

**SISTEM REKOMENDASI JASA TUKANG DENGAN METODE  
COLLABORATIVE FILTERING DAN ALGORITMA SLOPE ONE  
BERBASIS ANDROID (STUDI KASUS : DESA SINDANG SARI)**

*Recomendation System Jasa Tukang with Collaborative Filtering Method And  
Slope one Algorithm Based On Android (Study Case : Sindang Sari Village)*

Skripsi

Untuk memenuhi sebagian persyaratan  
Mencapai derajat sarjana S-1

Diajukan Oleh :  
**FERRY FERNANDO**  
17312021

Aa Renni  
12 Juni 2024

*[Handwritten signature]*  
Yuri P. Hment



All. Adh  
14/6 - 2024  
*[Handwritten signature]*

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK ILMU KOMPUTER  
UNIVERSITAS TEKNOKRAT INDONESIA  
BANDAR LAMPUNG  
2024**

**LEMBAR PENGESAHAN  
SKRIPSI**

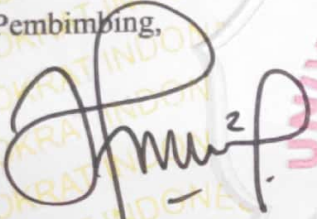
**SISTEM REKOMENDASI JASA TUKANG DENGAN METODE  
COLLABORATIVE FILTERING DAN ALGORITMA SLOPE ONE BERBASIS  
ANDROID (STUDI KASUS : DESA SINDANG SARI)**

Dipersiapkan dan disusun oleh :

**FERRY FERNANDO**  
**17312021**

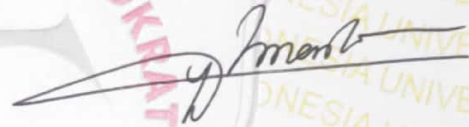
Telah dipertahankan di depan Dewan Penguji  
Pada tanggal 6 Juni 2024

Pembimbing,



**Dyah Ayu Megawaty, M.Kom.**  
NIK. 022 09 03 05

Penguji,

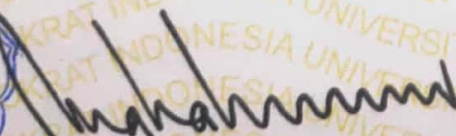


**Yuri Rahmanto, M.Kom.**  
NIK. 022 13 02 04

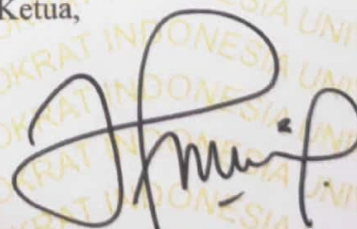
Skripsi ini telah diterima sebagai salah satu persyaratan  
Untuk memperoleh gelar sarjana  
Tanggal 12 Juni 2024

Fakultas Teknik dan Ilmu Komputer  
Dekan,

Program Studi S1 Informatika  
Ketua,



**Dr. H. Mahathir Muhammad, SE., MM.**  
NIK. 023 05 00 09



**Dyah Ayu Megawaty, M.Kom.**  
NIK. 022 09 03 05

## LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini:

Nama : Ferry Fernando

NPM : 17312021

Program Studi : Informatika

Dengan ini menyatakan bahwa tugas akhir :

Judul : Sistem Rekomendasi Jasa Tukang dengan Metode Collaborative Filtering dan Algoritma Slope One berbasis Android (Studi kasus : Desa Sindang Sari)

Pembimbing : Dyah Ayu Megawaty, M.Kom.

Belum pernah diajukan untuk diuji sebagai persyaratan untuk memperoleh gelar akademik pada berbagai tingkatan di Universitas atau perguruan tinggi manapun. Tidak ada bagian dalam skripsi ini yang pernah dipublikasikan oleh pihak lain, kecuali bagian yang digunakan sebagai referensi berdasarkan kaidah penulisan ilmiah yang benar.

Apabila dikemudian hari ternyata laporan tugas akhir yang saya tulis terbukti hasil saduran/plagiat, maka saya akan bersedia menanggung segala resiko yang akan saya terima

Demikian pernyataan ini dibuat dengan sebenar-benarnya.

Bandar Lampung, 21 Mei 2024



Ferry Fernando

NPM 17312021

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI  
UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Fakultas Teknik dan Ilmu Komputer Universitas Teknokrat Indonesia, Saya yang beryanda tangan dibawah ini :

Nama : Ferry Fernando

NPM : 17312021

Program Studi : Informatika

Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia, **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*)** atas karya ilmiah yang berjudul :

**“Sistem Rekomendasi Jasa Tukang dengan Metode *Collaborative Filtering* dan Algoritma *Slope One* berbasis Android (Studi kasus : desa Sindangsari)”**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia berhak menyimpan, mengalih media/formatkan, menggunakan asset, merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Bandar Lampung

Pada tanggal : 21 Mei 2024

Yang menyatakan,



Ferry Fernando  
17312021

## HALAMAN PERSEMBAHAN

Dengan mengucapkan syukur kepada Allah SWT atas semua karunia, berkah, rahmat dan nikmatnya. Serta keberhasilan ini merupakan hasil dari kerja keras, kesabaran dan doa dari orang-orang yang berada disekeliling saya dengan ketulusan dan keikhlasan. Sebagai ungkapan rasa sayang kepada mereka semua, maka saya persembahkan sebuah karya ini kepada :

1. Kedua orang tua saya yang tercinta, Bapak dan Ibu saya yang telah merawat dan mendidik dengan penuh kasih sayangnya. Bapak Tamsir dan Ibu Yeni Kusyanti, saya sangat mencintai kalian.
2. Sahabat terbaik saya, Agung Prayuda, S.Kom. yang telah senantiasa menemani saya dalam keadaan apapun bahkan saat halaman ini di ketik.
3. Wanita yang sangat saya cintai setelah ibu saya, Novitasari Purwanti, A.Md.Pi untuk motivasi dan semangat yang tak habis-habisnya untuk saya.
4. Adi, Karim, Dais, Dema, Wahyu pejuang skripsi yang tiada henti henti nya untuk mencari ACC dosen bersama saya.
5. Ilham Solehudin, S.Kom. atas bantuan aplikasi dikala saya sibuk dengan pekerjaan.
6. Rahma Aulia Putri, Irenne Sukardi. Sahabat serta saudara saya yang menemani malam revisi.
7. Bu Dwi yang selalu memberikan motivasi serta hidangan di dapur nya.
8. Teman-teman seperjuangan yang saling mendukung satu sama lain.

**Halaman Moto**

*“Selesaikan apa yang telah kamu mulai”*

*-Ferry Fernando*

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT, karena atas berkat rahmat-Nya, penulis dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar sarjana pada Program Studi S1 Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia. Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan laporan ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

- 1 Bapak Dr. H. M. Nasrullah Yusuf, S.E., M.B.A., selaku Rektor Universitas Teknokrat Indonesia
- 2 Bapak H. Mahathir Muhammad, S.E., M.M., selaku Dekan Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia.
- 3 Ibu Dyah Ayu Megawaty, S.Kom., M.Kom. selaku pembimbing yang telah meluangkan waktu untuk membimbing penulis menyelesaikan skripsi ini.
- 4 Bapak Yuri Rahmanto, S.Kom., M.Kom. selaku dosen penguji.

Akhir kata, penulis berharap semoga Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu dan semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Bandar Lampung, 21 Mei 2024

Penulis

## DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN .....	ii
LEMBAR PERNYATAAN .....	iii
HALAMAN PERSETUJUAN PUBLIKASI.....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTO .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR .....	xii
DAFTAR LAMPIRAN.....	xiii
ABSTRAK .....	xiv
1. BAB I.....	1
PENDAHULUAN .....	1
1.1. Latar Belakang .....	3
1.2. Rumusan Masalah .....	3
1.3. Batasan Masalah.....	4
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	4
2. BAB II.....	5
LANDASAN TEORI.....	5
2.1. Tinjauan Pustaka .....	5
2.1.1. Tinjauan Literatur 1 .....	5
2.1.2. Tinjauan Literatur 2 .....	6
2.1.3. Tinjauan Literatur 3 .....	6
2.1.4. Tinjauan Literatur 4.....	7
2.1.5. Tinjauan Literatur 5 .....	7
2.1.6. Kesimpulan Tinjauan Literatur.....	8
2.2. Android.....	9
2.3. Android Studio .....	9
2.4. Algoritma .....	9
2.5. Sistem Rekomendasi .....	10
2.6. <i>Collaborative Filtering</i> .....	10
2.7. Algoritma <i>Slope One</i> .....	12
2.8. Aplikasi .....	13
2.10. <i>Firebase</i> .....	14
2.11. <i>Tukang</i> .....	14
2.12. <i>Bangunan</i> .....	15
2.13. <i>Kuli</i> .....	15
2.14. <i>Extreme Programming</i> .....	15
2.15. <i>Java</i> .....	17
2.16. <i>Mobile Application</i> .....	17



2.17. Android SDK.....	17
2.18. IDE ( <i>intergrated development environment</i> ) .....	18
3. BAB III .....	19
METODE PENELITIAN.....	19
3.1. Kerangka Penelitian .....	20
3.1.1. Pengumpulan Data.....	20
3.1.1.1. Wawancara.....	20
3.1.1.2. Tinjauan Pustaka.....	20
3.1.2. Masalah.....	21
3.1.3. Pendekatan .....	21
3.1.4. <i>Planning</i> .....	21
3.1.4.1. <i>User Stories</i> .....	21
3.1.4.2. <i>Values</i> .....	22
3.1.4.3. <i>Accaptante Test Criteria</i> .....	23
3.1.4.4. <i>Iteration Plan</i> .....	24
3.1.5. <i>Design</i> .....	24
3.1.5.1. <i>CRC Cards</i> .....	24
3.1.5.2. Perancangan Rekomendasi .....	25
3.1.5.3. <i>Prototype</i> .....	27
3.1.6. <i>Coding</i> .....	45
3.1.7. <i>Testing</i> .....	45
3.1.8. <i>Release</i> .....	45
3.2. Jadwal Penelitian .....	45
4. BAB IV .....	47
4.1. Implementasi Pengkodean.....	47
4.1.1. Implementasi Koding Tampilan .....	49
4.1.2. Implementasi Koding Fitur Masuk dan Daftar .....	49
4.1.3. Implementasi Koding Kelengkapan Akun.....	50
4.1.4. Implementasi Koding Menu Utama.....	51
4.1.5. Implementasi Koding buat pekerjaan .....	53
4.1.6. Implementasi Koding Cari Tukang.....	53
4.1.7. Implementasi Koding Konfirmasi Pesanan .....	53
4.1.8. Implementasi Algoritma <i>Slope One</i> .....	55
5. BAB V.....	56
5.1. Hasil Pengujian .....	56
5.1.1. Pengujian Ketepatan .....	56
5.1.1.1. Pengujian Ketepatan Data Pada Fitur Pesanan Tukang .....	56
5.1.1.2. Pengujian Ketepatan Data Pada Pesanan Jasa Tukang.....	58
5.1.1.3. Pengujian Ketepatan Data Pada Pesanan jasa pada tukang.....	60
5.1.2. Pegujian Algoritma .....	62
5.1.3. Pengujian Sistem Aplikasi .....	67
5.1.3.1. Pengujian <i>Functional Suitability</i> .....	67
5.1.3.2. Pengujian <i>Usability</i> .....	70

6. BAB VI .....	71
6.1. Kesimpulan.....	71
6.2. Saran.....	71
DAFTAR PUSTAKA .....	72
LAMPIRAN.....	73

## DAFTAR TABEL

Tabel 3.1 Kerangka Penelitian .....	25
Tabel 3.2 Spesifikasi <i>Hardware</i> Pengembang .....	29
Tabel 3.3 Spesifikasi <i>Software</i> Pengembang .....	23
Tabel 3.4 Tabel Pertanyaan fungsional .....	30
Tabel 3.5 Tabel Pertanyaan kemudahan pengguna ( <i>usability</i> ) .....	32
Tabel 3.6 Jadwal Penelitian.....	45
Tabel 5.1 data pengujian fitur pesan tukang .....	56
Tabel 5.2 data pengujian gambar fitur pesan tukang .....	59
Tabel 5.3 hasil pengujian fitur pesan tukang .....	58
Tabel 5.4 atribut data untuk di tampilkan pada pesanan jasa tukang.....	58
Tabel 5.5 hasil pengujian fitur pesanan jasa tukang .....	60
Tabel 5.6 data pengujian pesanan jasa pada tukang.....	61
Tabel 5.7 rating enam tukang oleh enam orang pengguna.....	63
Tabel 5.8 Nilai rata-rata selisih rating antar tukang.....	64
Tabel 5.9 Nilai prediksi.....	65
Tabel 5.10 Bobot jawaban functional .....	67
Tabel 5.11 Hasil Pengujian .....	67
Tabel 5.12 Kriteria hasil uji .....	69
Tabel 5.13 Hasil Pengujian iso 25010 pada kriteria <i>functional suitability</i> .....	69
Tabel 5.14 Hasil Pengujian Usability.....	70
Tabel 5.15 Hasil Pengukuran Persentase .....	72
Tabel 5.16 Hasil Pengujian ISO 25010 pada kriteria <i>Usability</i> .....	72

## DAFTAR GAMBAR

Gambar 2.1 Logo Android .....	7
Gambar 2.2 Logo Android Studio.....	8
Gambar 2.3 Rekomendasi Youtube .....	9
Gambar 2.4 Proses <i>Collaborative Filtering</i> .....	10
Gambar 2.5 Skema <i>item-based</i> .....	11
Gambar 2.6 <i>Pseudocode slope one</i> .....	13
Gambar 2.7 <i>Extreme Programming</i> .....	15
Gambar 3.1 Kerangka Penelitian .....	19
Gambar 3.2. <i>CRC cards</i> .....	24
Gambar 3.3 Alur proses rekomendasi .....	25
Gambar 3.4 Skema Algoritama Slope One .....	26
Gambar 3.5 Tampilan Awal .....	27
Gambar 3.6 login-register .....	28
Gambar 3.7 Menu utama user dan pilih tukang .....	29
Gambar 3.8 Tampilan buat pesanan .....	30
Gambar 3.9 Pesanan aktif dan Rating .....	31
Gambar 3.10 Menu utama tukang dan pilih keahlian .....	32
Gambar 3.11 Lihat pemesanan.....	33
Gambar 4.1 Pembuatan project pada android studio .....	47
Gambar 4.2 Struktur Project android studi .....	48
Gambar 4.3 <i>dependecies</i> Aplikasi TuLen .....	48
Gambar 4.4 Penulisan kode tampilan.....	49
Gambar 4.5 Koding fitur masuk.....	50
Gambar 4.6 Koding fiur daftar.....	50
Gambar 4.7 Koding kelengkapan informasi untuk user .....	51
Gambar 4.8 Koding kelengkapan informasi untuk tukang .....	51
Gambar 4.9 Koding Menu utama user .....	52
Gambar 4.10 Koding Menu Utama Tukang.....	52
Gambar 4.12 Koding buat pekerjaan .....	53
Gambar 4.14 Koding cari tukang .....	55
Gambar 4.15 Koding Konfirmasi Pesanan.....	54
Gambar 4.16 Implementasi Algoritma.....	55
Gambar 5.1 Pengujian pada aplikasi .....	57
Gambar 5.2 hasil pengujian pada database .....	58
Gambar 5.7 Hasil Implementasi algoritma pada aplikasi .....	66

## DAFTAR LAMPIRAN

Lampiran 1 Surat izin penelitian .....	74
Lampiran 2 Dokumentasi wawancara .....	75
Lampiran 3 Rekaman Wawancara .....	76
Lampiran 4 Pengujian Functionality .....	78
Lampiran 5 Pengujian Usability.....	80

## ABSTRAK

### **SISTEM REKOMENDASI JASA TUKANG DENGAN METODE *COLLABORATIVE FILTERING* DAN ALGORITMA *SLOPE ONE* BERBASIS ANDROID (STUDI KASUS : DESA SINDANG SARI)**

*Recomendation System Jasa Tukang with Collaborative Filtering Method And Slope one Algorithm Based On Android (Study Case : Sindang Sari Village)*

**Oleh :  
Ferry Fernando  
17312021**

Penggunaan teknologi aplikasi semakin penting dalam membantu dan mempermudah pekerjaan manusia sehari-hari, termasuk dalam bidang pemesanan jasa. Masih banyak masyarakat yang kesulitan mencari tukang untuk berbagai pekerjaan seperti membangun atau memperbaiki rumah. Penelitian ini mengembangkan sistem rekomendasi jasa tukang berbasis Android menggunakan metode *Collaborative Filtering* dan algoritma *Slope One*, dengan studi kasus di Desa Sindang Sari.

Aplikasi ini dirancang untuk memberikan rekomendasi tukang dengan kebutuhan pengguna berdasarkan rating dan keahlian tukang. Namun rekomendasi yang diberikan kurang efektif karena cara kerja algoritma *slope one* yang hanya memprediksi rating dari tukang yang belum pernah dirating oleh *user*. Metode pengembangan yang digunakan adalah *Extreme Programming* (XP), meliputi tahap perencanaan, perancangan, pengkodean, pengujian, dan rilis.

Hasil penelitian menunjukkan bahwa aplikasi ini mempermudah pengguna dalam menemukan tukang yang tepat dan membantu tukang mendapatkan pekerjaan yang sesuai dengan keahlian mereka. Pengujian *functional* dan *usability* menunjukkan hasil 98,43% yang berarti “Sangat Layak” serta memiliki antarmuka yang mudah digunakan oleh masyarakat Desa Sindang Sari.

Kata kunci: Sistem rekomendasi, *Collaborative Filtering*, *Slope One*, Android, *Extreme Programming*, Jasa Tukang.

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

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 market store di berbagai bidang begitupun dalam bidang pemesanan jasa. Tidak bisa dipungkiri sudah terdapat banyak sekali aplikasi yang sudah dikembangkan 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 kemampuan sendiri seperti dalam hal membangun rumah, gedung, perbaikan rumah dan lain sebagainya (Janis, 2020). Untuk mengatasi hal tersebut dibutuhkan keahlian dan keterampilan khusus dalam melakukan pekerjaan tersebut yaitu seseorang pekerja yang memiliki jasa dalam pembangunan dan perbaikan.

