

# Mobile-Based Caregiver Ordering System Application Development

Pengembangan Sistem Aplikasi Pemesanan Jasa Pengasuh Berbasis Mobile

Albert Christian Tjakra, Sherwin R. U. A. Sompie, Dirko G.S. Ruindungan

Dept. of Electrical Engineering, Sam Ratulangi University Manado, Kampus Bahu St., 95115, Indonesia

e-mails : [17021106027@student.unsrat.ac.id](mailto:17021106027@student.unsrat.ac.id), [aldo@unsrat.ac.id](mailto:aldo@unsrat.ac.id), [dirko@unsrat.ac.id](mailto:dirko@unsrat.ac.id)

Received: 9 June 2021; revised: 15 July 2021; accepted: 15 July 2021

**Abstract** — *Busy of work and other important things make many people need caregivers to care for and care for families that require special attention in carrying out daily life. However, the search for caregiver services is still done manually through acquaintances. This makes caregiver services difficult to find and takes a lot of time to search and often does not find caregiver services that fit their needs. This research aims to build a caregiver service ordering system application that is expected to help bring together people and caregivers in the search process for caregiver services so that they can easily and quickly find caregivers that suit their needs compared to manual search. This application is built based on the Android operating system using Android Studio and Firebase. In its development, this application uses the Extreme Programming method. This study resulted in an application system for ordering caregiver services that was used to find caregiver services and could also be a forum for people who work as caregivers to offer their services. This research also provides a web-based management system.*

**Keywords** — *Android Application; Android Studio; Extreme Programming; Firebase.*

**Abstrak** — Kesibukan kerja dan hal-hal penting lainnya membuat banyak orang membutuhkan jasa pengasuh untuk mengasuh dan merawat keluarga yang memerlukan perhatian khusus dalam menjalani kehidupan sehari-hari. Namun pencarian jasa pengasuh masih dilakukan secara manual melalui kenalan. Hal tersebut membuat jasa pengasuh sulit ditemukan dan memakan banyak waktu dalam pencariannya dan seringkali juga tidak menemukan jasa pengasuh yang sesuai kebutuhan. Penelitian ini bertujuan untuk membangun sistem aplikasi pemesanan jasa pengasuh yang diharapkan dapat membantu mempertemukan masyarakat dengan pengasuh dalam proses pencarian jasa pengasuh agar dapat dengan mudah dan cepat menemukan jasa pengasuh yang sesuai dengan kebutuhannya dibandingkan pencarian secara manual. Aplikasi ini dibangun berbasis sistem operasi *Android* dengan menggunakan *Android Studio* dan *Firebase*. Dalam pengembangannya, aplikasi ini menggunakan metode *Extreme Programming*. Penelitian ini menghasilkan sistem aplikasi pemesanan jasa pengasuh yang digunakan untuk mencari jasa pengasuh dan juga dapat menjadi wadah bagi masyarakat yang berprofesi sebagai pengasuh dalam menawarkan jasanya. Penelitian ini juga tersedia sistem pengelolannya yang berbasis web.

**Kata kunci** — *Android Studio; Aplikasi Android; Extreme Programming; Firebase.*

## I. PENDAHULUAN

Pada tahun 2018 diketahui bahwa lebih dari setengah masyarakat Indonesia sudah memiliki telepon pintar atau

*smartphone* yaitu sebesar 66,31%. Sedangkan berdasarkan pendidikan sebesar 79,56% siswa Sekolah Menengah Atas memiliki telepon pintar. Juga berdasarkan profesi Pegawai Negeri Sipil 89,12% memiliki telepon pintar. [1]

Kesibukan kerja dan hal-hal penting lainnya membuat banyak orang membutuhkan jasa pengasuh untuk mengasuh dan merawat keluarga yang sedang sakit dan memerlukan perhatian khusus dalam menjalani kehidupan sehari-hari. Di kota Manado sendiri pada tahun 2017 terdapat 35.289 orang lansia dan lansia yang mendapat pelayanan kesehatan sebanyak 8.022 (22,73%). Sedangkan untuk kunjungan di Puskesmas dan Rumah sakit pada tahun 2017 di kota Manado sebanyak 513.227 orang yang terdiri dari 458.991 rawat jalan, 54.236 rawat inap. Namun pencarian jasa pengasuh masih dilakukan secara manual melalui kenalan. Karena pencarian jasa pengasuh masih secara manual. Maka jasa pengasuh sulit ditemukan dan memakan banyak waktu dalam pencariannya dan sering kali juga tidak menemukan jasa pengasuh yang sesuai kebutuhan mereka.

Adapun salah satu penelitian yang menerapkan pemesanan jasa pada aplikasi berbasis *Mobile* yaitu penelitian yang berjudul Rancang Bangun Aplikasi *Online* Sistem Pemesanan Jasa Tukang Bangunan Berbasis Lokasi. juga aplikasi yang saat ini beredar di pasar Indonesia yang menerapkan pemesanan jasa pada aplikasi berbasis *Mobile Android* seperti *Gojek*, *Grab* dan lainnya. Ini menunjukkan bahwa pemesanan jasa yang diterapkan pada aplikasi berbasis *Mobile* sudah sangat dikenal.[2]

Penelitian ini bertujuan membantu menyelesaikan permasalahan dengan sebuah Sistem Aplikasi Pemesanan Jasa Pengasuh Berbasis *Mobile* yang diharapkan bisa membantu dan mempercepat pengguna jasa pengasuh dalam mencari jasa pengasuh yang memenuhi kebutuhan mereka. Dalam hal ini jasa pengasuh yang dimaksud terbatas pada jasa pengasuh untuk Balita dan Lansia.

### A. Jasa Pengasuh

Jasa Pengasuh adalah orang awam yang dilatih untuk memberikan jasa dalam merawat atau mengasuh orang lain yang memerlukan perhatian khusus dan bantuan dalam menjalani kesehariannya yang di sebabkan oleh sakit, faktor umur yang sudah lanjut, dan sebagainya. Akan tetapi jasa

pengasuh tidak berkemampuan dan berkewenangan dalam melakukan tindakan medis keperawatan.

Tugas Pengasuh adalah untuk meringankan beban orang yang dirawat atau keluarga dari orang yang dirawat supaya dapat menjalani aktivitas sehari-hari seperti menyiapkan makanan, memandikan, membersihkan tempat tidur, membantu pasien buang air besar dan buang air kecil, memberi minum obat, dan sebagainya. [3]

### B. Pengembangan Aplikasi 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. Terdapat banyak lingkungan pengembangan terpadu untuk *Android* namun yang paling populer adalah *Android Studio*. [4]

*Android Studio* merupakan Lingkungan Pengembangan terpadu (*Integrated Development Environment/IDE*) resmi untuk pengembangan aplikasi *android*, yang didasarkan pada *IntelliJ IDEA*. *Android Studio* menghasilkan aplikasi untuk perangkat *smartphone* bersistem operasi *Android* dengan jenis ekstensi *APK* dan *XAPK*. *Android Studio* menawarkan banyak fitur yang meningkatkan produktivitas dalam membuat aplikasi *Android*. [2], [5] Itu sebabnya *Android Studio* dapat dikatakan menggantikan *Eclipse*. *Android studio* mendukung bahasa pemrograman *Java*, *Kotlin* dan lainnya. [6]

*Kotlin* merupakan salah satu bahasa pemrograman yang digunakan untuk pengembangan aplikasi *Android*. *Kotlin* juga termasuk baru dan *modern*, dan berjalan pada *platform Java Virtual Machine (JVM)* secara statis. *Kotlin* juga dapat dikompilasi ke dalam *JavaScript* menggunakan *compiler LLVM*. *Jetbrains* yang merupakan pengembang *tools* pemrograman *IntelliJ IDEA* yang adalah juga pengembang dari bahasa pemrograman *Kotlin*. *Google* mengadopsi bahasa pemrograman *Kotlin* ini untuk mengembangkan aplikasi *Android* melalui *Android Studio*. *Kotlin* menjadi bahasa pemrograman kelas utama dalam pengembangan aplikasi *Android* pada pagelaran *Google I/O*. Bahasa pemrograman *Kotlin* juga dapat digunakan dalam membangun aplikasi yang berbasis *Web*, *Desktop* dan juga dipakai pada *Backend*. [7]

### C. Firebase

*Firebase* adalah suatu layanan dari *Google* untuk memberikan kemudahan untuk para pengembang aplikasi dalam mengembangkan aplikasinya. *Firebase* yang adalah *BaaS (Backend as a Service)* merupakan solusi yang ditawarkan oleh *Google* untuk mempercepat pekerjaan *developer*. *Firebase* didirikan pertama kali pada tahun 2011 oleh Andrew Lee dan James Tamplin. Pada 2014, perusahaan tersebut diakuisisi oleh *Google*. *Firebase* memiliki banyak fitur layanan yang bisa digunakan untuk membantu proses pengembangan aplikasi. Layanan yang bisa digunakan pada *firebase* adalah *Firebase Analytics*, *Firebase Cloud Messaging and Notification*, *Firebase Cloud Firestore*, *Firebase Realtime Database*, *Firebase Hosting*, *Firebase Cloud Storage*. [8], [9]

