

Pembuatan Aplikasi *Crowdsourcing* Untuk Jasa Rumah Tangga Berbasis Android

Kristofer Danela¹, Liliana², Henry Novianus Palit³

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121-131 Surabaya 60236

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

E-mail: kristodanela@gmail.com¹, lilian@petra.ac.id², hnpalit@petra.ac.id³

ABSTRAK

Proses pencarian jasa untuk kebutuhan rumah tangga selama ini cenderung menyulitkan karena tidak adanya media informasi yang terpusat. Proses menawarkan jasa yang dilakukan penyedia jasa juga menyulitkan karena tidak adanya media promosi yang dapat diakses oleh pencari jasa. Berdasarkan kebutuhan ini, dibuatlah aplikasi *crowdsourcing* untuk jasa pekerja kebutuhan rumah tangga yang dapat menjadi wadah informasi serta menghubungkan pencari jasa dengan penyedia jasa.

Aplikasi *crowdsourcing* untuk jasa pekerja rumah tangga ini didesain menjadi aplikasi penyedia jasa, aplikasi pencari jasa, dan satu buah *website* pendaftaran dan administrator. *Website* pendaftaran berfungsi agar pengguna dapat mendaftar sebagai penyedia jasa. *Website* juga berfungsi bagi administrator untuk menerima atau menolak pendaftaran serta mengaktifkan atau menonaktifkan akun. Aplikasi penyedia jasa dan aplikasi pencari jasa berfungsi untuk saling berkomunikasi demi mencapai kesepakatan permintaan dan penawaran jasa.

Hasil aplikasi menunjukkan bahwa aplikasi penyedia jasa dan aplikasi pencari jasa dapat menyediakan informasi yang diperlukan serta membantu pengguna berkomunikasi, bernegosiasi, dan mencapai kesepakatan. Adanya fitur *review* dan pembuatan laporan pelanggaran membantu untuk mengevaluasi setiap unsur di dalam aplikasi, baik penyedia jasa, pencari jasa, permintaan, maupun penawaran.

Kata Kunci: *Crowdsourcing*, Penyedia Jasa, Pencari Jasa, Rumah Tangga, Cari Jasa.

ABSTRACT

The search process for domestic services have tended to be difficult because of the lack of centralized information media. The process of offering services performed by service providers is also difficult because of the absence of promotional media that can be accessed by service customers. Based on these requirements, crowdsourcing application was made for the services of domestic workers that can be a source of information as well as to help connecting service customers with service providers.

Crowdsourcing application for the services of domestic workers is designed into a Service Provider Application, a Service Customer Application, and a Registration and Administrator Website. Registration Website is used to allow users to enroll as a provider. The website also serves for the administrator to accept or reject registration and to activate or deactivate an account. Service Provider Application and Service Customer Application function to communicate with each other in order to reach an agreement of supply and demand services.

The results show that the Service Provider Application and Service Customer Application can provide the necessary information and help users communicate, negotiate, and reach an agreement. Review and report violations feature helps to evaluate every element in the application, both service providers, service customers, demand, and supply.

Keywords: *Crowdsourcing, Service Provider, Service Customer, Household, Search Services.*

1. PENDAHULUAN

Pencarian dan pemesanan jasa pekerja rumah tangga sudah menjadi kebutuhan sehari-hari. Dalam melakukan berbagai aktivitas rumah tangga, diperlukan pekerja rumah tangga yang dapat dipekerjakan secara cepat dan sesuai permintaan. Pekerja rumah tangga tersebut adalah pembantu rumah tangga, *baby sitter*, perawat manula, tukang reparasi, tukang kebun, pembersih rumah, pembasmi hama, dan sopir.

Proses pencarian sopir, pembantu rumah tangga, *baby sitter* atau perawat manula sekarang ini umumnya dilakukan dengan menghubungi kerabat terdekat, tetangga atau teman yang mengenal dan dapat memberi rekomendasi. Cara ini menyulitkan pencari jasa karena informasi yang sangat terbatas dan tersebar ke banyak orang sehingga sulit didapatkan. Cara lainnya adalah dengan memasang iklan di surat kabar. Namun, cara ini memakan biaya dan tidak memberi kepastian pencari jasa dapat menemukan informasi dan jasa yang diinginkan.

