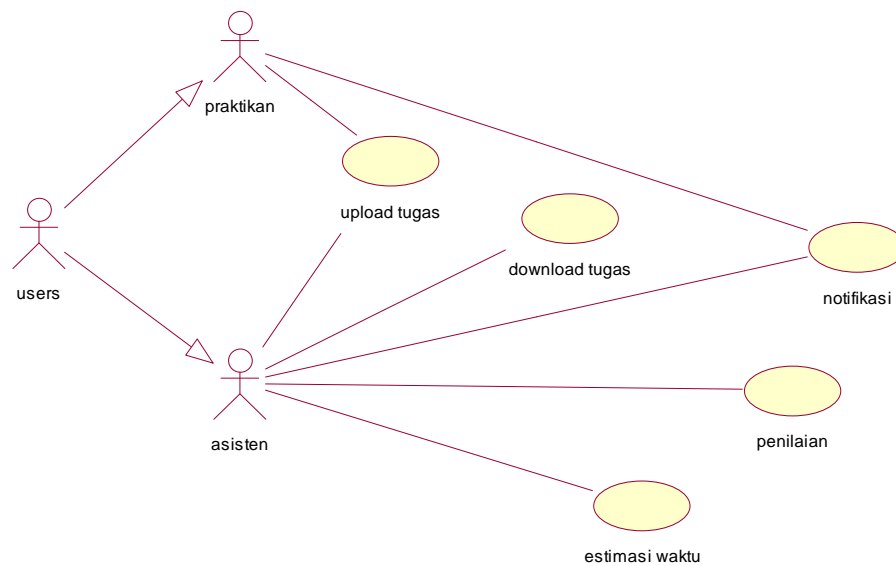
Analisis terhadap studi kasus menghasilkan kebutuhan sistem untuk membuat rancangan interaksi fungsional pengguna dengan aplikasi, antarmuka aplikasi dan penerapan teknologi *node.js* untuk *real – time system*. Peran dan objek yang terlibat dalam aplikasi ini di tunjukan pada tabel 1 (satu) berdasarkan pengelompokan objek.

Tabel 1. Pengelompokan objek

Kategori	Identifikasi Objek	Instansiasi Objek
<i>External entities</i>	-	-
Benda (<i>Things</i>)	Tugas	Tugas
Peristiwa (<i>Occurance or events</i>)	<i>Estimasi</i>	<i>estimasi</i>
	<i>Nofitikasi</i>	<i>Nofitikasi</i>
	<i>Log</i>	<i>Log</i>
Peran (<i>Roles</i>)	Asisten	<i>Users</i>
	Praktikan	
Tempat (<i>Places</i>)	-	-
Unit organisasi	-	-
Struktur	-	-

4.1. Interaksi fungsional

Kebutuhan sistem ini menjadi fungsional yang akan diterapkan pada aplikasi. Objek *users* dibagi menjadi dua asisten dan praktikan. Pembagian ini menjadi atribut peran dalam objek *users*. Peran asisten dapat mengelola estimasi waktu, download tugas praktikan, memberikan penilaian dan mendapatkan notifikasi. Peran praktikan dapat mengunggah tugas ke aplikasi. Apabila praktikan mengunggah tugas diluar waktu estimasi pengumpulan maka tugas tidak dapat di upload dan tidak dilakukan penilaian. Pada gambar 2 (dua) menunjukkan interaksi fungsional menggunakan *use-case diagram*.