Untuk konektivitas *Android* ke *Firebase realtime database* sangat penting untuk mengetahui dasar-dasarnya seperti

menulis, membaca data, dan bagaimana data terstruktur. Untuk operasi penulisan data ke basis data, digunakan *setValue()* untuk menyimpan data ke referensi yang telah ditentukan, juga dapat menggantikan data yang ada di jalur tersebut. Contoh penggunaan *setValue()* untuk menyimpan data *user* menggunakan *Kotlin* dapat dilihat pada Gambar 1. Sedangkan untuk membaca data dan membuat data aplikasi diperbarui secara *realtime*, harus ditambahkan *ValueEventListener* ke referensi yang telah di buat. dengan menggunakan metode *onDataChange()* akan membaca data sekali saat *listener* terpasang dan dipicu lagi setiap kali data berubah. Contoh penggunaan *onDataChange()* untuk membaca data *post* menggunakan *Kotlin* dapat dilihat pada Gambar 2. Untuk dapat menyimpan dan membaca data sangat penting untuk mengenal struktur dari *Firebase realtime database*. Semua data yang ada pada *Firebase realtime database* disimpan sebagai objek *JSON* dalam bentuk pohon *JSON*. Contoh struktur data yang tersimpan yaitu data profil *user* dimana profil pengguna terletak pada jalur */users/Suid* dapat dilihat pada Gambar 3. [10]

### D. Extreme Programming

*Extreme Programming (XP)* adalah salah satu metode pengembangan perangkat lunak yang biasanya menggunakan pendekatan berorientasi objek dan metode ini juga sesuai dengan *requirement* yang tidak jelas dan juga terjadi perubahan-perubahan *requirement* yang sangat cepat. Tim yang dibentuk dalam skala kecil sampai medium sangat cocok dengan metode ini. [11]

Pada maret 1996 *extreme programming* di usulkan pertama kali oleh Kent Beck dan Ward Cunningham, *extreme programming* digunakan pada saat itu karena cepatnya perubahan permintaan konsumen yang membuat metode tradisional tidak lagi cocok untuk digunakan karena memerlukan biaya yang sangat tinggi. Oleh sebab itu *extreme programming* bertujuan untuk memangkas biaya yang diperlukan ketika terjadi perubahan dalam pengembangan perangkat lunak. [12]

Terdapat 4 tahapan dalam model pengembangan *extreme programming* seperti pada Gambar 4 yaitu *Planning* yang adalah tahap perencanaan, *Design* yaitu tahap perancangan, *Coding* yang merupakan tahap penulisan kode program, dan terakhir adalah *Testing* yaitu tahap pengujian.

### E. Basis Data NoSQL

*NoSQL* adalah basis data tanpa relasi (*non-relational*) yang dirancang untuk menyimpan data yang tidak memiliki struktur yang tetap. *NoSQL* tidak membutuhkan data model relational dan bahasa *SQL* untuk beroperasi. Jenis alat pemodelan data standar yang sama dengan *SQL* tidak tersedia untuk pemodelan data *NoSQL*. Salah satu rekomendasi untuk memodelkan data *NoSQL* adalah memulai dengan model domain bisnis yang diekspresikan dalam bentuk yang dapat dimasukkan ke dalam aplikasi, seperti dokumen *JSON* yang terlihat pada Tabel I. Properti mendasar dari basis data *NoSQL* adalah kebutuhan untuk mengoptimalkan akses data, yang menempatkan fokus pada pola kueri dan alur kerja bisnis. Langkah pertama adalah menetapkan persyaratan bisnis, dan mencari tahu kebutuhan

spesifik orang-orang yang akan menggunakan aplikasi tersebut. Penyimpanan fisik dari basis data *NoSQL* akan didistribusikan dan dipartisi secara horizontal. [13]

#### F. Penelitian Terkait

Dalam Publikasi Janis dkk (2020) berjudul Rancang Bangun Aplikasi *Online* Sistem Pemesanan Jasa Tukang Bangunan Berbasis Lokasi. Persamaan yang ada terkait dengan penelitian skripsi ini yaitu penggunaannya untuk pengembangan aplikasi pemesanan di bidang jasa yang berbasis android dengan menggunakan *Android Studio* sebagai *Integrated Development Environment (IDE)* dan *Firestore* sebagai *back-end*. Perbedaan yang ada terkait dengan penelitian skripsi ini yaitu penggunaan bahasa pemrograman yang berbeda dimana penulis publikasi menggunakan bahasa pemrograman *Java* sedangkan peneliti menggunakan bahasa pemrograman *Kotlin* dan *javascript*, perbedaan juga terletak di jumlah *user* atau aktor dimana penelitian penulis publikasi memiliki jumlah *user* sebanyak 2(dua) sedangkan penelitian peneliti memiliki jumlah *user* sebanyak 3(tiga). Terdapat juga perbedaan penelitian dimana penulis publikasi menerapkan metode pengembangan *Rapid Application Development (RAD)* sedangkan peneliti menggunakan metode *Extreme Programming (XP)*. [2]

Dalam Publikasi Syani dan Werstantia (2018) berjudul Perancangan Aplikasi Pemesanan *Catering* Berbasis *Mobile Android*. Persamaan yang ada terkait dengan penelitian skripsi ini yaitu penggunaannya untuk pengembangan aplikasi pemesanan dengan menggunakan *Android Studio* sebagai *integrated development environment (IDE)* dengan metode pengembangan *Extreme Programming (XP)* yang menghasilkan aplikasi berbasis android yang terintegrasi dengan aplikasi yang berbasis web. Perbedaan dengan penelitian ini yaitu penggunaan basis data yang berbeda dimana penulis publikasi menggunakan *MySQL* sedangkan peneliti menggunakan basis data *Realtime database* dari *Firestore*, perbedaan juga terletak pada penggunaan bahasa pemrograman yang berbeda dimana penulis publikasi menggunakan bahasa pemrograman *Java* dan *php* sedangkan peneliti menggunakan bahasa pemrograman *Kotlin* dan *javascript*. Terdapat juga perbedaan dimana penulis publikasi ini mengembangkan aplikasi pemesanan untuk *catering* sedangkan peneliti mengembangkan aplikasi pemesanan untuk jasa pengasuh.[14]

Dalam Publikasi Wong dkk (2016) Rancang Bangun Aplikasi “Trip Alone” Berbasis *Android*. Persamaan yang ada terkait dengan penelitian skripsi ini yaitu penggunaannya untuk pengembangan yang berbasis android dengan menggunakan *Firestore* sebagai *back-end*. Perbedaan dengan penelitian ini yaitu penggunaan bahasa pemrograman yang berbeda dimana penulis publikasi menggunakan bahasa pemrograman *Javascript* sedangkan peneliti menggunakan bahasa pemrograman *Kotlin* dan *javascript*. Terdapat juga perbedaan penelitian dimana penulis publikasi menerapkan metode pengembangan *Disciplined Agile Delivery (DAD)* sedangkan peneliti menggunakan metode *Extreme Programming (XP)*. [15]

Dalam Publikasi Tompoh dkk (2016) Rancang Bangun Aplikasi Pemesanan Menu Makanan Restoran Berbasis

*Android*. Persamaan yang ada terkait dengan penelitian skripsi ini yaitu penggunaannya untuk pengembangan yang berbasis *Android* untuk *end-user* dan berbasis *Web* untuk admin. Perbedaan dengan penelitian ini yaitu penggunaan bahasa pemrograman yang berbeda dimana penulis publikasi menggunakan bahasa pemrograman *Javascript* dan *Php* sedangkan peneliti menggunakan bahasa pemrograman *Kotlin* dan *Javascript*. Terdapat juga perbedaan penelitian dimana penulis publikasi menerapkan metode pengembangan *Rapid Application Development (RAD)* sedangkan peneliti menggunakan metode *Extreme Programming (XP)*. [16]

## II. METODE

### A. Metode Pengembangan

Peneliti menggunakan metode pengembangan *Extreme Programming (XP)*. Berikut merupakan tahapan – tahapan dalam *Extreme Programming*:

#### 1) Planning

Pada tahap *Planning* (perencanaan) dilakukan identifikasi masalah dan analisa kebutuhan yaitu melalui observasi dan wawancara dengan narasumber, metode yang digunakan adalah menggunakan metode kuesioner dengan menggunakan *Google Form*.

#### 2) Design

Pada tahap *Design* (Perancangan) dilakukan berbagai macam pemodelan seperti pemodelan sistem dan pemodelan arsitektur menggunakan Desain Arsitektur Sistem dan *Unified Modeling Language (UML)* yaitu *Use case Diagram* dan *Activity Diagram*, dan pemodelan basis data yang disajikan dalam bentuk tabel yang menjelaskan tentang atribut dan struktur basis data *NoSQL*. Juga untuk menggambarkan interaksi antara objek digunakan *Sequence Diagram*.

#### 3) Coding

Tahapan *Coding* (Pengkodean) merupakan penerapan dari model yang telah dilakukan ke dalam eksekusi program menggunakan bahasa pemrograman *Kotlin* dengan *IDE Android studio* untuk *Android*, *JavaScript* dengan *IDE Visual Studio Code* untuk *Web* dan dengan basis data yang menggunakan *Firestore Realtime Database*.

