

Aplikasi “Golek Tukang” untuk Pencarian Jasa Perbaikan Rumah di Daerah Surabaya Berbasis Android

Andreas Endrahadi Wijaksono¹, Henry Novianus Palit², Anita Nathania Purbowo³

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 841765

E-mail: andreasew10@gmail.com, hnpalit@petra.ac.id, anitaforpetra@gmail.com

ABSTRAK

Pada realita kehidupan saat ini, banyak orang mempunyai smartphone yang terkoneksi dengan internet. Kondisi itu menyebabkan meningkatnya kebutuhan informasi melalui gadget. Salah satunya adalah untuk mencari jasa perbaikan rumah / tukang bangunan secara online melalui aplikasi yang ada pada smartphone. Namun di daerah Surabaya untuk mencari tukang masih menggunakan sistem manual dengan prinsip mencari informasi tukang bangunan lewat website ataupun informasi secara mulut ke mulut sehingga masyarakat Surabaya tidak dapat memenuhi kebutuhan perbaikan rumah yang sesuai.

Dengan adanya permasalahan itu maka akan dibuat aplikasi android “Golek Tukang” untuk pencarian jasa perbaikan rumah. Fitur yang disediakan di aplikasi Golek Tukang pencari jasa dapat memilih penyedia jasa yang sesuai, adanya fitur negosiasi harga untuk memberikan kenyamanan dari setiap pihak, adanya fitur chatting yang mempermudah pencari jasa dan penyedia jasa untuk berkomunikasi, dan adanya fitur maps agar penyedia jasa mudah mengetahui posisi pencari jasa berada.

Aplikasi Golek Tukang ini dibuat agar banyak masyarakat Surabaya yang ingin mencari jasa perbaikan rumah dengan mudah, cepat, dan nyaman. Dengan adanya aplikasi ini, membuat penyedia jasa mudah mendapatkan pekerjaan tanpa harus menggunakan sistem manual. Aplikasi ini dibuat dengan menggunakan bahasa pemrograman PHP dan Java, database yang digunakan adalah MySQL, dan library yang digunakan adalah Retrofit.

Kata Kunci: Android, Pencari Jasa, Penyedia Jasa, Negosiasi, Peta

ABSTRACT

In the reality of life today, many people have smartphones connected to the internet. This condition causes an increase in information needs through gadgets. One of them is to search for home improvement services / builders online through the application on the smartphone. But in the Surabaya area to find a handyman, they still use a manual system with the principle of finding information on builders through the website or by word of mouth so that the people of Surabaya do not meet the appropriate home improvement needs.

With that problem, the Android Golek Tukang application will be created to find home repair services. The feature provided in the “Golek Tukang” application is service seekers can choose the appropriate service provider, there is a price negotiation

feature to provide convenience to each party, a chat feature that makes it easier for service seekers and service providers to communicate, and the maps feature by which service providers can easily find out the location service seekers.

This Android “Golek Tukang” application enables many people of Surabaya to find home repair services easily, quickly, and comfortably. With this application, also enables service providers easily get jobs without having to use a manual system. This application is created using the programming language PHP and Java, the database used is MySQL, and the library used is Retrofit.

Keywords: Android, Service Seeker, Service Provider, Negotiation, Maps

1. PENDAHULUAN

Di zaman yang sudah menggunakan teknologi, sebagian orang mempunyai *smartphone* yang terkoneksi dengan internet. Kondisi itu menyebabkan meningkatnya kebutuhan informasi melalui *gadget*. Salah satunya adalah untuk mencari jasa perbaikan rumah / tukang bangunan secara *online* melalui aplikasi yang ada pada *smartphone*.

Tukang bangunan adalah suatu pekerjaan yang sering dicari oleh masyarakat Surabaya untuk memenuhi kebutuhan papan dalam kehidupan berumah tangga. Namun, proses bisnis yang dijalankan masih menggunakan sistem manual dengan cara mencari informasi tukang bangunan lewat *website* atau informasi dari mulut ke mulut sehingga masyarakat Surabaya susah untuk dapat memenuhi kebutuhan perbaikan rumah yang sesuai dengan masalah yang terjadi.