Untuk mendapatkan jasa lain seperti tukang kebun, pembersih rumah, tukang reparasi dan pembasmi hama, seorang pencari jasa terlebih dahulu harus mencari informasi mengenai perbandingan kualitas dari banyak vendor atau penyedia jasa tersebut melalui internet atau keluarga dan teman. Namun, metode ini membingungkan pencari jasa karena informasi yang tidak jelas dan tidak pasti kebenarannya. Selain itu, terbatasnya vendor penyedia jasa yang resmi juga menjadi kelemahan metode ini.

Dari sudut pandang penyedia jasa rumah tangga, proses memperkenalkan diri kepada pencari jasa dilakukan dengan menyampaikan kepada teman atau kerabat yang mungkin mengenal orang yang membutuhkan jasa tersebut. Namun, sangat kecil kemungkinan untuk mendapat pencari jasa yang sesuai dengan cara ini. Berdasarkan kelemahan-kelemahan tersebut, dapat disimpulkan bahwa hingga kini belum ada wadah informasi dan perantara untuk menghubungkan antara pencari jasa dengan penyedia jasa.

Untuk menjawab kebutuhan tersebut, pada skripsi ini dibuatlah aplikasi *crowdsourcing* sebagai wadah informasi dan perantara yang dapat menyalurkan kebutuhan pencari jasa kepada banyak pencari

jasa. Aplikasi ini dapat menampilkan macam-macam jasa dengan informasi yang lengkap dan mendetail sehingga memudahkan pencari jasa memilih jasa yang diinginkan secara spesifik. Aplikasi ini juga memudahkan pencari jasa bernegosiasi dengan penyedia jasa untuk mencari kesepakatan harga dan rincian pekerjaan.

Aplikasi yang dibuat akan berbasis *mobile device* Android untuk memudahkan pencari jasa dan penyedia jasa berkomunikasi secara cepat dan di mana saja menggunakan *mobile device*. Selain itu, dibuatlah juga *website* untuk membantu pendaftaran penyedia jasa dan untuk mengatur aplikasi. *Website* dapat berfungsi bagi administrator untuk mengatur penyedia jasa pada aplikasi.

2. TINJAUAN PUSTAKA

2.1 Crowdsourcing

Definisi *Crowdsourcing* atau *crowdsourcing* secara umum adalah model yang memungkinkan sejumlah besar orang berkontribusi dalam sebuah aktivitas secara online untuk menyelesaikan masalah atau menghasilkan ide inovatif [5]. *Crowdsourcing* muncul berdasarkan paradigma bahwa diperlukan aplikasi dari kemampuan manusia untuk menyelesaikan masalah yang tidak bisa diselesaikan oleh komputer [6]. Pada masa kini *crowdsourcing* menjadi lebih mudah dilakukan karena didukung oleh informasi dan komunikasi sebagai hasil dari perkembangan teknologi [7].

Dalam aplikasinya, kegunaan *crowdsourcing* adalah untuk memberikan solusi terhadap berbagai tugas seperti pengambilan keputusan atau proses pengolahan data yang umumnya berat untuk dilakukan komputer tapi cukup mudah dikerjakan oleh manusia [8]. *Crowdsourcing* juga berguna karena memanfaatkan perkembangan teknologi sehingga mampu mengurangi biaya pelatihan internal dan meningkatkan *support* terhadap proses-proses yang dikerjakan organisasi [1]. Selain itu, *crowdsourcing* memungkinkan sebuah perusahaan untuk lebih cepat dan lebih efisien dalam memecahkan masalah dimana dalam *crowdsourcing* masa kini, internet menjadi esensial dan sangat mendukung interaksi demi mewujudkan *crowdsourcing* [7]. Proses *crowdsourcing* secara umum dijabarkan dalam siklus yang terdiri dari 4 tahap sebagai berikut:

- *Pre-selection of contributors phase*: Organisasi *crowdsourcing* menentukan kriteria spesifik untuk mengumpulkan kontributor spesifik berpotensi atau tetap membuka proyek terhadap *crowd* secara umum.
- *Accessibility of peer contributions phase*: Organisasi *crowdsourcing* perlu menentukan apakah kontributor dari *crowd* diizinkan untuk melihat, melakukan *review*, melakukan *update*, atau *delete* terhadap kontribusi lain.
- *Aggregation phase*: Organisasi *crowdsourcing* mengumpulkan sekumpulan kontribusi untuk menghasilkan solusi terbaik sesuai kebutuhan.
- *Remuneration phase*: Bila dapat diaplikasikan, organisasi *crowdsourcing* memberi kompensasi pada kontributor (*crowd*) terhadap partisipasi mereka sesuai dengan aturan yang berlaku dari kesepakatan *crowdsourcing* yang harus dirumuskan di fase awal sebelum pengerjaan tugas [4].

2.2 Service Oriented Architecture

Service Oriented Architecture (SOA) adalah desain arsitektur untuk membangun *software* komputer dalam bentuk *service-service* dimana *service* tersebut diwadahi oleh dirinya sendiri dan bersifat independen dari konteks dan keadaan dari *service* lain. SOA juga berfungsi sebagai *framework* standar untuk membuat dan mengatur *service* dengan tujuan untuk meningkatkan kemampuan IT untuk bereaksi dan beradaptasi terhadap perubahan bisnis [2].

SOA membuat perubahan menjadi lebih mudah. Sebelum adanya SOA, proses *development* mengintegrasikan *software*, *hardware*, dan *networking* yang bersifat terintegrasi secara kuat. Ini menyebabkan perubahan pada sistem menjadi lebih susah. SOA menggunakan *service-service* untuk membangun sistem yang mudah disusun dan dikonfigurasi ulang.

Secara umum beberapa pokok tujuan SOA adalah sebagai berikut:

- i. Meningkatkan fleksibilitas dari komponen *software* dan kemampuan *software* untuk digunakan ulang.
- ii. Fleksibilitas didesain sehingga mampu merespon perubahan bisnis yang terus-menerus.
- iii. Standarisasi dan integrasi dari *platform* dan sub-struktur IT.
- iv. Peningkatan terhadap interaksi antar organisasi. [4]

Beberapa hal yang perlu diperhatikan terkait faktor keamanan dalam menerapkan *Service Oriented Architecture* sebagai berikut:

- i. *Authentication*: Berguna untuk mengindikasikan bahwa *user* yang mengakses informasi memang *user* yang seharusnya bertanggung jawab atau berhak terhadap informasi tersebut.
- ii. *Confidentiality*: Menjamin akses terhadap *service* atau informasi hanya dapat dilakukan oleh subjek yang terotorisasi.
- iii. *Integrity*: Menjamin informasi yang benar dan tidak korup
- iv. *Availability*: Menjamin *service* tersedia pada waktu dibutuhkan.
- v. *Authorization*
- vi. *Auditing* [4]

2.3 REST

REST adalah sebuah arsitektur sebagai *guideline* untuk mengorganisir *service*. Arsitektur menggunakan REST adalah arsitektur *client server* dimana *client* mengirim *request* kepada *server*, kemudian *server* mengolah *request* dan mengembalikan *response*. *Request* dan *response* dibangun pada sistem *transfer resource*. *Resource* didefinisikan oleh URI [3].

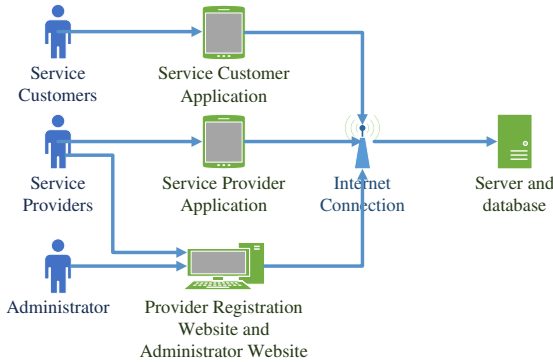
Prinsip desain REST umumnya adalah *addressability*, *statelessness* dan *interface uniform*. *Addressability* berarti REST adalah *dataset* untuk mengoperasikan *resource* yang ditandai dengan URI. *Uniform* dan *standart interface* berarti interface untuk mengakses *resource* menggunakan metode HTTP yang tetap. Setiap transaksi juga harus bersifat independen dan tidak berhubungan dengan transaksi lain dimana semua data yang dibutuhkan untuk menjalankan *request* terdapat di *request* itu sendiri.

