

Rancang Bangun Aplikasi *Online* Sistem Pemesanan Jasa Tukang Bangunan Berbasis Lokasi

Janiver W. Janis¹⁾, Dringhuzen J. Mamahit²⁾, Brave A. Sugiarto³⁾, Arthur M. Rumagit⁴⁾
Teknik Elektro, Universitas Sam Ratulangi Manado, Jl. Kampus Bahu-Unsrat Manado, 95115, Indonesia
E-mail: janiverjanis@gmail.com¹⁾, dringhuzen_mamahit@unsrat.ac.id²⁾, brave@unsrat.ac.id³⁾,
arthur_rumagit@unsrat.ac.id⁴⁾

Submitted: Januari 2020;

Received: Februari 2020

Publish: Maret 2020

Abstrak – Tukang Bangunan adalah pekerja yang mempunyai ketrampilan dalam bidang pembangunan. Kebutuhan masyarakat akan seseorang yang memiliki kemampuan khusus dalam masalah pembangunan masih sulit untuk dicari karena keterbatasan area atau jarak. Pada penelitian ini penulis membuat aplikasi untuk mencari tahu bagaimana seseorang dapat menemukan pekerja tukang bangunan melalui aplikasi *online*. Tujuan penelitian ini yaitu merancang aplikasi *online* sistem pemesanan jasa tukang bangunan berbasis lokasi untuk mendukung permasalahan kebutuhan masyarakat dan tukang bangunan dalam pekerjaan pembangunan. Dengan adanya aplikasi ini, masyarakat (selaku pemesan jasa) dapat dengan mudah melakukan pemesanan jasa tukang bangunan berdasarkan lokasi secara *online* (terkoneksi internet). Dalam pengembangan aplikasi digunakan metode *Rapid Application Development* yang memiliki tiga tahapan, yaitu perencanaan syarat-syarat, *workshop* desain *Rapid Application Development*, dan implementasi. Pengembangan aplikasi ini juga menggunakan *IDE Android Studio* dalam pembuatan sistem berbasis pemrograman berorientasi objek dengan bahasa pemrograman *java*. Penulis mempelajari tentang pencarian lokasi berdasarkan penelitian Pencarian Rumah Ibadah Di Kotamobagu Berbasis Android dan mengembangkan hal baru yaitu *costumer* dan *worker* dapat melakukan pendaftaran sendiri, *costumer* dapat melakukan pencarian jasa, dan *worker* dapat menawarkan jasa untuk melakukan pekerjaan. Kesimpulan yang dapat diambil dari perancangan dan pembangunan penelitian ini yaitu penulis berhasil merancang dan membangun aplikasi pemesanan jasa agar dapat digunakan oleh masyarakat selaku *costumer* sebagai pelanggan dan *worker* selaku penyedia jasa.

Kata kunci: *Android Studio; Java; Layanan; Pekerja; Pelanggan; Tukang Bangunan; Rapid Application Development.*

Abstract – Builders are workers who have skills in the field of constructions. The community needs for a person who has a specific ability in the building's problem is difficult to find because of the limited area or the distance. In this research the author makes an application to find out how a person can find a construction workers through an online application. The purpose of this research is to design an online application for the construction location-based of services ordering system to serve the problems of the community needs and the builders in their construction works. With this application, the community (as a customer of the service) can easily make an order for the services of a builder based on location by online (connected with the internet). This application development the author is using *Rapid Application Development* method which has three stages, those are planning requirements, *workshop* of *Rapid Application Development* design, and implementation. The development of this application are also using the *Android Studio IDE* in making an object-oriented programming system based on *java* programming language. The author has studied about the location searching based on the research of Finding Worship

Houses in Kotamobagu Based on Android and develops new features which are costumers and workers can register themselves, customers can search for services, and workers can offer services for works. The conclusion that can be drawn from the design and development of this research is the author successfully designed and built a service ordering application that it can be used by the community as a customer and workers as a service provider.

Keywords: *Android Studio; Builders; Costumer; Java; Rapid Application Development; Worker; Service.*

I. PENDAHULUAN

Teknologi aplikasi saat ini sangat penting sehingga dapat membantu serta mempermudah pekerjaan manusia sehari-hari yang tidak terlepas dengan internet. Sudah banyak teknologi aplikasi yang dikembangkan pada berbagai bidang begitupun dalam bidang pemesanan jasa. Tidak bisa dipungkiri sudah terdapat banyak sekali aplikasi-aplikasi yang sudah dikembangkan oleh pengembang dan telah digunakan banyak orang pada saat ini.

Pada saat ini masih banyak orang yang kesulitan dalam mencari seseorang untuk membantu pekerjaan yang tidak bisa dilakukan berdasarkan kemampuannya sendiri seperti dalam hal pembangunan rumah, gedung, perbaikan rumah/gedung dan lain sebagainya. Untuk mengatasi hal tersebut dibutuhkan keahlian dan keterampilan khusus dalam melakukan pekerjaan tersebut yaitu seorang pekerja yang memiliki jasa dalam pembangunan. Selain itu, beberapa pemilik jasa tukang bangunan juga memiliki keterbatasan area dalam melakukan pekerjaan mereka yang dikarenakan tidak adanya media yang efisien yang membantu mereka menemukan pekerjaan dengan mudah. Penerapan aplikasi *online* sistem pemesanan jasa tukang bangunan berbasis lokasi dapat mempermudah *user* selaku *costumer* sebagai pelanggan yang melakukan pemesanan dan *worker* yang selaku pekerja dalam menerima pesan hanya melalui *smartphone* yang terkoneksi dengan internet.

Aplikasi ini hampir mirip dengan aplikasi yang *trend* pada saat ini, yaitu Gojek dan Grab dimana pemesan melakukan pemesanan terhadap kendaraan yang akan ditumpangi dari lokasi pemesan sampai ditempat tujuan pemesan. Perbedaannya yaitu aplikasi ini hanya melakukan pemesanan terhadap jasa tukang bangunan dan jasa tukang bangunan itu sendiri yang akan pergi ke lokasi pemesan tersebut.

A. Penelitian Terkait

Berdasarkan artikel Teknik Elektro dan Komputer Reynold Rumimper tentang Rancang Bangun Alat Pengontrol Lampu Dengan *Bluetooth* Berbasis Android[1] penulis

mempelajari tentang perangkat android yang akan berhubungan dengan perancangan dan pengembangan aplikasi penelitian ini

Berdasarkan artikel Teknik Elektro dan Komputer dari Andri N. Lomboan tentang Rancang Aplikasi Pencarian Rumah Ibadah Di Kotamobagu Berbasis Android[2] penulis mempelajari mengenai pencarian lokasi pada perangkat android yang akan berhubungan dengan perancangan dan pengembangan aplikasi pada penelitian ini serta mengembangkan hal baru yaitu *costumer* dan *worker* dapat melakukan pendaftaran sendiri, *costumer* dapat melakukan pencarian jasa, dan *worker* dapat menawarkan jasa untuk melakukan pekerjaan.

Berdasarkan artikel Teknik Elektro dan Komputer Faisal Fatgehipon tentang Peta *Digital* Kota Manado Berbasis Android[3] penulis mempelajari mengenai *map* dan penempatan posisi kordinat yang akan berhubungan dengan pemesanan dan lokasi pada perancangan dan pengembangan penelitian ini.

Berdasarkan artikel Teknik Elektro dan Komputer dari Iwayan S. A. Mukti tentang Rancang Bangun Aplikasi Pembelajaran Untuk Anak Umur 6 – 9 Tahun Berbasis Android[4] penulis mempelajari mengenai android yang akan berhubungan dengan perancangan dan pengembangan aplikasi android pada penelitian ini.

B. Definisi Tukang Bangunan

Tukang Bangunan adalah pekerja yang mempunyai ketrampilan dalam bidang membangun rumah, membangun ruko dan bangunan yang lain. Biasanya tukang bangunan disebut juga tukang batu dalam istilah yang umum. Mengenai pengupahan kepada tukang bangunan umumnya ada yang harian dan ada juga yang bukan harian tergantung persetujuan antara tukang dan pengguna jasanya. Pada saat ini tugas dari tukang bangunan bukan hanya masalah membangun suatu gedung tapi bisa juga melakukan tugas lain seperti perbaikan, pengukiran, pengecatan dalam model bangunan yang ada. Tukang bangunan dipimpin oleh seorang yang disebut Pelaksana, yang bekerja sebagai pengarah dan pengatur pekerjaan dalam proyek. Tukang bangunan dibagi menjadi beberapa bagian, yaitu :

- 1) Tukang Batu adalah orang yang bekerja pasang batu bata, pasang pondasi batu kali dan pekerjaan cor (Untuk pekerjaan kasar)
- 2) Tukang Plaster Aci adalah tukang yang bekerja merapikan pasangan batu bata dengan campuran semen dan pasir (Untuk pekerjaan halus)
- 3) Tukang Pembesian adalah tukang yang bekerja merangkai besi di proyek, biasanya bila tukang spesialis ini tidak bisa untuk pekerjaan pasangan batu bata, plaster dan aci
- 4) Tukang Profil adalah tukang yang bekerja membuat motif pada tampak depan, biasanya tukang ini mempunyai tingkat keahlian yang lebih tinggi dibanding tukang batu
- 5) Tukang Keramik adalah yang bekerja untuk memasang keramik, tukang ini kebanyakan hanya menguasai bidang ini saja
- 6) Tukang Batu Alam adalah tukang yang bekerja untuk memasang batu alam biasanya spesialis dan tidak bisa mengerjakan pekerjaan yang lain

- 7) Tukang Marmer adalah tukang yang ahli dalam pemasangan marmer atau granit biasanya mempunyai keahlian di bidang ini saja dan tidak dimiliki tukang yang lain pada bagian dalam tukang bangunan yang lain.