Untuk mendapatkan kebutuhan tukang yang sesuai dengan pengguna biasanya dihadapkan dengan permasalahan harga yang tidak sesuai dengan *budget* pengguna dan ketidakpuasan pengguna dengan hasil yang diberikan oleh tukang karena hanya sembarangan mencari tukang untuk melakukan perbaikan rumah.

Tujuan dari pembuatan aplikasi ini adalah agar masyarakat Surabaya mencari jasa perbaikan rumah dengan mudah, cepat, dan terpercaya. Dan juga aplikasi ini juga dapat digunakan oleh tukang bangunan untuk mencari pekerjaan di Surabaya dengan mudah dan cepat.

Berdasarkan uraian dari permasalahan tersebut, penulis tertarik untuk membuat sebuah aplikasi berbasis android untuk melakukan pencarian tukang bangunan secara *online*.

2. LANDASAN TEORI

2.1 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka [8].

2.2 Android Studio

Android Studio adalah sebuah *Integrated Development Environment* (IDE) yang dapat digunakan untuk pengembangan aplikasi *Android*, dan dikembangkan oleh *Google*. *Android Studio* merupakan pengembangan dari *Eclipse* IDE, dan dibuat berdasarkan *IntelliJ* IDE. *Android Studio* menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi *Android* [7].

2.3 JavaScript Object Notation(JSON)

JSON (*JavaScript Object Notation*) yang dikutip dari website *JSON* (<https://www.json.org>) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh *computer*. *JSON* merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, *Java*, *JavaScript*, *Perl*, *Python*, dan banyak lainnya. *JSON* terbuat dari dua struktur :

- Kumpulan pasangan nama / nilai. Dalam berbagai bahasa, ini diwujudkan sebagai objek, *record*, *struct*, *dictionary*, *hash table*, *list key*, atau *array* asosiatif.
- Daftar nilai yang terurut. Dalam kebanyakan bahasa, ini diwujudkan sebagai *array*, vektor, *list*, atau *sequence* [4].

2.4 Hypertext Preprocessor (PHP)

PHP adalah bahasa pemrograman *script server side* yang didesain untuk pengembangan *web*. *PHP* digunakan sebagai *script* untuk memperindah tampilan *website*, selain itu *PHP* dapat digunakan bersamaan dengan bahasa pemrograman lainnya seperti *HTML* dan *JavaScript*. *PHP* dapat digunakan dalam tiga cara utama yaitu :

- *Server-side Scripting*

PHP pada awalnya dirancang untuk membuat konten *web* yang dinamis, dan masih paling cocok untuk tugas tersebut. Untuk menghasilkan *HTML*, diperlukan *PHP parser* dan *server web* yang akan digunakan untuk mengirim dokumen berkode.

- *Command-line Scripting*

PHP dapat menjalankan *script* dari *command-line*, seperti *Perl*, *awk*, atau *Unix shell*. Biasanya menggunakan *script* baris perintah untuk tugas-tugas administrasi sistem, seperti *backup* dan *log parsing*.

- *Client-side GUI Applications*

Menggunakan *PHP-GTK*, kita dapat menulis aplikasi *cross-platform GUI* di *PHP* [2].

2.5 Representational State Transfer (REST)

REST (*Representational State Transfer*) adalah sebuah arsitektur sebagai *guideline* untuk mengorganisir *service*. Arsitektur menggunakan *REST* arsitektur *client server* dimana *client* mengirim *request* kepada *server*, setelah itu *server*

mengolah *request* dan memberikan *response*. *Request* dan *response* dibangun pada sistem *transfer resource* yang didefinisikan oleh *URI*.

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.

Pendekatan *web service* dengan *REST* hanya menggunakan *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* mendukung berbagai jenis tipe data mulai objek *Javascript* sampai *binary blobs* yang digunakan dalam perintah *GET* dan *PUT* [5].

2.6 MySQL

MySQL adalah *Relational Database Management System* (*RDBMS*) yang bersifat *open source*. *MySQL server* dapat mengelola banyak *database* pada waktu yang sama. Bahkan, banyak orang mungkin memiliki *database* berbeda yang dikelola oleh *server MySQL*. *MySQL* merupakan *database* yang paling banyak digunakan dalam pembuatan *website*. *MySQL* sangat sering digunakan dalam pembuatan *website* yang ditulis dalam *PHP* [1].