Pendekatan *web service* dengan REST menggunakan hanya REST sebagai teknologi komunikasi untuk membangun SOA. *Service* akan didefinisikan dengan gaya dekomposisi SOA dan *web service* REST sebagai media transportasi. *Web service*

menggunakan REST sebenarnya sama seperti SOA menggunakan XML dan SOAP, kecuali REST mengukung berbagai jenis tipe data mulai objek *Javascript* sampai *binary blobs* yang digunakan dalam perintah GET dan PUT [3].

3. ANALISA DAN DESAIN SISTEM

3.1 Desain Sistem

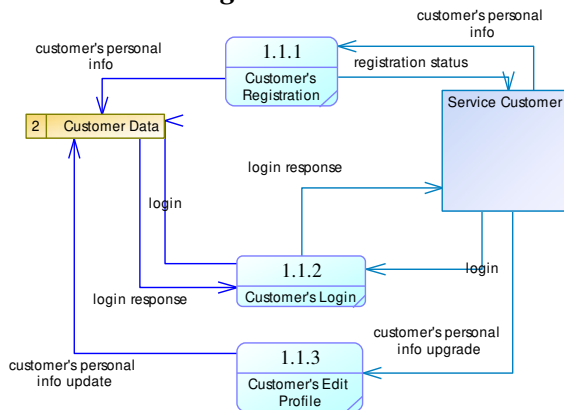


Gambar 1. Initial Structure Chart System Keseluruhan

Pada Gambar 1, ditampilkan *Initial Structure Chart* yang menggambarkan hubungan antar aplikasi dan *website* pada sistem. Pencari jasa menggunakan Aplikasi Pencari Jasa, sedangkan penyedia jasa menggunakan Aplikasi Penyedia Jasa dan *Website* Pendaftaran. Administrator menggunakan *Website* Administrator. Ketiganya harus terkoneksi internet terlebih dahulu untuk kemudian terhubung dengan *server* dan *database*.

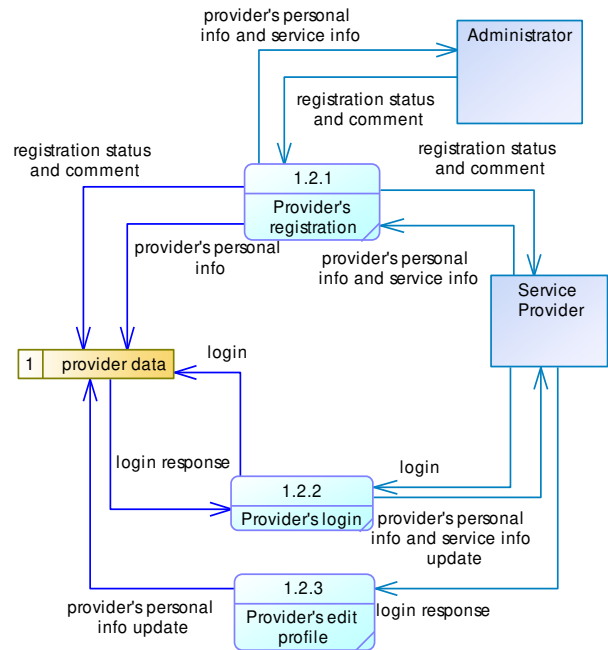
Terdapat tiga jenis pengguna dalam Sistem *Crowdsorce* Jasa Pekerja Rumah Tangga yaitu penyedia jasa (*service provider*), pencari jasa (*service customer*), dan administrator. Seorang penyedia jasa dapat mendaftarkan diri ke aplikasi, *log in*, menerima permintaan dan membuat penawaran balasan terhadap permintaan. Sedangkan, seorang pencari jasa juga dapat mendaftarkan diri menggunakan aplikasi, *log in*, membuat permintaan dan menerima penawaran balasan. Pengguna khusus selain penyedia jasa dan pencari jasa adalah administrator yang dapat menerima laporan aktivitas aplikasi, melakukan persetujuan atau penolakan terhadap pendaftaran penyedia jasa dan memasukkan data *master*.