C. Definisi Tukang Jasa

Jasa merupakan aktivitas ekonomi yang melibatkan sejumlah interaksi dengan konsumen atau dengan barang-barang milik, tetapi tidak menghasilkan transfer kepemilikan[5].

D. Android

Android merupakan Sistem Operasi berbasis Linux yang digunakan untuk *smartphone* dan juga tablet. Android bersifat open source dan juga Android menyediakan platform terbuka bagi para pengembang untuk menciptakan atau mengembangkan aplikasi mereka sendiri. Menurut artikel Vicky S. Gunawan tentang *Location-Based Information Berbasis QR Code Untuk Tourism*[6], Android didirikan oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White pada tahun 2003, dan pada tahun 2005 diambil alih keseluruhannya oleh google.

E. Global Positioning System

Global Positioning System merupakan sistem navigasi satelit yang dikembangkan dan pertama kali dikenalkan oleh *US DOD (United States Department of Defense)* yang merupakan Departemen Pertahanan Amerika Serikat. *GPS* adalah singkatan dari *Global Positioning System*, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi[2]. Sehingga dapat memungkinkan pengguna mengetahui letak posisi geografis baik itu dari segi lintang, bujur, dan ketinggian di atas permukaan laut dengan tepat.

Informasi dari *Global Positioning System (GPS)* akan ditransmisikan oleh beberapa satelit (tiga satelit misalnya) sehingga *GPS receiver* mampu mengkalkulasi dan menampilkan seakurat mungkin posisi, kecepatan dan informasi waktu kepada pengguna *GPS*. Keistimewaan yang dimiliki *Global Positioning System (GPS)* adalah mampu bekerja dalam berbagai kondisi cuaca, siang atau malam. Keakuratan sebuah perangkat *Global Positioning System* bisa mencapai 15 meter, bahkan model terbaru yang dilengkapi teknologi *Wide Area Augmentation System (WAAS)* keakuratannya sampai 3 meter.

Aplikasi *GPS* dapat dibagi kepada 5 kategori, yaitu aplikasi untuk di darat (*land applications*), aplikasi untuk di laut (*marine applications*), aplikasi untuk militer (*military applications*), aplikasi untuk di udara (*air applications*), aplikasi ringan (*recreation application*).

F. Unified Modeling Language (UML)

UML merupakan singkatan dari *Unified Modeling Language* yang merupakan sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. *UML* juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. Aplikasi atau sistem yang tidak terdokumentasi

biasanya dapat menghambat pengembangan karena *developer* harus melakukan penelusuran dan mempelajari kode program. *UML* juga dapat menjadi alat bantu untuk *transfer* ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainnya.

UML beberapa jenis diagram, antara lain :

TABEL I USE CASE DIAGRAM

NO	GAMBAR	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i>
2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>)
3		Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
4		Include	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit
5		Extend	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan
6		Association	Apa yang menghubungkan antara objek satu dengan objek yang lainnya
7		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas
8		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi)
10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

1) Use Case Diagram

Use case diagram merupakan gambaran *graphical* dari beberapa atau semua aktor, *use case*, dan interaksi yang memperkenalkan suatu sistem. Berdasarkan artikel dari Cristian O. Karundeng tentangancang Bangun Aplikasi Pengenalan Satwa Langka di Indonesia Menggunakan Augmented Reality[7] *Use case* secara sederhana sesungguhnya merupakan sebuah sarana bantu untuk mendefinisikan apa yang ada di luar sistem (aktor) dan apa yang harus dilakukan oleh sistem yang sedang dikembangkan. Simbol-simbol *use case* diagram dapat dilihat pada tabel 1 *use case diagram*.

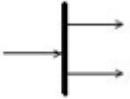
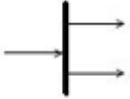
2) Activity Diagram

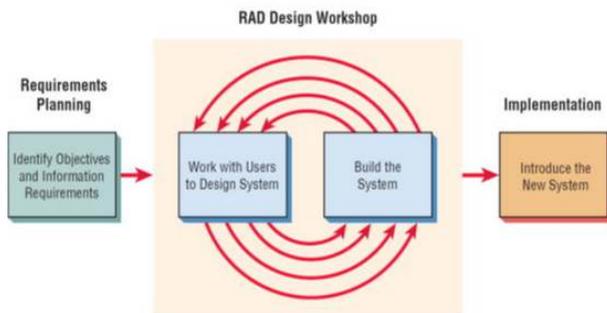
Activity diagram merupakan diagram ini menggambarkan tentang aktifitas yang terjadi pada sistem. Dari pertama sampai akhir, diagram ini menunjukkan langkah – langkah dalam proses kerja sistem yang kita buat. Simbol-simbol *activity diagram* dapat dilihat pada tabel 2 *activity diagram*.

3) Sequence Diagram

Sequence diagram merupakan diagram yang menjelaskan interaksi objek yang berdasarkan urutan waktu. Sequence diagram dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case* diagram.

TABEL II ACTIVITY DIAGRAM

NO	SIMBOL	KETERANGAN
1		Start Point
2		End Point
3		Activities
4		Fork (Percabangan)
5		Join (Penggabungan)
6		Decision
7	Swimlane	Sebuah cara untuk mengelompokkan <i>activity</i> berdasarkan actor (mengelompokkan <i>activity</i> dalam sebuah urutan yang sama)



Gambar 1 RAD Design Workshop

4) Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

G. Model Rapid Application Development (RAD)

RAD adalah sebuah metode pengembangan *software* yang digunakan untuk menekan waktu yang dibutuhkan dalam mendesain serta mengimplementasikan sistem informasi sehingga menghasilkan siklus pengembangan yang sangat pendek. RAD meliputi fase-fase seperti berikut :

1) Requirements Planning (Perencanaan Syarat-Syarat)

Dalam fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem dan mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah perusahaan. Sehingga fokusnya akan selalu tetap pada upaya pencapaian tujuan-tujuan perusahaan. Dalam fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem dan mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah perusahaan. Sehingga fokusnya akan selalu tetap pada upaya pencapaian tujuan-tujuan perusahaan.

2) RAD Design Workshop (Workshop Desain RAD)

Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Penganalisis dan pemrogram dapat bekerja membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna. *Workshop* desain ini dapat dilakukan selama beberapa hari tergantung dari ukuran aplikasi yang akan dikembangkan. Dapat dilihat pada gambar 1 RAD design workshop

3) Implementation (Implementasi)

Pada fase implementasi, penganalisis bekerja dengan para pengguna secara intens selama *workshop* dan merancang

aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui dan sistem-sistem dibangun dan disaring, sistem-sistem baru atau bagian dari sistem diujicoba dan kemudian diperkenalkan kepada organisasi[8].

H. Android Studio

Android Studio merupakan aplikasi yang digunakan untuk membuat aplikasi pada perangkat *smartphone* android dengan jenis ekstensi APK dan XAPK. Secara spesifik Android Studio adalah sebuah IDE untuk *Android Development* yang diperkenalkan pihak Google pada acara Google I/O tahun 2013. Menurut artikel dari Astrid A. A. Makiolor tentangancang Bangun Pencarian Rumah Sakit, Puskesmas dan Dokter Praktek Terdekat di Wilayah Manado Berbasis Android[9], menulis bahwa Android Studio merupakan pengembangan dari Eclipse IDE dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA[9]. Android Studio merupakan IDE resmi untuk pengembangan aplikasi android.