2.7 Retrofit

Retrofit merupakan *REST client library* yang aman untuk *android* dan *java*. *Retrofit* menyediakan cara yang aman untuk melakukan autentikasi dan interaksi dengan *API*. Sehingga memungkinkan pengiriman permintaan jaringan dengan *OkHttp*. *Retrofit* mengambil data *JSON* atau *XML* dari *web API* dan saat data diterima akan langsung diubah ke *Plain Old Java Object* (*POJO*). Sehingga harus ditentukan setiap *class* yang akan dipakai saat *response* diterima.

Retrofit juga bekerja sama dengan *REST API* menggunakan implementasi *java interface* yang dapat dihasilkan dengan bantuan *RestAdapter*. Implementasi dalam hal ini bertindak sebagai *local instance* dari layanan dan setiap panggilan sesuai dengan permintaan *HTTP* [6].

2.8 Push Notification

Push Notifications adalah penyampaian informasi dari aplikasi perangkat lunak ke perangkat komputasi tanpa melakukan permintaan khusus dari klien, di mana klien harus meminta informasi dari *server database* yang kemudian akan dikirimkan ke perangkat komputasi yang berisikan pesan sesuai yang diinputkan di awal., *push notification* berasal dari *server*. Keuntungan dari *push notification* dalam komputasi *mobile* adalah bahwa teknologi tidak memerlukan aplikasi spesifik pada perangkat seluler yang terbuka agar pesan diterima. Ini memungkinkan *smartphone* menerima dan menampilkan media sosial atau peringatan pesan teks bahkan saat layar perangkat dalam keadaan terkunci dan aplikasi media sosial yang mengirim pemberitahuan dalam kondisi aplikasi tertutup. *Push notification* dapat digunakan pada semua aplikasi [9].

2.9 Firebase Cloud Messaging (FCM)

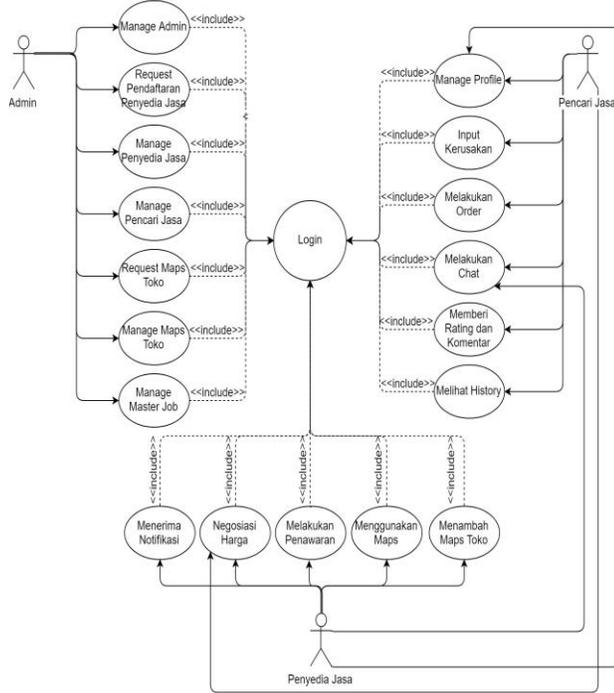
FCM adalah solusi pengiriman pesan lintas platform yang memungkinkan anda mengirim pesan tanpa biaya. Dengan menggunakan FCM, anda dapat memberi tahu aplikasi klien bahwa email baru atau data lainnya tersedia untuk disinkronkan. Anda dapat mengirim pesan notifikasi untuk mendorong interaksi kembali dan retensi pengguna. Untuk kasus penggunaan seperti pengiriman pesan instan, pesan dapat mentransfer payload hingga 4 KB ke aplikasi klien [3].