Tukang bangunan adalah suatu pekerjaan yang sering dicari oleh masyarakat untuk memenuhi pembuatan atau memperbaiki suatu hal yang berkaitan dengan bangunan (Wicaksono, 2019). Namun, proses bisnis yang dijalankan masih menggunakan sistem manual dengan cara mencari informasi tukang bangunan melalui *website* atau informasi dari mulut ke mulut sehingga masyarakat susah untuk dapat memenuhi kebutuhan perbaikan rumah yang sesuai dengan masalah yang terjadi

Selain itu berdasarkan hasil wawancara yang telah dilakukan oleh (Nafis, 2019) dengan judul aplikasi pemesanan jasa tukang dengan teknologi *web hook*, dapat disimpulkan bahwa kesulitan yang dialami bukan hanya dari pengguna saja

akan tetapi penyedia jasa juga memiliki kesulitan, salah satunya adalah kesulitan mendapatkan pesanan pekerjaan yang sesuai keahlian tukang.

Untuk itu diperlukan suatu rekomendasi yang dapat menangani masalah tersebut. Sistem rekomendasi adalah suatu sistem yang menyarankan informasi yang berguna atau menyarankan apa yang akan dipilih oleh pengguna jasa. Misalnya seperti memilih tukang tertentu. Sehingga pengguna jasa bisa memilih tukang yang tepat untuk melakukan pekerjaannya.

*Collaborative filtering* merupakan salah satu algoritma yang digunakan untuk menyusun *recommender system* dan telah terbukti memberikan hasil yang sangat baik (A Susanto, 2020). Rating produk merupakan elemen terpenting dari algoritma ini, rating diperoleh dari sebagian besar customer di mana customer secara explicit memberikan penilaiannya terhadap produk. Dari penelitian yang telah dilakukan oleh (Zahrotun, 2019) dengan judul penelitian Sistem perekomendasi dengan metode *item-based collaborative filtering* pada aplikasi *mobile* untuk wisata kuliner, dalam penelitian ini disebutkan bahwa *user-based* memanfaatkan histori pilihan *user*, metode ini mempunyai kelemahan yaitu seiring dengan bertambahnya *user* dan *item* maka bertambah pula kompleksitas perhitungannya. Selain itu, perhitungan dilakukan secara *online*. Berbeda dengan *item-based* metode ini tidak menggunakan profil *user* tetapi rekomendasi kepada *user* dihitung dengan menentukan item yang mirip dengan item lain yang disukai oleh *user* tersebut. Keuntungan lainnya, perhitungan *item-based* dapat dilakukan *offline*, sehingga pengguna sistem tidak merasa terganggu dengan lambatnya sistem dalam menampilkan hasil rekomendasi.



Kemudian penelitian yang dilakukan oleh (Pratama 2019) dengan judul penelitian aplikasi rekomendasi tempat makan menggunakan algoritma *slope one* pada platform android, dalam penelitian ini disebutkan bahwa hasil dari perhitungan manual, penelitian ini telah berhasil dibuat berdasarkan algoritma *slope one*, dengan kriteria jarak dan tingkat popularitas dari tempat makan dalam memberikan rekomendasi.

Sistem yang dibuat penulis akan di implementasikan pada perangkat *smartphone* berbasis android dikarenakan pada juni 2019, pengguna android mencapai 93.22% dari pengguna *smartphone* di Indonesia (Statista, 2019). Berdasarkan pemaparan diatas maka penulis mengangkat sebuah judul “**Sistem Rekomendasi Jasa Tukang Dengan Metode Collaborative Filtering dan Algoritma Slope One Berbasis Android**” penelitian ini dibuat dengan tujuan untuk membantu mencari jasa tukang berdasarkan rekomendasi dan membantu para penyedia jasa tukang untuk mendapatkan pesanan pekerjaan.

## 1.2. Rumusan Masalah

Berdasarkan pada pemaparan latar belakang yang sudah dijelaskan diatas, maka penulis mengambil sebuah rumusan masalah, sebagai berikut :

1. Bagaimana merancang dan membangun sistem rekomendasi menggunakan metode *collaborative filtering* dengan algoritma *slope one*?
2. Bagaimana membangun aplikasi yang dapat mencari tukang serta bagaimana agar tukang bisa mendapatkan pesanan melalui aplikasi?

### 1.3. Batasan Masalah

Berikut batasan masalah di dalam penelitian ini agar ruang lingkup permasalahan tidak meluas :

1. Aplikasi hanya dapat di gunakan pada Operasi sistem Android
2. Penelitian ini dilakukan di Desa Sindangsari, Kecamatan Tanjung Bintang, Kabupaten Lampung Selatan
3. Penelitian ini menggunakan pendekatan *collaborative filtering* dan algoritma *slope one* untuk memberikan nilai rating

### 1.4. Tujuan Penelitian

Adapun tujuan dari penelitian ini dilaksanakan adalah sebagai berikut :

1. Membangun aplikasi yang dapat berjalan pada *smartphone* berbasis android
2. Mengimplementasikan metode *collaborative filtering* dengan algoritma *slope one* untuk menghasilkan rekomendasi jasa tukang
3. Untuk memenuhi persyaratan tugas akhir atau skripsi

### 1.5. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut :

1. Dapat mengembangkan kemampuan penulis dalam membangun dan merancang sebuah aplikasi berbasis android
2. Dapat membantu pengguna dalam mencari jasa tukang dan membantu penyedia jasa tukang agar mendapatkan pesanan
3. Penelitian ini diharapkan menjadi bahan bacaan dalam menambah wawasan di perpustakaan Universitas dan dapat memberikan referensi bagi mahasiswa lain.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Tinjauan Pustaka**

Pada penelitian ini digunakan tinjauan studi yang nantinya akan mendukung penelitian, berikut ini merupakan tinjauan studi yang diambil oleh penulis sebagai berikut :

##### **2.1.1. Tinjauan Literatur 1**

Pada penelitian literatur 1 dengan judul Aplikasi Pemilihan Jasa Tukang Menggunakan Metode *Weighted Product* Berbasis Web pada tahun 2019 oleh Y Setiawan. Aplikasi tersebut meliputi sebuah sistem yang memberikan informasi mengenai jasa tukang. Sehingga aplikasi ini dapat membantu masyarakat dalam mencari atau menggunakan jasa tukang tanpa takut harga mahal dan kinerja yang tidak bagus. Aplikasi ini dapat mempermudah masyarakat yogyakarta dalam memilih jasa tukang dan melihat rekomendasi jasa tukang yang sudah disarankan berdasarkan ongkos dan rating. Lalu berdasar hasil pengamatan dan penelitian yang telah dilakukan, dapat disimpulkan bahwa dengan menggunakan metode *weighted product* pada aplikasi ini dapat mempermudah masyarakat yogyakarta dalam memilih jasa tukang. Kemudian metode *weighted product* yang digunakan sistem mampu melakukan pengurutan jasa tukang sebagai hasil rekomendasi jasa tukang yang disarankan berdasarkan ongkos dan rating.

##### **2.1.2. Tinjauan Literatur 2**

Pada penelitian literatur 2 dengan judul Aplikasi Pemesanan Jasa Tukang Menggunakan Teknologi *Webhook* dan *Google Event Calendar* pada tahun 2019 oleh Muhammad Nafis. Jasa tukang adalah sebuah layanan publik yang merupakan

orang yang mempunyai kepandaian dalam suatu pekerjaan, juga merupakan sebuah kegiatan dan manfaat yang di tawarkan oleh pihak penyedia jasa ke pihak pengguna jasa. Belum ada solusi mudah untuk proses pemesanan jasa tukang yang sesuai keahlian dan permintaan yang dibutuhkan, oleh karena itu dibutuhkan suatu wadah untuk mengatasi permasalahan tersebut yaitu dibuatkan aplikasi yang bernama Jakang. Aplikasi Jakang ini memiliki fungsi untuk mempermudah penggunaan jasa dalam memesan jasa tukang sesuai keahlian dan sesuai permintaan yang dibutuhkan oleh rekomendasi sistem. Adapun teknologi yang digunakan adalah *webhook* sebagai notifikasi otomatis sebelum pengerjaan, dan juga *google event calendar* digunakan untuk proses penjadwalan.

### **2.1.3. Tinjauan Literatur 3**

Pada penelitian literatur 3 dengan judul Rancang Bangun Sistem Rekomendasi Pariwisata Mobile dengan Menggunakan Metode *Collaborative Filtering* dan *Location Based Filtering* pada tahun 2019 oleh Assaf Arief. Penelitian ini bertujuan untuk membuat otomatis sistem rekomendasi pariwisata *mobile* menggunakan konsep personalisasi dengan metode *collaborative filtering* dan *location based filtering* sebagai pengganti dari sistem manual ( seorang pemandu). Langkah – langkah menyelesaikan penelitian ini adalah membuat sistem rekomendasi pariwisata *mobile* dengan metode tersebut dan membuat *prototype* aplikasi berbasis *mobile web*, melakukan perancangan, pembuatan, pengujian dan perbaikan sistem rekomendasi. Hasil penelitian ini berupa sebuah aplikasi sistem rekomendasi pariwisata berbasis *mobile web* menggunakan *platform query mobile*, HTML 5, javascript, Ajax, PHP dan MySQL. Dari proses pengujian fungsionalitas,

pengujian komparabilitas dan pengujian sistem rekomendasi dapat terlihat bahwa rancangan sistem rekomendasi pariwisata sudah dapat memberikan rekomendasi wisata sesuai dengan metode *Collaborative Filtering* dan *Location-based filtering* yang dirancang.

#### **2.1.4. Tinjauan Literatur 4**

Pada penelitian literatur 4 dengan judul Sistem Rekomendasi dengan Metode *Item-Based Collaborative Filtering* Pada Aplikasi *Mobile* Berbasis Android untuk Wisata Kuliner Kota Surakarta pada tahun 2019 oleh Vina Zahrotun. Penelitian ini bertujuan membangun sebuah sistem rekomendasi dengan *item-based collaborative filtering* yang diterapkan dalam teknologi *mobile*. Proses utama rekomendasi dalam sistem ini menggunakan item-based CF dengan 5 item tetangga terdekat (*neighborhood*) karena hasil analisis pengujian *MAE* dengan pengosongan acak sebesar 5%, 15%, 10%, dan 20% dengan variasi jumlah *neighborhood* dan menunjukkan bahwa metode *item-based collaborative filtering* dengan 5 tetangga dapat menghasilkan prediksi dengan kualitas akurasi yang baik.

#### **2.1.5. Tinjauan Literatur 5**

Pada penelitian literatur 5 dengan judul Aplikasi Rekomendasi Tempat Makan Menggunakan Algoritma *Slope One* pada Platform Android pada tahun 2020 oleh Dharma Pratama. Pada aplikasi ini, rekomendasi yang diberikan didapatkan dari perhitungan menggunakan algoritma *slope one* dan daftar tempat makan yang didapatkan berasal dari *Google Places API*. Algoritma *slope one* membuat rekomendasi dengan menjumlahkan rating dari suatu tempat dengan rata-

rata selisih suatu tempat dengan tempat lainnya. Aplikasi telah diuji coba pada pengguna menggunakan kuisioner yang dibuat berdasarkan kuisioner J.R. Lewis dengan katagori pertanyaan tingkat kegunaan aplikasi, kualitas informasi yang diberikan oleh aplikasi dan kualitas tampilan antar muka aplikasi. Hasil yang didapat dari uji coba melalui kuisioner adalah aplikasi berguna bagi user untuk menentukan pilihan tempat makan, kualitas informasi yang diberikan aplikasi baik dan tampilan antar muka dari aplikasi baik, yang berarti pengguna merasa puas dalam menggunakan aplikasi.

#### **2.1.6. Kesimpulan Tinjauan Literatur**

Kesimpulan yang dapat penulis ambil dari tinjauan literatur bahwa sistem rekomendasi dapat dilakukan dengan beberapa metode. Salah satu metode rekomendasi yaitu *item-based collaborative filtering* dinyatakan mampu memberikan hasil prediksi dengan kualitas akurasi yang baik dan telah di uji pada tinjauan literatur 4. Algoritma *Slope one* dinyatakan mampu membuat rekomendasi dengan menjumlah rating dari suatu tempat dengan rata-rata selisih suatu tempat dengan tempat yang lainnya dan telah di uji pada tinjauan literatur 5. Maka dari itu penulis menggunakan metode dan algoritma tersebut yaitu Metode *Collaborative Filtering* dan algoritma *Slope One* dalam penelitian ini dan untuk membangun sebuah aplikasi jasa tukang seperti pada tinjauan literatur 1 dan 2 dengan memasukan sistem rekomendasi. Perbedaan yang terdapat pada penelitian penulis adalah aplikasi jasa tukang yang akan dibuat oleh penulis berbasis android dan memberikan sistem rekomendasi untuk memilih jasa tukang di dalamnya.

## 2.2. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile* (M Nafis, 2019). Di lain pihak, google merilis kode-kode android dibawah lisensi Apache, sebuah lisensi perangkat lunak *open platform* perangkat seluler .

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat full support dari Google atau *Google Mail Service (GMS)* dan kedua tanpa dukungan Google atau *Open Handset Distribution (OHD)*. Pada masa saat ini kebanyakan vendor-vendor smartphone sudah memproduksi smartphone berbasis android, diantaranya di indonesia adalah Xiaomi, Samsung, Asus, Vivo, Oppo dan masih banyak lagi vendor yang memproduksi Andorid.

Pesatnya pertumbuhan Android selain faktor diatas adalah karena Android itu sendiri adalah *platform* yang sangat lengkap baik itu sistem operasinya, Aplikasinya dan *Tool* Pengembangan, *Market* serta dukungan sangat tinggi dari komunitas *Open Source* di dunia, sehingga android terus berkembang sangat pesat baik dari teknologi maupun dari segi jumlah *device* yang ada di dunia. Berikut adalah logo android yang ditunjukkan pada gambar 2.1 .



**Gambar 2.1** Logo Android

### 2.3. Android Studio

Android Studio merupakan sebuah *Integrated Development Environment (IDE)* khusus untuk membangun aplikasi yang berjalan pada platform android. Android studio ini berbasis pada *IntelliJ IDEA*, sebuah *IDE* untuk bahasa pemrograman *Java*. Bahasa pemrograman utama yang digunakan adalah *java*, sedangkan untuk membuat tampilan atau *layout*, digunakan bahasa *XML*. Android studio juga terintegrasi dengan *Android Software Development Kit (SDK)* untuk *deploy* ke perangkat android. Android studio juga merupakan pengembangan dari *eclipse*, dikembangkan menjadi lebih kompleks dan profesional yang telah tersedia didalamnya *Android Studio IDE*. (Yulianti dan Dita, 2019).



**Gambar 2.2** Logo Android Studio

Setiap proyek di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis modul mencakup :

- a. Modul aplikasi android
- b. Modul Pustaka
- c. Modul *Google App Engine*

Secara default, Android studio akan menampilkan file proyek dalam tampilan proyek android.



## 2.4. Algoritma

Algoritma adalah kumpulan instruksi atau perintah yang dibuat secara jelas dan sistematis berdasarkan urutan yang logis (logika) untuk penyelesaian suatu masalah. French, C.S (2020) menyatakan sejumlah konsep yang mempunyai relevansi dengan masalah rancangan program yaitu kemampuan komputer, kesulitan dan ketepatan. Knuth (2020) menyatakan algoritma fundamental untuk keperluan matematika dan program komputer.

Algoritma tidak selalu memberikan hasil terbaik yang mungkin diperoleh, maka diharapkan adanya suatu evaluasi mutu hasil dari algoritma tersebut (Liu, C.L, 2019). Sekali sebuah algoritma diberikan kepada sebuah permasalahan dan dijamin akan memberikan hasil yang diharapkan, maka langkah penting selanjutnya adalah menentukan besar biaya yang diperlukan algoritma tersebut untuk memperoleh hasil.

Dalam dunia komputer, algoritma sangat berperan penting dalam pembangunan suatu *software*. Dalam dunia sehari-hari, mungkin tanpa kita sadari algoritma telah masuk dalam kehidupan kita seperti cara memasak, membuat teh, cara membuat kopi dan lain-lain.

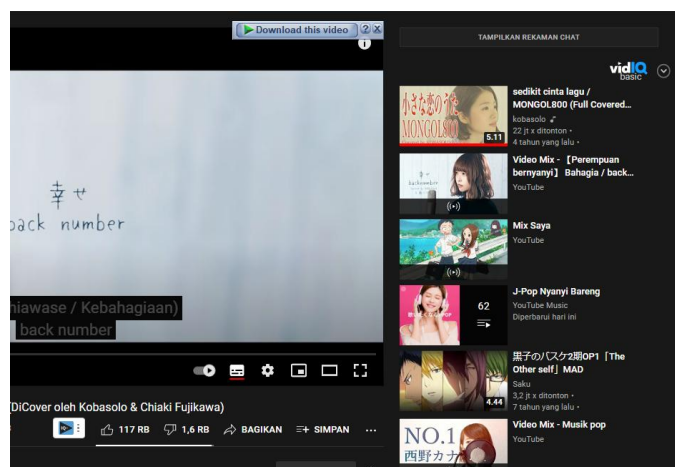
## 2.5. Sistem Rekomendasi

Sistem rekomendasi merupakan suatu sistem yang dirancang untuk memprediksi sekumpulan *item* yang sesuai dengan preferensi *user* yang mana nantinya item tersebut akan direkomendasikan pada *user* lain (Sanjung, 2019). Perkiraan informasi ini diperoleh melalui profil *user*, preferensi *item* dan aktivitas yang terjadi pada sistem. Profil *user* dapat berisi tentang informasi *user*,

ketertarikan *user* pada suatu *item* dan juga riwayat interaksi antara pengguna dengan *item*. Misalnya ketika pengguna melakukan peminjaman buku maka data riwayat peminjaman ini akan dicatat dan menjadi profil pengguna.

Pada sistem rekomendasi terdapat sekumpulan *item* yang ada akan disaring berdasarkan preferensi *item*, *user*, *rating*, catatan *history* transaksi dan lain lain sehingga menghasilkan beberapa *item* yang direkomendasikan kepada *user*. Pada saat ini sudah banyak yang menerapkan sistem rekomendasi pada *website* untuk mempermudah *user* mencari informasi, seperti youtube.com dan lain lain.

Gambar 2.3 merupakan *screenshot* dari halaman situs youtube.com. Pada halaman tersebut dapat dilihat pada sisi sebelah kanan terdapat beberapa video yang direkomendasikan kepada *user* berdasarkan *history* pencarian yang pernah dilakukan *user* tersebut.



