

Aplikasi *Crowdsorce* pada Sistem Pencarian Tutor Berbasis *Web* dan *Android*

Sontama Baktiano Ibe¹, Andreas Handojo², Adi Wibowo³

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121-131 Surabaya 60236

Telp. (031)-2983455, Fax. (031)-8417658

E-mail: amatnos@gmail.com, handojo@petra.ac.id, adiw@petra.ac.id

ABSTRAK

Saat ini murid mengalami kesulitan saat mencari jasa pengajar pelajaran tambahan, baik dari sulitnya mencari informasi, biaya yang tidak sesuai anggaran dan jadwal yang tidak fleksibel. Penyedia jasa pengajar juga kesulitan dalam mencari murid yang mau memakai jasanya.

Karena itu pada penelitian ini dibuat sebuah sistem berupa aplikasi *web* dan *Android* yang dapat mempermudah seorang tutor dan murid untuk bertemu dan melakukan kegiatan belajar mengajar. Aplikasi ini dapat dipergunakan untuk mencari tutor yang sesuai keinginan murid dan juga menampilkan daftar murid yang cocok dengan keinginan tutor. Aplikasi ini juga dapat menentukan dan menampilkan lokasi tempat pertemuan dengan bantuan peta. Berdasarkan hasil kuesioner dapat disimpulkan bahwa sebagian besar pengguna puas terhadap keseluruhan aplikasi ini.

Kata Kunci: *Crowdsorce*, *Android*, *web*

ABSTRACT

Nowadays, students are having difficulties looking for tutoring services. The obstacles are, for example, difficulty to find the tutor, fee that does not fit to the budget, and inflexible schedule. Furthermore, tutors are having difficulties looking for students who need his or her service.

Due to these reasons, this thesis is intended to create a system with web and Android apps that can facilitate tutors and students in meeting and holding learning activities. This application can be used by a student looking for an appropriate tutor and also by a tutor listing matching students. This application can display the meeting location with the help of Google Map. Based on the questionnaire result, it can be concluded that most of the users are satisfied with the application.

Keywords: *Crowdsorce*, *Android*, *web*

1. PENDAHULUAN

Bagi seorang pelajar, tidak jarang peran tutor tambahan diperlukan untuk mendukung materi atau pelajaran yang diterima di instansi pendidikan formal. Namun, tidak mudah bagi pelajar dan tutor yang sesuai dapat bertemu. Berbagai hal menjadi kendala, seperti kurangnya sarana informasi bagi pelajar untuk mencari tutor. Begitu pula sebaliknya, belum ada sarana informasi yang murah dan cepat bagi tutor untuk mencari pelajar yang membutuhkan jasanya.

Maka karena itu diperlukan sebuah aplikasi yang mampu memberikan solusi kepada permasalahan tersebut. Aplikasi ini haruslah bisa mempermudah pelajar dan tutor untuk bertemu. Aplikasi ini juga harus dapat membantu tutor untuk menemukan calon pelajar dan juga sebaliknya. Teori *crowdsourcing* juga akan diterapkan pada aplikasi ini memudahkan pelajar untuk mencari tutor yang sesuai dengan kriteria.

Aplikasi ini terdiri dari aplikasi *web* dan aplikasi *android*. Aplikasi *web* digunakan oleh pendaftar tutor untuk melakukan pendaftaran dan digunakan oleh staff untuk Admin database, sedangkan aplikasi *Android* akan digunakan untuk proses pencarian pelajar dan tutor. Terdapat peta yang menggunakan Google Maps API pada aplikasi *Android* untuk menentukan lokasi dimana tutor dan pelajar dapat bertemu. Pelajar akan dapat mencari tutor dengan kriteria yang sesuai melalui perangkat *Android* dimulai dari jenis kelamin, harga jasa tutor yang sesuai, umur, dan lain-lain. Aplikasi *android* juga memiliki notifikasi yang akan mempermudah tutor untuk mengetahui bahwa ada pelajar yang sedang membutuhkan jasanya. Pelajar juga akan mendapatkan jasa tutor dengan harga yang lebih bersaing karena adanya sistem *bidding*. Selain itu pelajar juga dapat memberikan dan melihat rating pada tutor untuk membantu menentukan pilihannya. Aplikasi ini juga menerapkan sistem *crowdsorce* dimana kebutuhan pelajar akan diumumkan dan dapat dipenuhi oleh sekelompok masyarakat luas.