3. ANALISA DAN DESAIN SISTEM

3.1 Analisa Permasalahan

Pembuatan aplikasi “Golek Tukang” untuk pencarian jasa perbaikan rumah di daerah Surabaya berbasis Android didasari oleh permasalahan yang dihadapi oleh banyak orang yang ingin mencari tukang untuk membetulkan rumah tetapi susah mendapatkan tukang karena masih harus mencari kontak tukang melalui orang lain. Dan juga masih banyak orang yang merasa tidak puas dengan hasil kerja yang dilakukan oleh tukang dalam membetulkan rumahnya.

3.2 Use Case Diagram

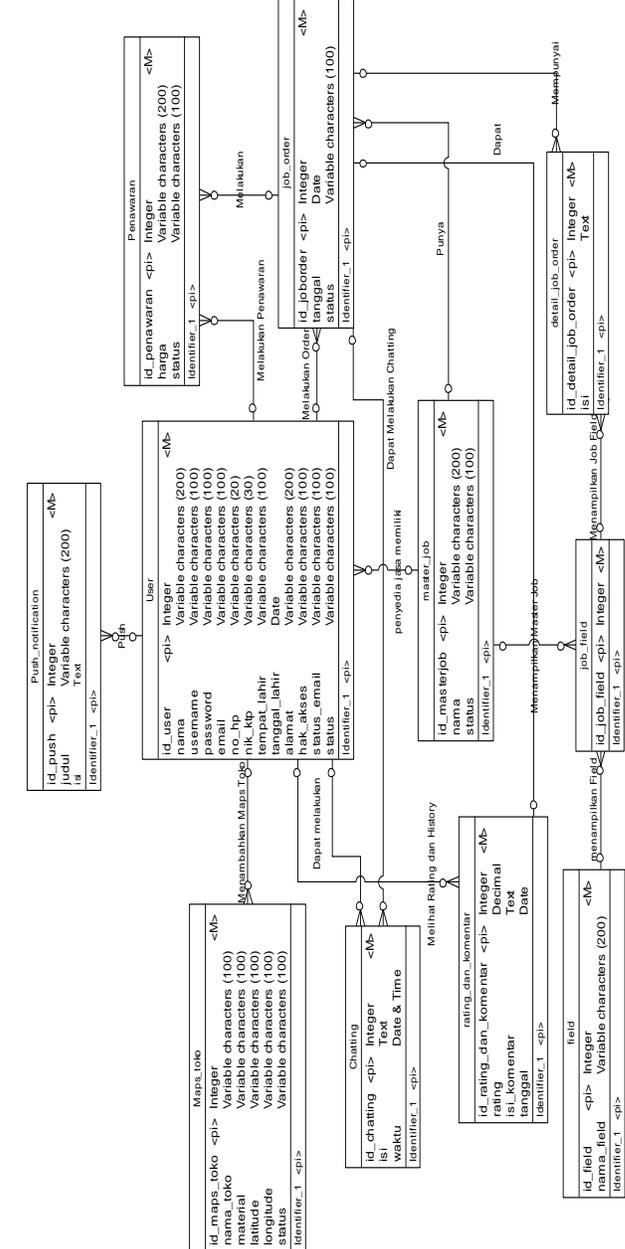


Gambar 1. Use Case Diagram Aplikasi Golek Tukang

Use Case Diagram pada Gambar 1 mendeskripsikan apa yang diperbuat aktor-aktor yang ada dalam aplikasi mobile dan website Aplikasi Golek Tukang. Dalam sistem yang dibuat terdapat 3 aktor yaitu User yang dibagi menjadi Admin, Pencari Jasa, dan Penyedia Jasa. Untuk use case admin terdiri dari manage admin, menerima request pendaftaran penyedia jasa yang nantinya akan dilakukan pengecekan data penyedia jasa yang melakukan pendaftaran dengan identitas yang tercantum di dalam KTP penyedia jasa, manage penyedia jasa (view dan delete), manage pencari jasa (view dan delete), menerima request maps toko yang telah dikirimkan oleh penyedia jasa,

manage maps toko (add, edit, dan delete), dan manage master job (add, edit, dan delete). Use case pencari jasa terdiri dari manage profile / edit profile, input kerusakan, melakukan order, negosiasi harga, melakukan chat, memberi rating dan komentar setelah order selesai, dan melihat history. Use case penyedia jasa terdiri dari menerima notifikasi disaat ada order dan chat, negosiasi harga, melakukan penawaran, menggunakan maps untuk menuju toko bangunan yang dituju, menambah maps toko.