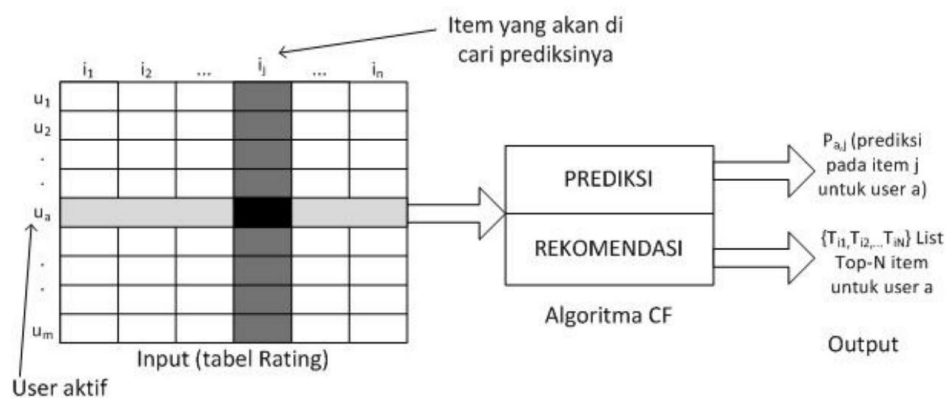
**Gambar 2.3** Rekomendasi Youtube

## 2.6. Collaborative Filtering

*Collaborative Filtering* merupakan salah satu cara yang diterapkan sistem rekomendasi untuk memberikan prediksi otomatis terhadap keinginan pelanggan dengan cara mengkoleksi informasi dari banyak pelanggan ( Ampaziz, 2020). Pada

*collaborative filtering*, rekomendasi yang diberikan berdasarkan keterkaitan (preferensi) antara *item* satu dengan *item* lainnya dan keterkaitan yang didapat diantara *user* yang telah *me-rating* atau berinteraksi dalam sistem.

*Collaborative filtering* memberikan rekomendasi berdasarkan kumpulan dari pendapat, minat dan keterkaitan beberapa *user* yang biasanya diberikan dalam bentuk *rating* yang diberikan *user* kepada suatu *item*. Pada metode ini, terdapat kumpulan  $m$  *user*  $U = \{u_1, u_2, u_3, \dots\}$  dan kumpulan  $n$  *item*  $I = \{i_1, i_2, i_3, \dots\}$  dimana setiap *user* ( $u_i$ ) mempunyai daftar *item*  $I_{u_i}$  yang merupakan ekspresi dari pendapatnya.



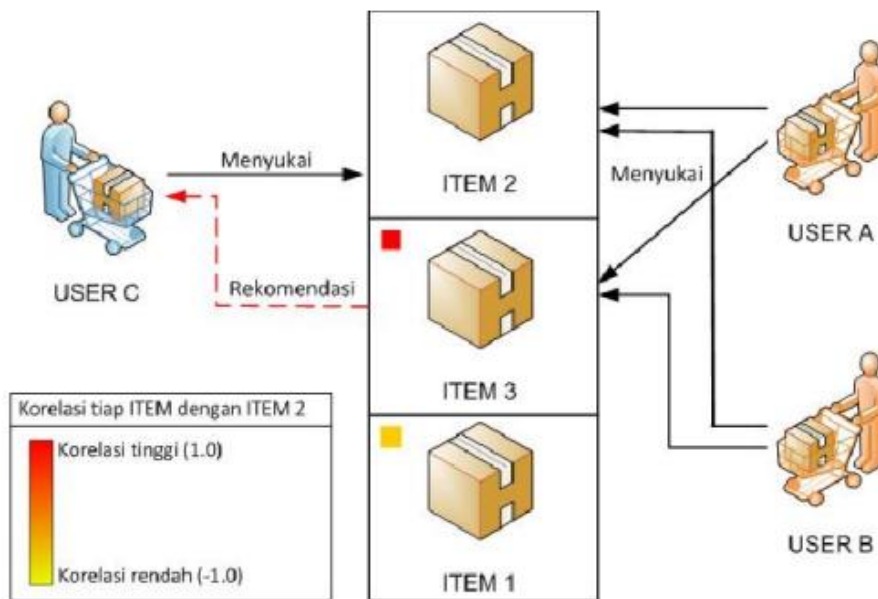
**Gambar 2.4** Proses *Collaborative Filtering* ( Hakim, 2020)

Gambar 2.4 diatas merupakan diagram skema dari metode *collaborative filtering*. Pada gambar tersebut dipresentasikan sejumlah  $m$  *user* x  $n$  *item* sebagai matriks rating dimana berisi nilai *rating* dari *user* untuk setiap *item*.

Pendekatan *collaborative filtering* pada dasarnya dibagi menjadi dua katagori yaitu *userbased collaborative filtering* disebut juga *memorybased*, dan *itembased collaborative filtering* yang disebut juga *model-based* ( Uyun, 2019). Pada pendekatan *userbased collaborative filtering* sistem memberikan rekomendasi kepada *user* *item-item* yang disukai atau di rating oleh *user-user* lain

yang memiliki banyak kemiripan dengannya. Misalnya, *user a* menyukai atau merating *item 1, 2, dan 3*, kemudian *user b* menyukai *item 1, 2, dan 4* maka sistem akan merekomendasikan *item 3* kepada *user b* dan *item 4* kepada *user a*. Kelebihan dari pendekatan *user based collaborative filtering* adalah dapat menghasilkan rekomendasi yang berkualitas baik. Sedangkan kekurangannya adalah kompleksitas perhitungan akan semakin bertambah seiring dengan bertambahnya pengguna sistem, semakin banyak pengguna yang menggunakan sistem, maka proses perkomendasi akan semakin lama.

Pendekatan *item based collaborative filtering* memberikan rekomendasi berdasarkan kemiripan antar *item*. Metode ini merupakan metode rekomendasi yang didasari atas adanya kesamaan antara pemberian *rating* terhadap suatu item dengan *item* yang pernah di-*rating user* lain (Purwanto, 2020). *Item* yang telah di-*rating* oleh *user* akan menjadi patokan untuk mencari sejumlah *item* lainnya yang berkorelasi dengan *item* yang telah di-*rating user*. Motivasi kunci dibalik metode ini adalah *user* akan cenderung menyukai *item* yang sejenis atau mempunyai korelasi dengan *item* yang telah disukainya. (Sarwar, 2019). Pada gambar 2.5 menggambarkan bagaimana *item 3* direkomendasikan kepada *user c* yang telah menyukai *item 2*, dimana *item 2* yang disukai *user c* memiliki korelasi yang kuat dengan *item 3*.



**Gambar 2.5** Skema *item-based collaborative filtering* (Hakim, 2020)

## 2.7. Algoritma *Slope One*

Algoritma *slope one* adalah salah satu algoritma untuk membuat sistem rekomendasi. *Slope one* memberikan prediksi berdasarkan nilai hasil pencarian dari item-item yang dibandingkan. Keunggulan algoritma ini dibandingkan dengan algoritma lainnya adalah algoritma *slope one* mudah untuk diimplementasi, efisien saat melakukan query, tidak memerlukan banyak requirement dikarenakan rekomendasi berdasarkan rating dari setiap *item*, dan cukup akurat.

Algoritma *slope one* melakukan perhitungan berdasarkan hubungan linear dari nilai preferensi atau *weight* dari setiap *item* yang dibandingkan. Estimasi umum dari dasar perhitungan algoritma ini adalah fungsi linear  $y = mx + b$ , dengan asumsi *gradient*  $m=1$ , sehingga fungsi menjadi  $b = y - x$ . cara kerja algoritma *slope one* adalah dengan mencari selisih dari suatu *item* dengan *item-item* lain yang

dibandingkan. Perhitungan algoritma *slope one* dapat diformulasikan dengan persamaan dengan

$$dev_{j,i} = \sum_{u \in S_{j,i}(x)} \frac{u_j - u_i}{card(S_{j,i}(x))}$$

Dimana :

$dev_{j,i}$  = rata-rata selisih rating item j dan i

$u_j$  = rating item j

$u_i$  = rating item i

$card(S_{j,i}(x))$  = banyaknya elemen yang dibandingkan

Apabila selisih sudah didapatkan, maka dapat dilakukan perhitungan rekomendasi untuk item j yang dapat dirumuskan dengan persamaan :

$$p^{SI}(u)_j = dev_{j,i} + u_j$$

Dimana :

$p^{SI}(u)_j$  = nilai rekomendasi untuk item j

Berdasarkan persamaan di atas, algoritma *slope one* memberikan rekomendasi dengan melakukan perhitungan selisih setiap item. Selisih yang didapatkan akan dirata-ratakan per *item* yang kemudian akan dijumlahkan dengan *value* dari masing-masing *item*. *Value* yang sudah dijumlahkan dengan rata-rata selisihnya akan digunakan sebagai point untuk memberikan rekomendasi. Gambar 2.6 memperlihatkan *pseudocode* algoritma *slope one*.

```

BEGIN
  For every item i
    For every other item j
      For every user u expressing preference for
        both i and j
        Add the difference in u's preference for
          i and j to an average
      //Prediction Part
      For every item i the user u expresses no
        preference for
        For every item j that user u expresses a
          preference for
          Find the average preference difference
            between j and i
          Add this diff to u's preference value for
            j
          Add this to a running average
      Return the top items, ranked by these averages
END

```

**Gambar 2.6** *Pseudocode slope one*

## 2.8. Aplikasi

Aplikasi merupakan perangkat lunak yang dijalankan oleh para pengguna atau biasa disebut dengan user untuk mendapatkan suatu tujuan tertentu. Aplikasi perangkat lunak adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk bertujuan melakukan suatu tugas yang diinginkan pengguna. Aplikasi perangkat lunak adalah program yang membuat komputer dapat digunakan untuk pekerjaan sehari-hari agar lebih efektif dan efisien.

Biasanya dibandingkan dengan perangkat lunak sistem yang menintegrasikan berbagai kemampuan komputer tetapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan. Program ini lah yang mengendalikan semua aktifitas yang ada pada pemroses. Program berisi konstruksi logika yang ada dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada *instuction set*. Program aplikasi merupakan program siap pakai yang dirancang untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi lain.

Contoh-contoh aplikasi ialah program pemroses kata dan *web browser*. Aplikasi akan menggunakan sistem operasi komputer dan aplikasi yang lainnya yang mendukung. Istilah ini mulai perlahan masuk ke dalam istilah Teknologi informasi.

## 2.9. Firebase

Firebase memiliki produk utama, yaitu menyediakan *database realtime* dan *backend* sebagai layanan. Layanan ini menyediakan pengembangan aplikasi *API* yang memungkinkan aplikasi data yang akan disinkronasikan di klien dan disimpan di cloud firebase ini. Firebase menyediakan library untuk berbagai *client platform* yang memungkinkan integrasi dengan Android, IOS, Javascript Node Js dan dapat juga disebut sebagai layanan *DbaaS (Database as a Service)* dengan konsep *realtime*.

Firebase digunakan untuk mempermudah dalam penambahan fitur – fitur yang akan dibangun oleh *developer*. Semua data firebase *realtime database* disimpan sebagai objek JSON. Tidak seperti basis data SQL, tidak ada tabel atau rekaman. Ketika ditambahkan ke JSON tree, data akan menjadi simpul dalam struktur JSON yang ada. Meskipun basis data menggunakan JSON tree, data yang tersimpan dalam basis data bisa diwakili sebagai tipe bawaan tertentu yang sesuai dengan tipe JSON yang tersedia untuk membantu menulis banyak kode.

## 2.10. Tukang

Terdapat 5 arti kata “Tukang” di Kamus Besar Bahasa Indonesia (KBBI) yang masuk kedalam kelas verba (kata kerja)



1. Orang yang mempunyai kepandaian dalam suatu pekerjaan tangan ( dengan alat atau bahan yang tertentu).

Contoh : batu, besi, kayu

2. Orang yang pekerjaannya membuat (menjual, memperbaiki, dan sebagainya)

Contoh : daging (sayur,susu), arloji, pedati (gerobak, becak)

3. Orang yang pekerjaannya melakukan sesuatu secara tetap

Contoh : pangkas (cukur), las, jahit, masak, cetak

4. Orang yang biasa suka melakukan sesuatu (yang kurang baik)

Contoh : mabuk, serabot, copet, tadah, catut

5. Ahli dalam ragam cakapan

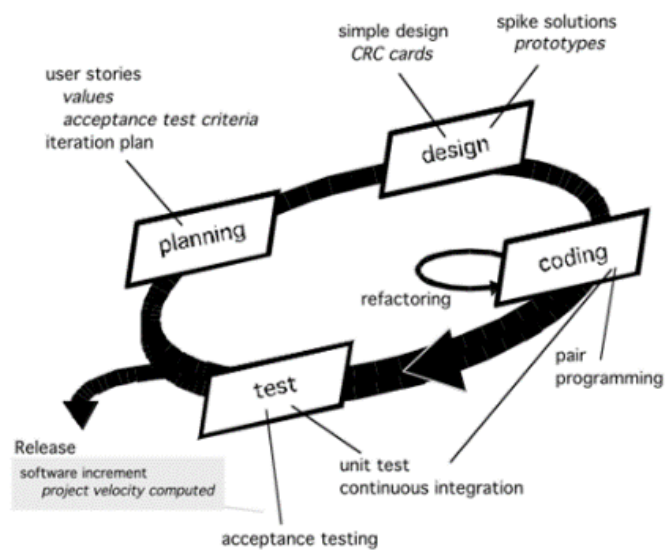
Contoh : menciptakan sajak, pidato

### **2.11. Bangunan**

Menurut Kamus Besar Bahasa Indonesia (KBBI) arti kata bangunan adalah bangunan yang didirikan, yang dibangun (seperti rumah, gedung, jembatan). Arti lainnya jika ditambah “—liar” adalah bangunan yang didirikan secara tidak sah tanpa memperoleh izin membangun atau yang didirikan di atas tanah bukan milik sendiri. Arti lain dengan tambahan “-permanen” adalah bangunan yang dibuat dengan bahan bangunan yang kuat dan tahan lama seperti dari baja, beton, batu bata

### 2.12. *Extereme Programming*

*Extreme Programming (XP)* merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan *requirement* yang tidak jelas maupun terjadi perubahan *requirement* yang cepat. Ilustrasi XP dapat dilihat pada gambar 2.3



**Gambar 2.7** Extreme Programming

Adapun tahapan dalam XP ini adalah sebagai berikut :

1. *Planning* (Perencanaan)

Tahapan ini dimulai dengan mendengarkan kumpulan kebutuhan aktifitas suatu sistem yang memungkinkan pengguna memahami proses bisnis untuk sistem dan mendapatkan gambaran yang jelas mengenai fitur utama, fungsionalitas dan keluaran yang di inginkan

## 2. *Design* (Perancangan)

Pada tahapan perancangan dilakukan pembuatan pemodelan sistem berdasarkan hasil analisa kebutuhan yang didapatkan. Selain itu dibuatkan juga pemodelan basis data untuk menggambarkan hubungan antar data. Pemodelan sistem yang digunakan yaitu *Simple Design CRC Card* dan *Spike solutions prototypes*

## 3. *Coding* (Pengkodean)

Tahapan ini merupakan implementasi dari perancangan model sistem yang telah dibuat kedalam kode program yang menghasilkan prototipe dari perangkat lunak.

## 4. *Testing* (Pengujian)

Tahapan ini merupakan tahapan pengujian terhadap aplikasi yang sudah dibangun, pada tahapan ini ditentukan oleh pengguna sistem dan berfokus pada fitur dan fungsionalitas dari keseluruhan sistem.

