Jurnal Informatika, Vol. 13, No. 1, Mei 2015, 25-32 DOI: 10.9744/informatika.13.1.25-32
ISSN 1411-0105

PEMBUATAN APLIKASI AUDIENCE RESPONSE SYSTEM BERBASIS WEB DAN ANDROID

Albert Leonardo Pisa[1*], Henry Novianus Palit[2], Justinus Andjarwirawan[3]
Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
E-mail: albert.leo93@gmail.com, hnpalit@petra.ac.id, justin@petra.ac.id
* Penulis korespondensi

Abstrak: *Audience Response System* banyak digunakan pada berbagai *event* acara baik sebagai media untuk mengumpulkan informasi, mengumpulkan *data*, dan mengetahui pendapat masyarakat umum terhadap suatu isu, topik, berita. *Audience Response System* juga dapat digunakan untuk mengetahui kepuasan *client* atau *customer* terhadap kualitas produk atau jasa. Sayangnya kebanyakan *audience response system* yang ada saat ini berupa perangkat fisik yaitu berupa sebuah *remote* yang dimana sangat terbatas penggunaannya. Seiring berkembangnya teknologi, masyarakat memiliki *smartphone* yang dapat melakukan berbagai hal. perangkat fisik tersebut dapat digantikan dengan aplikasi atau program yang dapat di-*install* di *smartphone*, sedangkan *polling* dibuat melalui *website*. *Audience response system* dalam penelitian ini dibangun di atas *platform* Node.js, PhoneGap, dengan menggunakan Socket.IO sebagai *Javascript library WebSocket* agar mendukung komunikasi data dua arah. Berdasarkan hasil pengujian, aplikasi ini dapat menjalankan fungsi *login*, membuat *polling*, mengikuti *polling*, melakukan *voting*, dan tingkat reliabilitas sistem adalah 93%.

Kata kunci: *Audience response system*, *voting*, *polling*, *website*, *android*

Abstract: *Audience Response Systems are widely used in various events as a media to gather information, to collect data, and to know the opinions of the general public towards particular issues, topics, news. Audience Response System can also be used to find out the client or customer satisfaction to the quality of products or services. Unfortunately most of the audience response systems that exist today are in the form of a physical (or remote) device which is severely limited in use. Along with the development of technology, almost all people have a smartphone to support their daily activities. The physical remote device called a clicker can be replaced with an application or program that can be installed on smartphones, while the polls are created through the website. In this research, the Audience response system was built using Node.js, PhoneGap platform, and the WebSocket Socket.IO as Javascript library to support bi-directional data communications. Based on testing result, this application can run properly all the functionalities that have been mentioned such as login to the system, create a poll, join a poll, and give a vote. The realiability of the system is 93%..*

Keywords*: Audience response system, voting, polling, website, android*

PENDAHULUAN

Saat ini sistem yang ada untuk mengolah hasil *polling* masih menggunakan cara tradisional, yaitu data dari partisipan dikumpulkan terlebih dahulu hingga mencapai batas yang telah ditentukan atau hingga batas waktu yang diberlakukan telah berakhir, baru kemudian data diolah untuk ditampilkan hasilnya dalam berbagai bentuk. Partisipan tidak dapat langsung mengetahui hasil *polling* secara *real-time*.

Sistem *polling* yang ada saat ini, masih menggunakan alat fisik untuk melakukan *polling*. Contohnya adalah salah satu acara televisi Mario Teguh *Golden Ways*, dan acara televisi korea *I Am a Man (Naneun Namja Da)*. Alat tersebut berupa sebuah *remote* yang dilengkapi dengan tombol pilihan jawaban untuk responden. Dengan demikian pihak narasumber harus mengeluarkan biaya lebih untuk membeli alat tersebut, dan semakin banyak jumlah responden yang ditargetkan, semakin banyak pula alat yang mesti disediakan. Akibatnya, semakin banyak pula biaya yang akan dikeluarkan untuk membeli alat tersebut beserta dengan perangkat lunaknya. Hal ini tentu merupakan sebuah pemborosan.