#### 4) Testing

Tahapan *Testing* (Pengujian) merupakan tahapan pengujian sistem dengan melakukan *Black box testing*.

## III. HASIL DAN PEMBAHASAN

### A. Tahap Perencanaan

Pada Tahapan ini akan menentukan apa saja yang menjadi kebutuhan dari setiap pengguna aplikasi dan fungsionalitas apa saja yang akan di implementasikan pada aplikasi sistem pemesanan jasa pengasuh sebagai fitur yang dibutuhkan. Untuk mengumpulkan data peneliti menggunakan metode kuesioner dan observasi.

Terdapat kuesioner untuk pengguna jasa pengasuh dan untuk penyedia jasa pengasuh dengan masing-masing pertanyaan yang berbeda. Pengumpulan kuesioner menggunakan *google form* sebagai media pemungutan dan pengolahan data kuesioner

secara daring.

1) Identifikasi Masalah

Berdasarkan hasil kuesioner dan observasi yang telah dilakukan oleh peneliti. maka dapat disimpulkan masalah yang ada sebagai berikut:

- a. Kesulitan dalam mencari dan menawarkan jasa pengasuh yang dilakukan dengan cara tradisional.
- b. Membutuhkan waktu yang lama dalam mencari dan menawarkan jasa pengasuh yang dilakukan dengan cara tradisional.

2) Analisa Kebutuhan

Berdasarkan indentifikasi masalah yang ada maka dapat di definisikan kebutuhan-kebutuhan fungsional yang ada pada sistem pemesanan jasa pengasuh yang dibagi menjadi 3 bagian berdasarkan user yaitu sebagai berikut:

a. Aplikasi Jasa Pengasuh Sisi Pengguna (berbasis Android)

Analisa kebutuhan aplikasi jasa pengasuh pada sisi pengguna yaitu sistem dapat memproses login dan registrasi; sistem dapat memproses pengelolaan profil; sistem dapat menampilkan daftar pengasuh yang dibutuhkan; sistem dapat menampilkan data diri dari pengasuh; sistem dapat memproses pemesanan pengasuh; sistem dapat menampilkan informasi pesanan; sistem mampu menghitung waktu dan pembayaran pesanan.

b. Aplikasi Jasa Pengasuh Sisi Pengasuh (berbasis Android)

Analisa kebutuhan aplikasi jasa pengasuh pada sisi pengasuh yaitu sistem dapat memproses login; sistem dapat memproses pengelolaan profil; sistem menyediakan konfigurasi kerja status ketersediaan pengasuh;

sistem dapat menampilkan tawaran kerja yang masuk; sistem dapat menampilkan informasi pesanan; sistem mampu menghitung waktu dan pembayaran pesanan.

c. Aplikasi Admin Jasa Pengasuh (berbasis Web)

Sistem dapat memproses login; Sistem dapat memproses pengelolaan data pengasuh (penyedia jasa); Sistem dapat memproses pengelolaan data pengguna jasa; Sistem dapat memproses pengelolaan data pesanan.

B. Tahap Perancangan

Pada tahap Perancangan ini peneliti akan melakukan perancangan untuk pengembangan aplikasi sistem pemesanan jasa pengasuh sesuai dengan apa yang telah direncanakan pada tahap perencanaan (planning). Pada tahap ini perancangan dibagi menjadi 2 bagian yaitu Pemodelan Sistem dan Pemodelan Basis Data.

1) Pemodelan Sistem

Pemodelan ini memodelkan bagaimana sistem aplikasi bekerja dengan menggunakan Desain Arsitektur Sistem dan beberapa Unified Modeling Language (UML) yaitu Use Case Diagram dan Activity Diagram.

a. Desain Arsitektur Sistem

Desain arsitektur sistem dari sistem aplikasi pemesanan jasa pengasuh berbasis mobile menggambarkan sebuah gambaran umum sistem yang akan dibangun dimana aplikasi klien bagian pengelola dan aplikasi klien bagian pengguna yang ada terhubung dengan firebase. seperti pada Gambar 6 hanya 4 dari banyak fitur layanan firebase yang digunakan.

```
fun writeNewUser(userId: String, name: String, email: String) {
    val user = User(name, email)

    database.child("users").child(userId).setValue(user)
}
```

Gambar 1. Contoh metode menyimpan data ke basis data [10]

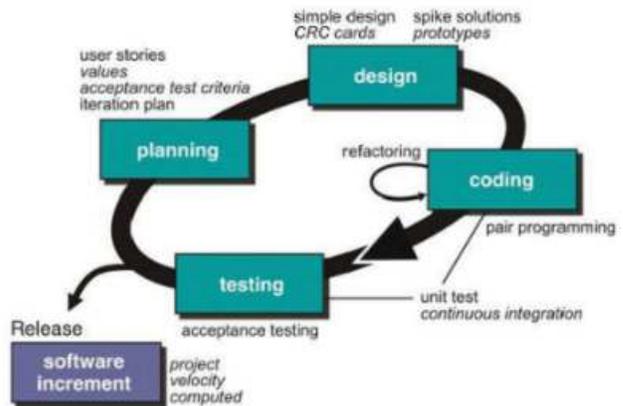
```
val postListener = object : ValueEventListener {
    override fun onDataChange(dataSnapshot: DataSnapshot) {
        // Get Post object and use the values to update the UI
        val post = dataSnapshot.getValue<Post>()
        // ...
    }

    override fun onCancelled(databaseError: DatabaseError) {
        // Getting Post failed, log a message
        Log.w(TAG, "loadPost:onCancelled", databaseError.toException())
    }
}
postReference.addValueEventListener(postListener)
```

Gambar 2. Contoh metode membaca data dari basis data [10]

```
{
  "users": {
    "alovelace": {
      "name": "Ada Lovelace",
      "contacts": { "ghopper": true },
    },
    "ghopper": { ... },
    "eclarke": { ... }
  }
}
```

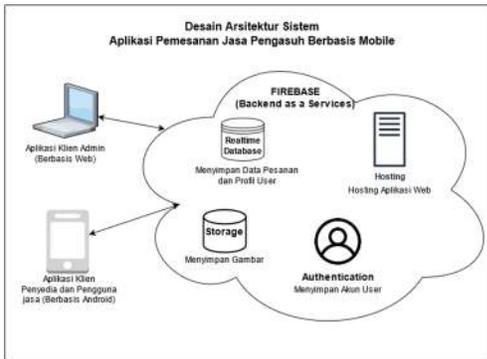
Gambar 3. Contoh struktur data firebase realtime database [10]



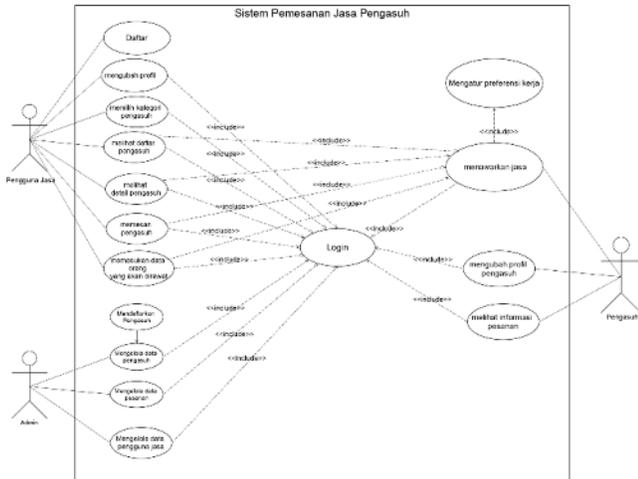
Gambar 4. Metode Extreme Programming sumber: (Ruhlessin dkk, 2020)[17]

TABEL I  
MODEL BASIS DATA NOSQL

Dokumen 1	Dokumen 2	Dokumen 3
{ "prop1":data, "prop2":data, "prop3":data, }	{ "prop1":data, "prop2":data, "prop3":data, }	{ "prop1":data, "prop2":data, "prop3":data, }



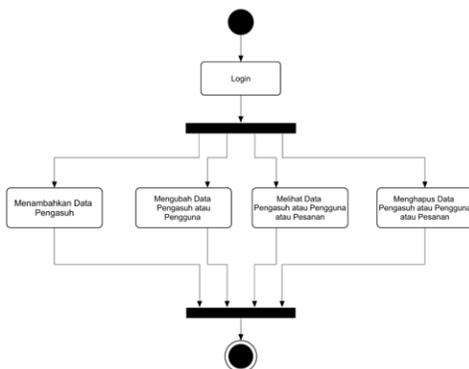
Gambar 6. Desain Arsitektur Sistem



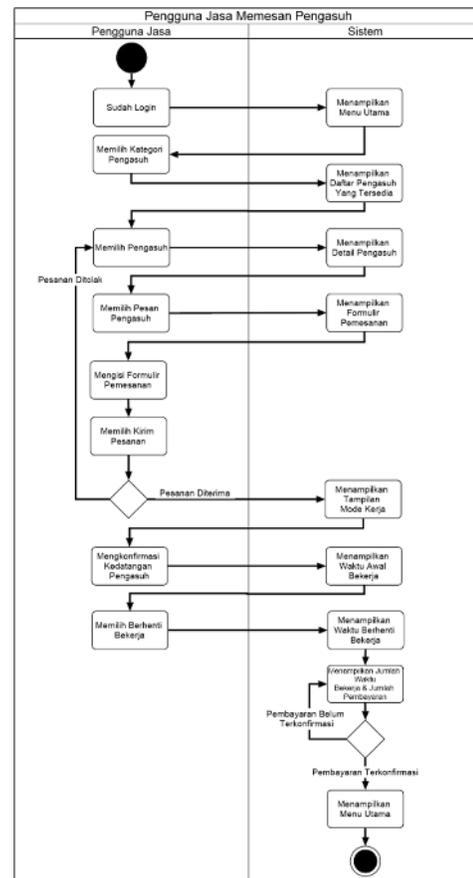
Gambar 7. Use Case Diagram Sistem Pemesanan Jasa Pengasuh