## 5. Software increment (Peningkatan Perangkat Lunak)

Tahapan ini merupakan tahap pengembang sistem yang sudah dibuat secara bertahap yang dilakukan setelah sistem diterapkan dalam organisasi dengan menambah layanan atau konten yang mengakibatkan bertambahnya kemampuan fungsionalitas dari sistem

### **2.13. Mobile Application**

*Mobile application* adalah sebuah aplikasi yang memungkinkan manusia melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler (*handphone*). Dengan menggunakan aplikasi *mobile*, manusia dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar,

mengerjakan pekerjaan kantor, browsing, chatting, e-mail dan sebagainya. (Yonarisa, 2019)

#### **2.14. Android SDK**

Android SDK (*software development kit*) merupakan *tools* bagi para programmer yang ingin mengembangkan aplikasi berbasis google android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode, dan tutorial (Onserda, 2020)

#### **2.15. IDE (Integrated Development Environment)**

Adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. Sebuah IDE atau secara bebas dapat diterjemahkan sebagai lingkungan pengembangan terpadu. Berikut beberapa fasilitas yang diberikan oleh IDE :

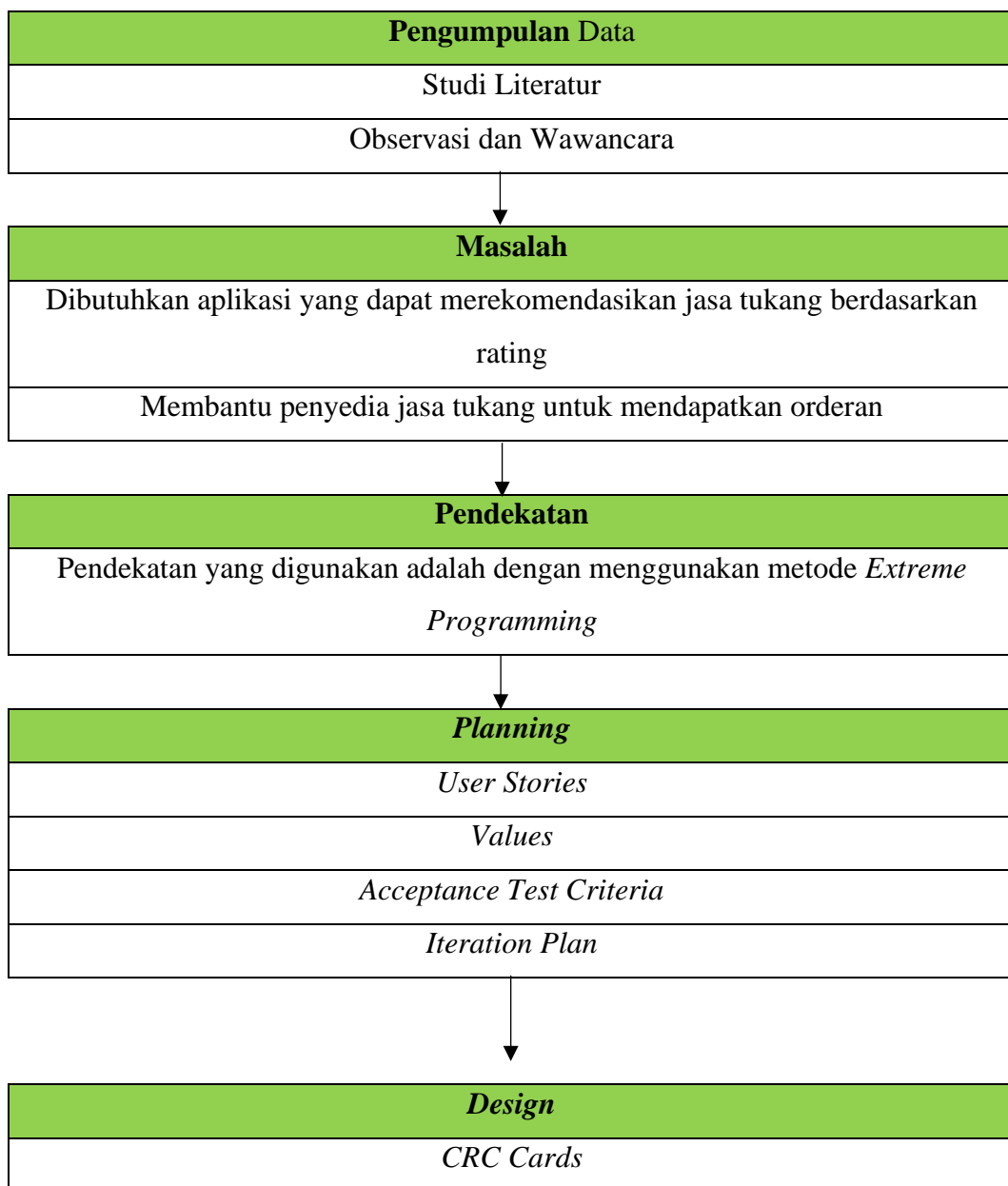
1. *Editor*, yaitu fasilitas untuk menulis kode sumber dari perangkat lunak.
2. *Compiler*, yaitu fasilitas untuk mengecek sintaks dari kode sumber kemudian mengubahh dalam bentuk binari yang sesuai dengan bahasa mesin.
3. *Linker*, yaitu fasilitas untuk menyatukan data binari dari beberapa kode sumber yang dihasilkan compiler sehingga data-data binari tersebut menjadi satu kesatuan dan menjadi suatu program siap dieksekusi.
4. *Debugger*, yaitu fasilitas yang mengetes jalannya program untuk mencari *bug* / kesalahan yang terdapat pada program.

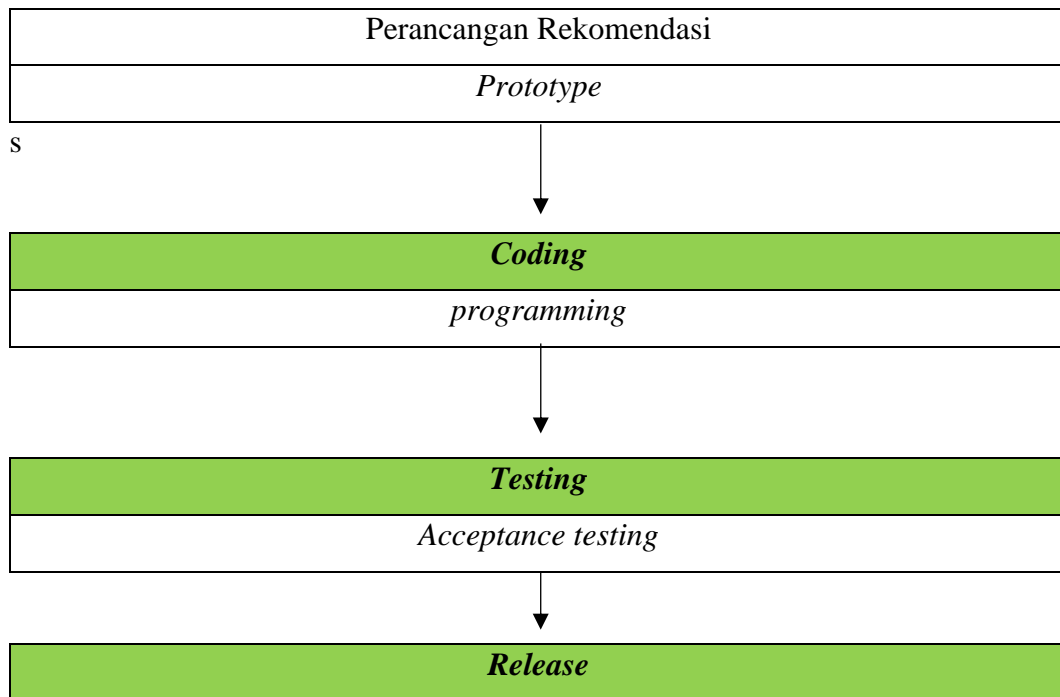
## BAB III

### METODE PENELITIAN

#### 3.1. Kerangka Penelitian

Kerangka penelitian merupakan sebuah konsep atau gambaran yang dibuat dan yang akan dilakukan oleh penulis dalam melaksanakan penelitian. Dari uraian yang sudah dijelaskan sebelumnya, maka dapat dibuat kerangka penelitian yang terdapat dibawah ini :





**Gambar 3.1** Kerangka Penelitian

### 3.1.1. Pengumpulan data

Dalam pengumpulan data, penulis menggunakan beberapa pengumpulan data seperti berikut:

#### 3.1.1.1. Wawancara

Wawancara adalah sebuah teknik pengumpulan data dengan melajukan interaksi tanya jawab secara langsung antara narasumber dan pewawancara, dengan pertanyaan-pertanyaan yang sudah disiapkan oleh pewawancara. Untuk mencari informasi dan mencari masalah penelitian dalam penelitian ini, penulis melakukan sebuah wawancara yang berguna untuk mengidentifikasi masalah, dan mencari solusi dari hasil wawancara tersebut. Penulis melakukan wawancara dengan beberapa pengguna serta penyedia Jasa Tukang / Kuli di desa sindangsari, Tanjung Bintang

### 3.1.1.2. Tinjauan Pustaka

Pengumpulan data dengan studi pustaka melakukan pengumpulan datanya dengan cara membaca, mencatat, mengutip, lalu mengumpulkan data dari jurnal, dan dari berbagai sumber lainnya seperti internet dengan *keyword* sistem rekomendasi, algoritma *slope one*, jasa tukang, *collaborative filtering*. Dengan melakukan studi pustaka, peneliti dapat memanfaatkan semua informasi dan pemikiran yang relevan dengan penelitian yang akan diteliti

### 3.1.2. Masalah

Dalam pengumpulan data pada penelitian ini, ditemukan masalah bahwa sulitnya masyarakat dalam mencari jasa tukang untuk membuat atau memperbaiki sesuatu, serta di sisi penyedia jasa tukang ditemukan masalah yaitu sulitnya mendapatkan pesanan atau pekerjaan. Kemudian penulis menemukan solusi dari masalah tersebut yaitu membuat sebuah aplikasi yang nantinya bisa membantu mempertemukan mereka sebagai pengguna dan penyedia jasa.

### 3.1.3. Pendekatan

Dalam penelitian ini pendekatan dilakukan dengan cara menggunakan metode *Extreme Programming (XP)*. *XP* merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek. Metode ini juga sesuai jika tim dihadapkan dengan *requirement* yang tidak jelas

maupun terjadi perubahan *requirement* yang sangat cepat. XP terdiri dari 5 tahapan yaitu *planning, design, coding, test, release*

#### 3.1.4. *Planning*

##### 3.1.4.1. *User Stories*

*User stories* adalah deskripsi singkat dan sederhana tentang fitur yang diceritakan dari sudut pandang orang yang menginginkan atau yang akan menggunakan sistem nanti nya. Ada pun rangkuman yang telah penulis buat dari *user stories* berdasarkan wawancara adalah :

##### A. Penyedia jasa

Bapak Ridho bercerita “Saya sebagai tukang, merasa kesulitan untuk mendapatkan pekerjaan atau pesanan dan bahkan ketika saya mendapatkan tawaran pekerjaan, terkadang pekerjaan tersebut tidak sesuai dengan keahlian saya. Maka dari itu saya membutuhkan aplikasi yang bisa membantu saya mendapatkan pesanan pekerjaan dan sesuai dengan keahlian saya ”.

Kemudian pak Heri bercerita “Iya saya sebagai tukang lebih sulit mencari pesanan atau pekerjaan dari pada mengerjakannya. Saya akan sangat amat terbantu jika ada aplikasi yang bisa membantu saya untuk mendapatkan pesanan. Ya karena itu tadi mas, sulit sekali cari pesanan”

Dan pak Sarno bercerita “Sebagai tukang, yang menyulitkan bagi saya adalah mencari pesanan atau pekerjaannya. Kalau masalah harga itu bisa di nego waktu sudah dapat pekerjaannya. Intinya saya butuh fitur yang bisa memudahkan saya untuk mendapatkan pesanan mas”



## B. Pengguna jasa

Bapak Piyan bercerita “Saya sebagai pengguna jasa, merasa kesulitan untuk mencari jasa tukang. Karena saya sendiri masih kekurangan informasi untuk mengetahui seseorang tukang serta saya tidak tahu kinerja dari tukang tersebut, pernah waktu itu memakai jasa tukang dari kenalan saudara tetapi hasil kerjanya tidak memuaskan. Maka dari itu saya menginginkan aplikasi yang bisa mencarikan saya jasa tukang yang berkompeten dalam bidangnya”.

Kemudian pak nawang bercerita “Sebagai pengguna jasa, hal-hal sulit yang saya alami ketika ingin memesan tukang itu yang paling utama adalah mencari tukang yaang ulet atau jago. Iya saya kenal tukang yang seperti itu, tetapi kalau dia sudah di pesan orang lain dan sedang mengerjakan pesanan, saya nunggu dulu sampai dia selesai. Akan sangat membantu kalau ada aplikasi yang bisa mencarikan saya tukang ulet lainnya, apalagi kalau pekerjaannya darurat mas”

### 3.1.4.2. *Values*

*Values* adalah serangkaian nilai tertentu yang menggambarkan fitur fungsionalitas yang dibuat dari *user stories* dan mengarahkan kerja sama yang lebih efektif di dalam tim untuk pembuatan aplikasi dan pada akhirnya membuat kualitas produk yang lebih tinggi. Berikut adalah *values* yang penulis ambil berdasarkan *user stories* :

#### A. Dibagian pengguna jasa

1. Fitur untuk mencari dan memesan jasa tukang
2. Memberikan rekomendasi tukang yang akan dipakai

B. Dibagian penyedia jasa

1. Pilih keahlian yang sesuai dengan bidangnya
2. Mendapatkan pesanan pekerjaan

Adapun kebutuhan sistem dan alat - alat yang diperlukan untuk penelitian dan pembuatan fitur diatas, berikut adalah tabel-tabel kebutuhan yang digunakan dalam penelitian ini :

**Tabel 3.1** Spesifikasi Minimum Target *Smartphone*

<i>Smartphone</i>		
No	<i>Hardware</i>	Spesifikasi
1	Sistem Operasi	Android
2	Versi Android	Versi 5.0 ( <i>lolipop</i> ) ke atas
3	RAM ( <i>Random Access Memory</i> )	1GB

**Tabel 3.2** Spesifikasi *Hardware* Pengembang

<i>Personal Komputer</i>			
No	<i>Hardware</i>	Nama	Spesifikasi
1	<i>Processor</i>	Intel i5 vPro-2520m	2.50Ghz (4CPUs)
2	<i>Memory</i>	DDR 3 vgen 4x2	8192MB-1600Mhz
3	<i>Graphic Card</i>	IntelHd Family	IntelHd 3000
4	<i>Storage</i>	Seagate	512GB

**Tabel 3.3** Spesifikasi *Software* Pengembang

No	Nama	Spesifikasi
1	Sistem Operasi	Windows 10 Pro 64bit
2	Aplikasi	Android Studio
		Adobe Photoshop
		Adobe XD

### 3.1.4.3. *Acceptance Test Criteria*

Berdasarkan hasil kesepakatan dengan pengguna, selanjutnya menentukan kriteria tes sebagai acuan terhadap kebutuhan sistem yang akan dibangun, tes dilakukan oleh peneliti dan hasil akan dinilai oleh pengguna. Kriteria yang digunakan berdasarkan dua aspek pengujian ISO 25010 adalah sebagai berikut :

#### 1. *Functional Suitability*

*Functional Suitability* atau pengujian pada bagian fungsional, memiliki tujuan untuk melakukan pengecekan terhadap fungsi-fungsi yang ada pada sistem yang telah dibuat. Berikut ini merupakan karakteristik pengujian dibagi menjadi beberapa karakteristik yaitu :

- a. *Functional completeness*, sejauh mana fungsi yang telah disediakan mencakup seluruh tugas dan tujuan pengguna secara spesifik.
- b. *Functional correctness*, sejauh mana produk atau sistem menyediakan hasil yang benar sesuai dengan kebutuhan pengguna.
- c. *Functional appropriateness*, sejauh mana fungsi yang telah disediakan mampu memfasilitasi penyelesaian tugas atau suatu tujuan tertentu.

Berikut pertanyaan yang akan diujikan pada pengujian fungsional

(*functional suitability*) :

**Tabel 3.4** Tabel pertanyaan fungsional

Pertanyaan
<b><i>Functional Completeness</i></b>
Apakah aplikasi ini dapat terkoneksi ke database?
Apakah aplikasi dapat menampilkan form pesan tukang?
Apakah aplikasi dapat memesan tukang?
Apakah pesanan sampai ke tukang?

<b><i>Functional Correctness</i></b>
Apakah aplikasi ini dapat diakses dengan mudah?
Apakah aplikasi ini dapat menampilkan list rekomendasi tukang?
Apakah aplikasi ini dapat memilih tukang?
<b><i>Functional Appropriateness</i></b>
Apakah aplikasi ini dibuat sesuai dengan kebutuhan?
Apakah dengan dibuatkannya aplikasi ini dapat mempermudah tukang, pengguna tukang dalam mencari dan mendapatkan pesanan pekerjaan?
Apakah aplikasi ini menyimpan data sesuai dengan fungsinya?

## 2. Kemudahan pengguna (*Usability*)

Kemudahan pengguna atau *usability* merupakan tingkat dimana sistem yang akan dibuat memiliki tujuan agar mudah dimengerti, mudah dipakai dan menarik untuk digunakan oleh pengguna. Karakteristik dalam pengujian *usability* ini dibagi menjadi beberapa karakteristik yaitu :

- a. *Appropriateness recognisability*, sejauh mana pengguna mengetahui apakah sistem atau produk sesuai kebutuhan mereka.
- b. *Operability*, sejauh mana produk atau sistem mudah dioperasikan dan dikendalikan.
- c. *Learnability*, sejauh mana produk atau sistem dapat digunakan oleh pengguna untuk mencapai suatu tujuan tertentu, belajar menggunakan sistem atau produk dengan efisien, efektif dan kebebasan dari resiko serta dalam hal kepuasan pada konteks tertentu.
- d. *User interface aesthetics*, sejauh mana tampilan antarmuka pengguna dari sistem memungkinkan interaksi yang menyenangkan dan memuaskan pengguna.

- e. *User error protection*, sejauh mana produk atau sistem melindungi pengguna terhadap suatu kesalahan
- f. *Accessibility*, sejauh mana produk atau sistem dapat digunakan oleh semua kalangan untuk mencapai suatu tujuan tertentu sesuai dengan konteks pengguna.

Berikut pertanyaan yang akan diujikan pada pengujian kemudahan pengguna (*Usability*) :

**Tabel 3.5** Tabel pernyataan kemudahan pengguna (*usability*)

<b><i>Appropriateness Recognisability</i></b>	
1	Sistem ini membantu saya dalam mencaari tukang / pekerjaan
2	Sisstem ini sudah sesuai dengan kebutuhan saya
3	Sistem ini berjalan sesuai yang saya harapkan
<b><i>Operability</i></b>	
4	Sistem ini mudah dipakai
5	Tidak terdapat kesulitan dalam menggunakan aplikasi ini
6	Saya daapat melihat list pilihan rekomendasi tukang
7	Saya berhasil menggunakan aplikasi ini berjalan sesuai dengan kemampuan setiap kali digunakan
<b><i>Learnability</i></b>	
8	Saya belajar menggunakan aplikasi ini dengan cepat dan mudah
9	Saya mudah mengingat cara menggunakan sistem ini
10	Sistem ini mudah untuk dipelajari
<b><i>User Interface Aesthetics</i></b>	
11	Saya puas dengan aplikasi ini
12	Saya sangat merekomendasikan aplikasi ini bagi tukang dan pengguna jasa tukang
<b><i>User error Protection</i></b>	
13	Jika frmulir pesanan ataupun inputan tidak lengkap, akan ada peringatan
<b><i>Accesibility</i></b>	

14	Sistem ini dapat digunakan oleh warga sindang sari
15	Sistem ini dapat digunakan untuk jangka waktu yang panjang
16	Kemudahan dalam aplikasi ini akan membuat kemudahan pada tukang dan pengguna tukang dalam mencari dan mendapatkan pekerjaan

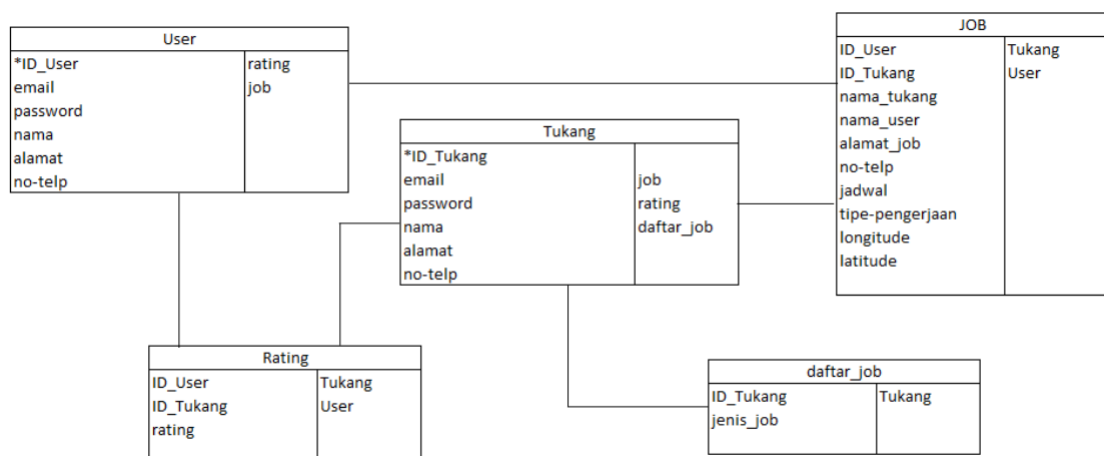
#### 3.1.4.4. Iteration Plan

Iterasi atau perulangan, rencana iterasi berlangsung selama 1 sampai 3 minggu. *User stories* dipilih untuk iterasi berdasarkan urutan prioritas pembuatan fitur, lalu dibuatkan *acceptance test criteria* agar fitur yang dibuat nanti sesuai dengan harapan pengguna.