3.3 Entity Relationship Diagram (ERD)



Gambar 2. ERD Conceptual Model

ERD pada Gambar 2 mendeskripsikan tabel apa saja yang akan digunakan pada database. Dan juga menjelaskan tipe data yang akan digunakan dari setiap data yang dibutuhkan untuk disimpan di dalam database.

4. IMPLEMENTASI SISTEM



Gambar 3. Tampilan Pendaftaran Akun

Penyedia jasa dan pencari jasa diharuskan melakukan pendaftaran akun terlebih dahulu sebelum *login* ke dalam aplikasi. Penyedia jasa dan pencari jasa harus menekan tombol daftar setelah itu akan menuju ke halaman pendaftaran. Penyedia jasa dan pencari jasa harus mengisi data sesuai *form* yang tersedia pada Gambar 3. Setelah selesai penyedia jasa dapat menekan tombol daftar.



Gambar 4. Tampilan Login Pencari Jasa dan Penyedia Jasa

Setelah pencari jasa atau penyedia jasa selesai melakukan pendaftaran maka dapat melakukan *login* ke dalam aplikasi dengan mengisi *username* dan *password* yang sesuai dengan yang didaftarkan, seperti pada Gambar 4.



Gambar 5. Tampilan Home Pencari Jasa



Gambar 6. Tampilan Home Penyedia Jasa

Jika *login* berhasil, maka pencari jasa atau penyedia jasa akan diarahkan ke halaman awal. Disaat masuk ke halaman awal maka tampilan pencari jasa dengan penyedia jasa berbeda. Tampilan halaman awal pencari jasa dapat dilihat pada Gambar 5 dan tampilan halaman awal penyedia jasa dapat dilihat pada Gambar 6.



Gambar 7. Tampilan Order Pencari Jasa

Setelah masuk ke halaman *home*, maka pencari jasa dapat melakukan *order* untuk mencari tukang yang dapat memperbaiki kerusakan bangunan yang ringan. Pencari jasa dapat menekan salah satu tombol menu sesuai dengan yang ingin diperbaiki. Setelah memilih maka akan masuk ke halaman *order* dan pencari jasa harus mengisi *form* secara lengkap seperti pada Gambar 7.

Setelah melihat detail pengajuan dari tiap penyedia jasa, maka pencari jasa dapat menentukan penyedia jasa mana yang akan dipilih. Ketika pencari jasa sudah yakin dengan penyedia jasa yang dipilih tetapi tidak yakin dengan harga yang diajukan maka pencari jasa dapat melakukan negosiasi harga kepada penyedia jasa sampai *deal*. Negosiasi harga dapat dilihat pada Gambar 8.



Gambar 8. Tampilan Negosiasi Harga

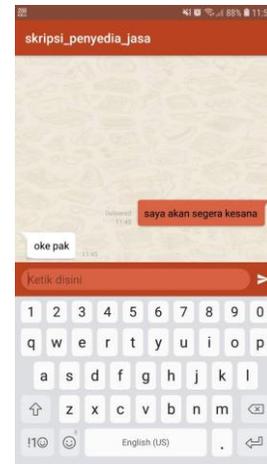
Penyedia jasa dapat melihat *detail* dari *order* yang sedang dikerjakan seperti pada Gambar 9. Di dalam *detail* tersebut penyedia jasa dapat melihat *maps* untuk menuju rumah pencari jasa dengan menekan tombol *maps* rumah pencari jasa. Tampilan *maps* rumah pencari jasa dapat dilihat pada Gambar 10.



Gambar 9. Tampilan *Detail* Proses Pengerjaan



Gambar 10. Tampilan Maps Menuju Lokasi Pencari Jasa



Gambar 11. Tampilan *Chatting*

Penyedia jasa dapat melakukan *chatting* dengan pencari jasa pada saat order sudah dalam proses pengerjaan. Adanya fitur ini memudahkan pencari jasa atau penyedia jasa untuk bertanya.