3.2 Data Flow Diagram



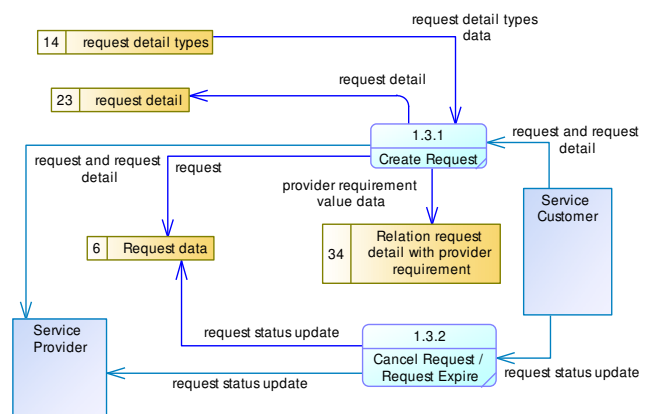
Gambar 2. DFD Proses Customer's Registration, Login & Edit Profile

Proses *Customer's Registration, Login & Edit Profile* berfungsi agar *customer* dapat melakukan registrasi, melakukan *log in* dan melakukan *edit profile* pada aplikasi. Proses ini lalu dijabarkan menjadi tiga proses seperti ditunjukkan pada Gambar 2. Ketiga proses ini dilakukan menggunakan aplikasi.



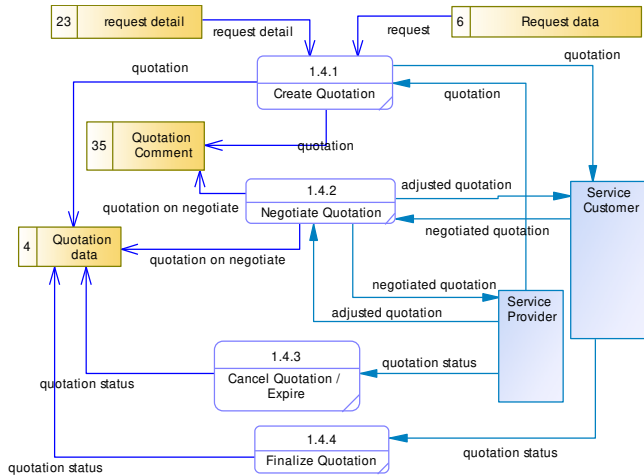
Gambar 3. DFD Proses Provider's Registration, Login & Edit Profile

Proses *Provider's Registration, Login & Edit Profile* dibagi menjadi 3 proses yaitu Proses *Provider's Registration, Provider's Login*, dan *Provider's Edit Profile* seperti ditunjukkan pada Gambar 3



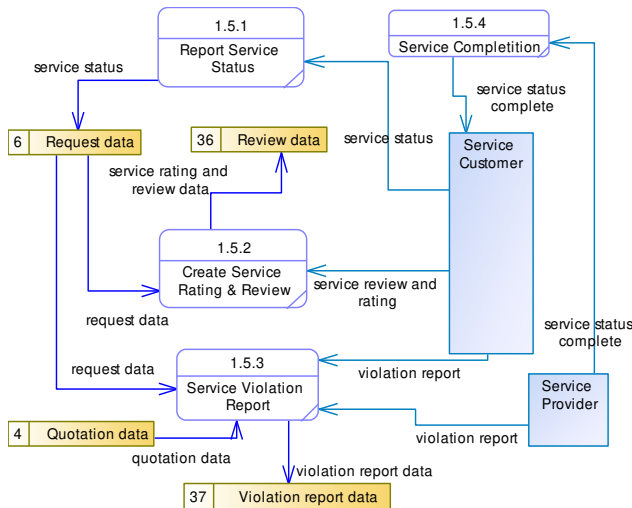
Gambar 4. DFD Proses Service Request

Pada Proses *Service Request*, pencari jasa mengirimkan permintaan kepada sistem untuk diteruskan ke *database request* dan kepada penyedia jasa dengan kategori yang sesuai dengan permintaan pencari jasa. Proses ini dijabarkan menjadi empat proses seperti ditunjukkan pada Gambar 4 yaitu Proses *Create Request, Proses Cancel Request/Request Expire, Proses Finalize Request* dan Proses *Request Completion*.