TABEL II  
DESKRIPSI AKTOR

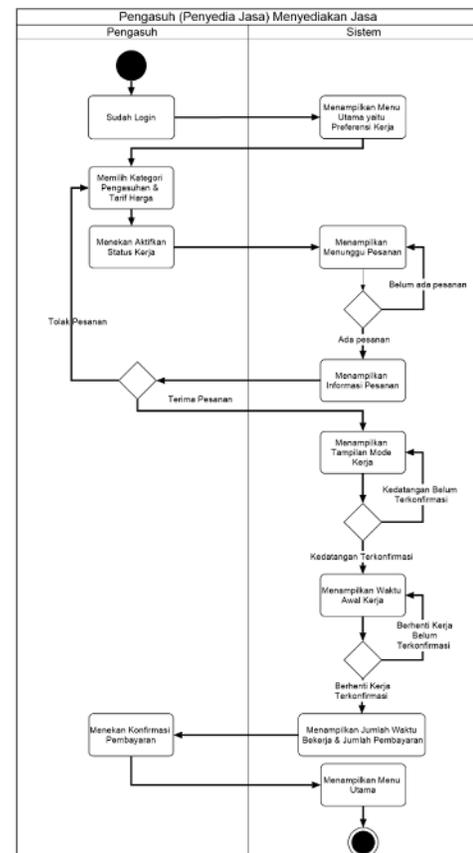
No	Aktor	Deskripsi
1.	Admin	Admin adalah orang yang bertugas dan memiliki hak akses untuk melakukan pekerjaan mengelola data pengasuh termasuk mendaftarkan pengasuh, mengelola data pesanan, mengelola data pelanggan.
2.	Pengguna Jasa	Pengguna Jasa adalah orang yang bisa memilih kategori pengasuh, bisa melihat daftar pengasuh yang tersedia, pesan pengasuh, dan yang mengisi data orang yang akan dirawat.
3.	Pengasuh (Penyedia Jasa)	Pengasuh (Penyedia Jasa) adalah orang yang bisa mengatur preferensi kerja dan menawarkan jasanya, juga bisa melihat informasi pesanan.



Gambar 8. Activity Diagram Admin Mengelola Sistem Pemesanan Jasa Pengasuh



Gambar 9. Activity Diagram Pengguna Jasa Memesan Pengasuh



Gambar 10. Activity Diagram Pengasuh (Penyedia Jasa) menyediakan Jasa

TABEL III  
PENDEFINISIAN DOMAIN DATA SISTEM PEMESANAN JASA PENGASUH

Attribute	Nama Domain	Keterangan	Pendefinisian Domain
<i>_id</i>	<i>Key id</i>	Kumpulan karakter unik untuk membedakan data dalam objek	Karakter unik berukuran 28 digit
<i>email</i>	Alamat <i>email</i>	Alamat <i>Email</i> dari pengguna	Karakter 20 digit, Memiliki simbol "@"
<i>nama</i>	Nama	Nama lengkap dari pengguna	Karakter 30 digit
<i>agama</i>	Agama	Agama dari pengguna	Karakter 10 digit
<i>alamat</i>	Alamat tinggal	Alamat lengkap dari pengguna	Karakter 100 digit
<i>available</i>	Ketersediaan	Indikator ketersediaan pengasuh	Objek
<i>balita</i>	Balita	Indikator ketersediaan pengasuh kategori balita	<i>Boolean</i>
<i>lansia</i>	Lansia	Indikator ketersediaan pengasuh kategori lansia	<i>Boolean</i>
<i>birth</i>	Tanggal lahir	Tanggal lahir dari pengguna	<i>Date</i> , dengan format yyyy-mm-dd
<i>foto</i>	Foto profil	Alamat <i>link</i> dari foto profil pengguna	Karakter 20 digit, berisi alamat <i>link</i> foto profil
<i>gender</i>	Jenis kelamin	Jenis kelamin dari pengguna	Bernilai "Laki-laki" atau "perempuan"
<i>nik</i>	Nomor induk kependudukan	NIK dari pengguna	Angka 16 digit
<i>pendidikan</i>	Pendidikan terakhir	Pendidikan terakhir dari pengguna yang adalah pengasuh	Karakter 10 digit
<i>phone</i>	Nomor telpon	Nomor telpon dari pengguna	Angka 15 digit
<i>resume</i>	Resume	Resume dari pengguna yang adalah pengasuh yang menceritakan kemampuan, kelebihan, pengalaman kerja.	Karakter 100 digit
<i>alamatDirawat</i>	Alamat orang yang dirawat	Alamat tinggal dari orang yang dirawat	Karakter 100 digit
<i>catatanDirawat</i>	Catatan orang yang dirawat	Catatan yang berisi informasi pribadi dari orang yang dirawat	Karakter 100 digit
<i>catatanKerja</i>	Catatan kerja/tugas	Catatan yang berisi informasi pekerjaan atau tugas dari pengasuh	Karakter 100 digit
<i>fotoDirawat</i>	Foto orang yang di rawat	Alamat <i>link</i> foto orang yang dirawat	Karakter 20 digit, berisi alamat <i>link</i> foto orang yang dirawat
<i>genderDirawat</i>	Jenis kelamin orang yang dirawat	Jenis kelamin dari orang yang dirawat	Bernilai "Laki-laki" atau "perempuan"
<i>idPengasuh</i>	<i>Key id</i> Pengasuh	<i>Key id</i> dari pengasuh	Karakter unik 28 digit
<i>idPengguna</i>	<i>Key id</i> Pengguna	<i>Key id</i> dari pengguna	Karakter unik 28 digit
<i>idPesanan</i>	<i>Key id</i> Pesanan	<i>Key id</i> dari pesanan	Karakter unik 28 digit
<i>kategori</i>	Kategori pengasuh	Kategori pengasuh pada pesanan	Bernilai "balita" atau "lansia"
<i>namaDirawat</i>	Nama orang yang dirawat	Nama lengkap dari orang yang dirawat	Karakter 30 digit
<i>namaPengasuh</i>	Nama pengasuh	Nama lengkap dari pengasuh	Karakter 30 digit
<i>namaPengguna</i>	Nama pengguna	Nama lengkap dari pengguna	Karakter 30 digit
<i>status</i>	status	Keterangan status dari pesanan	Bernilai "deal", "bekerja", "berhenti", "selesai"
<i>tarif</i>	tarif	Tarif perbulan pengasuh	Angka 10 digit
<i>timeDeal</i>	Waktu <i>deal</i>	Rekam waktu ketika pesanan diterima	<i>Date</i> , dengan format yyyy-mm-dd
<i>timeWork</i>	Waktu <i>work</i>	Rekam waktu ketika pengasuh bekerja	<i>Date</i> , dengan format yyyy-mm-dd
<i>timeDone</i>	Waktu <i>done</i>	Rekam waktu ketika pesanan selesai	<i>Date</i> , dengan format yyyy-mm-dd
<i>timePay</i>	Waktu <i>pay</i>	Rekam waktu Ketika pesanan telah dibayar	<i>Date</i> , dengan format yyyy-mm-dd
<i>totalPay</i>	Jumlah pembayaran	Jumlah pembayaran dari pesanan	Angka 10 digit
<i>umurDirawat</i>	Umur orang yang dirawat	Umur dari orang yang dirawat	Angka 3 digit
<i>workDuration</i>	Durasi kerja	Durasi kerja dari pesanan	Bernilai durasi dengan format, mm bulan dd hari hh jam mm menit ss detik
<i>buktiJudul</i>	Gambar bukti	Gambar yang dilampirkan bersama pengalaman atau kualifikasi	Karakter 20 digit, berisi alamat <i>link</i> gambar
<i>judulLampiran</i>	Judul lampiran	Pengalaman atau kualifikasi	Karakter 20 digit

#### a. Use case Diagram

*Use Case Diagram* digunakan untuk menggambarkan hubungan antara aktor dan sistem yang akan dibuat. Gambar 7 merupakan *Use Case Diagram* dari sistem pemesanan jasa pengasuh yang akan dibuat dan juga penjelasan tentang aktor yang terlibat dalam sistem pada *Use Case Diagram* terdapat pada Tabel I.