5. HASIL KUESIONER

Untuk mengetahui penilaian pengguna tentang program ini, dilakukan penelitian dan pengujian terhadap penggunaan program ini melalui kuesioner yang diberikan kepada 20 (dua puluh) pencari jasa dan 10 (sepuluh) penyedia jasa / tukang bangunan.

Dari hasil kuesioner yang telah dikumpulkan, maka detail penilaian terhadap penggunaan program oleh pencari jasa pada Tabel 1.

Tabel 1. Kuesioner Pencari Jasa

Pertanyaan	Penilaian			
	1	2	3	4
Aplikasi <i>user friendly</i>		1	12	7
User Interface mudah dipahami	1	2	12	5
Tampilan aplikasi secara keseluruhan		2	14	4
Kemudahan dalam melakukan proses pemesanan		1	14	5
Kemudahan dalam melakukan pemilihan penyedia jasa		1	9	10
Fitur negosiasi harga mudah dipahami			12	8
Apakah fitur negosiasi sangat membantu?		2	10	8
Apakah fitur chat sangat membantu?		3	5	12
Pemahaman untuk penggunaan keseluruhan aplikasi		1	15	4

* Keterangan Penilaian: 1 = kurang; 2 = sedang; 3 = baik; 4 = sangat baik.

Persentase penilaian pengguna terhadap **aplikasi yang user friendly** adalah sebagai berikut.

- Nilai 2 = $(1/20) * 100\% = 5\%$

- Nilai 3 = $(12/20) * 100\% = 60\%$
- Nilai 4 = $(7/20) * 100\% = 35\%$
- Nilai Rata-rata = $(2 * 1) + (3 * 12) + (4 * 7) / 20 = 3,3$

Persentase penilaian pengguna terhadap **user interface mudah dipahami** adalah sebagai berikut.

- Nilai 1 = $(1/20) * 100\% = 5\%$
- Nilai 2 = $(2/20) * 100\% = 10\%$
- Nilai 3 = $(12/20) * 100\% = 60\%$
- Nilai 4 = $(5/20) * 100\% = 25\%$
- Nilai Rata-rata = $(1 * 1) + (2 * 2) + (3 * 12) + (4 * 5) / 20 = 3,05$

Persentase penilaian pengguna terhadap **tampilan aplikasi secara keseluruhan** adalah sebagai berikut.

- Nilai 2 = $(2/20) * 100\% = 10\%$
- Nilai 3 = $(14/20) * 100\% = 70\%$
- Nilai 4 = $(4/20) * 100\% = 20\%$
- Nilai Rata-rata = $(2 * 2) + (3 * 14) + (4 * 4) / 20 = 3,1$

Persentase penilaian pengguna terhadap **kemudahan dalam melakukan proses pemesanan** adalah sebagai berikut.

- Nilai 2 = $(1/20) * 100\% = 5\%$
- Nilai 3 = $(14/20) * 100\% = 70\%$
- Nilai 4 = $(5/20) * 100\% = 25\%$
- Nilai Rata-rata = $(2 * 1) + (3 * 14) + (4 * 5) / 20 = 3,2$

Persentase penilaian pengguna terhadap **kemudahan dalam melakukan pemilihan penyedia jasa** adalah sebagai berikut.

- Nilai 2 = $(1/20) * 100\% = 5\%$
- Nilai 3 = $(9/20) * 100\% = 45\%$
- Nilai 4 = $(10/20) * 100\% = 50\%$
- Nilai Rata-rata = $(2 * 1) + (3 * 9) + (4 * 10) / 20 = 3,45$

Persentase penilaian pengguna terhadap **fitur negosiasi harga mudah dipahami** adalah sebagai berikut.

- Nilai 3 = $(12/20) * 100\% = 60\%$
- Nilai 4 = $(8/20) * 100\% = 40\%$
- Nilai Rata-rata = $(3 * 12) + (4 * 8) / 20 = 3,4$

Persentase penilaian pengguna terhadap **apakah fitur negosiasi sangat membantu?** adalah sebagai berikut.