I. Firebase

Firebase adalah *BaaS (Backend as a Service)* yang saat ini dimiliki oleh Google. Firebase ini merupakan solusi yang ditawarkan oleh Google untuk mempermudah pekerjaan *Mobile Apps Developer*. Dengan adanya Firebase, *apps developer* bisa fokus mengembangkan aplikasi tanpa harus memberikan *effort* yang besar untuk urusan *backend*. Menurut artikel dari Kelvin Wong tentang Rancang Bangun Aplikasi ‘Trip Alone?’ Berbasis Android[10], Firebase berdiri sejak April 2012, dan berbasis di San Francisco, Amerika Serikat. Firebase memberikan API kepadadeveloper untuk mengirim dan mensinkronisasi data di setiap *client* yang terhubung ke Firebase. Kemudian berkembang menjadi layanan penyedia pengembangan aplikasi. Pada Oktober 2014, perusahaan tersebut diakuisi oleh Google. Berbagai fitur terus dikembangkan hingga diperkenalkan pada Mei 2016 di Google I/O. Beberapa fitur yang dimiliki oleh Firebase adalah sebagai berikut :

1) Firebase Analytics

Firebase *Analytics* merupakan solusi pengukuran aplikasi yang memberikan informasi mengenai penggunaan aplikasi dan interaksi pengguna.

2) Firebase Cloud Messaging

FCM (Firebase *Cloud Messaging*) merupakan solusi pengiriman pesan lintas *platform*. FCM 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.

3) Firebase Authentication

Dengan mengetahui identitas pengguna, aplikasi dapat menyimpan data pengguna secara aman di *cloud* dan memberikan pengalaman personal yang sama di setiap perangkat pengguna. Firebase *Authentication* menyediakan layanan *backend*, SDK yang mudah digunakan, dan *library* UI yang siap pakai untuk mengautentikasi pengguna ke aplikasi. Firebase *Authentication* mendukung autentikasi menggunakan sandi, nomor telepon, penyedia identitas gabungan yang populer, seperti Google, Facebook, dan Twitter, dan lain-lain.

4) *Firestore Remote Config*

Firestore Remote Config adalah layanan *cloud* yang dapat digunakan untuk mengubah perilaku dan tampilan aplikasi tanpa mengharuskan pengguna mengunduh *update* aplikasi.

5) *Firestore Real Time Database*

Firestore Realtime Database adalah database yang di *host* pada *cloud*. Data disimpan sebagai JSON dan disinkronkan secara *realtime* ke setiap klien yang terhubung. Ketika Anda membuat aplikasi lintas-*platform* dengan SDK Android, iOS, dan JavaScript, semua klien akan berbagi sebuah *instance Realtime Database* dan menerima *update* data terbaru secara otomatis.

6) *Firestore Crashlytics*

Firestore Crashlytics adalah solusi pelaporan *error* yang ringan dan bekerja secara *realtime* untuk membantu Anda melacak, memprioritaskan, serta memperbaiki masalah stabilitas yang mengurangi kualitas aplikasi. *Crashlytics* menghemat waktu pemecahan masalah dengan mengelompokkan *error* secara cerdas dan menunjukkan keadaan yang menyebabkannya.

J. *Java Programming*

Bahasa pemrograman Java merupakan Bahasa pemrograman yang populer dan sering digunakan banyak *programmer* dalam mengembangkan aplikasi khususnya pada aplikasi. Terdapat sebuah perusahaan bernama Sun *Microsystems* yang berambisi untuk membuat *platform* universal yang dapat mengintegrasikan berbagai mesin. Hal tersebut merupakan proyek rahasia yang diberi nama *Green Project*. Pengembangan proyek tersebut melibatkan Patrick Naughton, Mike Sheridan, dan James Gosling serta dibantu oleh 13 orang staf yang bekerja secara tertutup dan mengasingkan diri pada sebuah gedung di Sand Hill Road, Menlo Park, California, AS.

II. METODE

A. *Waktu dan Tempat Penelitian*

Waktu penelitian dimulai pada bulan Agustus 2018 berlokasi di Lab. Teknologi Basis Data, Fakultas Teknik Universitas Sam Ratulangi, Manado.

B. *Alat dan Bahan*

Alat dan bahan yang digunakan dalam perancangan dan pengembangan penelitian ini yaitu perangkat keras Laptop Acer Aspire 3 dengan spesifikasi Processor AMD Ryzen 5 2.00 GHz, Operating System Windows 10 64bit, Memori RAM 8 GB DDR4, Hardisk 1000 GB. Perangkat lunak yang digunakan adalah Microsoft Office 2016, Adobe Illustrator, Adobe Photoshop, Android Studio, Java programming, *Firestore Database*.

C. *Teknik Pengumpulan Data*

Metode pengambilan data yang dilakukan peneliti yaitu berdasarkan data yang dibagi atas :

1) *Studi Literatur*

Studi literatur digunakan dengan mengumpulkan referensi terkait permasalahan yang diteliti.

2) *Studi Pustaka*

Metode studi pustaka adalah memperoleh data-data atau informasi dari dokumen yang di tulis dari seseorang yang telah mengalami peristiwa (*primer*) ataupun peristiwa yang di laporkan dari orang lain selanjutnya di tulis oleh orang ini (*sekunder*).

D. *Wawancara*

Penggunaan metode wawancara untuk bisa memperoleh informasi dari narasumber yang dianggap dapat memberikan informasi mengenai jasa tukang bangunan.

III. HASIL DAN PEMBAHASAN

Pada Aplikasi *Online* Sistem Pemesanan Jasa Tukang Bangunan Berbasis Lokasi ini terdapat *costumer* selaku pengguna yang akan menggunakan aplikasi untuk memesan dan *worker* selaku pekerja yang akan melakukan pekerjaan tukang bangunan berdasarkan pemesanan yang dilakukan oleh *costumer*. Aplikasi ini dapat digunakan masyarakat untuk melakukan pemesanan jasa tukang bangunan berdasarkan daftar pekerjaan yang terdapat pada aplikasi.

A. *Implementasi Model RAD*

Berdasarkan artikel David Lumingkewas tentang Peta Aset Universitas Sam Ratulangi Berbasis Web[11], *Rapid Application Development (RAD)* adalah proses model perangkat lunak *incremental* yang menekankan siklus pengembangan yang singkat. Dalam proses perancangan aplikasi pemesanan jasa ini dibuat berdasarkan model *RAD (Rapid Application Development)* dengan fase-fase berikut :

1) *Perencanaan Syarat-Syarat*

Pengumpulan informasi dan data-data penelitian dilakukan melalui studi referensi. Dalam hal ini, penulis melakukan pembelajaran dari berbagai media referensi mengenai android studio, bahasa pemrograman java, aplikasi-aplikasi terkait pemesanan, aplikasi-aplikasi terkait yang menggunakan lokasi dalam perancangan dan pembangunan awal aplikasi *online* sistem pengembangan jasa tukang bangunan berbasis lokasi ini. dalam penelitian ini.

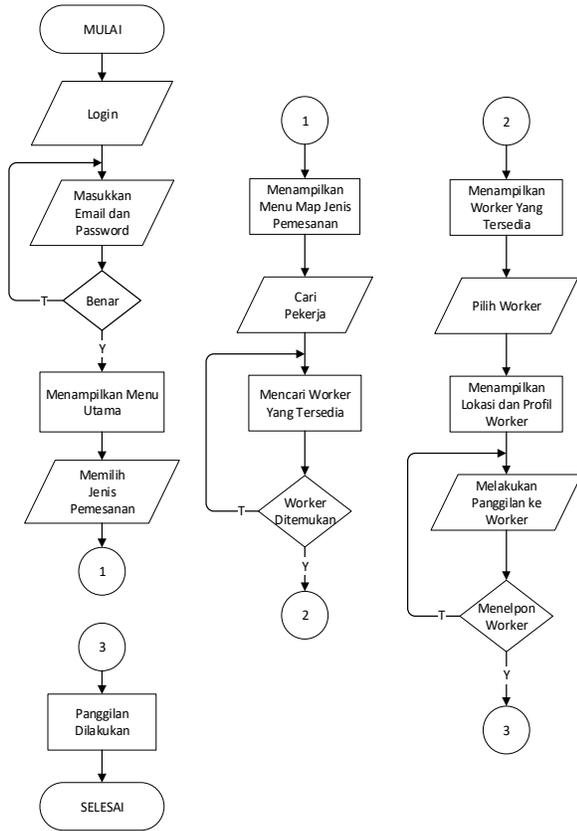
Adapun data tukang (*worker*) dan data user (*costumer*) yang dikumpulkan dari beberapa sumber yang dapat membantu dalam perancangan dan pembangunan awal pada penelitian ini dapat berupa ID (nomor KTP), nama, nomor telepon, alamat dan email.