#### 3.1.5. Design

##### 3.1.5.1. CRC Cards

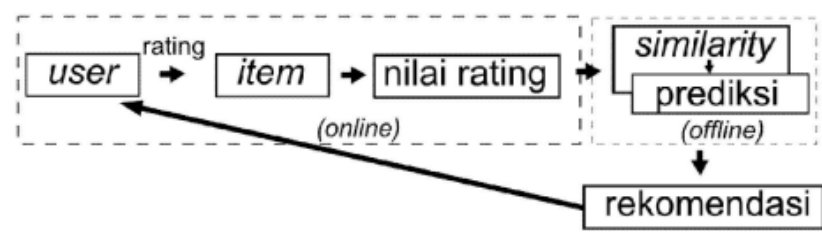
CRC adalah kumpulan kartu indeks standar yang telah dibagi menjadi tiga bagian (*class, responsibilities, collaborator*). *Class* merupakan koleksi benda-benda yang sama, *responsibilities* adalah sesuatu yang diketahui class atau tidak, dan *collaborator* yang berinteraksi untuk mengisi dari *responsibilities*. Berikut adalah rancangan *CRC Cards* yang di tunjukan pada gambar 3.2 .



**Gambar 3.2** CRC cards

### 3.1.5.2. Perancangan Rekomendasi

Dalam perencanaan ini, penulis telah merancang metode *collaborative filtering*. Alur yang menggambarkan proses dari sistem perekomendasi dengan metode tersebut dapat ditunjukkan pada gambar 3.3.



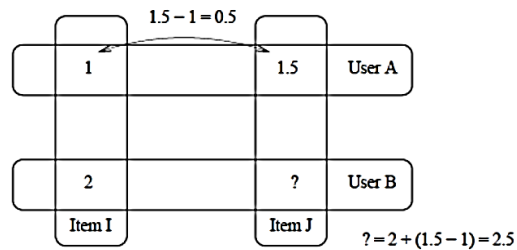
**Gambar 3.3** Alur proses rekomendasi

Tahap pertama dalam proses utama menghasilkan rekomendasi adalah sistem menampilkan *item* / tukang yang akan dilihat *user* dan diberi rating oleh *user*. Sistem akan menangkap masukan rating oleh *user*  $u$  untuk *item*  $i$  yang kemudian akan dilakukan perhitungan saat *user* menutup aplikasi (*offline* / lokal). Karena proses perhitungan *offline*, *user* yang baru pertama kali meminta rekomendasi akan diberikan semua *item* untuk posisi *user* saat itu. Nilai hasil rating yang diberikan oleh *user* untuk *item* berupa bilangan bulat antara 1-5. Sedangkan nilai untuk *item-item* dimana *user* belum pernah memberi rating adalah 0.

Tahap selanjutnya adalah perhitungan *similarity* dengan menggunakan algoritma *slope one*. Algoritma ini memerlukan informasi dari *user* lain yang merating *item* / tukang yang sama dengan *user* yang akan diprediksi ratingnya (contoh *user* A) dan informasi *item* yang telah diberi rating oleh *user* A. Algoritma *slope one* berjalan dengan menghitung selisih rating antara dua *item* yang ada.

Selisih rating tersebut digunakan untuk memprediksi berapa besar nilai rating yang diberikan terhadap sebuah *item* untuk *user* A. Jadi ada dua input untuk algoritma slope one yaitu rating dari *user* A dan *item* mana yang akan diprediksi. Oleh karena itu jika *user* belum pernah memberi satu rating pun terhadap *item*, maka prediksi tidak dapat dilakukan.

Contoh bila terdapat dua *user* A dan B, serta dua *item* I dan J. *user* A memberi rating 1 untuk *item* I dan 1,5 untuk *item* J. sedangkan *user* B memberi rating 2 untuk *item* I saja. Dari sini dapat dihitung selisih rating pelanggan A untuk *item* I dan *item* J adalah  $1.5 - 1 = 0,5$ . Berdasarkan nilai selisih ini dapat diprediksi bahwa *user* B akan memberi rating terhadap *item* J sebesar  $2 + 0,5 = 2,5$ . Ilustrasi contoh ini dapat dilihat pada gambar 3.4.



**Gambar 3.4.** Skema Algoritma Slope One

Pendekatan algoritma *slope one* ada dua tahap, pertama menghitung rata-rata selisih rating antara *item* berdasarkan data rating dari pelanggan, kedua menghitung prediksi menggunakan data rata-rata selisih rating pada tahap pertama dan data rating *item* dari pelanggan. Rumus untuk menghitung rata-rata selisih rating adalah sebagai berikut :

$$dev_{j,i} = \sum_{u \in S_{j,i}(\chi)} \frac{u_j - u_i}{card(S_{j,i}(\chi))}$$

Dimana :  $dev_{j,i}$  adalah rata-rata selisih rating *item* I dan J

$u$  = *user* yang merating *item* I dan J



$u_j$  = rating *item* j

$u_i$  = rating *item* i

$X$  = himpunan rating *item* dari para *user*

$S_{j,i}(X)$  = himpunan bagian dari  $X$  yang mengandung *item* i dan j

$Card(S_{j,i}(X))$  = jumlah elemen di  $S_{j,i}(X)$

Sedangkan rumus untuk menghitung nilai prediksi adalah sebagai berikut :

$$P(u)_j = \frac{1}{card(R_j)} \sum_{i \in R_j} (dev_{j,i} + u_i)$$

Dimana :  $P(u)_j$  = nilai prediksi algoritma slope one untuk *item* j pada *user* P

$R_j$  = himpunan *item* dari pelanggan yang memberi rating pada *item* j yang bersesuaian dengan *item* yang diberi rating oleh *user* P

$Card(R_j)$  = jumlah elemen di  $R_j$

### 3.1.5.3. *Prototype*

*Prototype* atau prototipe adalah sebuah metode dalam pengembangan produk dengan cara membuat rancangan, sampel, atau model dengan tujuan pengujian konsep atau proses kerja dari produk. Berikut adalah prototipe yang sudah penulis buat beserta penjelasannya :



**Gambar 3.5** Tampilan Awal

Pada tampilan awal, *user* akan diberi tampilan selamat datang yang menandakan bahwa aplikasi telah terpasang, lalu *user* diberikan 2 pilihan login yaitu login sebagai pengguna jasa atau login sebagai tukang sang penyedia jasa untuk para pengguna aplikasi Tulen

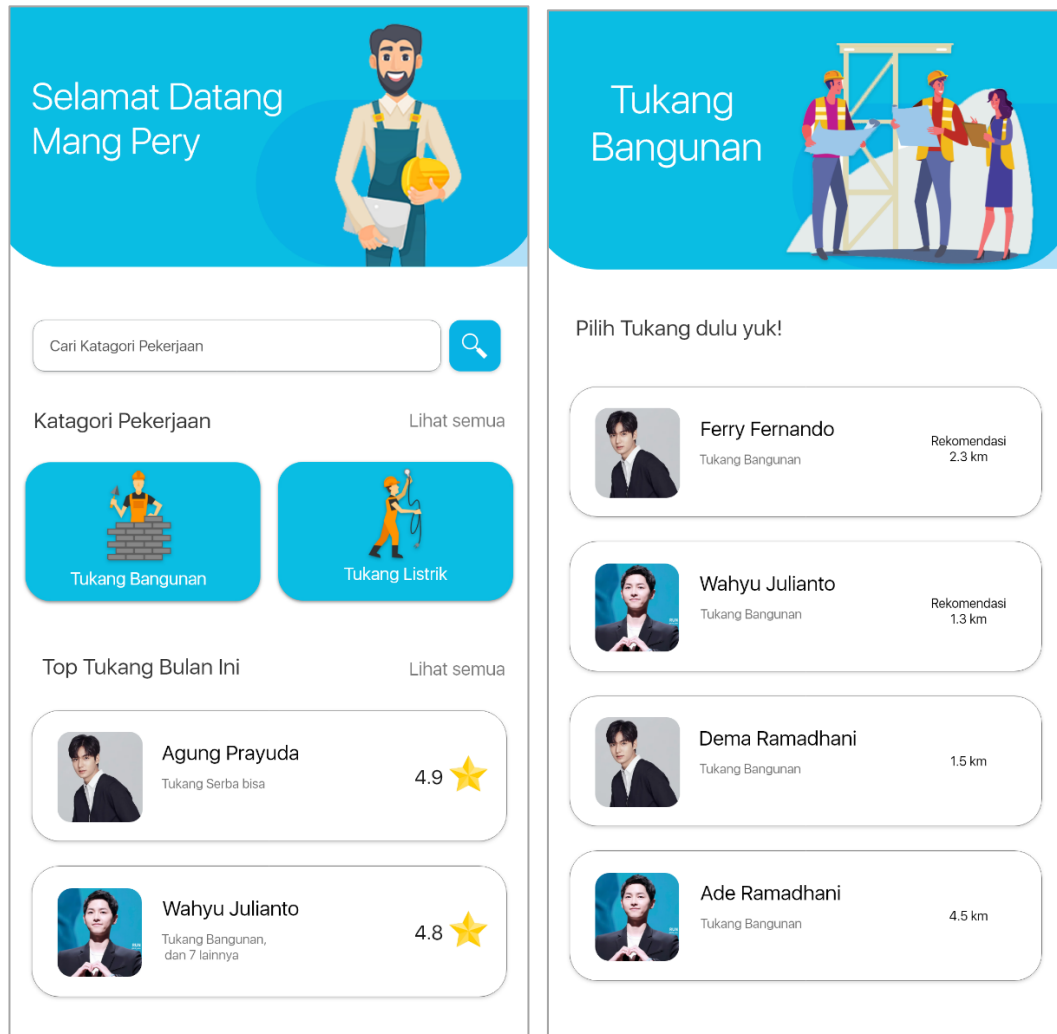
The image shows two mobile app screens side-by-side. The left screen is titled "Yuk Login Dulu" and features a construction worker icon. It has input fields for "masukan email" and "masukan password", a blue "Login" button, and social login options for Google and Facebook. The right screen is titled "Daftar dulu ya" and features a construction worker icon holding blueprints. It has input fields for "masukan email", "masukan password", and "masukan ulang password", a blue "Daftar" button, and social login options for Google and Facebook.

**Gambar 3.6** login – register

Setelah memilih bagaimana *user* login, maka akan ditampilkan halaman loginnya, tampilan login untuk Tukang dan Pengguna sama saja. Login di aplikasi ini menggunakan *email* dan *password* atau bisa juga dengan akun google dan facebook. Jika *user* belum memiliki akun, maka wajib mendaftarkan dirinya dengan cara memilih tulisan “daftar sini” untuk mendaftar lalu akan di arahkan ke halaman register atau daftar.

Disini *user* diwajibkan memasukan *email* yang aktif untuk menerima kode serta memasukan *password* 2 kali untuk konfirmasi bahwa password tidak salah.

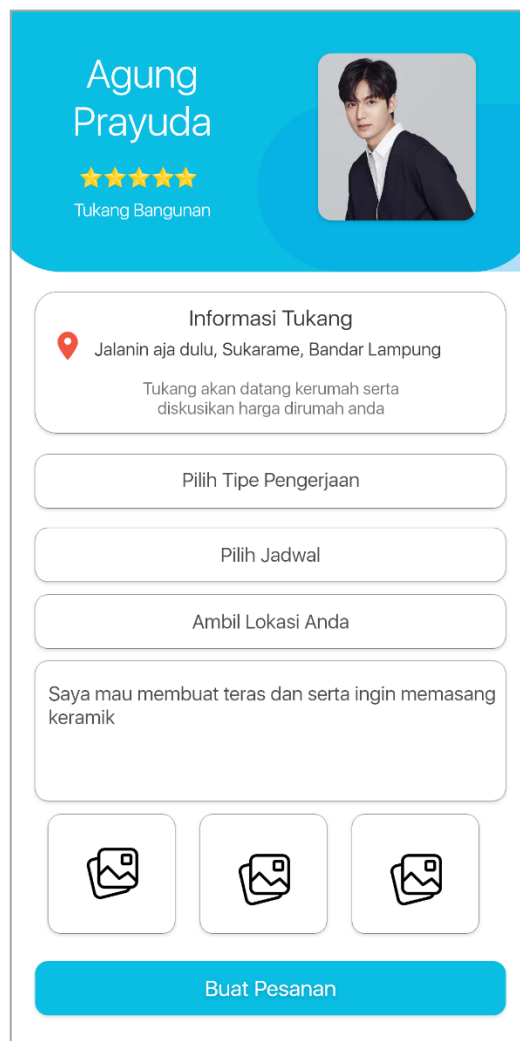
Selain itu tersedia juga cara mendaftar dengan akun google dan facebook untuk memudahkan *user* login tanpa memasukan email dan password.



**Gambar 3.7** Menu utama *user* dan pilih tukang

Akan penulis jelaskan untuk menu *user* terlebih dahulu, setelah *user* berhasil *login* menggunakan akunnya maka akan ditampilkan menu utama khusus *user* atau pengguna jasa tukang. Di menu utama *user* bisa melihat nama, katagori pekerjaan, *search box* untuk mencari katagori pekerjaan, dan *top* tukang berdasarkan ratingnya.

Kemudian setelah *user* memilih pekerjaan dari katagori, selanjutnya akan ditampilkan menu pilih tukang, disini rekomendasi tukang ditampilkan berdasarkan rating si tukang dan *user* bisa melihat jarak rumah tukang dari lokasinya. Tukang yang direkomendasikan akan terdapat tulisan “Rekomendasi” di atas jarak, ini akan mempermudah *user* untuk mencari tukang yang kompeten. *User* juga tidak harus memilih dari rekomendasi, *user* bisa melihat dan memilih tukang lain jika *user* mau.

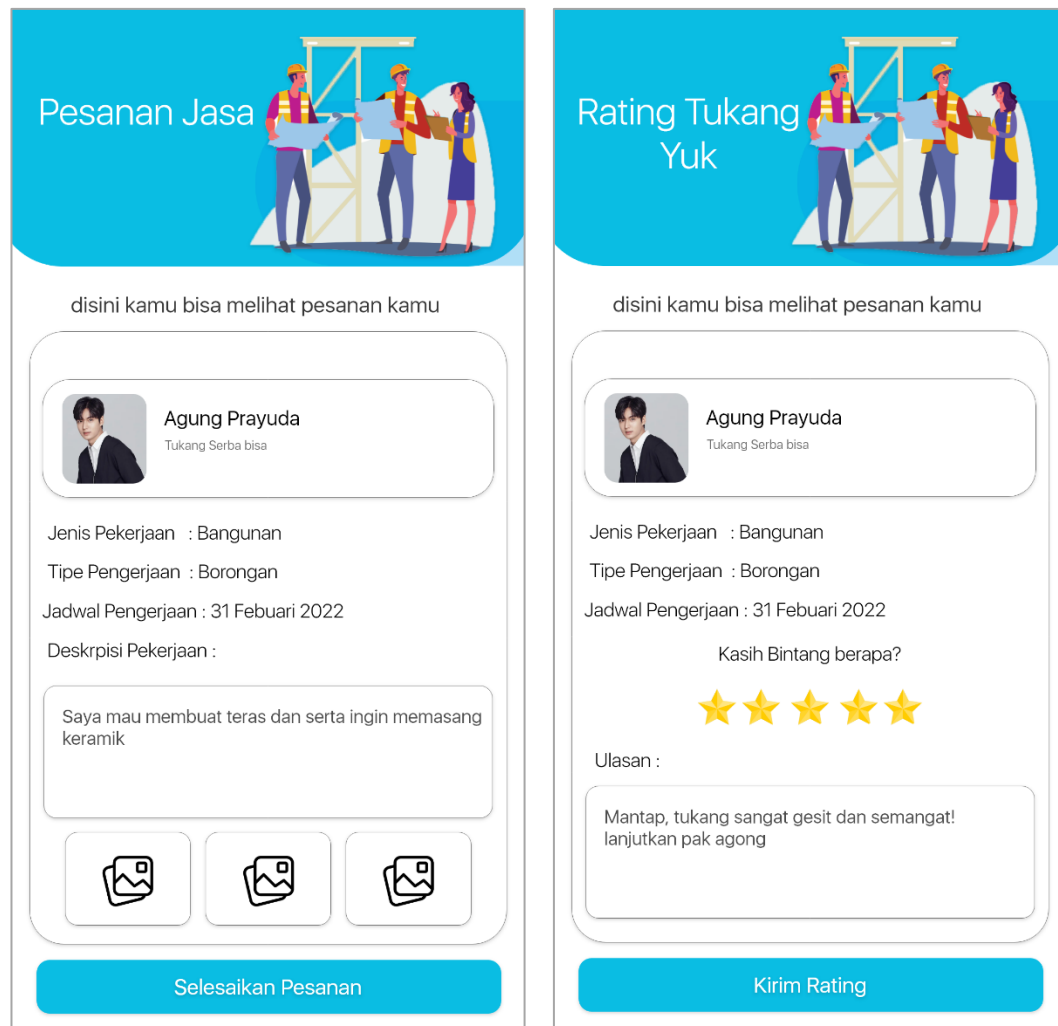


**Gambar 3.8** tampilan buat pesanan

Setelah *user* memilih tukang dari menu pilih tukang, selanjutnya *user* akan di arahkan ke halaman buat pesanan, halaman ini berisi nama tukang yang dipilih

beserta informasi tukang tersebut. Informasi bisa berisikan alamat rumah, harga jasa mulai dari sekian dan lain-lain. Kemudian *user* diminta untuk memilih tipe pengerjaan. Berdasarkan wawancara yang sudah penulis lakukan, tukang memiliki 2 tipe pengerjaan yaitu Borongan dan Harian.