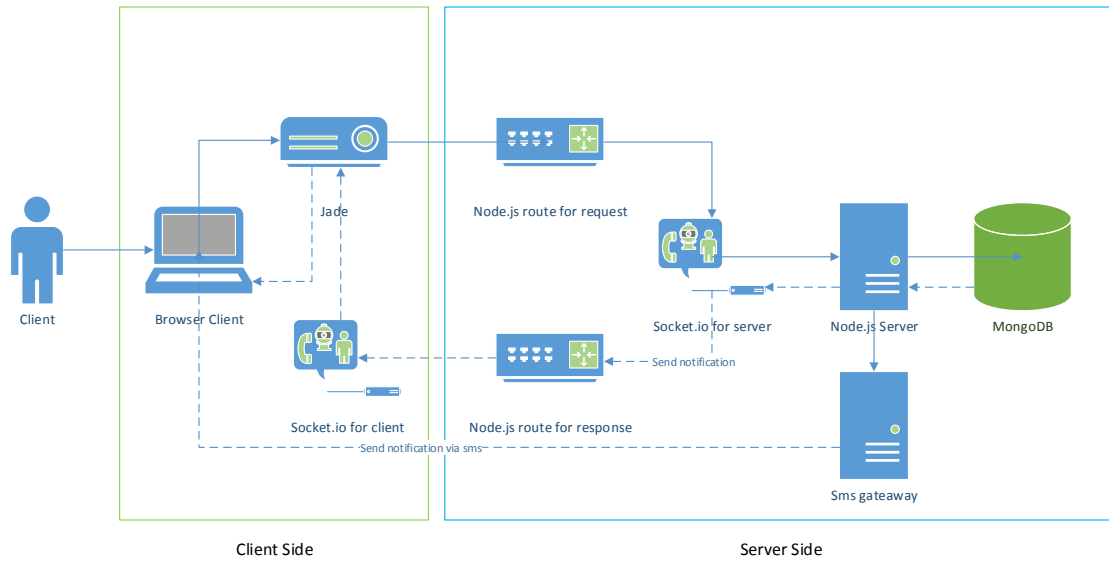


Gambar 2. Interaksi fungsional aplikasi dengan pengguna

4.2. Arsitektur aplikasi

Untuk dapat berinteraksi dengan fungsional, users membutuhkan antarmuka. Praktikan diberikan antarmuka yang dapat mengunggah tugas kemudian melihat tugas yang telah di unggah tersebut. Asisten diberikan antarmuka yang dapat mengunggah tugas, melihat tugas, unduh tugas. Selain itu, antarmuka untuk mengelola estimasi waktu, mulai dari membuat, menghapus dan mengubah estimasi waktu. Asisten dapat melakukan penilaian pada tabel tugas. Praktikan dan asisten akan mendapatkan notifikasi masing – masing. Praktikan mendapatkan notifikasi tugas sudah di upload dan diterima oleh asisten serta notifikasi tugas sudah dinilai. Asisten mendapatkan notifikasi secara akumulatif berdasarkan jumlah praktikan yang mengumpulkan tugas ke aplikasi. Setiap tugas disimpan pada objek tugas. Setiap peristiwa yang terjadi dicatat pada objek log. Estimasi waktu dicatat pada objek waktu estimasi.

Node.js berperan pada pengelolaan aplikasi di sisi server. Untuk mengelola navigasi aplikasi, pencatatan setiap peristiwa ke log dan pengelolaan estimasi waktu. Untuk real-time notifikasi node.js perlu tambahan modul socket.io. Pendekatan *real-time* yang diterapkan adalah *event-triggered*. Saat ada peristiwa upload atau download atau tugas sudah dinilai maka aplikasi akan segera memberikan notifikasi kepada *Users*. Pada gambar 3 (tiga) mendeskripsikan arsitektur aplikasi.



Gambar 3. Arsitektur aplikasi

5. KESIMPULAN

Kondisi ideal memang akan sulit di terapkan dalam waktu singkat. Rancangan arsitektur aplikasi ini diharapkan mampu mengurangi hal – hal yang menjadi kendala untuk meraih kondisi ideal tersebut. Aplikasi ini harus dikembangkan dan menerapkan *real-time system* dari node.js, karena manfaat yang diberikan berupa efisiensi sumber daya dan efektifitas pengelolaan tugas yang lebih baik.

DAFTAR PUSTAKA

Hernandez, Alejandro, "An Introduction to Full-Stack Javascript".
<http://coding.smashingmagazine.com/2013/11/21/introduction-to-full-stack-javascript/#more-129418>, 21
November 2013.

Canadian International Development Agency, "Knowledge Sharing Methods, Meetings and Tools", Canada
International Development Agency, Canada, 2003.

Kiesling, Manuel, "The Node Beginner Book, A comprehensive Node.js tutorial".
<http://www.leanpub.com/nodebeginner>, 22 April 2013.

O'Doherty, Mike, "Object-Oriented Analysis & Design understanding system development with uml 2.0",
WILEY, England, 2005

Netscape Communications Corporation, "Server-Side Javascript Guide version 1.2", Netscape Communication
Corporation, USA, 1998.

Pressman, Roger, S. "Software Engineering A Practitioner's Approach Fifth Edition", The McGraw-Hill, New
York, 2001

Pressman, Roger, S. Lowe, David. "Web Engineering A Practitioner's Approach", The McGraw-Hill, New
York, 2009

Sridadi, Bambang, "Sistem Waktu Nyata", Informatika, Bandung, 2010.

Suehring, Steve, "Javascript Step by Step Third Edition", California, 2013