2. TINJAUAN PUSTAKA

2.1 *Crowdsorce*

Crowdsorce adalah sebuah aktivitas, proses ataupun model bisnis dimana sebuah individu, organisasi maupun perusahaan mengajukan sebuah permasalahan kepada sekumpulan masyarakat luas secara terbuka untuk dicarikan solusinya [7]. Dengan sistem *crowdsorce* ini perusahaan akan mendapatkan akses kepada tenaga kerja yang sangat besar sehingga dapat menyelesaikan sebuah permasalahan dengan biaya yang lebih sedikit dan hasil yang memuaskan. Dalam waktu kedepan akan ada banyak perusahaan yang menggunakan *crowdsorce* untuk menyelesaikan berbagai macam pekerjaan [10].

2.2 *Android*

Android adalah sebuah sistem operasi berbasis *Linux* yang dikembangkan oleh Google [1]. *Android* dikembangkan secara khusus untuk perangkat *mobile* yang memiliki perangkat *input* berupa layar sentuh. Pada awalnya, *android* dikembangkan khusus pada perangkat berbasis arsitektur *ARMv7* saja, namun setelah beberapa lama sebuah proyek bernama *Android-x86* memungkinkan *android* untuk mendukung perangkat berbasis

arsitektur x86. Android dikembangkan secara *open source* sehingga para pengembang bisa mendistribusikan dan memodifikasi ulang secara bebas.

2.3 Google Maps API

Google Maps API (*Application Programming Interface*) adalah sebuah servis yang dikembangkan oleh Google [6]. Servis ini memungkinkan pengembang aplikasi untuk menampilkan peta, gambar dari satelit, dan tampilan gambaran perspektif dari permukaan tanah dalam perangkat lunak yang sedang dikembangkan. Servis ini juga memiliki berbagai macam fungsi untuk membantu pengembangan aplikasi perangkat lunak contohnya adalah fitur routing pada peta, visualisasi data dan fitur untuk mengambil dan memberikan comment pada sebuah lokasi. API ini dapat dipakai tanpa biaya sampai dengan jumlah batasan akses tertentu yaitu 25.000 akses setiap harinya. Saat ini sudah ada lebih dari 400.000 website yang telah menggunakan Google Maps API ini [9].

2.4 AngularJS

AngularJS adalah sebuah JavaScript *framework* berbasis *open source* yang didukung oleh Google beserta komunitasnya [2]. *Framework* ini dikembangkan dengan tujuan untuk mempermudah pengembangan aplikasi berbasis *web* dengan cara menyediakan struktur baik dimulai dari desain aplikasi sampai testing [8]. AngularJS menggunakan metode MVC (*Model View Controller*) untuk membuat *source code* menjadi terstruktur, dimana dengan metode tersebut *code* akan dibagi menjadi tiga bagian sesuai fungsinya yaitu *model*, *view* dan *controller*.

2.5 Ionic

Ionic adalah sebuah *framework* untuk user interface berbasis JavaScript dan CSS yang dikembangkan secara *open source*. *Framework* ini dikembangkan dengan tujuan mempermudah developer untuk membuat user interface yang optimal untuk digunakan pada device yang memiliki input berupa layar sentuh [4]. Dengan Ionic maka developer tidak perlu membuat user interface yang sesuai dengan perangkat berbasis layar sentuh melainkan developer tinggal memakai user interface yang telah disediakan. Ionic juga menyediakan *command line tool* yang dapat dipergunakan untuk melakukan *scaffolding* aplikasi baru dan juga menyediakan konversi sebuah *project* aplikasi berbasis *web* menjadi aplikasi berbasis *mobile* dengan menggunakan Cordova.