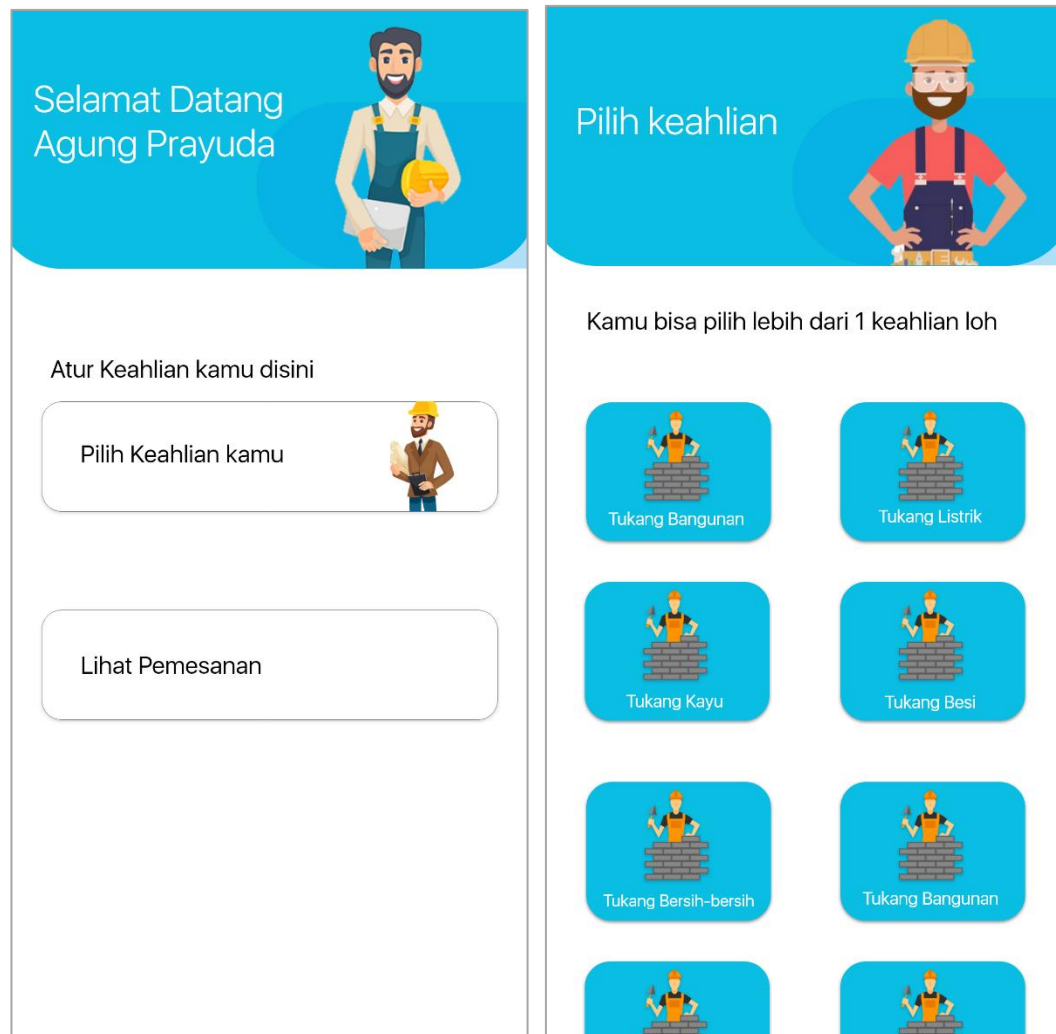
Borongan adalah membayar semua harga jasa dan menentukan berapa hari kerja tukang kemudian Harian adalah upah tukang sesuai dengan berapa hari pengerjaannya. Setelah *user* memilih tipe pengerjaan maka dilanjutkan dengan mengisi jadwal untuk pengerjaan proyek lalu dengan menekan tombol “ambil lokasi” untuk menentukan lokasi proyek yang akan dikerjakan. Tombol ini akan mengambil koordinat si *user*, setelah itu *user* mengisi deskripsi pekerjaan serta foto untuk mendukung deskripsinya.



**Gambar 3.9** Pesanan aktif dan Rating

Kemudian *user* yang telah membuat pesanan bisa melihat pesanan aktif di menu utama. Halaman ini berfungsi untuk melihat detail pesanan *user* yang telah dibuat sebelumnya. Di bawah halaman terdapat tombol selesaikan pesanan jika tukang telah menyelesaikan pekerjaannya.

Setelah *user* menekan tombol tersebut, maka *user* akan diarahkan ke halaman rating tukang, di halaman ini berisi informasi detail pesanan dan *user* diwajibkan memberi rating 1-5 bintang serta memberi ulasan kepada tukang. Data ini akan sangat berfungsi untuk memberikan rekomendasi selanjutnya kepada *user* lain



**Gambar 3.10** Menu utama tukang dan pilih keahlian

Setelah tukang login dengan akunnya, maka akan diarahkan menu utama khusus tukang, di menu ini terdapat 2 pilihan yaitu pilih keahlian dan lihat pemesanan. Tukang diwajibkan untuk memilih keahliannya terlebih dahulu agar namanya terdaftar di database kategori pekerjaan yang dipilih nya. Untuk saat ini ada beberapa katagori pekerjaan yaitu Bangunan untuk keperluan rumah atau gedung, listrik untuk instalasi atau memperbaiki kelistrikan, kayu untuk furniture rumah, besi untuk pagar atau tralis rumah, dan tukang bersih-berish untuk membersihkan halaman atau dalam rumah.



Tukang bisa memilih lebih dari 1 keahlian di atas lalu menyimpan pilihannya dengan tombol simpan dibawah. Kemudian di menu utama terdapat menu “lihat pemesanan” untuk melihat siapa yang memesan jasa si tukang. Menu ini dapat dilihat pada gambar 3.11 dibawah ini



**Gambar 3.11** lihat pemesanan

Di fitur ini terdapat informasi pemesanan dan informasi pengguna jasa seperti no telpon yang bisa di hubungi dan alamat rumah yang di jadikan tempat proyek pengerjaan.

### 3.1.6. Coding

Pada tahap ini pengkodean dengan mengikuti rancangan yang telah dibuat sebelumnya di tahap design. Untuk pengkodean menggunakan Android Studio dan menggunakan bahasa pemrograman Java.

### 3.1.7. Testing

#### 3.1.7.1. Acceptance Testing

Tes penerimaan menggunakan ISO 925010 Setiap test harus memiliki hasil yang diharapkan dari sistem berdasarkan kriteria. *user* bertanggung jawab untuk memverifikasi test dan meninjau hasil test untuk memutuskan lolos atau tidaknya sebuah produk untuk di rilis.

### 3.1.8. Release

Pada tahap ini akan dilakukan perilsan penuh terhadap aplikasi yang telah dibuat dengan melewati 4 tahapan sebelumnya.

## 3.2. Jadwal Penelitian

Berikut adalah jadwal penelitian proposal skripsi yang dibuat oleh penulis, dapat dilihat pada Tabel 3.5 sebagai berikut :

**Tabel 3.5** Jadwal Penelitian

No	Kegiatan Penelitian	Tahun 2024				
		Januari	Febuari	Maret	April	Mei
	Studi Literatur					
	Wawancara					
	Design Tampilan					
	Penyusunan Proposal Skripsi					
	Seminar Proposal					

	Produksi					
	Pengujian					
	Rilis					
	Penyusunan Skripsi					
	Sidang Skripsi					

Keterangan :

Warna hijau : Telah dilaksanakan

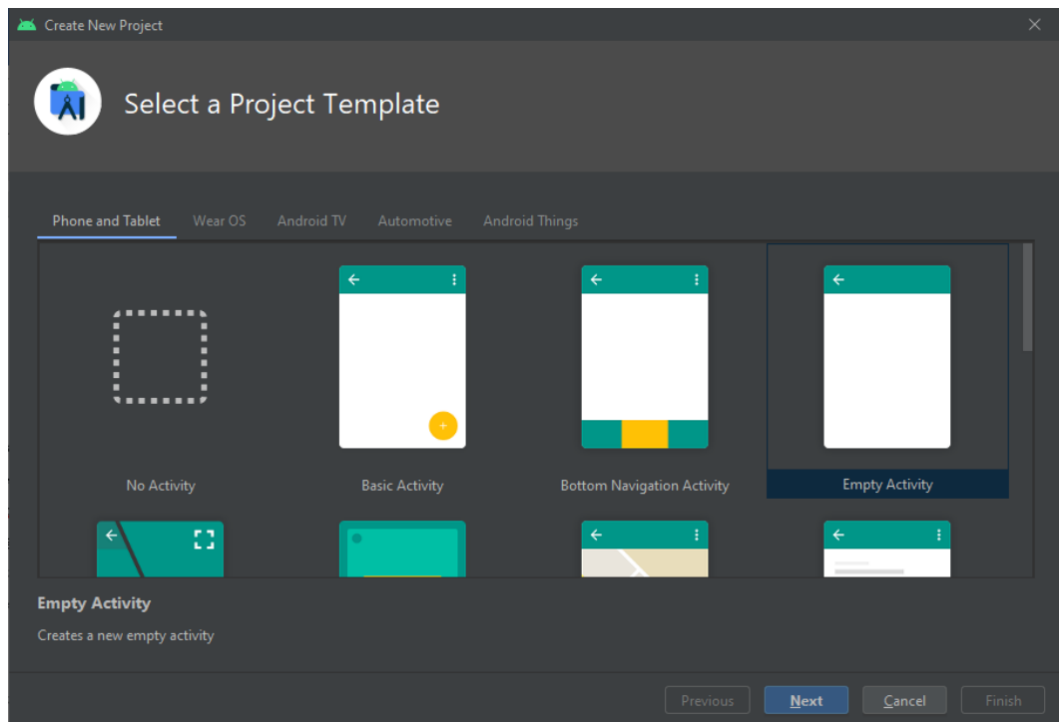
Warna merah : Belum dilaksanakan

## BAB IV

### IMPLEMENTASI

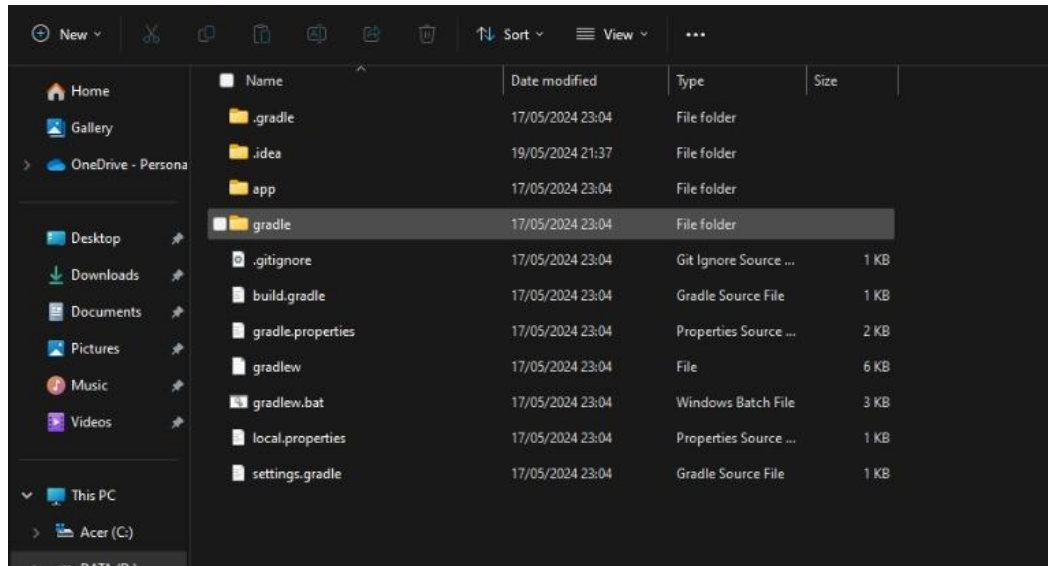
#### 4.1 Implementasi Pengkodean

Pada tahap koding ini adalah tahapan dari langkah – langkah pemrograman dengan menulis kode dalam bahasa pemrograman dan menjadi sebuah aplikasi. Aplikasi TuLen dibuat berdasarkan *design interface* dan rancangan yang telah di rencanakan sebelumnya. Semua yang telah di *design* dan dirancang akan dibuat menggunakan android studio dengan bahasa pemrograman java. Langkah awal dalam pembuatan aplikasi menggunakan android studio adalah dengan membuat *project* awal pada android studio seperti gambar berikut :



**Gambar 4.1** Pembuatan *project* pada Android Studio

Setelah project dibuat maka akan terdapat folder project yang telah kita buat tadi dan memiliki struktur folder projek sebagai berikut :



**Gambar 4.2** Struktur Project Android Studio

Setelah membuat *project* pada android studio, langkah selanjutnya adalah menambahkan *dependencies* yang diperlukan untuk kepentingan *project*. Berikut adalah *dependencies* yang digunakan untuk membangun aplikasi TuLen :

```
dependencies {

    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'com.google.android.material:material:1.12.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'

    // firebase
    implementation platform("com.google.firebase:firebase-bom:29.3.1")
    implementation 'com.google.firebase:firebase-analytics'
    implementation 'com.google.firebase:firebase-firestore'
    implementation 'androidx.activity:activity:1.9.0'
    implementation 'com.google.firebase:firebase-storage'

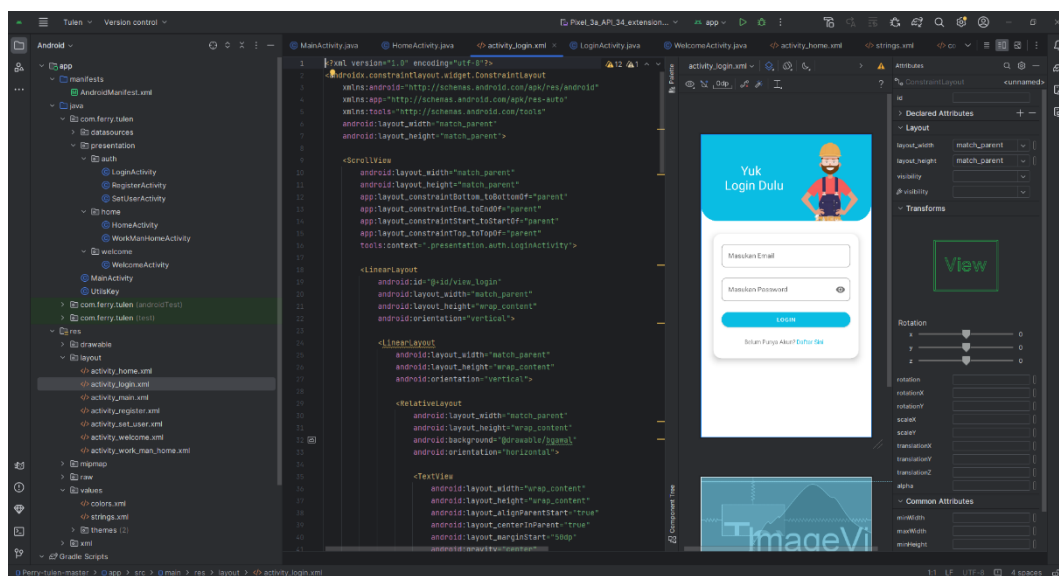
    implementation 'com.google.android.gms:play-services-location:21.2.0'
    // Add this to your app/build.gradle
    implementation 'com.firebase:geofire-android-common:3.1.0'

    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
```

**Gambar 4.3** *Dependencies* Aplikasi TuLen

### 4.1.1 Implementasi Koding Tampilan

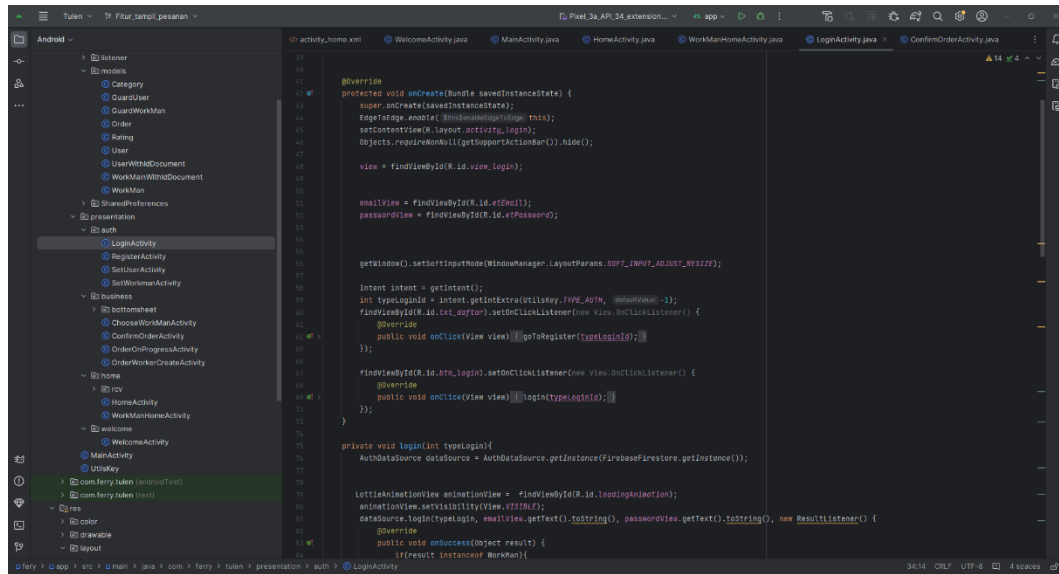
Pada tahap ini merupakan tahapan menulis (*coding*) untuk tampilan aplikasi dari aplikasi pertama kali dibuka (*login*). Penulisan kode untuk *design interface* ini berdasarkan *design* yang telah dirancang sebelumnya. *Design* yang telah dirancang pada Adobe XD akan di *export* kedalam bentuk *file .png* lalu akan di *import* ke dalam folder *asset* di *project* android studio dan kemudian dituliskan menjadi kode di android studio seperti yang di tunjukan pada gambar 4.4 di bawah ini :