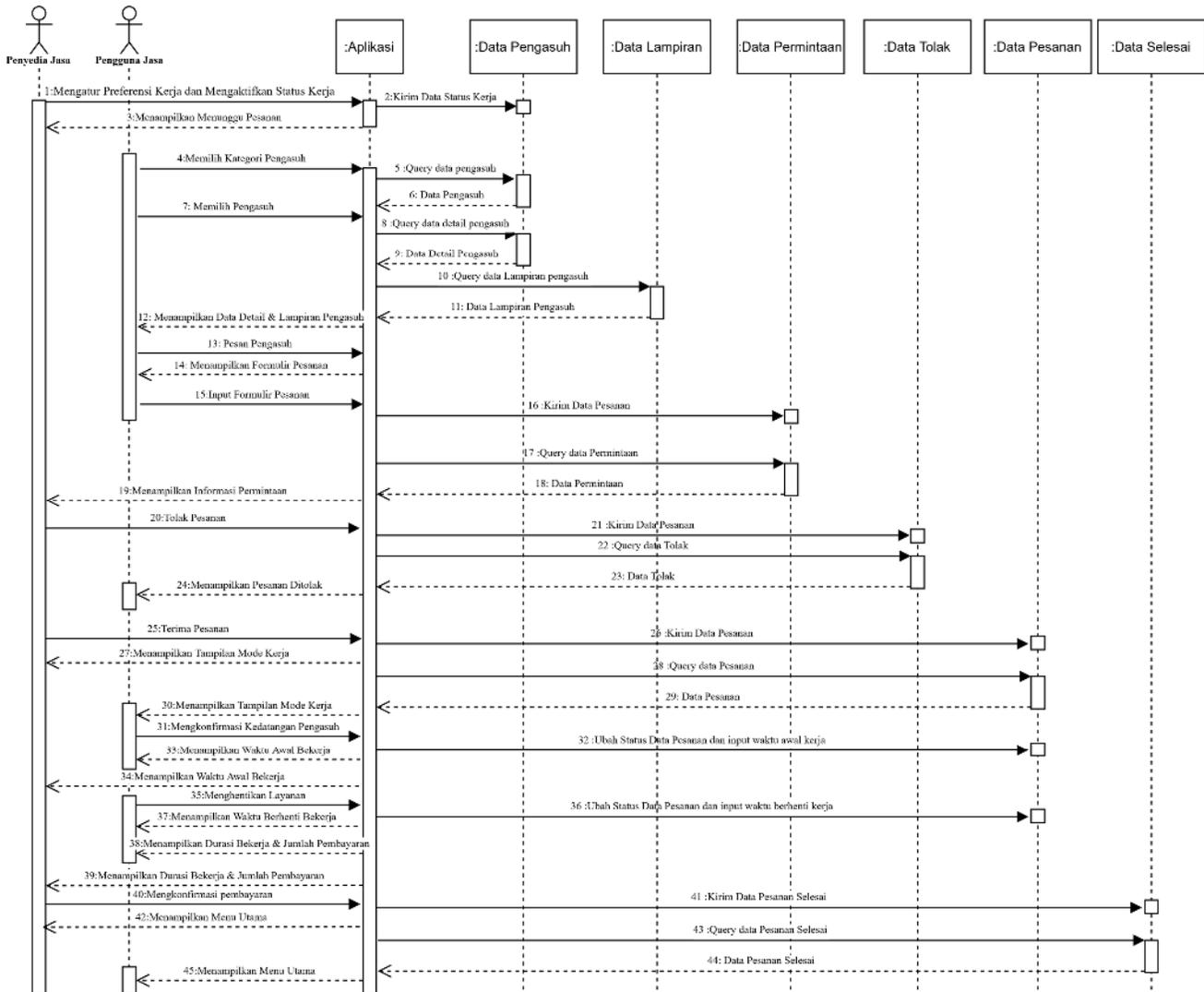
#### b. Activity Diagram

*Activity Diagram* digunakan untuk menggambarkan setiap aktivitas atau alur kerja dari sistem secara berurutan mulai dari awal sampai dengan akhir. Aktifitas Admin mengelola sistem pemesanan jasa pengasuh yang dimulai dari login. Lalu dapat mengelola data pengasuh, data pengguna, dan data pesanan seperti pada Gambar 8. Aktifitas Pengguna Jasa untuk memesan pengasuh yang dimulai dari login. Lalu memilih kategori pengasuh yang

menampilkan daftar pengasuh yang tersedia. Setelah pengguna memilih pengasuh dan memesannya pengguna harus mengisi formulir pesanan. Setelah itu menunggu konfirmasi pesanan jika diterima maka akan tampil mode kerja. Lalu ketika pengguna memulai dan menghentikan pekerjaan akan menampilkan rekam waktu, durasi dan jumlah pembayaran. Setelah pembayaran terkonfirmasi maka sistem menampilkan menu utama. Aktifitas ini terlihat pada Gambar 9. Aktifitas Pengasuh (Penyedia Jasa) untuk menyediakan jasanya yang dimulai dari login. Lalu mengatur preferensi kerja yaitu tarif dan kategori pengasuh. Lalu menyediakan jasa dengan cara mengaktifkan kerja. Setelah pesanan masuk sistem menampilkan informasi pesanan. Jika pengasuh menerima pesanan maka sistem menampilkan mode kerja.

TABEL IV  
 MODEL BASIS DATA SISTEM PEMESANAN JASA PENGASUH

user <<root>>	selesai <<root>>	lampiran <<root>>
<pre>{   "admin": [     {       "_id": ObjectID,       "email": String,       "nama": String     }   ]   "pengasuh": [     {       "_id": ObjectID,       "agama": String,       "alamat": String,       "available": [         {           "balita": Boolean           "lansia": Boolean           "tarif": int         }       ]       "birth": String,       "email": String,       "foto": String,       "gender": String,       "nama": String,       "nik": String,       "pendidikan": String,       "phone": String,       "resume": String     }   ]   "pengguna": [     {       "_id": ObjectID,       "agama": String,       "alamat": String,       "birth": String,       "email": String,       "foto": String,       "gender": String,       "nama": String,       "nik": String,       "phone": String     }   ] }</pre>	<pre>{   "_id(idPesanan)": ObjectID,   "alamatDirawat": String,   "catatanDirawat": String,   "catatanKerja": String,   "fotoDirawat": String,   "genderDirawat": String,   "idPengasuh": String,   "idPengguna": String,   "idPesanan": String,   "kategori": String,   "namaDirawat": String,   "namaPengasuh": String,   "namaPengguna": String,   "status": String,   "tarif": String,   "timeDeal": String,   "timeDone": String,   "timePay": String,   "timeWork": String,   "totalPay": String,   "umurDirawat": String,   "workDuration": String }</pre>	<pre>{   "_id(idPengasuh)": [     {       "_id": ObjectID,       "buktiLampiran": String,       "judulLampiran": String     }   ] }</pre>
pesanan <<root>>	permintaan <<root>>	ditolak <<root>>
<pre>{   "_id": ObjectID,   "alamatDirawat": String,   "catatanDirawat": String,   "catatanKerja": String,   "fotoDirawat": String,   "genderDirawat": String,   "idPengasuh": String,   "idPengguna": String,   "idPesanan": String,   "kategori": String,   "namaDirawat": String,   "namaPengasuh": String,   "namaPengguna": String,   "status": String,   "tarif": String,   "timeDeal": String,   "timeDone": String,   "timePay": String,   "timeWork": String,   "totalPay": String,   "umurDirawat": String,   "workDuration": String }</pre>	<pre>{   "_id(idPengasuh)": ObjectID,   "alamatDirawat": String,   "catatanDirawat": String,   "catatanKerja": String,   "fotoDirawat": String,   "genderDirawat": String,   "idPengasuh": String,   "idPengguna": String,   "idPesanan": String,   "kategori": String,   "namaDirawat": String,   "namaPengasuh": String,   "namaPengguna": String,   "status": String,   "tarif": String,   "timeDeal": String,   "timeDone": String,   "timePay": String,   "timeWork": String,   "totalPay": String,   "umurDirawat": String,   "workDuration": String }</pre>	<pre>{   "_id(idPengguna)": Boolean }</pre>



Gambar 11. Sequence Diagram Pemesanan Jasa Pengasuh

Lalu ketika terkonfirmasi mulai dan berhenti bekerja akan menampilkan rekam waktu, durasi dan jumlah pembayaran dan setelah menekan konfirmasi pembayaran maka sistem menampilkan menu utama. aktifitas ini terlihat pada Gambar 10.

2) Pemodelan Basis Data

Pemodelan ini memodelkan domain data dari aplikasi ini dapat dilihat pada Tabel III. Sementara struktur dari basis data sistem pemesanan jasa pengasuh yang merupakan basis data NoSQL yang dapat dilihat pada Tabel IV. Gambar 11 memperlihatkan interaksi antar objek yang ada yaitu 2 user, aplikasi yang adalah aplikasi jasa pengasuh dan 6 objek basis data yang ada sesuai pada Tabel III pada aktifitas pemesanan jasa pengasuh.

C. Tahap Penulisan Kode Program

Pada tahapan ini peneliti melakukan pembangunan aplikasi yang telah direncanakan dan dirancangan pada dua tahap sebelumnya.