2.6 Cordova

Cordova adalah sekumpulan API yang dapat digunakan oleh *developer* untuk mengakses fungsi *native* dari sebuah perangkat, sebagai contohnya adalah kamera dan *accelerometer* menggunakan javascript [3]. Dengan Cordova, *developer* tidak perlu mempelajari bahasa pemrograman baru yang berkaitan dengan *platform* tujuan. *Developer* cukup membuat aplikasi menggunakan JavaScript, HTML dan CSS selanjutnya aplikasi *web* biasa kemudian cukup menggunakan Cordova sehingga aplikasi tersebut dapat berjalan selanjutnya aplikasi *native* pada perangkat *mobile*. Aplikasi yang dihasilkan dari Cordova adalah sebuah aplikasi *hybrid*, karena proses rendering dilakukan pada *web view* namun juga memiliki akses pada API *native* perangkat. *Platform* yang didukung Cordova sampai saat ini adalah iOS, Android, Blackberry, Windows Phone, Palm WebOS, Bada, dan Symbian.

2.7 Firebase

Firebase adalah sebuah penyedia layanan berupa database *realtime* dan *backend* yang dapat digunakan pada berbagai *platform*. *Backend* sendiri adalah sebuah bagian dalam code aplikasi yang berhubungan langsung dengan isi *database*. Dengan Firebase, pengembang aplikasi tidak perlu membuat *backend* sendiri melainkan memakai API yang telah disediakan oleh Firebase sehingga pengembangan aplikasi dapat dipersingkat. Firebase dikembangkan dengan menggunakan database MongoDB sehingga Firebase menggunakan tipe database NoSQL. Karena memakai tipe database NoSQL maka struktur database dari Firebase bersifat fleksibel dan cepat sehingga cocok untuk digunakan pada aplikasi berbasis *mobile* [5].

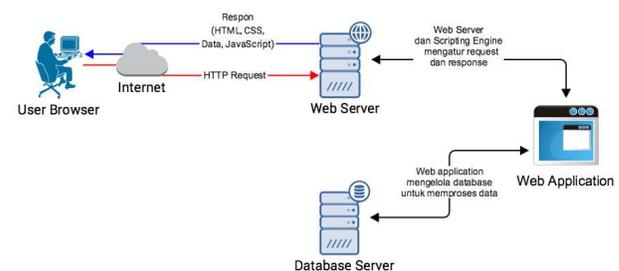
3. ANALISIS SISTEM

Dalam penelitian ini terdapat dua buah aplikasi. Aplikasi yang pertama adalah aplikasi berbasis *Android* yang digunakan oleh murid dan tutor untuk melakukan sesi *tutoring*. Aplikasi yang kedua adalah aplikasi berbasis *Web* yang digunakan oleh *Staff* dan *Admin* untuk melakukan manajemen data dan verifikasi pendaftaran tutor.

Pada aplikasi Android terdapat fitur untuk mencari tutor, mencari murid, *chat*, *report course*, dan verifikasi *course*. Seluruh fitur ini memerlukan sebuah sistem yang memiliki komponen aplikasi Android dan *Database Server*. Aplikasi Android pada sistem ini dibuat dengan menggunakan Cordova. Untuk melakukan fungsi-fungsinya aplikasi ini akan berhubungan dengan *Database Server* yang menggunakan Firebase. Untuk lebih jelasnya dapat dilihat pada Gambar 1.



Gambar 1. Sistem Aplikasi Android



Gambar 2. Sistem Aplikasi Web

Pada aplikasi *website* terdapat fitur *manage course*, *manage tutor*, dan *manage murid*. Agar semua fungsi ini dapat berjalan dibutuhkan sistem yang memiliki komponen *Web Server*, *Web Application*, dan *Database Server*. *Admin* dan *staff* akan mengakses *website* dari *browser*. *Browser* akan melakukan *request* yang diterima oleh *Web Server*. *Web Server* kemudian akan mengatur *request* dan *response* sehingga *Web Application* dapat diakses oleh *admin* dan *staff*. *Web Application* membutuhkan akses pada *Database Server* untuk menjalankan

seluruh fungsinya. Untuk lebih jelasnya dapat dilihat pada Gambar 2.