2) *Workshop Model Rapid Application Development (RAD)*

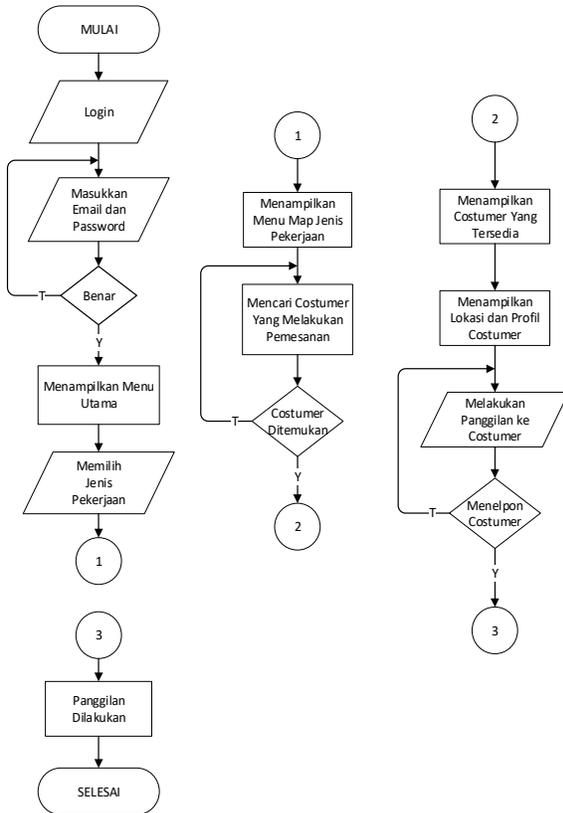
Pada tahap ini, pengembang akan merancang sistem dari aplikasi yang akan dibangun terlebih dahulu.

a. *Flowchart*

Flowchart merupakan suatu *diagram* alir yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam suatu sistem. *Flowchart costumer* dan *worker* untuk melakukan dan menerima pemesanan dapat dilihat pada gambar 2 *flowchart costumer* melakukan pemesanan dan pada gambar 3 *flowchart worker* mendapatkan pemesanan.



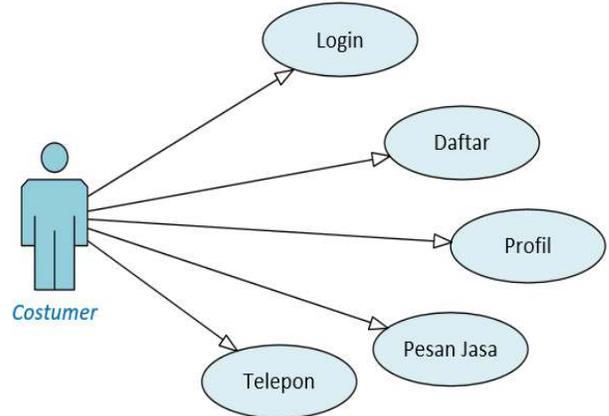
Gambar 2 flowchart costumer melakukan pemesanan



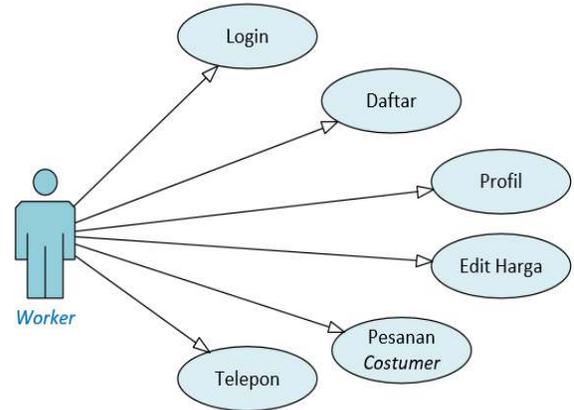
Gambar 3 flowchart worker mendapatkan pesanan

b. Use Case Diagram

Use case diagram costumer dan worker dapat dilihat pada Gambar 4 Use Case Diagram Costumer untuk pemesan dan Gambar 5 Use Case Diagram Worker untuk pekerja. Untuk memperjelas use case diagram pada gambar 4 use case diagram costumer dan gambar 5 use case diagram worker diatas dapat dilihat pada tabel III-VII untuk costumer dan tabel VIII-XIII untuk worker berikut.



Gambar 4 Use Case Diagram Costumer



Gambar 5 Use Case Diagram Worker

TABEL III DESKRIPSI USE CASE DIAGRAM LOGIN PADA COSTUMER

Nama use case	Login	
Aktor	Costumer dan Sistem	
Deskripsi	Costumer yang telah melakukan pendaftaran akun dapat masuk ke menu utama melalui proses login	
Normal course	Costumer	Sistem
	1. Memasukkan data email dan password.	1. Menampilkan form login
2. Menekan tombol MASUK pada tampilan aplikasi	2.a. Berpindah ke menu utama saat data benar	
	2.b. Menampilkan pesan peringatan apabila data yang dimasukkan salah	
Alternatif course		
Pre-Cond.	Menampilkan form login	
Post-Cond.	Memasukkan email dan password	
Assumption	-	

TABEL IV DESKRIPSI USE CASE DIAGRAM DAFTAR PADA COSTUMER

Nama use case	Daftar								
Aktor	Costumer dan Sistem								
Deskripsi	Costumer harus melakukan pendaftaran akun agar dapat masuk ke menu utama melalui proses login								
<i>Normal course</i>	<table border="1"> <thead> <tr> <th>Costumer</th> <th>Sistem</th> </tr> </thead> <tbody> <tr> <td>1. Memasukkan data yang diperlukan.</td> <td>1. Menampilkan form pendaftaran</td> </tr> <tr> <td>2. Menekan tombol DAFTAR pada tampilan aplikasi</td> <td>2.a. Kembali ke menu login ketika selesai melakukan pendaftaran</td> </tr> <tr> <td></td> <td>2.b. Menampilkan pesan peringatan apabila terjadi kesalahan dalam memasukkan data pengguna</td> </tr> </tbody> </table>	Costumer	Sistem	1. Memasukkan data yang diperlukan.	1. Menampilkan form pendaftaran	2. Menekan tombol DAFTAR pada tampilan aplikasi	2.a. Kembali ke menu login ketika selesai melakukan pendaftaran		2.b. Menampilkan pesan peringatan apabila terjadi kesalahan dalam memasukkan data pengguna
	Costumer	Sistem							
1. Memasukkan data yang diperlukan.	1. Menampilkan form pendaftaran								
2. Menekan tombol DAFTAR pada tampilan aplikasi	2.a. Kembali ke menu login ketika selesai melakukan pendaftaran								
	2.b. Menampilkan pesan peringatan apabila terjadi kesalahan dalam memasukkan data pengguna								
<i>Alternatif course</i>									
<i>Pre-Cond.</i>	Menampilkan form pendaftaran								
<i>Post-Cond.</i>	Memasukkan data pengguna yang diperlukan								
<i>Assumption</i>	-								

TABEL V DESKRIPSI USE CASE DIAGRAM PROFIL PADA COSTUMER

Nama use case	Profil						
Aktor	Costumer dan Sistem						
Deskripsi	Costumer dapat melihat dan mengubah data profil yang dimiliki						
<i>Normal course</i>	<table border="1"> <thead> <tr> <th>Costumer</th> <th>Sistem</th> </tr> </thead> <tbody> <tr> <td>1. Melihat data profil.</td> <td>1. Menampilkan profil costumer</td> </tr> <tr> <td>2. Mengubah data profil</td> <td>2. Menampilkan jendela ubah profil</td> </tr> </tbody> </table>	Costumer	Sistem	1. Melihat data profil.	1. Menampilkan profil costumer	2. Mengubah data profil	2. Menampilkan jendela ubah profil
	Costumer	Sistem					
1. Melihat data profil.	1. Menampilkan profil costumer						
2. Mengubah data profil	2. Menampilkan jendela ubah profil						
<i>Alternatif course</i>							
<i>Pre-Cond.</i>	Menampilkan profil dari costumer						
<i>Post-Cond.</i>	Memasukkan data profil						
<i>Assumption</i>	-						

TABEL VI DESKRIPSI USE CASE DIAGRAM PESAN JASA PADA COSTUMER

Nama use case	Pesan Jasa								
Aktor	Costumer dan Sistem								
Deskripsi	Costumer melakukan pemesanan jasa melalui menu pemesanan yang sudah disediakan								
<i>Normal course</i>	<table border="1"> <thead> <tr> <th>Costumer</th> <th>Sistem</th> </tr> </thead> <tbody> <tr> <td>1. Memilih jasa tukang bangunan yang disediakan aplikasi</td> <td>1. Menampilkan list menu dari jasa tukang bangunan</td> </tr> <tr> <td>2. Menekan tombol Cari Tukang untuk melakukan pemesanan</td> <td>2.a. Mengirim notifikasi pemesanan ke aplikasi worker</td> </tr> <tr> <td></td> <td>2.b. Menampilkan pesan pebritahuan pemesanan</td> </tr> </tbody> </table>	Costumer	Sistem	1. Memilih jasa tukang bangunan yang disediakan aplikasi	1. Menampilkan list menu dari jasa tukang bangunan	2. Menekan tombol Cari Tukang untuk melakukan pemesanan	2.a. Mengirim notifikasi pemesanan ke aplikasi worker		2.b. Menampilkan pesan pebritahuan pemesanan
	Costumer	Sistem							
1. Memilih jasa tukang bangunan yang disediakan aplikasi	1. Menampilkan list menu dari jasa tukang bangunan								
2. Menekan tombol Cari Tukang untuk melakukan pemesanan	2.a. Mengirim notifikasi pemesanan ke aplikasi worker								
	2.b. Menampilkan pesan pebritahuan pemesanan								
<i>Alternatif course</i>									
<i>Pre-Cond.</i>	Menampilkan list menu jasa tukang bangunan								
<i>Post-Cond.</i>	Melakukan pemesanan jasa tukang bangunan								
<i>Assumption</i>	-								