Terdapat sebuah *website* yang cukup terkenal di *internet,* yang menyediakan jasa pembuatan *polling* gratis melalui *website* mereka, dengan berbagai fitur yang disediakan dengan alamat www.pollevery-where.com. Salah satu fitur yang paling ditonjolkan adalah pemanfaatan media SMS *(Short Message Service)* sebagai media penyebaran dan pengumpulan jawaban dari dan ke responden. Namun terdapat kelemahan dari sistem ini, yaitu untuk memberikan

jawabannya responden harus rela mengeluarkan uang untuk membayar biaya SMS sesuai tarif biasa operator.

Terdapat sebuah solusi untuk berbagai masalah diatas, yakni dengan membuat sebuah aplikasi sistem *polling* yang *real-time* berbasis *web* dan Android. Responden dapat memberikan jawabannya melalui sebuah aplikasi sederhana yang dibuat untuk menggantikan peran *remote*, sehingga meminimalkan biaya yang digunakan untuk memfasilitasi sistem *polling* yang dibutuhkan.

TINJAUAN PUSTAKA

Audience Response System

Audience Response Systems (ARS) merupakan alternatif media pengumpulan data berupa perangkat sistem informasi untuk memberikan respon dalam bentuk data numerik maupun kategorik terhadap pertanyaan yang diajukan. Hasil data selanjutnya akan disajikan sebagai *feedback* dari responden atau partisipan. [5]

Penggunaan umum ARS adalah berupa alat kontrol yang berada di tangan hadirin dihubungkan dengan perangkat lunak. Pada saat yang sama halaman presentasi menampilkan pertanyaan serta beberapa pilihan jawaban. Hadirin dapat berpartisipasi dengan memilih jawaban yang mereka anggap benar dengan cara menekan tombol koresponden pada alat nirkabel individu yang mereka pegang. Kemudian jawaban mereka akan dikirimkan ke stasiun dasar atau penerima pesan yang juga terhubung dengan komputer presentator. Dalam beberapa waktu yang telah ditentukan, ketika semua hadirin telah memberikan jawabannya, sistem akan mengakhiri pemungutan suara untuk pertanyaan terkait dan merangkumnya sebagai hasil dari seluruh jawaban peserta di ruangan tersebut.

Node.js

Node.js adalah sebuah *platform software* yang dipakai untuk membangun aplikasi – aplikasi *server-side* yang fleksibel di sebuah *network* / jaringan. Node.js menggunakan *JavaScript* sebagai bahasa pemrograman dan dapat dengan mudah menghasilkan *throughput* / pemrosesan tingkat tinggi melalui *non-blocking I/O*. Node.js memiliki fitur *built-in* HTTP *server library* yang menjadikannya mampu menjadi sebuah *web server* tanpa bantuan perangkat lunak lainnya seperti Apache atau Nginx. [8]

Node.js pertama kali dibuat oleh Ryan Dahl pada tahun 2009 yang kemudian berkembang pesat di bawah lisensi *Open Source* MIT oleh sebuah perusahaan bernama Joyent Inc.

Pada hakekatnya Node.js dikembangkan berdasarkan teknologi Google V8 *JavaScript engine* serta berisi kompilasi *script* inti dan banyak modul siap pakai yang bermanfaat sehingga pengguna (dalam hal ini *web* pengembang) tidak perlu melakukan *coding* dan mendesain segalanya dari awal. [8]

MongoDB

Mongo DB merupakan *database open-source* yang berorientasi pada dokumen, atau yang lebih populer disebut "NoSQL *(Not Only* SQL*)*". MongoDB bukanlah *database* relasional berbasis tabel yang telah lazim dikenal seperti MySQL, namun berorientasi pada dokumen. MongoDB menyimpan data dalam format serupa dengan notasi JSON (*Java Script Object Notation*) yang disebut BSON *(Binary* JSON*)* karena selain menangani data teks, juga mampu untuk menyimpan data biner (*binary*). Berikut ini adalah terminologi dan konsep yang perlu anda tahu di MongoDB, untuk memudahkan anda, maka akan saya buat tabel perbandingan dengan konsep yang ada di SQL *database*. [2]

Untuk *primary key*, walaupun penamaan istilahnya sama, namun dalam sisi implementasi sangat berbeda. *Primary key* di RDBMS adalah kolom unik di tabel yang didefinisikan sendiri oleh *user*. *MongoDB* secara otomatis membuatkan *primary key* di *field _id* dan akan terisi secara otomatis.