- Nilai 2 = $(2/20) * 100\% = 10\%$
- Nilai 3 = $(10/20) * 100\% = 50\%$
- Nilai 4 = $(8/20) * 100\% = 40\%$
- Nilai Rata-rata = $(2 * 2) + (3 * 10) + (4 * 8) / 20 = 3,3$

Persentase penilaian pengguna terhadap **apakah fitur chat sangat membantu** adalah sebagai berikut.

- Nilai 2 = $(3/20) * 100\% = 15\%$
- Nilai 3 = $(5/20) * 100\% = 25\%$
- Nilai 4 = $(12/20) * 100\% = 60\%$
- Nilai Rata-rata = $(2 * 3) + (3 * 5) + (4 * 12) / 20 = 3,45$

Persentase penilaian pengguna terhadap **pemahaman untuk penggunaan keseluruhan aplikasi** adalah sebagai berikut.

- Nilai 2 = $(1/20) * 100\% = 5\%$
- Nilai 3 = $(15/20) * 100\% = 75\%$

- Nilai 4 = $(4/20) * 100\% = 20\%$
- Nilai Rata-rata = $(2 * 1) + (3 * 15) + (4 * 4) / 20 = 3,15$

Dari hasil kuesioner yang telah dikumpulkan, maka detail penilaian terhadap penggunaan program oleh penyedia jasa pada Tabel 2.

Tabel 2. Kuesioner Penyedia Jasa

Pertanyaan	Penilaian			
	1	2	3	4
Aplikasi <i>user friendly</i>		2	8	
Kemudahan dalam melihat memahami isi dari pemesanan		1	9	
Kemudahan dan manfaat dari fitur :				
1. Fitur negosiasi		1	8	1
2. Fitur chat		3	5	2
3. Fitur maps ke lokasi pencari jasa			9	1
4. Fitur maps ke lokasi toko bangunan			10	
Pemahaman untuk penggunaan keseluruhan aplikasi		2	6	2

* Keterangan Penilaian: 1 = kurang; 2 = sedang; 3 = baik; 4 = sangat baik.

Persentase penilaian pengguna terhadap **aplikasi yang user friendly** adalah sebagai berikut.

- Nilai 2 = $(2/10) * 100\% = 20\%$
- Nilai 3 = $(8/10) * 100\% = 80\%$
- Nilai Rata-rata = $(2 * 2) + (3 * 8) / 10 = 2,4$

Persentase penilaian pengguna terhadap **kemudahan dalam melihat memahami isi dari pemesanan** adalah sebagai berikut.

- Nilai 2 = $(1/10) * 100\% = 10\%$
- Nilai 3 = $(9/10) * 100\% = 90\%$
- Nilai Rata-rata = $(2 * 1) + (3 * 9) / 10 = 2,9$

Persentase penilaian pengguna terhadap **kemudahan dan manfaat dari fitur negosiasi** adalah sebagai berikut.

- Nilai 2 = $(1/10) * 100\% = 10\%$
- Nilai 3 = $(8/10) * 100\% = 80\%$
- Nilai 4 = $(1/10) * 100\% = 10\%$
- Nilai Rata-rata = $(2 * 1) + (3 * 8) + (4 * 1) / 10 = 3$

Persentase penilaian pengguna terhadap **kemudahan dan manfaat dari fitur chat** adalah sebagai berikut.

- Nilai 2 = $(3/10) * 100\% = 30\%$
- Nilai 3 = $(5/10) * 100\% = 50\%$
- Nilai 4 = $(2/10) * 100\% = 20\%$
- Nilai Rata-rata = $(2 * 3) + (3 * 5) + (4 * 2) / 10 = 2,9$

Persentase penilaian pengguna terhadap **kemudahan dan manfaat dari fitur maps ke lokasi pencari jasa** adalah sebagai berikut.

- Nilai 3 = $(9/10) * 100\% = 90\%$
- Nilai 4 = $(1/10) * 100\% = 10\%$
- Nilai Rata-rata = $(3 * 9) + (4 * 1) / 10 = 2,8$

Persentase penilaian pengguna terhadap **kemudahan dan manfaat dari fitur maps ke lokasi toko bangunan** adalah sebagai berikut.