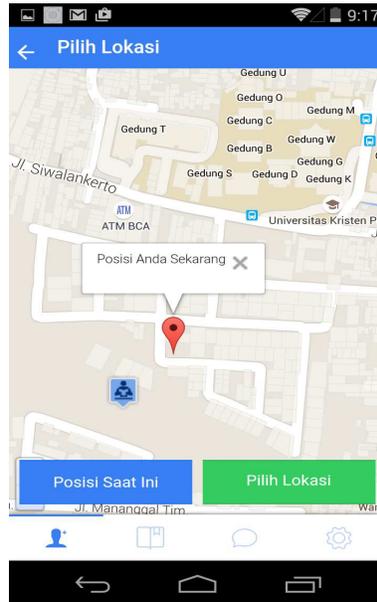
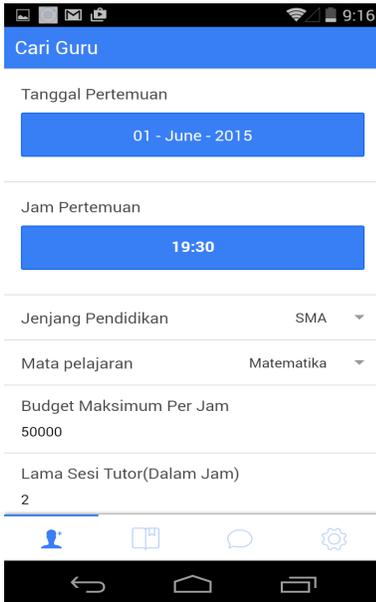
4. HASIL

Hasil aplikasi berupa aplikasi *web* dan Android. Setelah murid melakukan *login*, murid diarahkan pada halaman utama.

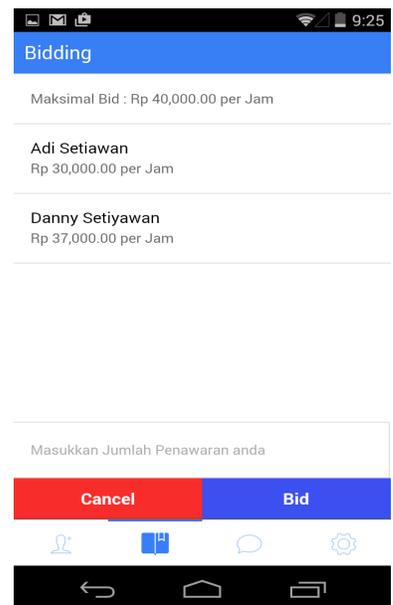
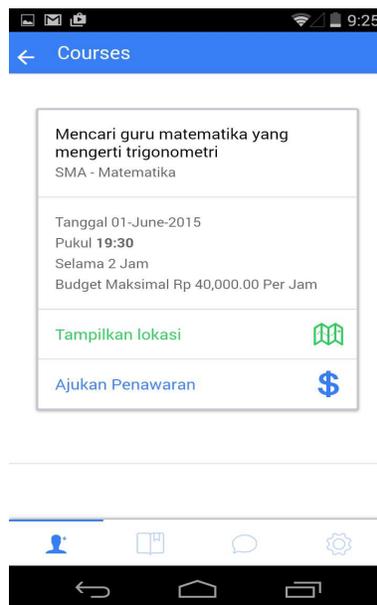
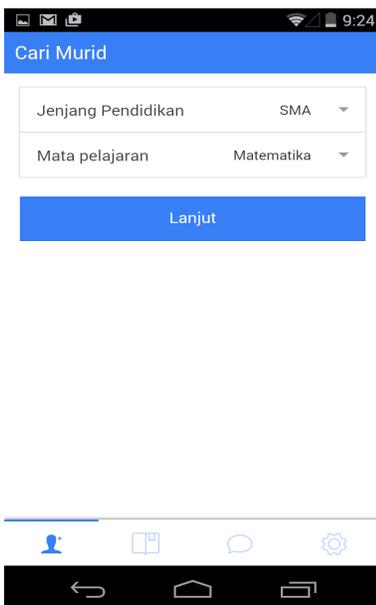
Murid dapat melakukan pencarian tutor dengan tiga langkah yaitu memasukkan informasi pencarian, memilih tempat pertemuan dan

menentukan preferensi tutor. Tahapan tersebut dapat dilihat pada Gambar 3.