Gambar 5. DFD Proses Service Quotation

Pada Proses *Service Quotation*, sistem menerima data permintaan dari *Database Request*. Kemudian, penyedia jasa yang sesuai dengan kriteria permintaan akan membuat penawaran sebagai tanggapan terhadap permintaan untuk dikirimkan oleh sistem ke pencari jasa yang membuat permintaan tersebut. Nantinya, pencari jasa dan penyedia jasa akan saling balas berbalas penawaran. Jika sudah sepakat, maka pencari jasa akan mengirimkan *quotation status* pada sistem sebagai tanda penawaran sudah disepakati. Proses ini dijabarkan menjadi empat proses seperti ditunjukkan pada Gambar 5.



Gambar 6. DFD Proses Service Reports

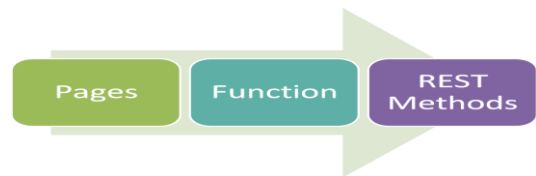
Pada proses ini, pencari jasa dapat memberikan *service review* dan *rating* yang dapat dilakukan setelah jasa selesai atau batal dilakukan. Selain itu, pencari jasa juga memberikan laporan bahwa jasa sudah selesai dilakukan dengan memberikan *service review* dan *rating*. Selain itu, pada proses ini pencari jasa dan penyedia jasa bisa memberikan *violation report* atau laporan pelanggaran. Proses ini dibagi menjadi tiga proses seperti ditunjukkan pada Gambar 6 yaitu Proses *Create Service Review & Rating*, Proses *Report Service Status*, dan Proses *Service Violation Report*.

4. IMPLEMENTASI SISTEM

Implementasi sistem yang dibuat adalah sebagai berikut:

- *Website* pendaftaran penyedia jasa dan *website* administrator menggunakan framework CodeIgniter versi 3.0.3 sehingga membutuhkan PHP minimal versi 5.4 atau lebih lanjut.
- Aplikasi penyedia jasa dan aplikasi pencari jasa dibuat menggunakan Android Studio versi 2.2. Kedua aplikasi tersebut harus dijalankan pada Android dengan minimal API level 16 atau lebih lanjut sehingga Android yang mengoperasikan aplikasi minimal harus menggunakan OS versi Jelly Bean.
- *Server* yang digunakan sebagai perantara antara aplikasi dengan *database* mengimplementasikan metode REST melalui framework *Slim 3 Application*.

4.1 Mekanisme Pemanggilan Metode REST



Gambar 7. Mekanisme Pemanggilan Metode REST

Mekanisme pemanggilan metode REST pada aplikasi ditunjukkan pada Gambar 7. Pada aplikasi, halaman-halaman yang menampilkan data seperti Halaman Profil, Halaman Membuat Permintaan, dan lain lain akan memanggil fungsi *asynchronous* yang terdapat pada aplikasi seperti Fungsi *GetCustomer()*, *GetReview()*, *PostRequest()*, dll. Parameter-parameter yang diperlukan dari sebuah fungsi juga terlebih dahulu dikirimkan ke fungsi *asynchronous* tersebut. Kemudian, fungsi tersebut akan mengakses URL yang berfungsi memanggil metode-metode REST seperti POST, PUT, GET, atau DELETE yang tersedia pada *server*. Contoh: URL (*alamat hosting*)/*search/request/agree/{prov_id}* akan memanggil metode GET untuk mendapatkan data permintaan yang sudah disetujui.