Tabel 1. Tabel Perbedaan MongoDB dan MYSQL

Istilah RDBMS	Istilah Mongo
database	*database*
table	*collection*
row	*document or BSON document*
column	*field*
index	*index*
table join	*embedded document*
primary key	*primary key*

npm.js

npm, adalah singkatan dari *node package manager*, yang mana merupakan sebuah *online repository*, untuk mempublikasikan dan mendistribusikan kode-kode Node.js dari sebuah proyek. Selain itu npm adalah sebuah *command-line utility* yang digunakan oleh pengembang, untuk berinteraksi dengan *repository*. *Command-line* juga biasa digunakan untuk instalasi, mengelola versi, dan manajemen dependensi paket-paket kode dari *repository* secara langsung. Kebanyakan *library* dan aplikasi yang terbuat dari Node.js dipublikasikan di npm. Namun sayangnya untuk mengunduh modul-modul yang ada di *online repository*, dibutuhkan akses *internet*. [8]

Untuk menggunakan modul, pengembang harus mengimpor modul tersebut terlebih dahulu dengan fungsi seperti ini: **var module = require ('nama_ module')** lalu variabel modul dapat digunakan pada *file* dimana anda mengimpor *file* tersebut.

Express.js

Express.js adalah *web application framework* Node.js yang paling sering digunakan dan paling populer diantara yang lain. Express.js memiliki keunggulan yaitu performa yang cepat karena *source code* yang minimalis dan didesain sedemikian rupa demi mendapatkan performa *website* yang baik. [3]

Mongoose

Mongoose adalah sebuah ODM (*Object Data Modelling* untuk MongoDB*)*. Awamnya, Mongoose lah yang bertugas menjadi jembatan antara aplikasi dengan *database*, karena Mongoose memodelkan data dari aplikasi ke dalam *collection-collection* dan mendesain struktur *document* di dalamnya [2].

Untuk menggunakan Mongoose, pertama-tama pengembang harus membuat *schema*. *Schema* berperan untuk memetakan *collection* pada MongoDB dan mendesain *document* yang ada di dalamnya. Berikut adalah contoh *schema* berdasarkan *website* resminya http://mongoosejs.com/docs/guide.html:

```
var mongoose = require('mongoose');
var Schema = mongoose.Schema;
var blogSchema = new Schema({
  title:  String,
  author: String,
  body:   String,
  comments: [{ body: String, date: Date
}],
  date: { type: Date, default: Date.now },
  hidden: Boolean,
  meta: {
    votes: Number,
    favs:  Number
  }
});
```

Socket.IO

Socket.IO merupakan *JavaScript library* untuk membuat *website* yang *real-time*, dan memungkinkan komunikasi dua arah yang *real-time* antara *web client* dan *server*. Memiliki 2 bagian *library* untuk *client-side* yang berjalan di *browser* dan *server-side library* untuk Node.js. berikut adalah cara menggunakan Socket.IO pada *server* Node.js. [10]

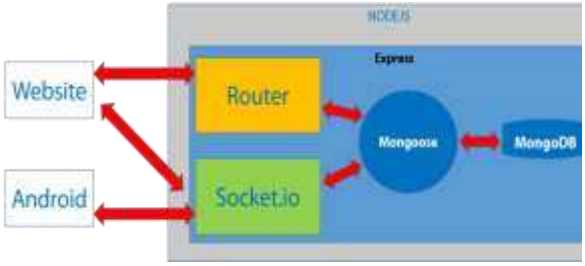
PhoneGap

PhoneGap adalah sebuah kerangka kerja/*framework open source* yang dipakai untuk membuat aplikasi *cross-platform mobile* dengan *HTML5, CSS*, dan *JavaScript*. PhoneGap menjadi suatu solusi yang ideal untuk seorang *web developer* yang tertarik dalam pembuatan aplikasi di *smartphone*. Dengan *framework* PhoneGap kita hanya perlu melakukan satu kali pemrograman dan dapat langsung di-*compile* ke semua *platform* sekaligus. [7]