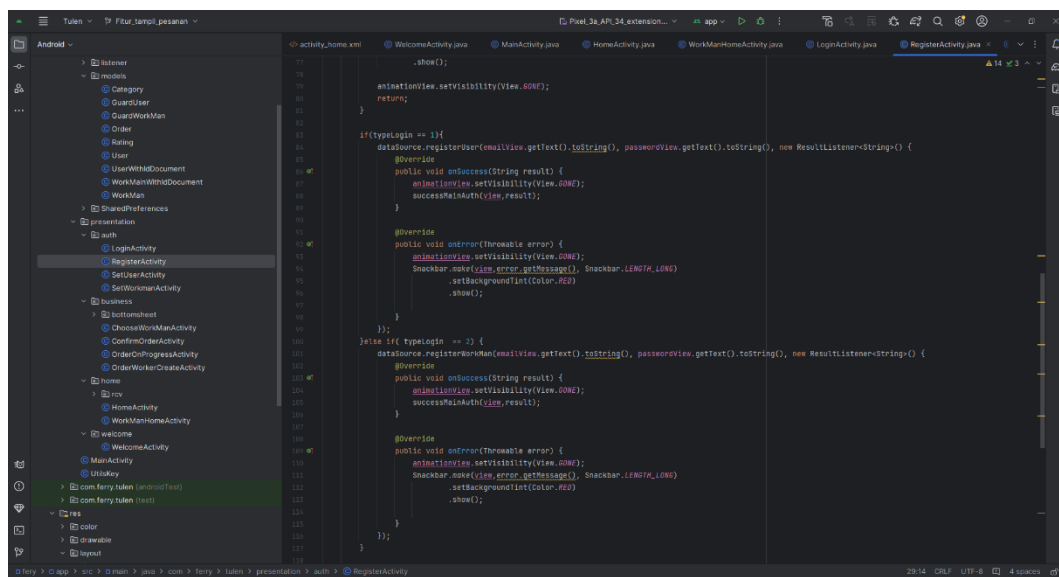
Gambar 4.4 penulisan kode tampilan

### 4.1.2 Implementasi Koding Fitur Masuk dan Daftar

Tahapan ini adalah tahapan koding untuk *user* masuk dan *verifikasi* dasar. Fitur ini digunakan untuk memasuki menu utama untuk menggunakan seluruh fitur yang ada pada aplikasi TuLen. Jika *user* belum mempunyai akun, maka *user* wajib untuk melakukan pendaftaran akun dengan email. Berikut adalah gambar koding fitur masuk dan daftar yang ditunjukkan pada gambar 4.5 dan 4.6 :



Gambar 4.5 Koding fitur masuk



Gambar 4.6 Koding fitur daftar

### 4.1.3 Implementasi Koding Kelengkapan Akun

Pada tahapan ini jika user belum pernah atau baru saja melakukan login akun untuk pertama kali, maka akan diminta untuk mengisi beberapa informasi tambahan untuk keperluan aplikasi nanti. Implementasi ini bisa dilihat pada gambar 4.7 dan gambar 4.8 :

```

    typeLogin;
    idUser;
    String iduser;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_set_user);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        }};

        SharedPreferencesHelper sharedPreferences = new SharedPreferencesHelper(this);
        String typeLoginString = sharedPreferences.getString(SharedPreferencesHelper.KEY_TYPE_LOGIN, null);
        typeLogin = Integer.parseInt(typeLoginString);

        idUser = sharedPreferences.getString(SharedPreferencesHelper.KEY_ID_USER, null);

        System.out.println("Debug: typeLogin : " + typeLogin + " iduser : " + iduser);

        fullKame = findViewById(R.id.txtFullKame);
        alamatKontak = findViewById(R.id.txtAlamat);
        nomorPhone = findViewById(R.id.txtPhoneNumber);
        job = findViewById(R.id.txtJob);

        findViewById(R.id.btnSave).setOnClickListener(new View.OnClickListener() {
            @Override
    
```

Gambar 4.7 Koding Kelengkapan informasi untuk User

```

    String iduser;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_set_workman);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        }};

        SharedPreferencesHelper sharedPreferences = new SharedPreferencesHelper(this);
        String typeLoginString = sharedPreferences.getString(SharedPreferencesHelper.KEY_TYPE_LOGIN, null);
        typeLogin = Integer.parseInt(typeLoginString);

        iduser = sharedPreferences.getString(SharedPreferencesHelper.KEY_ID_USER, null);

        System.out.println("Debug: typeLogin : " + typeLogin + " iduser : " + iduser);

        fullKame = findViewById(R.id.txtFullKame);
        alamatKontak = findViewById(R.id.txtAlamat);
        nomorPhone = findViewById(R.id.txtPhoneNumber);
        // job = findViewById(R.id.txtJob);
        spinnerJob = findViewById(R.id.txtJob);

        CategoryDataSource dataSource = CategoryDataSource.getInstance(FirebaseFirestore.getInstance());
        dataSource.getAllCategory(new ResultListenerList<Category>() {
            @Override
            public void onSuccess(List<Category> result) {
                ArrayAdapter<Category> cate = new ArrayAdapter<Result>();
    
```

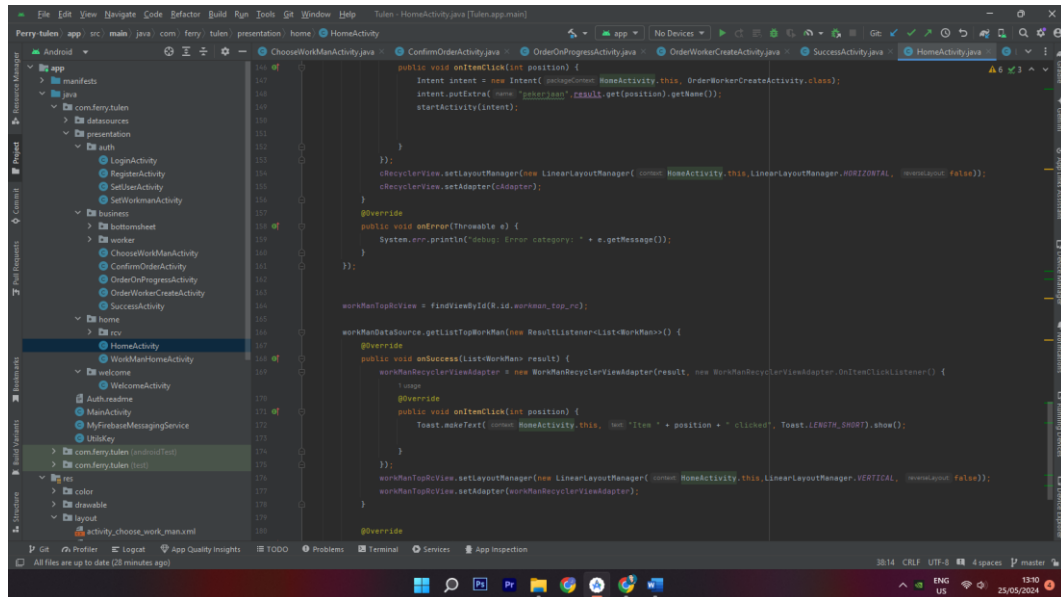
Gambar 4.8 Koding Kelengkapan informasi untuk Tukang

#### 4.1.4 Implementasi Koding Menu Utama

Pada tahapan ini merupakan tahapan penulisan kode untuk tampilan menu utama setelah user atau tukang melakukan login, di dalam menu utama *user* terdapat tombol untuk memilih katagori pekerjaan yang diinginkan dan beberapa tukang dengan *rating* yang tinggi pada bulan ini. Kemudian di menu utama tukang akan



terdapat dua tombol menu yaitu pesanan dan keahlian, koding dua menu ini dapat di lihat pada gambar 4.9 dan gambar 4.10 di bawah ini :



```

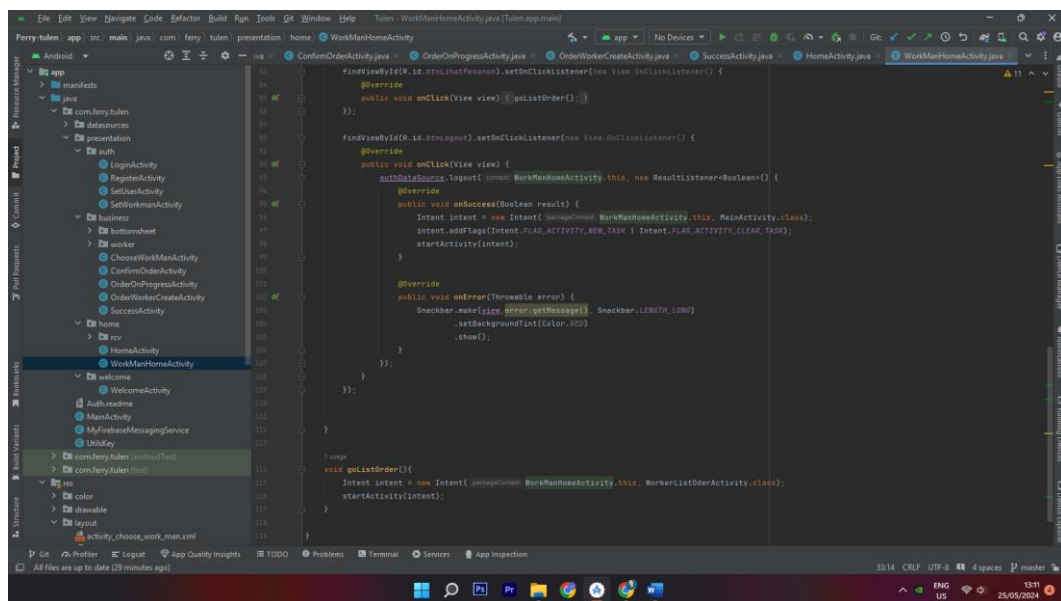
public void onItemClick(int position) {
    Intent intent = new Intent(getApplicationContext(), OrderWorkerCreateActivity.class);
    intent.putExtra("new_orderman", dataList.get(position).getName());
    startActivity(intent);
}

@Override
public void onError(Throwable e) {
    System.err.println("Error category: " + e.getMessage());
}

workManTopRecyclerView = findViewById(R.id.workman_top_rcv);
workManDataSource.getListWorkMan(new ResultListener<List<WorkMan>>() {
    @Override
    public void onSuccess(List<WorkMan> result) {
        workManRecyclerViewAdapter = new WorkManRecyclerViewAdapter(result, new WorkManRecyclerViewAdapter.OnItemClickListener() {
            @Override
            public void onItemClick(int position) {
                Toast.makeText(getApplicationContext(), "Item " + position + " clicked", Toast.LENGTH_SHORT).show();
            }
        });
        workManTopRecyclerView.setLayoutManager(new LinearLayoutManager(getApplicationContext(), LinearLayoutManager.VERTICAL, false));
        workManTopRecyclerView.setAdapter(workManRecyclerViewAdapter);
    }
});
}

```

**Gambar 4.9** Koding Menu Utama User



```

findViewById(R.id.btnJustPresman).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        getListOrder();
    }
});

findViewById(R.id.btnLogout).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        authDataSource.logout(getApplicationContext(), new ResultListener<Boolean>() {
            @Override
            public void onSuccess(Boolean result) {
                Intent intent = new Intent(getApplicationContext(), MainActivity.class);
                intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
                startActivity(intent);
            }
        });
    }
});

@Override
public void onError(Throwable error) {
    Snackbar.make(view, error.getMessage(), Snackbar.LENGTH_LONG)
        .setBackgroundTint(Color.RED)
        .show();
}

@Override
void getListOrder() {
    Intent intent = new Intent(getApplicationContext(), WorkerListOrderActivity.class);
    startActivity(intent);
}
}

```

**Gambar 4.10** Koding Menu Utama Tukang

#### 4.1.5 Implementasi Koding Buat Pekerjaan

Tahapan ini adalah tahapan penulisan kode untuk fitur Buat Pekerjaan yang telah dirancang sebelumnya. Fitur buat pekerjaan dibuat oleh *user* untuk membuat pekerjaan dan memesan tukang dengan mengisi beberapa informasi yang diperlukan pada form yang telah disediakan dan memilih tukang yang direkomendasikan atau tukang yang dipilih oleh user sendiri. Berikut koding pada fitur buat pekerjaan bisa dilihat pada gambar 4.11 dibawah ini :

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_order_creator);

    Objects.requireNonNull(getSupportActionBar()).hide();
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets } -> {
        @Override
        Insets windowInsetsCompat = insets.getWindowInsets().getInsetsIgnoringSystemBars();
        Insets navigationBars = insets.getInsets(WindowInsetsCompat.Type.navigationBars());
        v.setPadding(v.getPaddingLeft(), v.getPaddingTop(), v.getPaddingRight(), navigationBars.bottom);
        return insets;
    });
}

etDescription = findViewById(R.id.textJob);
TextView pekerjaanTxt = findViewById(R.id.textPekerjaan);
Intent intent = getIntent();

pekerjaan = intent.getStringExtra("nama_pekerjaan");
pekerjaanTxt.setText(pekerjaan);

FusedLocationClient = LocationServices.getFusedLocationProviderClient(this);

findViewById(R.id.btnAbdiLokasi).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) { getLocation(); }
});

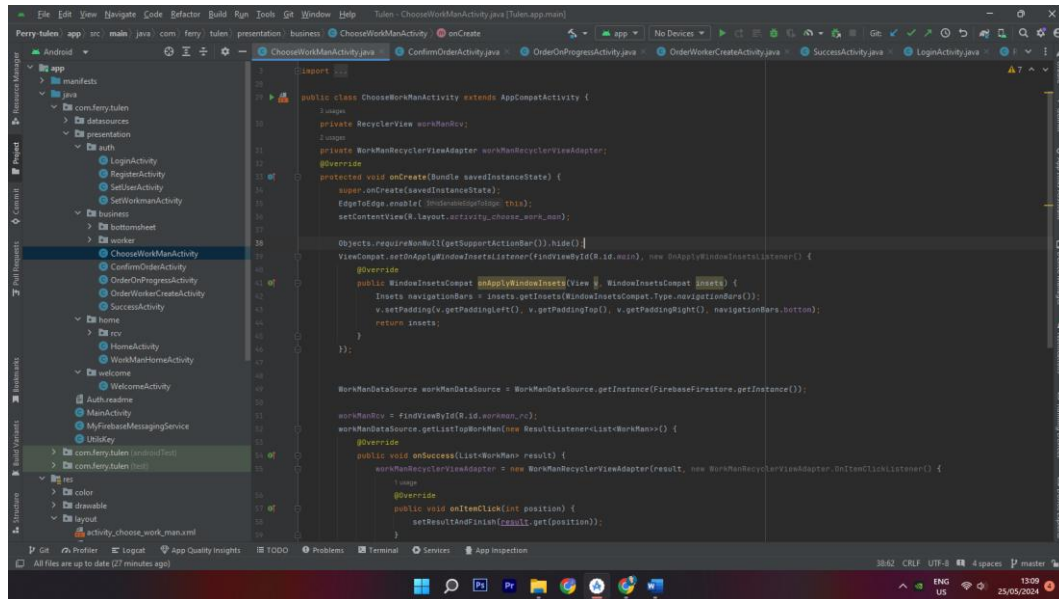
findViewById(R.id.btnPilihTukang).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

```

Gambar 4.12 Koding Buat Pekerjaan

#### 4.1.6 Implementasi Koding Cari Tukang

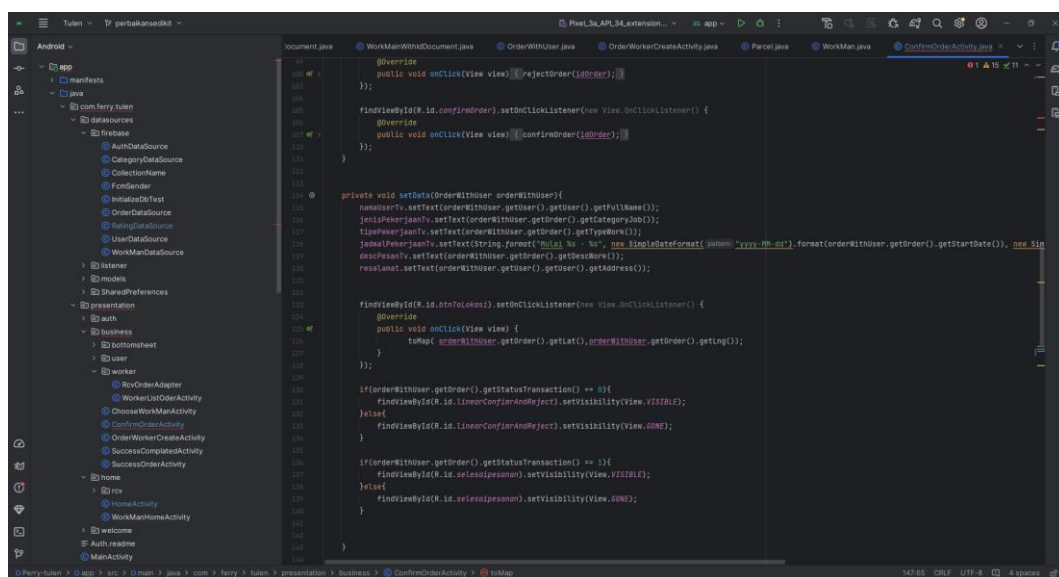
Tahapan ini adalah tahapan penulisan kode pada fitur cari tukang. Dimana pada tahapan ini akan menampilkan list tukang pada fitur buat pekerjaan yang dibuat oleh *user*. List tukang didapat dari *firebase* dan di *filter* oleh algoritma *slope one* sebagai rekomendasi dan ditampilkan dibagian paling atas, namun jika *user* belum pernah melakukan rating sama sekali, maka algoritma tidak bisa di jalankan dan list tukang yang ditampilkan adalah acak. Implementasi koding cari tukang dapat dilihat pada gambar 4.13 dibawah ini :



Gambar 4.14 Koding Cari Tukang

#### 4.1.7 Implementasi Koding Konfirmasi Pesanan di Tukang

Implementasi koding konfirmasi pesanan pada tukang merupakan tahapan penulisan kode pemrograman agar tukang bisa melihat pesanan yang masuk dan yang berlangsung atau yang sedang dikerjakan serta terdapat tombol untuk menyelesaikan pekerjaannya. Berikut adalah koding konfirmasi pesanan pada gambar 4.15



Gambar 4.15 Koding Konfirmasi Pesanan

### 4.1.8 Implementasi Algoritma *Slope One*

Algoritma *Slope One* digunakan untuk menghitung dan memberikan rekomendasi list tukang kepada *user* untuk memilih tukang pada saat membuat pesanan pekerjaan. Berikut adalah implementasi algoritma *slope one* yang di tunjukkan pada gambar 4.16