1) Konfigurasi Firebase

Sistem pemesanan jasa pengasuh yang akan dibangun menggunakan layanan backend dari firebase. Setelah membuat

project baru pada Firebase maka peneliti akan menghubungkan setiap modul aplikasi yang akan di bangun pada sistem aplikasi pemesanan jasa pengasuh dalam satu project Firebase. Dengan menggunakan Firebase Authentication seluruh autentikasi pengguna pada satu project tersebut dilakukan dengan cara menggunakan metode email dan password saja. Pada Firebase Realtime Database dilakukan pengaturan keamanan. Juga pada Firebase Storage dilakukan pengaturan keamanan yang nantinya akan menyimpan gambar dari aplikasi. Sedangkan pada Firebase Hosting digunakan menyimpan data sumber dari sistem pengelolaan bagian web agar bisa di akses lewat internet.

2) Pengembangan Sistem Pemesanan Jasa Pengasuh

Pada bagian ini peneliti hanya membahas beberapa bagian penulisan kode program yang merupakan inti dari proses sistem pemesanan jasa pengasuh yang dibangun.

b. Aplikasi Jasa Pengasuh

Aplikasi ini berbasis mobile dengan system operasi Android. Aplikasi ini juga dibangun dengan IDE Androdi Studio. Peneliti menggunakan Kotlin sebagai bahasa pemrograman yang menggerakkan apikasi ini dan XML sebagai struktur penyusun tampilan antarmuka dari

aplikasi ini yang dapat dilihat pada Gambar 12. *User* pada aplikasi ini adalah pengguna jasa dan pengasuh (penyedia jasa).

#### b. Aplikasi Admin Jasa Pengasuh

Aplikasi ini adalah aplikasi yang dibangun berbasis *web*. Aplikasi ini juga dibangun dengan *IDE Visual Studio Code*. Peneliti menggunakan *Javascript* sebagai bahasa pemrograman yang menggerakkan logika aplikasi ini dan juga *HTML* dan *CSS* sebagai struktur penyusun tampilan antarmuka dari aplikasi ini yang dapat dilihat pada Gambar 13. *User* pada aplikasi ini adalah Admin.

#### D. Tahap Pengujian

Pada tahap ini peneliti akan melakukan pengujian atas sistem pemesanan jasa pengasuh yang telah dibangun pada tahap sebelumnya. pengujian ini dilakukan dengan metode *Black Box Testing* atau yang biasa disebut dengan *Functional Testing* yang akan menguji seluruh fungsi kerja aplikasi. Pengujian sistem pemesanan jasa pengasuh terdiri dari 3 bagian berdasarkan *user* dengan masing-masing kebutuhan fungsionalnya.

##### 1) Aplikasi Jasa Pengasuh Sisi Pengguna

Ada 8 fungsionalitas atau fitur pada sisi pengguna yang berjalan sesuai fungsinya yaitu pengguna dapat registrasi atau mendaftar dan juga *login* lewat aplikasi jasa pengasuh. Pengguna juga dapat mengelola profil dirinya seperti pada Gambar 14. Pengguna juga dapat memilih kategori pengasuh dan melihat daftar pengasuh yang dibutuhkan dengan cara menekan menu navigasi sesuai kategori yang dibutuhkan seperti terlihat pada Gambar 15. Kemudian pengguna melihat detail data diri pengasuh dengan cara menekan salah satu pengasuh (penyedia jasa) pada daftar pengasuh yang tersedia seperti terlihat pada Gambar 16. Ketika pengguna memesan pengasuh, pengguna melakukan proses pemesanan dengan cara mengisi formulir pesanan dengan memasukkan data orang yang akan dirawat dan menekan tombol “*kirim*” seperti terlihat pada Gambar 17. Setelah pengasuh menerima permintaan pesanan, sistem akan menampilkan informasi pesanan. Dan ketika tombol “*mulai bekerja*” ditekan akan menampilkan pemberitahuan “*mode kerja*” seperti terlihat pada Gambar 18. Setelah selesai sistem akan menampilkan hasil hitung waktu penggunaan jasa dan total harga yang harus dibayar setelah tombol “*berhenti kerja*” ditekan seperti terlihat pada Gambar 19.

##### 2) Aplikasi Jasa Pengasuh Sisi Pengasuh

Ada 5 fungsionalitas atau fitur pada sisi pengasuh yang berjalan sesuai fungsinya yaitu pengasuh dapat *login* lewat aplikasi jasa pengasuh. Pengasuh juga dapat mengelola profil dirinya. Yang didalamnya juga Pengasuh dapat menambahkan, menghapus, dan mengubah lampiran pengalaman yang termasuk dalam profilnya seperti terlihat pada Gambar 20. Pengasuh juga dapat menyediakan atau menawarkan jasanya dengan cara mengatur preferensi kerja. Yaitu tarif perbulan dari jasanya dan juga kategori jasa pengasuh yang dia tawarkan. Lalu menekan tombol “*aktifkan*” dengan pemberitahuan kalau jasa sedang tersedia seperti terlihat pada Gambar 21. Ketika ada pesanan sistem akan menampilkan informasi pesanan pada saat pengguna jasa melakukan pemesanan pada pengasuh tersebut

dengan pemberitahuan kalau ada tawaran pekerjaan yang kemudian akan di arahkan ke tampilan informasi tawaran pekerjaan seperti terlihat pada Gambar 22. Setelah itu sistem akan menampilkan informasi pesanan yang sedang berlangsung. setelah pengguna jasa mengaktifkan mode kerja, rekam waktu mulai kerja akan tampil seperti terlihat pada Gambar 23. Kemudian sistem akan menampilkan hasil hitung waktu penggunaan jasa dan total harga yang harus di bayar setelah pengguna jasa menghentikan penggunaan jasa seperti terlihat pada Gambar 24.

#### E. Aplikasi Admin Jasa Pengasuh

Ada 5 fungsionalitas atau fitur pada sisi admin yang berjalan sesuai fungsinya yaitu admin dapat *login* lewat modul bagian *web*. Setelah *login* admin dapat mengelola data pengasuh termasuk menambahkan atau mendaftarkan pengasuh. admin juga dapat mengelola data pengguna namun tidak dapat menambahkan atau mendaftarkan pengguna jasa, dan juga mengelola data pesanan yang sedang berlangsung maupun pesanan yang telah selesai seperti terlihat pada Gambar 25.

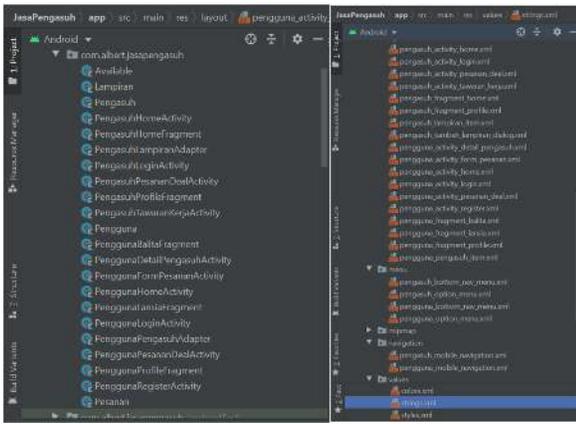
## IV. KESIMPULAN DAN SARAN

### A. Kesimpulan

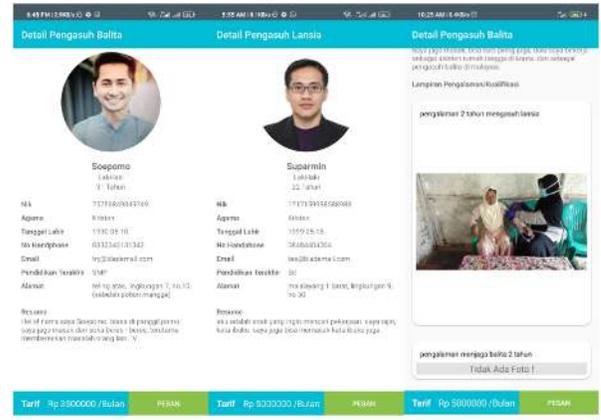
Berdasarkan hasil dari penelitian ini dapat disimpulkan beberapa kesimpulan yaitu penelitian ini telah menghasilkan aplikasi sistem pemesanan jasa pengasuh yang dirancang dan dibangun mengikuti alur metode pengembangan berdasarkan dengan analisa kebutuhan yang ada. Juga sistem aplikasi pemesanan jasa pengasuh yang dirancang dan dibangun telah menghasilkan sebuah sistem aplikasi yang saling terintegrasi dalam satu basis data. Dalam sistem aplikasi tersebut pengguna jasa dapat memesan jasa pengasuh menggunakan aplikasi jasa pengasuh. Sedangkan pengasuh (penyedia jasa) dapat menawarkan jasanya menggunakan aplikasi tersebut. Sedangkan pengelola dapat mengelola data pengasuh, data pengguna, dan data pesanan pada bagian *web*. Untuk dapat menggunakan aplikasi penyedia jasa pengasuh, pengasuh (penyedia jasa) harus didaftarkan/ditambahkan oleh admin (pengelola). Sedangkan pengguna jasa dapat melakukan registrasi dengan sendirinya pada aplikasi jasa pengasuh. Aplikasi yang telah dibangun telah melewati pengujian dengan metode pengujian *Black Box Testing* atau disebut juga dengan pengujian fungsionalitas. Dari Pengujian tersebut disimpulkan semua fungsionalitas berhasil berjalan sesuai fungsinya.