- Nilai 3 = $(10/10) * 100\% = 100\%$
- Nilai Rata-rata = $(3 * 10) / 10 = 3$

Persentase penilaian pengguna terhadap **pemahaman untuk penggunaan keseluruhan aplikasi** adalah sebagai berikut.

- Nilai 2 = $(2/10) * 100\% = 20\%$
- Nilai 3 = $(6/10) * 100\% = 60\%$
- Nilai 4 = $(2/10) * 100\% = 20\%$
- Nilai Rata-rata = $(2 * 2) + (3 * 6) + (4 * 2) / 10 = 3$

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari hasil pengujian sistem yang telah dilakukan, dapat diambil beberapa kesimpulan antara lain:

- Aplikasi dapat memudahkan pencari jasa untuk mencari jasa perbaikan rumah yang pengerjaannya cukup 1 hari saja.
- Aplikasi dapat memudahkan penyedia jasa mendapatkan pekerjaan karena tidak perlu menunggu untuk dikontak secara manual oleh pencari jasa.
- Dengan adanya fitur-fitur yang disediakan maka sangat mempermudah proses *order* tanpa harus bertatap muka untuk memastikan kerusakan yang terjadi.
- Berdasarkan kuesioner yang dibagikan kepada pencari jasa, 75% pencari jasa menilai baik terhadap keseluruhan aplikasi.
- Berdasarkan kuesioner yang dibagikan kepada penyedia jasa, 60% penyedia jasa menilai baik terhadap keseluruhan aplikasi.

6.2 Saran

Saran yang dapat diberikan untuk penyempurnaan dan pengembangan program lebih lanjut antara lain:

- Aplikasi di buat versi iOS, sehingga tidak hanya pengguna android, namun pengguna iOS juga dapat menggunakan aplikasi ini.
- *User Interface* dirubah menjadi yang lebih bagus.

- Adanya pembayaran secara digital.

7. DAFTAR REFERENSI

- [1] Budi, Y.S. 2016. MySQL. Desain Aplikasi Antrian Berbasis Android. *Jurnal Infra*, 4(2): 1-7. URI=<http://publication.petra.ac.id/index.php/teknik-informatika/article/viewFile/4533/4155>.
- [2] Budi, Y.S. 2016. PHP. Desain Aplikasi Antrian Berbasis Android. *Jurnal Infra*, 4(2): 1-7. URI=<http://publication.petra.ac.id/index.php/teknik-informatika/article/viewFile/4533/4155>.
- [3] FCM. URI=<https://firebase.google.com/docs/cloud-messaging/?hl=ID>
- [4] JSON. URI=<https://www.json.org/json-id.html>
- [5] Lukito, K.D. 2017. REST. Pembuatan Aplikasi Crowdsourc untuk Jasa Rumah Tangga Berbasis Android. *Jurnal Infra*, 5(1): 1-6. Retrieved Mei 03, 2018. From <http://publication.petra.ac.id/index.php/teknik-informatika/article/viewFile/5132/4725>.
- [6] Ferryansyah, M.S. 2018. Retrofit. Analisis Performansi HTTP Networking Library pada Android. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(5): 2025-2033. URI=<http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/1454/513/>.
- [7] Juansyah, A. 2015. Android Studio. Pembangunan Aplikasi Child Tracker Berbasis Assisted – Global Positioning System (A-GPS) dengan Platform Android. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, 1(1): 1-8. URI=<http://elib.unikom.ac.id/files/disk1/673/jbptunikompp-gdl-andijuansy-33648-11-20.unik-a.pdf>.
- [8] Murtiwiayati, dan Glenn L. 2013. Android. Rancang Bangun Aplikasi Pembelajaran Budaya Indonesia untuk Anak Sekolah Dasar Berbasis Android. *Jurnal Ilmiah Komputasi*, 12(2): 1-10. URI=<http://murtiwiayati.staff.gunadarma.ac.id/Publications/files/2058/jurnal+Android.pdf>.
- [9] Rouse, M. 2016. Push Notification. URI=<https://searchmobilecomputing.techtarget.com/definition/push-notification>.