ANALISA DAN DESAIN SISTEM

Desain Sistem

Skema desain sistem dan alur komunikasi data antar *platform* dapat dilihat pada Gambar 1. *Website* melakukan komunikasi dengan *server* melalui *router*, sedangkan dengan Android menggunakan Socket.IO. Sedangkan untuk mengakses *database*, baik Android maupun *website* harus melalui *mongoose*. Router, Socket.IO dan Mongoose semua merupakan bagian dari sebuah framework Node.js bernama Express.js.



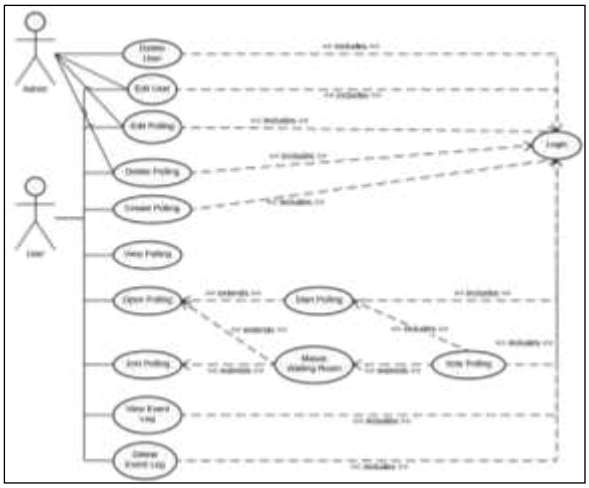
Gambar 1. Desain Sistem

Desain Aplikasi

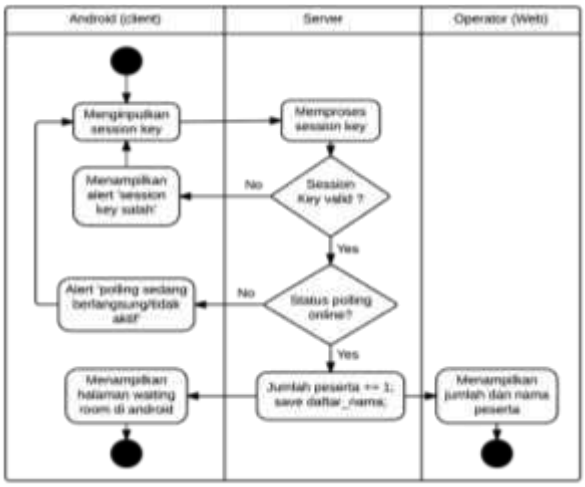
Pada Gambar 2 adalah use case diagram sistem yang dibuat dalam penelitian ini. Seperti dapat dilihat pada gambar 2, bahwa pada aplikasi ini *admin* hanya dapat menggunakan fitur *delete user, edit user, edit polling, delete polling* pada *website*. Sedangkan *user* biasa dapat menggunakan fitur *edit user, edit polling, delete polling, create polling, view polling, open polling, join polling, view event log,* dan *delete event log*. Untuk masuk ke *waiting room* dan memulai *polling (start polling)*, *user* terlebih dahulu harus memilih menu *open polling*. Sedangkan pada Android untuk bisa mengikuti *polling* dan melakukan *vote*, terlebih dahulu harus masuk ke *waiting room*, dan menunggu *operator* (*user* yang membuat *polling*) untuk memulai *polling (start polling)*.

Fungsi Search & Join Polling

Fungsi *search & join polling* oleh *user* di Android menggunakan *socket* yang di-*emit* dari *client* menuju *server*, menggunakan *parameter session key*. *Flowchart* dari fungsi *search & join polling* dapat dilihat pada Gambar 3.