4.2 Perubahan Status Permintaan dan Penawaran

Berdasarkan Tabel 1, dan Tabel 2, perlakuan yang dapat dilakukan pada sistem baik oleh *server*, aplikasi pencari jasa, dan aplikasi penyedia jasa dalam proses pemesanan jasa adalah sebagai berikut:

- Jika pencari jasa membuat sebuah permintaan. Permintaan mendapat status 0.
- Jika penyedia jasa memberi balasan pada permintaan dengan mengajukan penawaran. Permintaan mendapat status 3. Penawaran baru mendapat status 1.
- Jika pencari jasa membalas penawaran tersebut, penawaran mendapat status 0.
- Jika pencari jasa menyepakati permintaan dan penawaran, permintaan mendapat status 5. Penawaran mendapat status 2.
- Jika tanggal waktu mulai permintaan sudah lewat, permintaan mendapat status 8.
- Jika pencari jasa melaporkan permintaan dikerjakan, permintaan mendapat status 6. Jika pencari jasa melaporkan permintaan dikerjakan, permintaan mendapat status 7.

Tabel 1. Status Permintaan

Status	Deskripsi
0	Permintaan aktif: Permintaan baru yang dibuat oleh pencari jasa dan ditunjukkan ke penyedia jasa dengan kategori tertentu.
1	Permintaan dibatalkan oleh pencari jasa.
2	Permintaan telah kadaluarsa (tidak mendapat penawaran yang difinalisasi sampai batas waktu tertentu).
3	Permintaan aktif mendapat jawaban berupa penawaran dari penyedia jasa untuk pencari jasa.
4	Permintaan aktif mendapat jawaban <i>quotation</i> dari pencari jasa untuk penyedia jasa.
5	Permintaan telah difinalisasi (satu buah permintaan terhadap penawaran ini telah difinalisasi).
6	Permintaan yang sudah difinalisasi telah selesai dikerjakan oleh penyedia jasa.
7	Permintaan yang sudah difinalisasi tidak dikerjakan atau dibatalkan.
8	Permintaan sudah melewati batas mulai pengerjaan + 1 hari sehingga pencari jasa dapat melaporkan status pengerjaan dari permintaan tersebut.

Tabel 2. Status Penawaran

Status	Deskripsi
0	Penawaran dikirimkan oleh pencari jasa untuk penyedia jasa.
1	Penawaran dikirimkan oleh penyedia jasa untuk pencari jasa.
2	Penawaran sudah disetujui oleh pencari jasa.
3	Penawaran sudah tidak berlaku atau dibatalkan.

4.3 HASIL KUESIONER

- Kecepatan aplikasi sudah menghubungkan penyedia jasa dengan pencari jasa dengan baik. Aplikasi juga dianggap cepat, serta menyediakan informasi yang memadai. Selain itu, informasi yang disediakan sudah sesuai untuk penyedia jasa.
- Pengguna sebagai penyedia jasa masih kesulitan memahami sistem aplikasi yang tergolong baru dan belum disosialisasikan.
- Kecepatan aplikasi sudah menghubungkan penyedia jasa dengan pencari jasa dengan baik. Aplikasi juga dianggap cepat, serta menyediakan informasi yang memadai. Selain itu, informasi yang disediakan sudah sesuai untuk penyedia jasa.
- Pengguna sebagai penyedia jasa masih kesulitan memahami sistem aplikasi yang tergolong baru dan belum disosialisasikan.

5. KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil pengujian dapat disimpulkan beberapa hal sebagai berikut:

- Sistem yang terdiri atas aplikasi penyedia jasa, aplikasi pencari jasa, dan *website* pendaftaran dan administrator mampu menjadi wadah informasi, menghubungkan pencari jasa dan penyedia jasa, serta mendukung proses pembuatan permintaan, penawaran, negosiasi, dan kesepakatan jasa antar pencari jasa dan penyedia jasa sehingga menjadi lebih mudah.
- Dengan adanya fitur *rating*, *review*, laporan pelanggaran, serta kontribusi administrator, maka keberlangsungan sistem dapat dijaga, mulai dari dengan memeriksa pendaftaran dan memeriksa proses permintaan dan penawaran antara pencari jasa dan penyedia jasa.
- Bagi penyedia jasa, informasi pada Aplikasi Penyedia Jasa sudah cukup lengkap dan memudahkan proses menawarkan jasa jika dibandingkan dengan proses menawarkan jasa dari mulut ke mulut. Namun, aplikasi ini masih tidak mudah dipahami penyedia jasa karena sistemnya yang masih baru.
- Bagi pencari jasa, informasi pada Aplikasi Pencari Jasa untuk memsani jasa masih kurang lengkap. Aplikasi ini dianggap mampu memudahkan pencarian jasa dan menghubungkan dengan penyedia jasa. Namun, kemudahan penggunaan aplikasi ini masih kurang dipahami pencari jasa.