Tutor dapat mencari murid dengan melalui tiga tahapan yaitu menentukan mata pelajaran, melihat daftar murid yang tersedia kemudian melakukan *bidding*. Tahapan pencarian murid dapat dilihat pada Gambar 4.

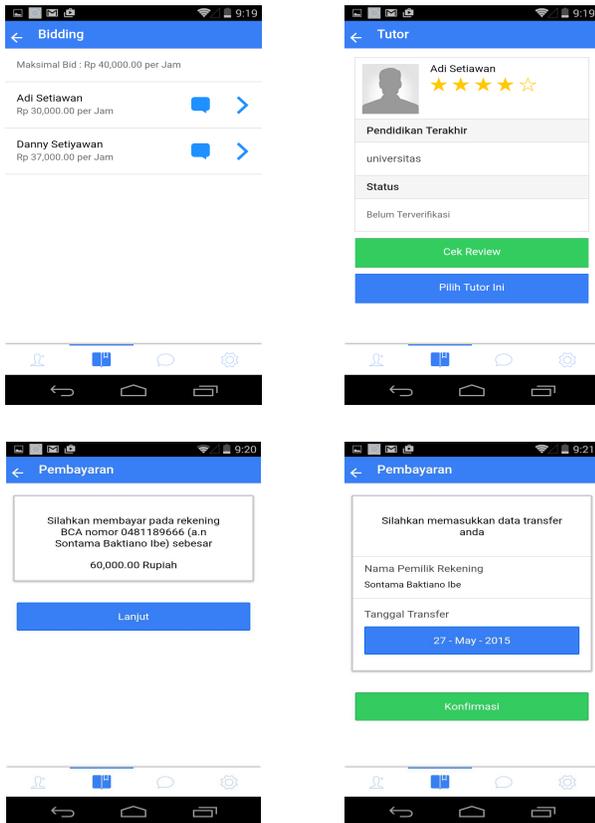


Gambar 3. Tahapan Pencarian Tutor



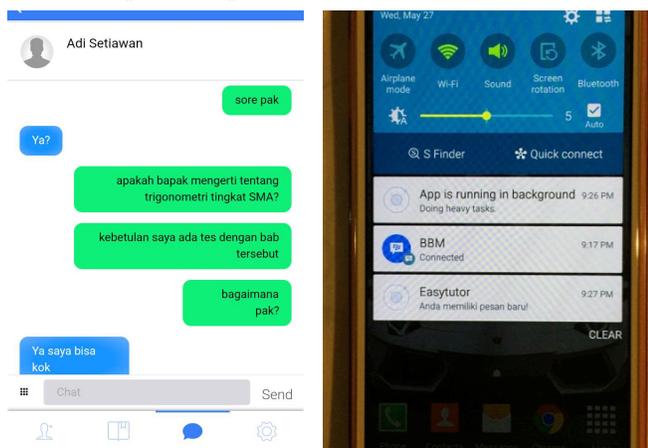
Gambar 4. Tahapan Pencarian Murid

Setelah tutor melakukan *bidding*, murid akan dapat memilih tutor dengan memilih pemenang *bid*, melakukan transfer dan melakukan konfirmasi transaksi. Tahapan tersebut dapat dilihat pada Gambar 5.