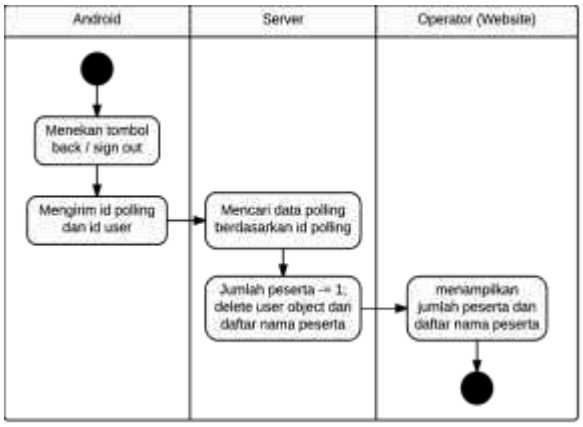
Gambar 2. *Use Case Diagram* Desain Aplikasi



Gambar 3. Fungsi Search Polling

Fungsi *Leave Room*

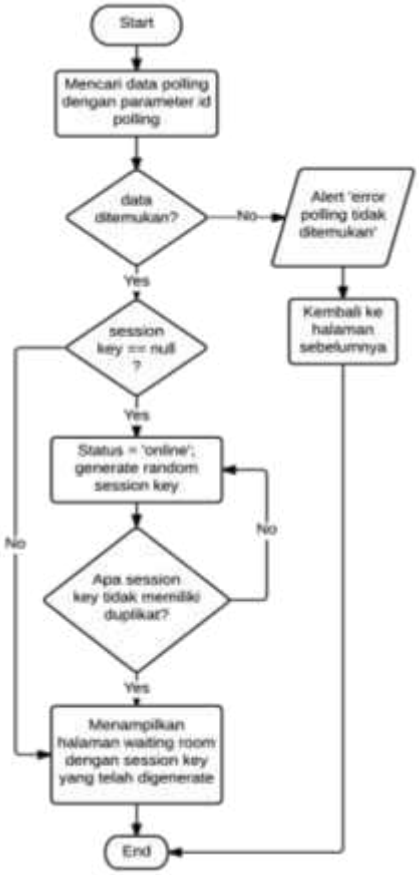
Fungsi *leave room* oleh *user* di Android dijalankan ketika *client* menekan tombol *back* ataupun *sign out* di Android. *Activity Diagram* dari fungsi *leave room* dapat dilihat pada Gambar 4.



Gambar 4. Fungsi *Leave Room*

Fungsi *Open Polling*

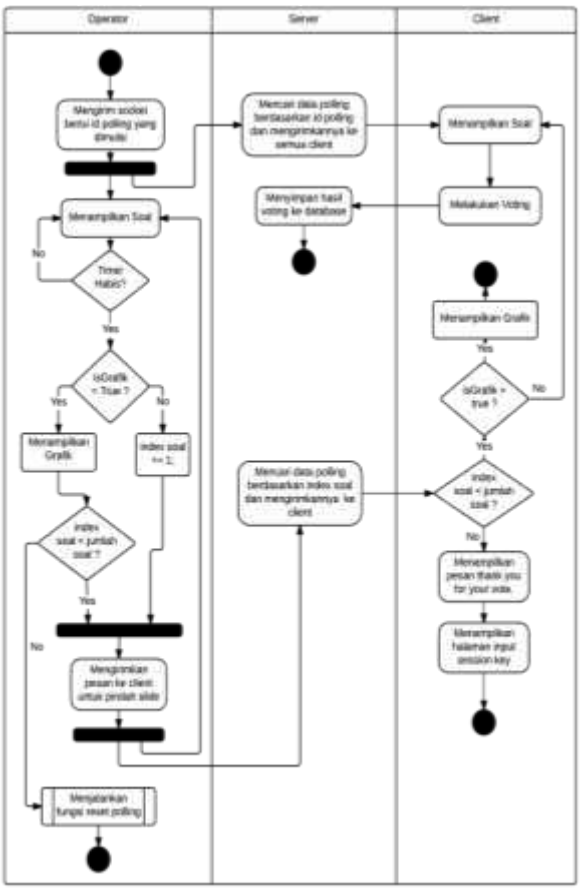
Pada Gambar 5 adalah *flowchart* yang menampilkan proses yang terjadi saat *operator* di *website* menekan tombol *open polling*. ID *polling* dikirim sebagai parameter untuk mencari *data polling* yang ada, jika ditemukan maka selanjutnya *server* akan mengecek apakah *polling* tersebut telah memiliki *session key* atau belum, jika belum maka *server* akan men-*generate session key* secara acak, untuk kemudian dicek dengan *session key* dari *polling* lain, jika ada yang sama, maka akan dilakukan *generate* ulang, setelah itu *server* akan menampilkan halaman *waiting room*, beserta data berupa *session key* yang telah di-*generate* sebelumnya.