5.2 Saran

Beberapa hal yang dapat dijadikan saran dalam proses pengembangan selanjutnya antara lain:

- Menemukan solusi terhadap batasan-batasan aplikasi yang belum ditemukan solusinya sebagai berikut:
 - Aplikasi tidak mampu memenuhi kebutuhan pembayaran yang diperlukan dalam proses pemesanan jasa rumah tangga.
 - Aplikasi tidak mampu memenuhi kebutuhan pencatatan dan pengaturan perpajakan sesuai dengan ketentuan hukum terhadap penyedia jasa yang terdaftar pada aplikasi tanpa lembaga resmi.
 - Aplikasi tidak mampu memenuhi kebutuhan validasi informasi yang akurat dari data pribadi penyedia jasa maupun data pribadi pencari jasa.
 - Aplikasi tidak mampu memenuhi kebutuhan pelaporan pekerjaan yang sudah disepakati namun secara insidental berhalangan dikerjakan penyedia jasa.
 - Aplikasi tidak mampu memenuhi kebutuhan pemesanan jasa secara rutin, melainkan hanya mampu memenuhi pemesanan jasa untuk sekali pakai.
- Menambahkan fitur konfirmasi pendaftaran menggunakan email sebelum pendaftaran disetujui.
- Menggunakan server dengan kapasitas lebih tinggi untuk dapat diakses aplikasi secara massal.
- Menambahkan fitur *push-notification*.
- Mensosialisasikan sistem aplikasi yang masih baru kepada pencari jasa dan penyedia jasa yang akan menggunakan aplikasi.

6. REFERENCES

- [1] Allahbakhsh, M. et al. 2013. *Quality Control in Crowdsourcing Systems*. (IEEE Internet Computing 17 (2): 76-81). URI=<http://www.infosys.tuwien.ac.at/staff/sd/papers/Zeitschriftenartikel%20-%20Quality%20Control%20SD%202013.pdf>.
- [2] Babu, D.V., Darsi, M.P. 2013. *A Survey on Service Oriented Architecture and Metrics to Measure Coupling*. (International Journal on Computer Science and Engineering Vol. 5 (08): 726-733). URI= www.enggjournals.com/ijcse/doc/IJCSE13-05-08-058.pdf.
- [3] Dahiya, N., Parmar, N. 2014. *SOA AND REST Synergistic Approach*. (International Journal of Computer Science and Information Technologies 5 (6): 7045-7049). URI = www.ijcsit.com/docs/Volume%205/vol5issue06/ijcsit2014050633.pdf.
- [4] Kamoun, F. et al. 2015. *Weaving Risk Identification Into Crowdsourcing Lifecycle*. (Procedia Computer Science 56: 41-48). URI = www.sciencedirect.com/science/article/pii/S1877050915016622.
- [5] Khalid, M. et al. 2015. *A Case of Mobile App Reviews as a Crowdsorce*. (I.J. Information Engineering and Electronic Business, 5: 39-47). URI = www.mecs-press.org/ijieeb/ijieeb-v7-n5/IJIEEB-V7-N5-6.pdf.
- [6] Marcus, A., Parameswaran, A. 2013. *Crowdsourced Data Management: Industry and Academic Perspectives*. (Foundations and Trends R in Databases 6: 1-161). URI = www.nowpublishers.com/article/Details/DBS-044.
- [7] Seltzer, S., Mahmoudi, D. 2012. *Citizen Participation, Open Innovation, and Crowdsourcing: Challenges and Opportunities for Planning*. (Journal of Planning Literature 00 (0): 1-16). URI = jpl.sagepub.com/content/early/2012/12/10/0885412212469112.
- [8] Zhao, Z. et al. 2015. *Crowd-Selection Query Processing in Crowdsourcing Databases: A Task-Driven Approach*. (International Conference on Extend: 397-408). URI = <https://openproceedings.org/2015/conf/edbt/paper-87.pdf>.