TABEL VII DESKRIPSI USE CASE DIAGRAM TELEPON PADA COSTUMER

Nama use case	Telepon				
Aktor	Costumer dan Sistem				
Deskripsi	Costumer melakukan panggilan kepada worker				
<i>Normal course</i>	<table border="1"> <thead> <tr> <th>Costumer</th> <th>Sistem</th> </tr> </thead> <tbody> <tr> <td>1. Menekan tombol telepon untuk melakukan</td> <td>1. Membuat panggilan ke nomor worker terkait</td> </tr> </tbody> </table>	Costumer	Sistem	1. Menekan tombol telepon untuk melakukan	1. Membuat panggilan ke nomor worker terkait
	Costumer	Sistem			
1. Menekan tombol telepon untuk melakukan	1. Membuat panggilan ke nomor worker terkait				

	panggilan ke nomor worker
<i>Alternatif course</i>	
<i>Pre-Cond.</i>	Menampilkan informasi worker beserta tombol panggilan
<i>Post-Cond.</i>	Melakukan panggilan ke worker
<i>Assumption</i>	-

TABEL VIII DESKRIPSI USE CASE DIAGRAM LOGIN PADA WORKER

Nama use case	Login								
Aktor	Worker dan Sistem								
Deskripsi	Worker yang telah melakukan pendaftaran akun dapat masuk ke menu utama melalui proses login								
<i>Normal course</i>	<table border="1"> <thead> <tr> <th>Worker</th> <th>Sistem</th> </tr> </thead> <tbody> <tr> <td>1. Memasukkan data email dan password.</td> <td>1. Menampilkan form login</td> </tr> <tr> <td>2. Menekan tombol MASUK pada tampilan aplikasi</td> <td>2.a. Berpindah ke menu utama saat data benar</td> </tr> <tr> <td></td> <td>2.b. Menampilkan pesan peringatan apabila data yang dimasukkan salah</td> </tr> </tbody> </table>	Worker	Sistem	1. Memasukkan data email dan password.	1. Menampilkan form login	2. Menekan tombol MASUK pada tampilan aplikasi	2.a. Berpindah ke menu utama saat data benar		2.b. Menampilkan pesan peringatan apabila data yang dimasukkan salah
	Worker	Sistem							
1. Memasukkan data email dan password.	1. Menampilkan form login								
2. Menekan tombol MASUK pada tampilan aplikasi	2.a. Berpindah ke menu utama saat data benar								
	2.b. Menampilkan pesan peringatan apabila data yang dimasukkan salah								
<i>Alternatif course</i>									
<i>Pre-Cond.</i>	Menampilkan form login								
<i>Post-Cond.</i>	Memasukkan email dan password								
<i>Assumption</i>	-								

TABEL IX DESKRIPSI USE CASE DIAGRAM DAFTAR PADA WORKER

Nama use case	Daftar								
Aktor	Worker dan Sistem								
Deskripsi	Worker harus melakukan pendaftaran akun agar dapat masuk ke menu utama melalui proses login								
<i>Normal course</i>	<table border="1"> <thead> <tr> <th>Worker</th> <th>Sistem</th> </tr> </thead> <tbody> <tr> <td>1. Memasukkan data yang diperlukan.</td> <td>1. Menampilkan form pendaftaran</td> </tr> <tr> <td>2. Menekan tombol DAFTAR pada tampilan aplikasi</td> <td>2.a. Kembali ke menu login ketika selesai melakukan pendaftaran</td> </tr> <tr> <td></td> <td>2.b. Menampilkan pesan peringatan apabila terjadi kesalahan dalam memasukkan data pengguna</td> </tr> </tbody> </table>	Worker	Sistem	1. Memasukkan data yang diperlukan.	1. Menampilkan form pendaftaran	2. Menekan tombol DAFTAR pada tampilan aplikasi	2.a. Kembali ke menu login ketika selesai melakukan pendaftaran		2.b. Menampilkan pesan peringatan apabila terjadi kesalahan dalam memasukkan data pengguna
	Worker	Sistem							
1. Memasukkan data yang diperlukan.	1. Menampilkan form pendaftaran								
2. Menekan tombol DAFTAR pada tampilan aplikasi	2.a. Kembali ke menu login ketika selesai melakukan pendaftaran								
	2.b. Menampilkan pesan peringatan apabila terjadi kesalahan dalam memasukkan data pengguna								
<i>Alternatif course</i>									
<i>Pre-Cond.</i>	Menampilkan form pendaftaran								
<i>Post-Cond.</i>	Memasukkan data pengguna yang diperlukan								
<i>Assumption</i>	-								

TABEL X DESKRIPSI USE CASE DIAGRAM PROFIL PADA WORKER

Nama use case	Profil						
Aktor	Worker dan Sistem						
Deskripsi	Worker dapat melihat dan mengubah data profil yang dimiliki						
<i>Normal course</i>	<table border="1"> <thead> <tr> <th>Worker</th> <th>Sistem</th> </tr> </thead> <tbody> <tr> <td>1. Melihat data profil.</td> <td>1. Menampilkan profil worker</td> </tr> <tr> <td>2. Mengubah data profil</td> <td>2. Menampilkan jendela ubah profil</td> </tr> </tbody> </table>	Worker	Sistem	1. Melihat data profil.	1. Menampilkan profil worker	2. Mengubah data profil	2. Menampilkan jendela ubah profil
	Worker	Sistem					
1. Melihat data profil.	1. Menampilkan profil worker						
2. Mengubah data profil	2. Menampilkan jendela ubah profil						
<i>Alternatif course</i>							
<i>Pre-Cond.</i>	Menampilkan profil worker						
<i>Post-Cond.</i>	Memasukkan data profil						
<i>Assumption</i>	-						

TABEL XI DESKRIPSI USE CASE DIAGRAM EDIT HARGA PADA WORKER

Nama use case	Edit Harga
Aktor	Worker dan Sistem
Deskripsi	Worker dapat melihat dan mengubah data harga pemesanan jasa dari worker tersebut

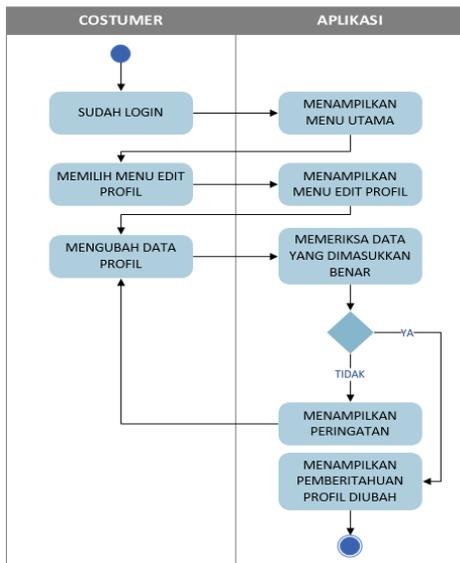
	Worker	Sistem
Normal course	1. Melihat data harga.	1. Menampilkan data harga <i>worker</i>
	2. Mengubah data harga	2. Menampilkan jendela ubah harga
Alternatif course		
Pre-Cond.	Menampilkan harga jasa <i>worker</i>	
Post-Cond.	Memasukkan data harga jasa <i>worker</i>	
Assumption	-	