Gambar 5. Fungsi *Open Polling*

Proses Sinkronisasi

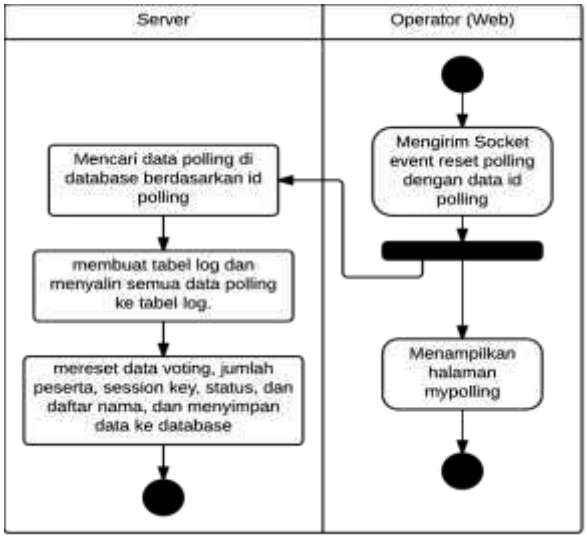
Pada Gambar 6 adalah Proses sinkronisasi dan *voting* antara *operator* dengan *client*. Proses sinkronisasi menggunakan *socket* yang di-*emit* dari *operator* di *website* menuju *server*, kemudian dilakukan *query* pada *database* untuk mencari data *polling* sesuai id *polling* dan *index* soal. Jika id *polling valid*, maka *index* soal dan data *polling* (soal dan pilihan jawaban) dikirimkan secara *broadcast* melalui *socket*, ke seluruh *client* (Android).



Gambar 6. Proses Sinkronisasi & *Voting*

Fungsi *Reset Polling*

Pada 7 adalah fungsi *reset polling*. Fungsi ini dijalankan hanya setelah *polling* selesai dan *operator* menekan tombol *finish*. Fungsi ini berperan untuk menyalin semua data *voting*, jumlah peserta, daftar nama, dari tabel *polling* ke tabel *log* baru dan me-*reset* tabel *polling*.



Gambar 7. Fungsi *Reset Polling*

PENGUJIAN & EVALUASI

Lingkungan pengujian

Pengujian dilaksanakan menggunakan jaringan *wifi speedy fiber optic* dengan kecepatan 12 *mbps*. Semua *memory* RAM dan *running apps device* peserta juga telah dibersihkan terlebih dahulu untuk menghindari banyaknya komunikasi data yang berjalan di-*system*.

Pengujian dilakukan dengan menggunakan 15 *smartphone* yang berbeda, berikut ini adalah daftar spesifikasi *smartphone* yang digunakan dalam pengujian.

Tabel 2. Tabel daftar *smartphone* peserta dan spesifikasinya

No	Nama	Versi Android	RAM	Processor
1	Polytron W2430	ICS	512 MB	Dualcore
2	Samsung galaxy A5	Lollipop	2 GB	Quadcore
3	Samsung galaxy S4 mini	Kitkat	1,5 GB	Dualcore
4	Samsung Galaxy Tab 2 (2 buah)	JB	1 GB	Dualcore
5	Oppo	Jelly bean	1 GB	Dualcore
6	Polytron	Jelly bean	512 MB	Dualcore
7	Lenovo A6000 (2 buah)	Kitkat	1 GB	Quadcore
8	Lenovo A369i	Jelly bean	512 MB	Dualcore
9	Evercoss A66A	Jelly bean	1 GB	Quadcore
10	Galaxy s3 mini	Jelly bean	512 MB	Dualcore
11	Galaxy note 10.1	Lollipop	3 GB	Octacore
12	Samsung galaxy Grand 2	Kitkat	1.5 GB	Quadcore
13	Huawei Ascend Mate 7	Lollipop	2 GB	Octa core