### B. Saran

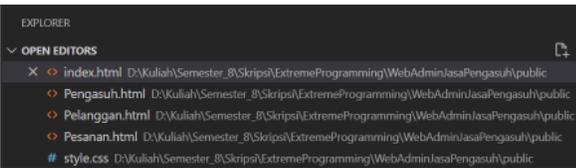
Adapun penelitian ini tidaklah lepas dari kelemahan dan kekurangan, dalam meningkatkan aplikasi ini untuk kedepannya peneliti mempunyai beberapa saran yaitu *user interface* pada aplikasi ini masih harus di tingkatkan lagi, juga menambahkan fitur *Chat* atau fitur untuk komunikasi antara pengguna jasa dengan penyedia jasa pada aplikasi ini, juga menambahkan fitur *push notification* pada aplikasi ini, aplikasi ini juga bisa dikembangkan keberbagai *platform* lain, aplikasi ini bisa dikembangkan menjadi aplikasi yang berbasis lokasi.



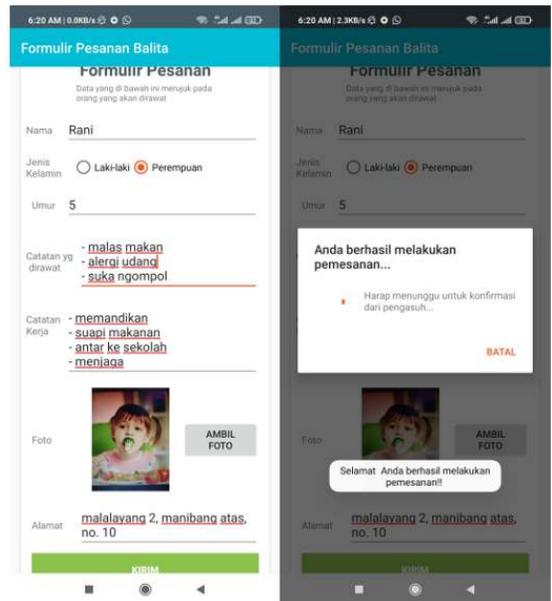
Gambar 12. Struktur Program Aplikasi Android Pemesanan Jasa Pengasuh



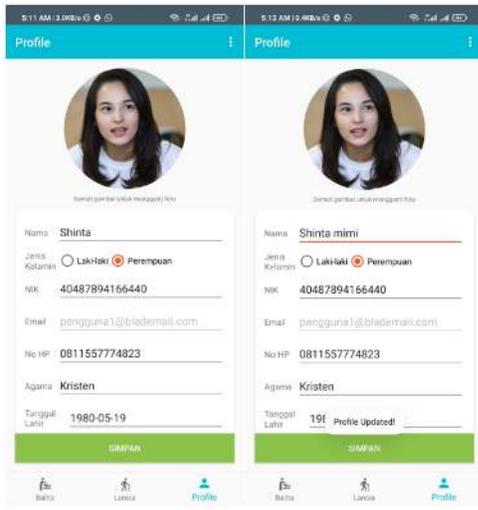
Gambar 16. Uji Tampil Detail Pengasuh Tersedia



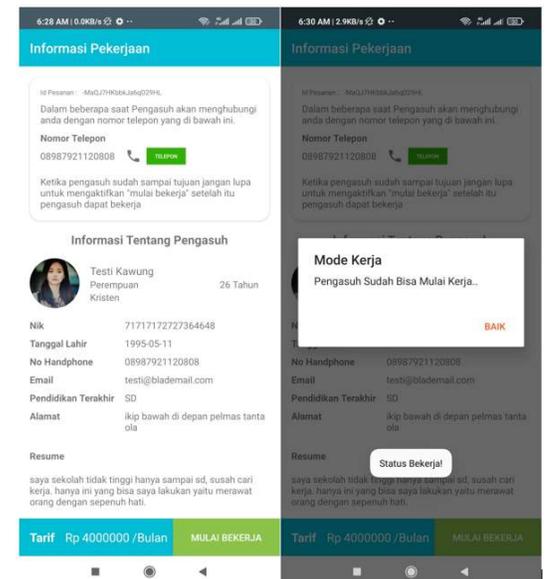
Gambar 13. Struktur Program Aplikasi Web Sistem Pemesanan Jasa Pengasuh untuk Admin (Pengelola)



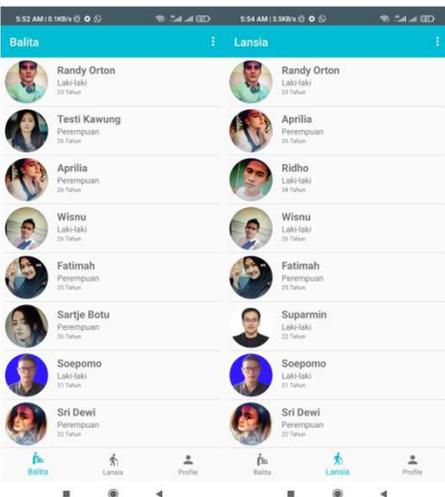
Gambar 17. Uji Tampil Formulir Pemesanan



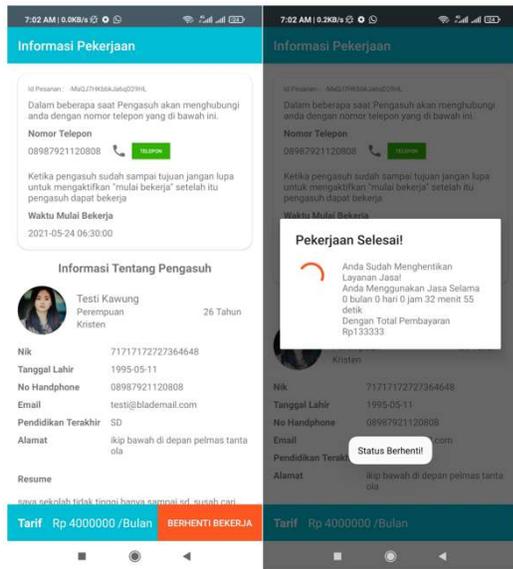
Gambar 14. Uji Profil Sisi Pengguna



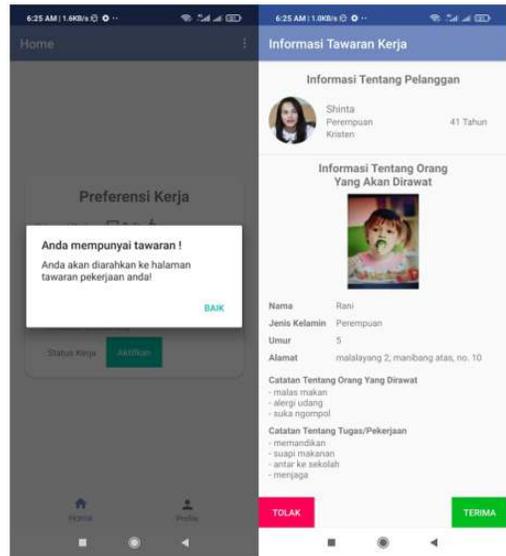
Gambar 18. Uji Tampil Informasi Pemesanan



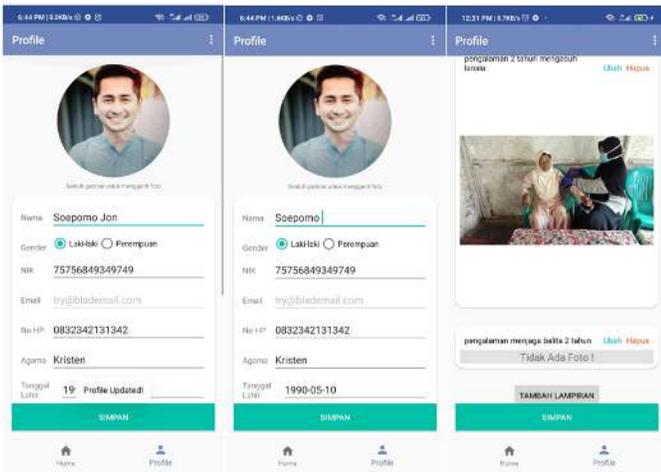
Gambar 15. Uji Tampil Daftar Pengasuh Tersedia