TABEL XII DESKRIPSI USE CASE DIAGRAM PESANAN *COSTUMER* PADA *WORKER*

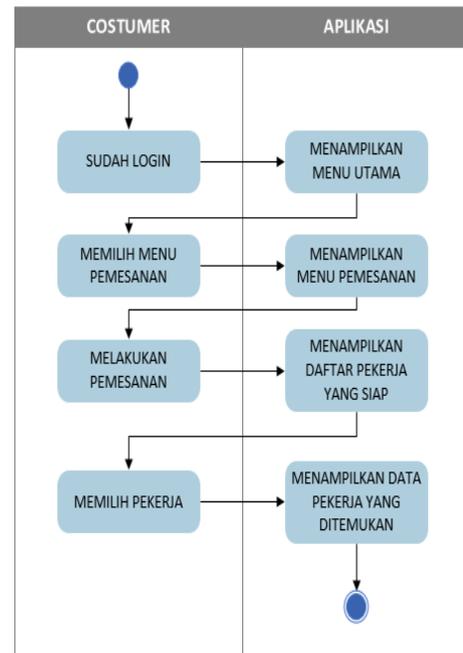
Nama use case	Pesanan <i>Costumer</i>	
Aktor	<i>Worker</i> dan Sistem	
Deskripsi	<i>Worker</i> mendapat pemesanan jasa dari <i>costumer</i>	
	Worker	Sistem
Normal course	1. Menerima pemesanan dari <i>costumer</i>	1. Menampilkan pemberitahuan pemesanan 2.a. Menampilkan lokasi <i>costumer</i> yang memesan 2.b. Menampilkan pesan pemberitahuan jika terjadi kesalahan
	Alternatif course	
Pre-Cond.	Menampilkan <i>list menu</i> jasa tukang bangunan	
Post-Cond.	Melakukan pemesanan jasa tukang bangunan	
Assumption	-	

TABEL XIII DESKRIPSI USE CASE DIAGRAM TELEPON PADA *WORKER*

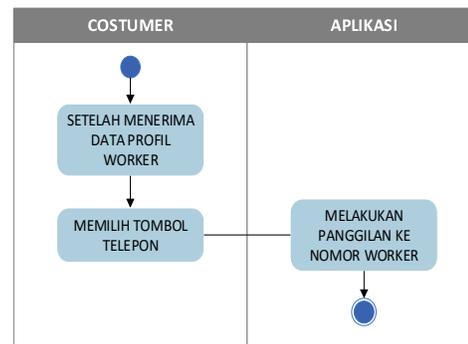
Nama use case	Telepon	
Aktor	<i>Worker</i> dan Sistem	
Deskripsi	<i>Worker</i> melakukan panggilan kepada <i>costumer</i>	
	Worker	Sistem
Normal course	1. Menekan tombol telepon untuk melakukan panggilan ke nomor <i>costumer</i>	1. Membuat panggilan ke nomor <i>costumer</i> terkait
	Alternatif course	
Pre-Cond.	Menampilkan informasi <i>costumer</i> beserta tombol panggilan ke <i>costumer</i>	
Post-Cond.	Melakukan panggilan ke <i>costumer</i>	
Assumption	-	



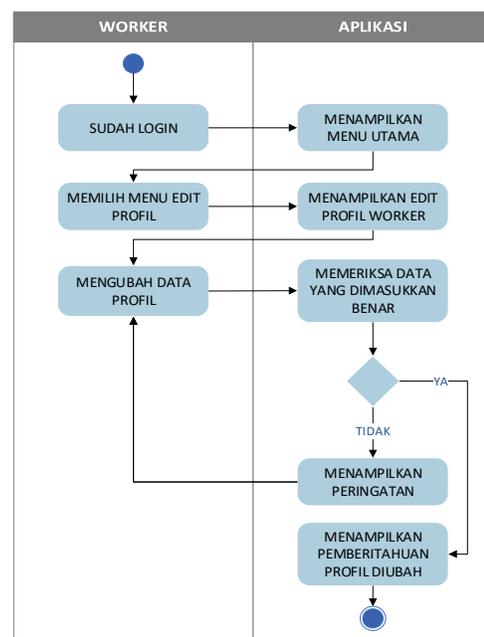
Gambar 6 Activity Diagram *Costumer* Untuk Membuka Menu Profil



Gambar 7 Activity Diagram *Costumer* Untuk Melakukan Pemesanan Jasa



Gambar 8 Activity Diagram *Costumer Worker* Untuk Melakukan Panggilan kepada *Worker*



Gambar 9 Activity Diagram *Worker* Untuk Membuka ke Menu Profil

c. Activity Diagram

Activity Diagram menggambarkan tentang aktifitas antara user dan sistem yang terjadi pada aplikasi dari awal sampai akhir. Pada gambar 6 Activity Diagram Costumer Untuk Membuka Menu Profil menunjukkan langkah-langkah dimana setelah *costumer* melakukan *login* dan masuk ke menu utama, *costumer* dapat menekan ikon *setting* yang terletak pada bagian bawah kanan dari menu utama aplikasi dan selanjutnya sistem akan mengarahkan *costumer* ke menu *edit* profil. Setelah itu *costumer* dapat mengubah data profil berupa nama dan nomor telpon dari *costumer* dan selanjutnya sistem akan memeriksa data yang dimasukkan benar atau tidak. Jika data benar maka sistem akan menampilkan pesan pemberitahuan bahwa profil berhasil diubah, jika tidak maka sistem akan menampilkan pesan pemberitahuan bahwa terjadi kesalahan.

Pada gambar 7 Activity Diagram Costumer Untuk Melakukan Pemesanan Jasa menunjukkan langkah-langkah ketika *costumer* sudah login dan masuk ke menu utama, *costumer* dapat memilih menu pesanan yang diinginkan untuk melakukan pemesanan jasa. Setelah itu *costumer* dapat mencari pekerja dan sistem akan menampilkan pekerja yang ditemukan oleh aplikasi sehingga *costumer* dapat memilih pekerja yang diinginkan.

Pada gambar 8 Activity Diagram Costumer Untuk Melakukan Panggilan kepada Worker menunjukan dimana setelah *costumer* menerima data dari *worker* yang ditemukan, *costumer* dalam melakukan panggilan ke *worker* dengan menekan ikon telepon pada profil *worker*.

Pada gambar 9 Activity Diagram Worker Untuk Masuk ke Menu Profil menunjukan langkah-langkah dimana setelah *worker* melakukan *login* dan masuk ke menu utama, *worker* dapat menekan ikon *setting* yang terletak pada bagian bawah kanan dari menu utama aplikasi dan selanjutnya sistem akan mengarahkan *worker* ke menu *edit* profil. Setelah itu *worker* dapat mengubah data profil berupa nama dan nomor telpon dari *worker* dan selanjutnya sistem akan memeriksa data yang dimasukkan benar atau tidak. Jika data benar maka sistem akan menampilkan pesan pemberitahuan bahwa profil berhasil diubah, jika tidak maka sistem akan menampilkan pesan pemberitahuan bahwa terjadi kesalahan.

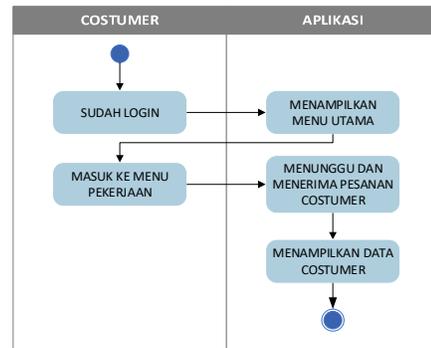
Pada gambar 10 Activity Diagram Worker Menerima Pesanan Costumer dibawah menunjukan langkah-langkah dimana setelah *worker* melakukan *login* dan masuk ke menu utama, *worker* masuk ke menu pekerjaan untuk menerima pesanan dari *costumer*. Setelah *worker* berhasil mendapat pesanan, sistem akan menampilkan informasi profil dari pemesan pada aplikasi.

Pada gambar 11 Activity Diagram Worker Untuk Melakukan Panggilan kepada Costumer menunjukan dimana setelah *worker* mendapatkan informasi *costumer*, *worker* dapat melakukan panggilan ke *costumer* terkait dengan menekan ikon telepon yang terletak pada informasi *costumer*. Setelah itu sistem akan melakukan panggilan ke *costumer*.