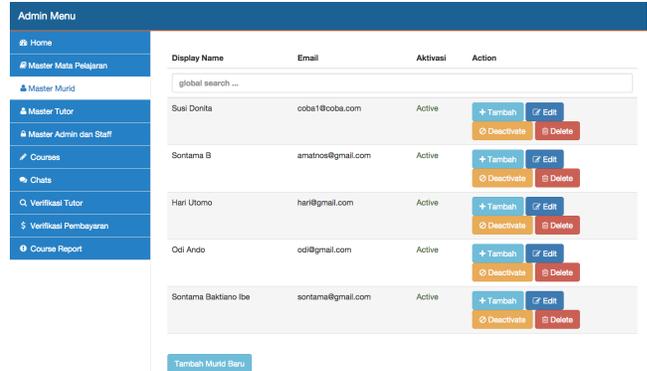
Gambar 5. Tahapan Pemilihan Tutor

Murid dan tutor dapat saling berkirim pesan dengan menggunakan *chat*. Pesan yang dikirim juga mengaktifkan notifikasi apabila aplikasi berjalan dalam posisi *background*. Tampilan chat dan notifikasi dapat dilihat pada Gambar 6.



Gambar 6. Chat dan Notifikasi

Staff dapat melakukan *maintain database* dengan menggunakan aplikasi *web*. Contoh tampilan *maintain database* dapat dilihat pada Gambar 7.



Gambar 7. Maintain Database

Penulis memberikan kuesioner pada pengguna untuk mengetahui tingkat kepuasan pengguna terhadap aplikasi. Rincian hasil kuesioner dapat dilihat pada Tabel 1.

Tabel 1. Rincian Hasil Kuesioner

Topik Pertanyaan	Tanggapan				
	Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
Pemilihan warna			2	4	4
Tampilan mudah dipahami			1	7	2
Tampilan keseluruhan				6	4
Fitur mudah digunakan			1	7	2
Navigasi mudah dipahami			1	6	3
Membantu dalam mencari jasa tutor			2	4	1
Membantu dalam mencari murid				1	1
Rata-rata			12 %	60%	28%

5. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut

- Aplikasi ini telah mampu melakukan fungsi pendaftaran, pengunggahan dokumen, pengiriman notifikasi dan menampilkan tempat tujuan bentuk peta.
- Dibutuhkan *plugin* Cordova Background untuk mencegah aplikasi masuk mode *sleep* sehingga fitur notifikasi tetap berjalan dalam mode *background*.
- Fitur *query* yang dimiliki Firebase tidak cukup untuk memenuhi kebutuhan skripsi ini sehingga masih membutuhkan sebuah *web service* untuk mengolah data dari Firebase.
- Dari hasil kuesioner dapat disimpulkan bahwa sebanyak 28% responden sangat puas terhadap aplikasi yang dibuat dan 60% menyatakan puas.

6. DAFTAR REFERENSI

- [1] Android Developers. *Android, the world's most popular mobile platform*. Retrieved Oktober 26, 2014, from: <https://developer.android.com/about/index.html>
- [2] AngularJs. *AngularJS — Superheroic JavaScript MVW Framework*. Retrieved Oktober 26, 2014, from: <https://angularjs.org/>
- [3] Apache. *About Apache Cordova*. Retrieved Oktober 28, 2014, from Apache Cordova: <http://cordova.apache.org/>
- [4] Drifty. *Ionic Documentation Overview*. Retrieved Oktober 26, 2014, from: <http://ionicframework.com/docs/overview>
- [5] Fotache, M., & Cogean, D. 2013. NoSQL and SQL Databases for Mobile Applications. *Informatica Economică* , 17, 41-58.
- [6] Google Developers. *Google Maps Web APIs*. Retrieved Oktober 26, 2014, from: <https://developers.google.com/maps/web/>
- [7] Howe, J. 2008. *Crowdsourcing: How the Power of the Crowd is Driving the Future of Business*. Great Britain: Cornerstone Publishing.
- [8] Lerner, A. 2013. *The Complete Book on AngularJS*. Iowa: Fullstack io.
- [9] Svennerberg, G. 2010. *Beginning Google Maps API 3*. Sweden: Apress.
- [10] Whitla, P. 2009. Crowdsourcing and Its Application in Marketing Activities. *Contemporary Management Research* , 5, 15-28.