Sedangkan untuk spesifikasi *server (virtual machine)* yang digunakan adalah sebagai berikut:
Prosesor: Intel Xeon CPU E5504 2Ghz
Jumlah Core: 2
Memori : 2GB

Pengujian Sistem

Gambar 8 menampilkan potongan halaman *mypolling* di *website* yang digunakan untuk menampilkan hasil *polling* yang telah dibuat sebelumnya oleh *user*.

Gambar 8. Halaman *Mypolling*

Melalui halaman *mypolling*, pengguna dapat melihat *polling* yang telah dibuat dengan menekan tombol *view*, atau meng-*edit* dan menghapus *polling* dengan menekan tombol *delete*.

Gambar 9. Halaman *View Polling*

Halaman *view* pada Gambar 9 menampilkan berbagai *detail polling* yang telah dibuat mulai dari *title*, daftar soal, jenis pertanyaan, pilihan jawaban, *timer*, jenis grafik dan tombol *delete*. Baik *timer* maupun jenis grafik dapat diatur sebelumnya oleh pembuat *polling*. Minimum *timer* yang diperbolehkan adalah 5 detik, sedangkan maksimalnya 60 detik. Jenis grafik yang diperbolehkan adalah *bar, column*, dan *pie chart*.

Gambar 10. Halaman *Waiting Room Operator*

Pada Gambar 10 adalah tampilan halaman *waiting room* pada *website*. pada halaman ini ditampilkan *session key* yang harus dimasukkan oleh peserta di Android untuk mengikuti *polling* seperti yang ditampilkan pada Gambar 13.

Gambar 11. Tampilan Halaman *Operator*

Sedangkan pada Gambar 11 adalah tampilan halaman *operator* di *website* saat *polling* dimulai. Setelah *timer* habis, maka ditampilkan grafik jumlah *vote* seperti pada Gambar 12.

Gambar 12. Hasil *Voting* Pada Website

Pada aplikasi Android, halaman *input session key* akan ditampilkan setelah pengguna berhasil *login* atau *register*. Tampilan halaman *input session key* dapat dilihat pada Gambar 13.

Gambar 13. Halaman *Input Session Key*

Jika *session key* yang di-*input*-kan *user* valid, maka halaman *waiting room* akan ditampilkan seperti pada Gambar 14.

Gambar 14. Halaman *Waiting Room* Android

Gambar 15. Tampilan *Voting* di android

Gambar 15 merupakan tampilan ketika responden melakukan *vote* pada suatu *polling*.

Gambar 16. *Polling Chart*

Pada Gambar 16, *Chart* akan ditampilkan seketika setelah *timer* habis.

Evaluasi Hasil Pengujian

Berdasarkan hasil pengujian, dari 15 peserta hanya 1 peserta yang gagal dalam mengikuti *voting*, yaitu *device* no 1 pada tabel 2, dikarenakan *device* peserta yang bersangkutan mengalami *hang* atau *error,* dan dari 14 peserta semua dapat melakukan voting, tidak ada yang gagal.