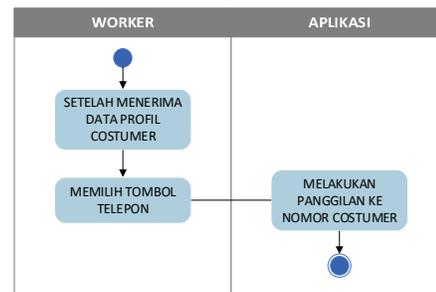
d. Database Firebase

Pada perancangan dan pengembangan *database* dalam aplikasi penelitian ini adalah *Firebase*. Didalam *Firebase* terdapat *database costumer* dan *worker* yang diberikan ID

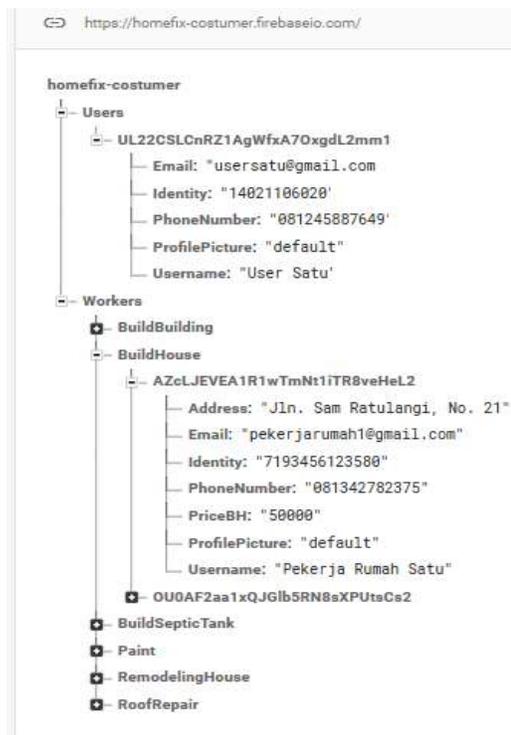
husus secara acak untuk tiap pengguna yang terdapat didalam *database child Users (costumer)* dan *Worker*. Dapat dilihat pada gambar 12 *database costumer* dan *worker*. Didalam ID *key* pengguna terdapat data profil dari tiap-tiap pengguna yang ada berupa *address* (alamat), *email*, *identity* (identitas pengguna), *phone number* (nomor telepon pengguna), *priceBH* (harga jasa pengguna berdasarkan pekerjaan), *profile picture* (foto profil) dan *username* (nama pengguna).



Gambar 10 Activity Diagram Worker Menerima Pesanan Costumer



Gambar 11 Activity Diagram Worker Untuk Melakukan Panggilan kepada Costumer



Gambar 12 Database Firebase Pengguna

e. Kode Program

Pada perancangan dan pengembangan aplikasi dalam penelitian ini terdapat beberapa penggalan kode-kode program yang digunakan antara lain, yaitu penggalan kode-kode program pemesanan jasa dapat dilihat pada gambar 13 dan penggalan kode-kode program lokasi pekerja pada gambar 14. Pada penggalan kode-kode program gambar 13 terdapat kodingan yang digunakan untuk menemukan jasa tukang bangunan dari *database firebase* seperti kode `rvReferenceWdata=FirebaseDatabase.getInstance().getReference().child("Workers").child("BuildHouse");` memiliki fungsi untuk menunjukkan lokasi data dari pekerja pada *database BuildHouse* terdapat didalam *database Workers* untuk dipesan. Pada kodingan `rvReferenceWdata.addValueEventListener(new ValueEventListener())` memiliki fungsi untuk menangkap data dengan fungsi dari `dataSnapshot.getChildren()` melalui *database* yang telah ditunjukkan oleh kodingan `rvReferenceWdata` sebelumnya. Pada penggalan kode-kode program gambar 14 terdapat kodingan `LatLng workerLatLatLng = new LatLng(locationLat, locationLng);` yang memiliki fungsi untuk membuat *latitude* (garis lintang) dan *longitude* (garis bujur) yang baru pada *map*. Selain itu terdapat juga kodingan `Location location = new Location("");`; dan `Location location2=new Location("");`; memiliki fungsi untuk mendapatkan lokasi melalui *google map* sehingga aplikasi dapat menampilkan lokasi dari pengguna terkait. Pada kodingan `workerMarker = mMap.addMarker(new MarkerOptions().position(workerLatLatLng).title("Your Worker").icon(BitmapDescriptorFactory.fromResource(R.drawable.map_icon_worker)));`; memiliki fungsi untuk menampilkan lokasi dengan ikon lokasi yang telah dibuat oleh pengembang pada aplikasi.

3) Implementasi Antarmuka Aplikasi

Tampilan *interface* atau antarmuka dari perancangan dan pengembangan aplikasi pada penelitian ini, dapat dilihat pada gambar 15-20. Gambar 15 merupakan tampilan antarmuka *profile edit* yang dapat digunakan *costumer* untuk mengubah data profil dari *costumer* pada *database*. Pada menu ini juga pengguna dapat keluar ke menu *login* dengan menekan tombol KELUAR AKUN. Pada gambar 16 merupakan tampilan menu *map* pemesanan *costumer* ini merupakan menu yang muncul ketika *costumer* menekan tombol CARI PEKERJA untuk mencari *worker* atau pekerja jasa tukang bangunan. Pada tampilan menu daftar *user* pada gambar 17 adalah menu yang akan digunakan *worker* yang belum memiliki akun untuk memilih jenis pekerjaan dan melakukan pendaftaran akun agar bisa mengakses masuk ke menu utama aplikasi. Pada gambar 18 merupakan tampilan antarmuka *edit* harga yang dapat digunakan *worker* untuk mengubah data harga atau biaya jasa dari pekerjaan *worker*. Pada gambar 19 merupakan tampilan antarmuka menu *map* aplikasi. Pada tampilan menu ini *worker* dapat menerima pesanan dari *costumer* berdasarkan jenis pemesanan yang dipilih. Pada gambar 20 dapat dilihat tampilan antarmuka *profile edit* yang dapat digunakan *worker* untuk mengubah data profil yang dimiliki *worker*. Pada menu ini juga pengguna dapat keluar dari menu utama ke menu *login*.

```
private void getListWorker()

recycleListLayout.setVisibility(View.VISIBLE);
recyclerView =
findViewById(R.id.recyclerViewWorker);
rvWorkerListLayout =
findViewById(R.id.rvLyWorkerInfo);
recyclerView.setLayoutManager(new
LinearLayoutManager(this));

list = new ArrayList<WorkerDataBH>();
rvReferenceWdata =
FirebaseDatabase.getInstance()
.getReference().child("Workers")
.child("BuildHouse");

rvReferenceWdata.addValueEventListener(
new ValueEventListener() {

@Override
public void onDataChange(@NonNull
DataSnapshot dataSnapshot) {
for (DataSnapshot dataSnapshot1 :
dataSnapshot.getChildren()) {
WorkerDataBH p =
dataSnapshot1.getValue(WorkerDataBH.class);
list.add(p);
}
adapter = new WorkerAdapter(CostumerMapBH.this,
list);
recyclerView.setAdapter(adapter);
recyclerView.setLayoutManager(new
LinearLayoutManager(getApplicationContext());
requestW.setText("Batalkan");
});
}
```

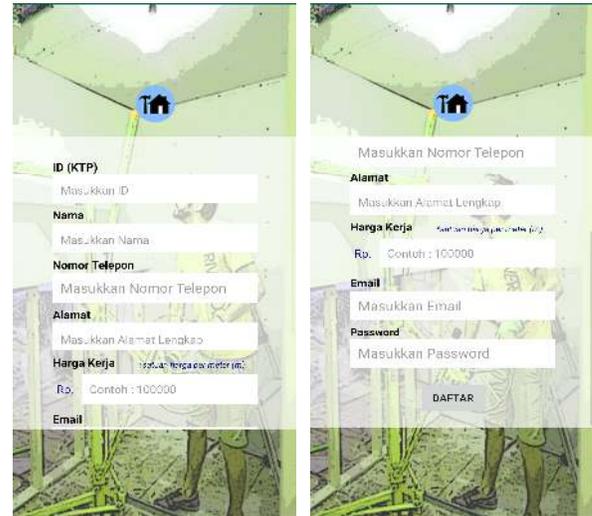
Gambar 13 Penggalan Kode-Kode Program Pemesanan Jasa

```
private void getWorkerLocation() {
workerLocationRef = FirebaseDatabase
.getInstance().getReference()
.child("workersWorkingBH").child(workerFoundID)
.child("1");
workerLocationRefListener = workerLocationRef
.addValueEventListener(new ValueEventListener() {
@Override
public void
onDataChange(@NonNull DataSnapshot dataSnapshot) {
if (dataSnapshot.exists() && reqBoolean) {
List<Object> map = (List<Object>) dataSnapshot
.getValue();
double locationLat = 0; double locationLng = 0;
if (map.get(0) != null) {
locationLat = Double.parseDouble
(map.get(0).toString()); } if (map.get(1) != null)
{ locationLng = Double.parseDouble
(map.get(1).toString()); }
LatLng workerLatLatLng = new LatLng(locationLat,
locationLng);
if (workerMarker != null) {workerMarker.remove(); }
Location location1 = new Location("");
location1.setLatitude (targetLocation.latitude);
location1.setLongitude (targetLocation.longitude);
Location location2 = new Location("");
location2.setLatitude (workerLatLatLng.latitude);
location2.setLongitude (workerLatLatLng.longitude);
workerMarker = mMap.addMarker(new MarkerOptions()
.position(workerLatLatLng) .title("Your Worker")
.icon(BitmapDescriptorFactory.fromResource(R.drawa
ble.map_icon_worker))); }
}
```