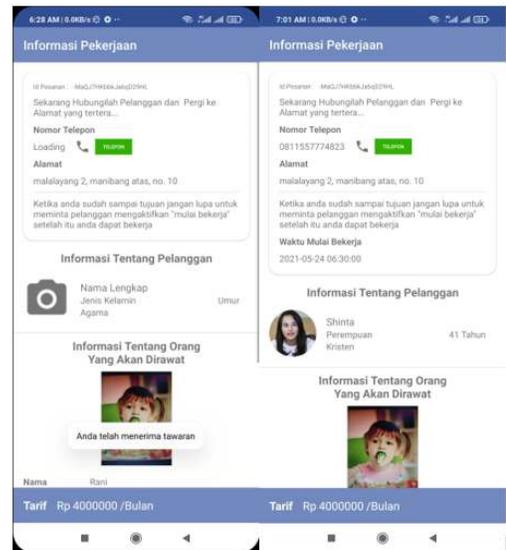
Gambar 19. Uji Tampil Pesanan Selesai



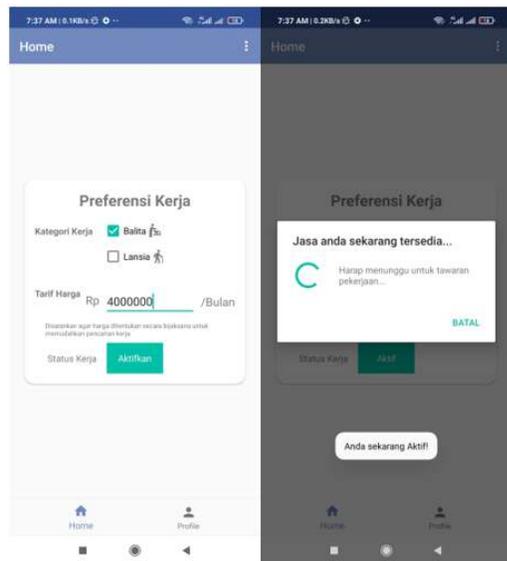
Gambar 22. Uji Tampil Tawaran Kerja



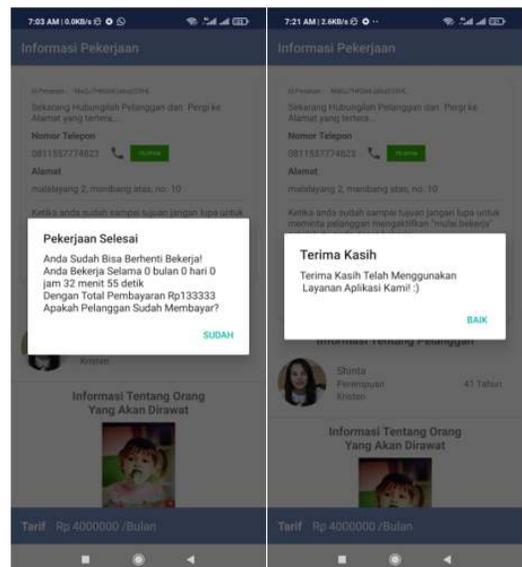
Gambar 20. Uji Profil Sisi Pengasuh



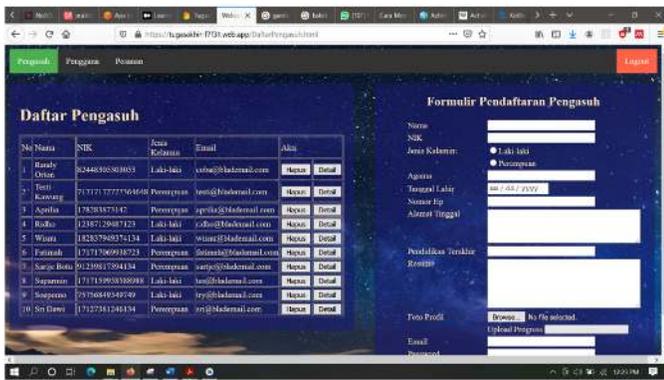
Gambar 23. Uji Tampil Informasi Kerja



Gambar 21. Uji Tampil Preferensi Kerja



Gambar 24. Uji Tampil Selesai Kerja



Gambar 25. Uji Tampilan Admin Mengelola

### V.KUTIPAN

- [1] A. Syaifulhalla, "66,3% masyarakat Indonesia Memiliki Smartphone," *Indonesiabaik.id*, 2018. <http://indonesiabaik.id/infografis/663-masyarakat-indonesia-memiliki-smartphone-8> (accessed Nov. 10, 2020).
- [2] J. W. Janis, D. J. Mamahit, B. A. Sugiarto, and A. M. Rumagit, "Rancang Bangun Aplikasi Online Sistem Pemesanan Jasa Tukang Bangunan Berbasis Lokasi," *J. Tek. Inform.*, vol. 15, no. 1, pp. 1–12, 2020, doi: 10.35793/jti.15.1.2020.29023.
- [3] N. Resna and R. Utari, "Memahami Perbedaan Caregiver dan Perawat, Berikut Penjelasan Lengkapnya," *sehatq.com*, 2020. <https://www.sehatq.com/artikel/memahami-caregiver-adalah-dan-perbedaannya-dengan-perawat> (accessed Nov. 10, 2020).
- [4] K. J. Baso, Y. D. Y. Rindengan, and R. Sengkey, "Perancangan Aplikasi Catering Berbasis Mobile," *J. Tek. Elektro dan Komput.*, vol. 9, no. 2, pp. 81–90, 2020, doi: 10.31962/jiitr.v1i2.22.
- [5] H. N. Kai, S. R. U. A. Sompie, and A. M. Sambul, "Aplikasi Layanan Pengangkutan Sampah Berbasis Android," *J. Tek. Inform.*, vol. 13, no. 4, pp. 1–12, 2018, doi: 10.35793/jti.13.4.2018.28088.
- [6] R. T. Tjoetama, S. R. U. A. Sompie, and X. B. N. Najoran, "Aplikasi Informasi Harga Material Bangunan Berbasis Android," *J. Tek. Inform.*, vol. 13, no. 2, 2018, doi: 10.35793/jti.13.2.2018.22496.
- [7] N. S. Sibarani, G. Munawar, and B. Wisnuadhi, "Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin," *9th Ind. Res. Work. National Semin.*, no. Juli, pp. 319–324, 2018.
- [8] A. Prasetya, A. H. Brata, and M. T. Ananta, "Pengembangan Aplikasi Pemesanan Lapangan Futsal Di Kota Malang Berbasis Android Menggunakan Metode Pengembangan Extreme Programming (Studi Kasus Champion Tidar, Zona SM Futsal, dan Viva Futsal)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 12, pp. 7293–7301, 2018.
- [9] G. C. E. Sumera, S. R. U. A. Sompie, and D. J. Mamahit, "Aplikasi Pelaporan Pelanggaran dan Pelayanan Publik di Lingkungan Universitas Sam Ratulangi Berbasis Mobile," *J. Tek. Inform.*, vol. 15, no. 2, pp. 145–154, 2020, doi: 10.35793/jti.15.2.2020.29535.
- [10] Firebase, "Firebase Realtime Database Documentation," *firebase.google.com*. <https://firebase.google.com/docs/database/> (accessed Jul. 14, 2021).
- [11] I. Carolina, A. M. H. Pardede, and A. Supriyatna, "Penerapan Metode Extreme Programming Dalam Perancangan Aplikasi Perhitungan Kuota Sks Mengajar Dosen," *J. IKRA-ITH Inform.*, vol. 3, no. 1, pp. 106–113, 2019, doi: 10.31227/osf.io/se6f9.
- [12] Febri Nur Rohman, "Pengembangan Aplikasi Web Pengolah Data Nilai Lomba Baris Berbaris Menggunakan Metodologi Extreme Programming," 2015.
- [13] MongoDB, "NoSQL Database Design & Data Modeling," *mongodb.com*. <https://www.mongodb.com/nosql-explained/data-modeling> (accessed Jul. 13, 2021).
- [14] M. Syani and N. Werstantia, "Perancangan Aplikasi Pemesanan Catering Berbasis Mobile Android," *J. Ilm. Ilmu dan Teknol. Rekayasa*, vol. 2, no. 1, pp. 109–117, 2018, doi: 10.31962/jiitr.v1i2.22.
- [15] K. Wong, A. A. E. Sinsuw, and X. B. N. Najoran, "Rancang Bangun Aplikasi 'Trip Alone?' Berbasis Android," *J. Tek. Inform.*, vol. 9, no. 1, 2016, doi: 10.35793/jti.9.1.2016.14809.
- [16] J. F. Tompoh, S. R. Sentinuwo, and A. A. E. Sinsuw, "Rancang Bangun Aplikasi Pemesanan Menu Makanan Restoran Berbasis Android," *J. Tek. Inform.*, vol. 9, no. 1, pp. 1–9, 2016, doi: 10.35793/jti.9.1.2016.13749.
- [17] C. B. Ruhulessin and X. B. N. Najoran, "Aplikasi Layanan Informasi Tempat Wisata di Kota Ambon Berbasis Lokasi," *J. Tek. Inform.*, vol. 15, no. 2, pp. 83–92, 2020, doi: 10.35793/jti.15.2.2020.29511.



**Albert Christian Tjakra**, lahir di Tomohon, 21 April 1999 merupakan anak yang sulung dari 4 bersaudara. Penulis menempuh pendidikan taman kanak-kanak di TK Katolik Sta. Theresia Manado (2004–2005), selanjutnya penulis melanjutkan ke pendidikan sekolah dasar di SD Katolik Sta. Theresia 01 Manado (2005–2011), kemudian penulis melanjutkan pendidikan sekolah menengah pertama di SMP Katolik Pax Christi Manado (2011–2014), setelah itu penulis melanjutkan pendidikan sekolah menengah atas di SMA Negeri 1 Manado (2014–2017). Setelah lulus SMA pada Tahun 2017 penulis melanjutkan pendidikan S1 di Program Studi Informatik, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Sam ratulangi Manado.