Gambar 17 adalah *screenshot* tampilan *log* saat peserta melakukan *voting*. Rata-rata waktu yang dibutuhkan bagi sistem untuk mengirimkan jawaban peserta ke *server* adalah <1 detik. Berdasarkan hasil pengujian, didapati tingkat reliabilitas sistem adalah 93%, yang didapat dari jumlah peserta yang berhasil mengikuti *voting*/jumlah total peserta X 100%.

```
wawan memilih Wanita
Sat Jan 16 2016 18:23:03 GMT+0700 (WIB) Ujicoba 2
papa memilih Pria
Sat Jan 16 2016 18:23:04 GMT+0700 (WIB) Ujicoba 2
hermin memilih Pria
Sat Jan 16 2016 18:23:04 GMT+0700 (WIB) Ujicoba 2
rama memilih Pria
Sat Jan 16 2016 18:23:04 GMT+0700 (WIB) Ujicoba 2
adi memilih Wanita
Sat Jan 16 2016 18:23:04 GMT+0700 (WIB) Ujicoba 2
jonathan memilih Pria
Sat Jan 16 2016 18:23:04 GMT+0700 (WIB) Ujicoba 2
juszam memilih Wanita
Sat Jan 16 2016 18:23:05 GMT+0700 (WIB) Ujicoba 2
embun memilih Pria
Sat Jan 16 2016 18:23:05 GMT+0700 (WIB) Ujicoba 2
Vina memilih Wanita
Sat Jan 16 2016 18:23:05 GMT+0700 (WIB) Ujicoba 2
ai memilih Pria
Sat Jan 16 2016 18:23:05 GMT+0700 (WIB) Ujicoba 2
Eva memilih Pria
Sat Jan 16 2016 18:23:05 GMT+0700 (WIB) Ujicoba 2
Helena memilih Pria
Sat Jan 16 2016 18:23:05 GMT+0700 (WIB) Ujicoba 2
Diana memilih Wanita
Sat Jan 16 2016 18:23:06 GMT+0700 (WIB) Ujicoba 2
mama memilih Wanita
Sat Jan 16 2016 18:23:06 GMT+0700 (WIB) Ujicoba 2
```

Gambar 17. Screenshot Log Server

KESIMPULAN

Dari hasil pembuatan *audience response system* dan pengimplementasian Node.js pada aplikasi *GampangPoll* dapat diambil beberapa kesimpulan antara lain:

- Sistem dapat melakukan *vote* secara *real-time* dengan menggunakan Socket.IO.
- Berdasarkan hasil ujicoba, diketahui tingkat reliabilitas sistem adalah 93% bergantung pada tinggi rendahnya komunikasi data dan banyaknya *device* yang terhubung pada jaringan *wifi* yang digunakan.
- Jaringan *wifi* pribadi atau *closed network* dengan *modem router* digunakan untuk dapat memperoleh hasil terbaik.
- Tidak disarankan untuk menggunakan *wifi tethering* dari *smartphone android* dikarenakan batasan maksimum *user* yang hanya berjumlah 5 – 8.

DAFTAR PUSTAKA

[1] Android Architecture, ND, URI = http://www.tutorialspoint.com/android/android_architecture.htm

[2] Chodorow, Kristina. 2013. *MongoDB: The Definitive Guide*. Retrieved Januari 16, 2016, from: usuaris.tinet.cat/bertolin/pdfs/mongodb_the_definitive_guide.pdf

[3] Express.js Website. http://expressjs.com. Retrieved Januari 10, 2015.

[4] Green, Adams. 2013. *Single-User Twitter OAuth Programming*. Retrieved Januari 16, 2016. From 140dev.com/download/single-user-oauth.pdf

[5] Laili, N. 2009. *Audience Response Systems Sebagai Alternatif Media Pembelajaran Interaktif Di Universitas*.ND, URI = jurnal.fmipa.unila.ac.id/index.php/sains/article/download/218.pdf

[6] Materializecss Website. http://materializecss.com. Retrieved Januari 10, 2015.

[7] Myer, Thomas. 2012. *Beginning PhoneGap*. Retrieved Januari 16, 201, from: https://www.ebooks-it.net/ebook/beginning-phonegap-1

[8] Nguyen, Don. 2012. *Jump Start Node.js*. Retrieved 26 October, 2014, from http://it-ebooks.info/book/2246/

[9] Lerner, M. R. 2011. At the Forge – NodeJS. Retrieved Januari 22, 2016, from: http://www.linuxjournal.com/article/11014.

[10] Zhangling, Y., Mao, D. A Real-Time Group Communication Architecture Based on Web Socket. Retrieved Januari 22, 2016, from: http://www.ijcce.org/papers/100-T0063.pdf