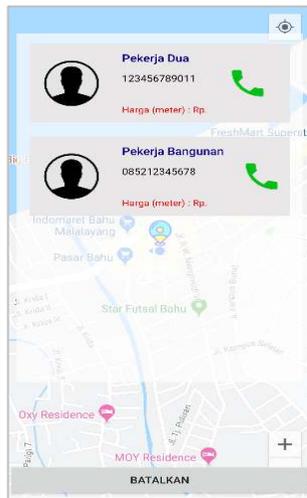
Gambar 14 Penggalan Kode-Kode Program Lokasi Pekerja



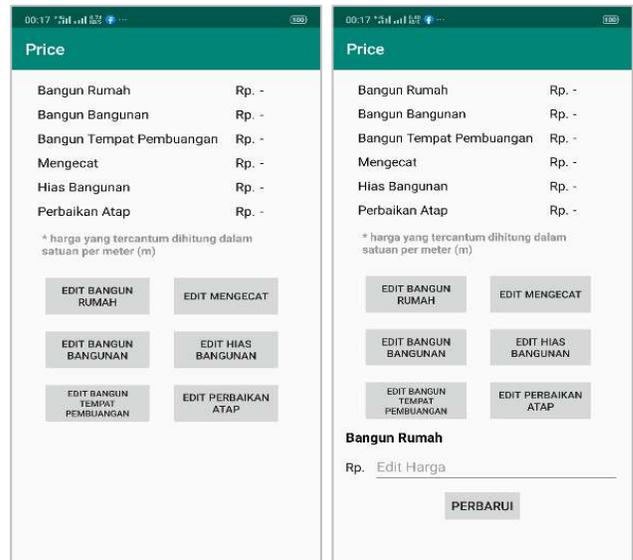
Gambar 15 Interface Profile Edit



Gambar 17.b Interface Daftar Worker



Gambar 16 Interface Menu Map Pemesanan Pada Costumer



Gambar 18 Interface Edit Harga



Gambar 17.a Interface Daftar Worker



Gambar 19 Interface Menu Map Pemesanan Pada Worker



Gambar 20 Interface Profil Edit Worker

4) Kendala-Kendala Yang Dihadapi

Adapun kendala yang dihadapi dalam perancangan dan pengembangan aplikasi *online* sistem pemesanan jasa tukang bangunan berbasis lokasi ini, antara lain :

- Update* profil dari aplikasi *worker*. Tidak menemukan kode program yang tepat pada saat memperbarui profil *worker* dikarenakan perbedaan *child database* pada *Firestore* yang dikategorikan berdasarkan *database* pekerjaan sehingga aplikasi hanya bisa merubah data pada satu *database* dari jenis pekerjaan yang ada.
- Update* harga dari aplikasi *worker* karena tidak menemukan kode program yang tepat pada saat memperbarui harga dari pekerjaan *worker* dan perbedaan *child database* dari lokasi tiap-tiap *worker* yang dikategorikan berdasarkan *database* pekerjaan sehingga aplikasi tidak bisa merubah harga berdasarkan ID khusus dari *Firestore*.
- Lokasi *worker* tidak muncul pada map *costumer* saat memilih *worker* yang tersedia dikarenakan tidak menemukan kode program yang tepat.
- Foto profil *worker* dan *costumer* tidak terbaca dari *database* dikarenakan tidak menemukan kode program yang tepat.

IV. PENUTUP

A. Kesimpulan

Dari hasil dan pembahasan perancangan dan pembuatan Aplikasi *Online* Sistem Pemesanan Jasa Tukang Bangunan Berbasis Lokasi yang telah dilakukan, dapat disimpulkan bahwa penulis telah berhasil merancang dan membangun aplikasi *online* sistem pemesanan jasa tukang bangunan berbasis lokasi dan dapat bermanfaat untuk masyarakat dalam mencari jasa tukang bangunan yang ada di Kota Manado

B. Saran

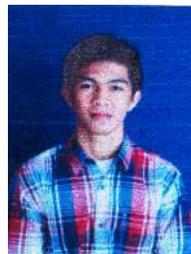
Berdasarkan hasil pembahasan Aplikasi *Online* Sistem Pemesanan Jasa Tukang Bangunan Berbasis Lokasi yang dirancang dan dibuat penulis maka penulis dapat mengajukan saran pada aplikasi ini yaitu *developer* dapat mengembangkan

fitur *setting* untuk mengatur tampilan, bahasa, dan pengaturan lainnya pada aplikasi, *developer* dapat mengembangkan fitur interaksi pesan untuk *costumer* dan *worker*, *developer* dapat mengembangkan fitur *history* untuk mengatur riwayat pemesanan dari sisi *worker* dan *costumer*, *developer* dapat mengembangkan fitur-fitur pada sisi *worker* dan *costumer* untuk mempermudah penggunaan pada *user*.

KUTIPAN

- R. Rumimper, S. R. U. A. Sompie, and D. J. Mamahit, "Rancang Bangun Alat Pengontrol Lampu Dengan Bluetooth Berbasis Android," *J. Tek. Inform.*, vol. 5, no. 3, pp. 24–33, 2016.
- A. N. Lomboan, D. J. Mamahit, and Y. D. Y. Rindengan, "Rancang Bangun Aplikasi Pencarian Rumah Ibadah Di Kotamobagu Berbasis Android," *J. Tek. Inform.*, vol. 13, no. 2, 2018, doi: 10.35793/jti.13.2.2018.22552.
- F. Fatgehipon, A. S. M. Lumenta, and B. A. Sugiarto, "Peta Digital Kota Manado Berbasis Android," *E-Journal Tek. Elektro Dan Komput.*, vol. 4, no. 7, pp. 73–78, 2015.
- I. S. A. Mukti, A. S. M. Lumenta, and B. A. Sugiarto, "Rancang Bangun Aplikasi Pembelajaran Untuk Anak Umur 6 – 9 Tahun Berbasis Android," *J. Tek. Inform.*, vol. 7, no. 1, pp. 1–6, 2018, doi: 10.35793/jti.7.1.2016.10772.
- Sumberpengertian, "Pengertian Jasa Lengkap! (Ciri-ciri, Jenis-jenis Jasa dan Contoh Jasa)," *sumberpengertian.id*. [Online]. Available: <https://www.sumberpengertian.id/pengertian-jasa-lengkap>.
- V. S. Gunawan, A. A. E. Sinsuw, and A. M. Sambul, "Location-Based Information Berbasis QR Code Untuk Tourism," *J. Tek. Inform.*, vol. 13, no. 1, pp. 1–9, 2018, doi: 10.35793/jti.13.1.2018.20197.
- C. O. Karundeng, D. J. Mamahit, and B. A. Sugiarto, "Rancang Bangun Aplikasi Pengenalan Satwa Langka di Indonesia Menggunakan Augmented Reality," *J. Tek. Inform.*, vol. 13, no. 1, pp. 1–8, 2018, doi: 10.35793/jti.13.1.2018.20852.
- Vlado, "Rapid Application Development (RAD)," *Adikristanto*, 2010. [Online]. Available: <http://adikristanto.net/rapid-application-development-rad/>.
- A. A. A. Makiolor, A. Sinsuw, and X. B.N. Najoan, "Rancang Bangun Pencarian Rumah Sakit, Puskesmas dan Dokter Praktek Terdekat di Wilayah Manado Berbasis Android," *J. Tek. Inform.*, vol. 10, no. 1, 2017, doi: 10.35793/jti.10.1.2017.16552.
- K. Wong, "Rancang Bangun Aplikasi 'Trip Alone?' Berbasis Android," *J. Tek. Inform.*, vol. 9, no. 1, 2016, doi: 10.35793/jti.9.1.2016.14809.
- D. Lumingkewas, Y. D. Y. Rindengan, and X. B. N. Najoan, "Peta Aset Universitas Sam Ratulangi Berbasis Web," *J. Tek. Inform.*, vol. 14, no. 1, pp. 53–62, 2019, doi: 10.35793/jti.14.1.2019.23837.

TENTANG PENULIS



Nama penulis adalah Janiver W. Janis. Lahir pada tanggal 27 Janiver 1997 di kota Manado. Saya merupakan anak pertama dari Imelda N. Tadete. Pendidikan yang telah saya tempuh dimulai dari TK Katolik Ondong (2001), Saya melanjutkan pendidikan di SDN INPRES Paniki (2002–2008), dan melanjutkan studi di SMP Negeri 1 Siau Barat (2009–2011) dan melanjutkan pendidikan di SMA Negeri 1 Siau Barat (2011–2014). Di tahun 2014 saya menyelesaikan studi SMA kemudian melanjutkan pendidikan S1 di perguruan tinggi Universitas Sam Ratulangi dengan mengambil Program Studi Teknik Informatika di Jurusan Elektro Fakultas Teknik. Hingga akhirnya pada Mei 2020 saya dapat menyelesaikan studi S1 dengan hasil yang baik.