```

30 // }
31 // }
32 // }
33 // }
34 // Update the diffs and freq maps with the new data
35 public void update(Map<String, Map<String, Double>> data) {
36 // Iterate over each user in the dataset
37 for (Map.Entry<String, Map<String, Double>> userEntry : data.entrySet()) {
38 String user = userEntry.getKey();
39 Map<String, Double> userRatings = userEntry.getValue();
40 // Iterate over each pair of items rated by the user
41 for (Map.Entry<String, Double> itemEntry1 : userRatings.entrySet()) {
42 String item1 = itemEntry1.getKey();
43 Double rating1 = itemEntry1.getValue();
44 for (Map.Entry<String, Double> itemEntry2 : userRatings.entrySet()) {
45 if (itemEntry1 != itemEntry2) {
46 String item2 = itemEntry2.getKey();
47 Double rating2 = itemEntry2.getValue();
48 // Calculate the difference in ratings between the two items
49 Double diff = rating1 - rating2;
50 // Update the diffs and freq maps with the new difference
51 diffs.computeIfAbsent(item1, k -> new HashMap<>()).put(item2, diffs.getOrDefault(item1, Collections.emptyMap()).getOrDefault(item2, 0.0) + diff);
52 freqs.computeIfAbsent(item1, k -> new HashMap<>()).put(item2, freqs.getOrDefault(item1, Collections.emptyMap()).getOrDefault(item2, 0.0) + 1);
53 }
54 }
55 }
56 }
57 // }
58 // }
59 // }
60 // }
61 // }
62 // }
63 // }
64 // }
65 // }
66 // }
67 // }
68 // }
69 // }
70 // }
71 // }
72 // }
73 // }
74 // }
75 // }
76 // }
77 // }
78 // }
79 // }
80 // }
81 // }
82 // }
83 // }
84 // }
85 // }
86 // }
87 // }
88 // }
89 // }
90 // }
91 // }
92 // }
93 // }
94 // }
95 // }
96 // }
97 // }
98 // }
99 // }
100 // }
101 // }
102 // }
103 // }
104 // }
105 // }
106 // }
107 // }
108 // }
109 // }
110 // }
111 // }
112 // }
113 // }
114 // }
115 // }
116 // }
117 // }
118 // }
119 // }
120 // }
121 // }
122 // }
123 // }
124 // }
125 // }
126 // }
127 // }
128 // }
129 // }
130 // }
131 // }
132 // }
133 // }
134 // }
135 // }
136 // }
137 // }
138 // }
139 // }
140 // }
141 // }
142 // }
143 // }
144 // }
145 // }
146 // }
147 // }
148 // }
149 // }
150 // }
151 // }
152 // }
153 // }
154 // }
155 // }
156 // }
157 // }
158 // }
159 // }
160 // }
161 // }
162 // }
163 // }
164 // }
165 // }
166 // }
167 // }
168 // }
169 // }
170 // }
171 // }
172 // }
173 // }
174 // }
175 // }
176 // }
177 // }
178 // }
179 // }
180 // }
181 // }
182 // }
183 // }
184 // }
185 // }
186 // }
187 // }
188 // }
189 // }
190 // }
191 // }
192 // }
193 // }
194 // }
195 // }
196 // }
197 // }
198 // }
199 // }
200 // }
201 // }
202 // }
203 // }
204 // }
205 // }
206 // }
207 // }
208 // }
209 // }
210 // }
211 // }
212 // }
213 // }
214 // }
215 // }
216 // }
217 // }
218 // }
219 // }
220 // }
221 // }
222 // }
223 // }
224 // }
225 // }
226 // }
227 // }
228 // }
229 // }
230 // }
231 // }
232 // }
233 // }
234 // }
235 // }
236 // }
237 // }
238 // }
239 // }
240 // }
241 // }
242 // }
243 // }
244 // }
245 // }
246 // }
247 // }
248 // }
249 // }
250 // }
251 // }
252 // }
253 // }
254 // }
255 // }
256 // }
257 // }
258 // }
259 // }
260 // }
261 // }
262 // }
263 // }
264 // }
265 // }
266 // }
267 // }
268 // }
269 // }
270 // }
271 // }
272 // }
273 // }
274 // }
275 // }
276 // }
277 // }
278 // }
279 // }
280 // }
281 // }
282 // }
283 // }
284 // }
285 // }
286 // }
287 // }
288 // }
289 // }
290 // }
291 // }
292 // }
293 // }
294 // }
295 // }
296 // }
297 // }
298 // }
299 // }
300 // }
301 // }
302 // }
303 // }
304 // }
305 // }
306 // }
307 // }
308 // }
309 // }
310 // }
311 // }
312 // }
313 // }
314 // }
315 // }
316 // }
317 // }
318 // }
319 // }
320 // }
321 // }
322 // }
323 // }
324 // }
325 // }
326 // }
327 // }
328 // }
329 // }
330 // }
331 // }
332 // }
333 // }
334 // }
335 // }
336 // }
337 // }
338 // }
339 // }
340 // }
341 // }
342 // }
343 // }
344 // }
345 // }
346 // }
347 // }
348 // }
349 // }
350 // }
351 // }
352 // }
353 // }
354 // }
355 // }
356 // }
357 // }
358 // }
359 // }
360 // }
361 // }
362 // }
363 // }
364 // }
365 // }
366 // }
367 // }
368 // }
369 // }
370 // }
371 // }
372 // }
373 // }
374 // }
375 // }
376 // }
377 // }
378 // }
379 // }
380 // }
381 // }
382 // }
383 // }
384 // }
385 // }
386 // }
387 // }
388 // }
389 // }
390 // }
391 // }
392 // }
393 // }
394 // }
395 // }
396 // }
397 // }
398 // }
399 // }
400 // }
401 // }
402 // }
403 // }
404 // }
405 // }
406 // }
407 // }
408 // }
409 // }
410 // }
411 // }
412 // }
413 // }
414 // }
415 // }
416 // }
417 // }
418 // }
419 // }
420 // }
421 // }
422 // }
423 // }
424 // }
425 // }
426 // }
427 // }
428 // }
429 // }
430 // }
431 // }
432 // }
433 // }
434 // }
435 // }
436 // }
437 // }
438 // }
439 // }
440 // }
441 // }
442 // }
443 // }
444 // }
445 // }
446 // }
447 // }
448 // }
449 // }
450 // }
451 // }
452 // }
453 // }
454 // }
455 // }
456 // }
457 // }
458 // }
459 // }
460 // }
461 // }
462 // }
463 // }
464 // }
465 // }
466 // }
467 // }
468 // }
469 // }
470 // }
471 // }
472 // }
473 // }
474 // }
475 // }
476 // }
477 // }
478 // }
479 // }
480 // }
481 // }
482 // }
483 // }
484 // }
485 // }
486 // }
487 // }
488 // }
489 // }
490 // }
491 // }
492 // }
493 // }
494 // }
495 // }
496 // }
497 // }
498 // }
499 // }
500 // }
501 // }
502 // }
503 // }
504 // }
505 // }
506 // }
507 // }
508 // }
509 // }
510 // }
511 // }
512 // }
513 // }
514 // }
515 // }
516 // }
517 // }
518 // }
519 // }
520 // }
521 // }
522 // }
523 // }
524 // }
525 // }
526 // }
527 // }
528 // }
529 // }
530 // }
531 // }
532 // }
533 // }
534 // }
535 // }
536 // }
537 // }
538 // }
539 // }
540 // }
541 // }
542 // }
543 // }
544 // }
545 // }
546 // }
547 // }
548 // }
549 // }
550 // }
551 // }
552 // }
553 // }
554 // }
555 // }
556 // }
557 // }
558 // }
559 // }
560 // }
561 // }
562 // }
563 // }
564 // }
565 // }
566 // }
567 // }
568 // }
569 // }
570 // }
571 // }
572 // }
573 // }
574 // }
575 // }
576 // }
577 // }
578 // }
579 // }
580 // }
581 // }
582 // }
583 // }
584 // }
585 // }
586 // }
587 // }
588 // }
589 // }
590 // }
591 // }
592 // }
593 // }
594 // }
595 // }
596 // }
597 // }
598 // }
599 // }
600 // }
601 // }
602 // }
603 // }
604 // }
605 // }
606 // }
607 // }
608 // }
609 // }
610 // }
611 // }
612 // }
613 // }
614 // }
615 // }
616 // }
617 // }
618 // }
619 // }
620 // }
621 // }
622 // }
623 // }
624 // }
625 // }
626 // }
627 // }
628 // }
629 // }
630 // }
631 // }
632 // }
633 // }
634 // }
635 // }
636 // }
637 // }
638 // }
639 // }
640 // }
641 // }
642 // }
643 // }
644 // }
645 // }
646 // }
647 // }
648 // }
649 // }
650 // }
651 // }
652 // }
653 // }
654 // }
655 // }
656 // }
657 // }
658 // }
659 // }
660 // }
661 // }
662 // }
663 // }
664 // }
665 // }
666 // }
667 // }
668 // }
669 // }
670 // }
671 // }
672 // }
673 // }
674 // }
675 // }
676 // }
677 // }
678 // }
679 // }
680 // }
681 // }
682 // }
683 // }
684 // }
685 // }
686 // }
687 // }
688 // }
689 // }
690 // }
691 // }
692 // }
693 // }
694 // }
695 // }
696 // }
697 // }
698 // }
699 // }
700 // }
701 // }
702 // }
703 // }
704 // }
705 // }
706 // }
707 // }
708 // }
709 // }
710 // }
711 // }
712 // }
713 // }
714 // }
715 // }
716 // }
717 // }
718 // }
719 // }
720 // }
721 // }
722 // }
723 // }
724 // }
725 // }
726 // }
727 // }
728 // }
729 // }
730 // }
731 // }
732 // }
733 // }
734 // }
735 // }
736 // }
737 // }
738 // }
739 // }
740 // }
741 // }
742 // }
743 // }
744 // }
745 // }
746 // }
747 // }
748 // }
749 // }
750 // }
751 // }
752 // }
753 // }
754 // }
755 // }
756 // }
757 // }
758 // }
759 // }
760 // }
761 // }
762 // }
763 // }
764 // }
765 // }
766 // }
767 // }
768 // }
769 // }
770 // }
771 // }
772 // }
773 // }
774 // }
775 // }
776 // }
777 // }
778 // }
779 // }
780 // }
781 // }
782 // }
783 // }
784 // }
785 // }
786 // }
787 // }
788 // }
789 // }
790 // }
791 // }
792 // }
793 // }
794 // }
795 // }
796 // }
797 // }
798 // }
799 // }
800 // }
801 // }
802 // }
803 // }
804 // }
805 // }
806 // }
807 // }
808 // }
809 // }
810 // }
811 // }
812 // }
813 // }
814 // }
815 // }
816 // }
817 // }
818 // }
819 // }
820 // }
821 // }
822 // }
823 // }
824 // }
825 // }
826 // }
827 // }
828 // }
829 // }
830 // }
831 // }
832 // }
833 // }
834 // }
835 // }
836 // }
837 // }
838 // }
839 // }
840 // }
841 // }
842 // }
843 // }
844 // }
845 // }
846 // }
847 // }
848 // }
849 // }
850 // }
851 // }
852 // }
853 // }
854 // }
855 // }
856 // }
857 // }
858 // }
859 // }
860 // }
861 // }
862 // }
863 // }
864 // }
865 // }
866 // }
867 // }
868 // }
869 // }
870 // }
871 // }
872 // }
873 // }
874 // }
875 // }
876 // }
877 // }
878 // }
879 // }
880 // }
881 // }
882 // }
883 // }
884 // }
885 // }
886 // }
887 // }
888 // }
889 // }
890 // }
891 // }
892 // }
893 // }
894 // }
895 // }
896 // }
897 // }
898 // }
899 // }
900 // }
901 // }
902 // }
903 // }
904 // }
905 // }
906 // }
907 // }
908 // }
909 // }
910 // }
911 // }
912 // }
913 // }
914 // }
915 // }
916 // }
917 // }
918 // }
919 // }
920 // }
921 // }
922 // }
923 // }
924 // }
925 // }
926 // }
927 // }
928 // }
929 // }
930 // }
931 // }
932 // }
933 // }
934 // }
935 // }
936 // }
937 // }
938 // }
939 // }
940 // }
941 // }
942 // }
943 // }
944 // }
945 // }
946 // }
947 // }
948 // }
949 // }
950 // }
951 // }
952 // }
953 // }
954 // }
955 // }
956 // }
957 // }
958 // }
959 // }
960 // }
961 // }
962 // }
963 // }
964 // }
965 // }
966 // }
967 // }
968 // }
969 // }
970 // }
971 // }
972 // }
973 // }
974 // }
975 // }
976 // }
977 // }
978 // }
979 // }
980 // }
981 // }
982 // }
983 // }
984 // }
985 // }
986 // }
987 // }
988 // }
989 // }
990 // }
991 // }
992 // }
993 // }
994 // }
995 // }
996 // }
997 // }
998 // }
999 // }
1000 // }

```

Gambar 4.16 Implementasi Algoritma

## BAB V

### HASIL DAN PEMBAHASAN

#### 5.1 Hasil Pengujian

Pada bab ini membahas pengujian algoritma *slope one* dan data data yang akan dimasukan oleh pengguna di dalam aplikasi. Pengujian ini untuk mengetahui apakah aplikasi sudah bisa digunakan oleh pengguna atau tidak. Pengujian yang dilakukan terdiri dari ketepatan data yang di *inputkan* pengguna ke dalam *database firebase* dan setelah pengujian maka akan dilakukan pengujian fungsional dan *usability* dengan menggunakan metode *ISO 25010*.

##### 5.1.1 Pengujian Ketepatan

Pengujian ini dilakukan untuk mengetahui apakah data yang di *inputkan* pengguna ketika di *firebase* sesuai dengan yang dimasukan atau tidak dan tanpa adanya perubahan atau kerusakan setelah dimasukan pengguna ke *database firebase*.

##### 5.1.1.1 Pengujian Ketepatan Data Pada Fitur Pesan Tukang

Untuk mendapatkan hasil yang akurat maka diperlukan data yang akan digunakan untuk menguji fitur pesan tukang. Berikut adalah data yang akan digunakan :


**Tabel 5.1** data pengujian fitur pesan tukang

No	Atribut data	Tipe data	Karakter
1	Nama Tukang	String	Jamal
2	Tipe Pengerjaan	String	Borong
3	Jadwal	Date	1
4	Lokasi	Number	105.21837, -5.3438142

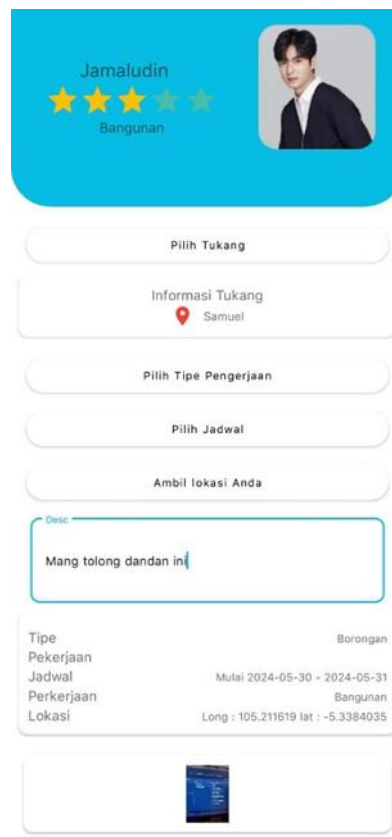
5	Deskripsi	String	Mang tolong dandan ini
---	-----------	--------	------------------------

Serta gambar yang digunakan sebagai pendukung keterangan permintaan pemesanan tukang yang di *input* akan masuk kedalam *storage firebase* lalu diambil *link* lokasi foto dan dimasukkan kedalam *database firebase*.

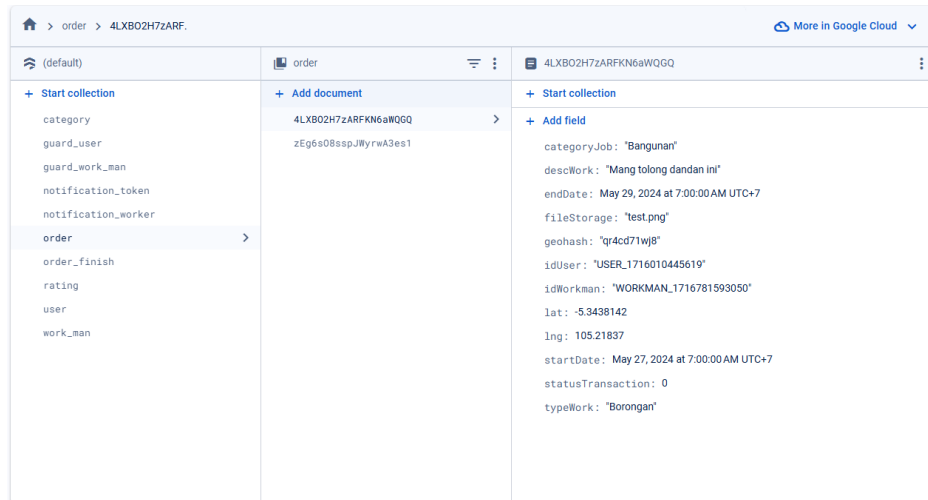
**Tabel 5.2** data pengujian gambar fitur pesan tukang

No	Atribut data	Tipe data	Data gambar
1	Gambar	Gambar (.png/jpg)	

Berikut adalah hasil pengujian dari data pada tabel-tabel diatas :



**Gambar 5.1** Pengujian pada aplikasi



**Gambar 5.2** Hasil Pengujian pada *database*

**Tabel 5.3** Hasil Pengujian fitur pesan tukang

Ketepatan Nama Tukang	Berhasil
Ketepatan Tipe Pengerjaan	Berhasil
Ketepatan Jadwal	Berhasil
Ketepatan Lokasi	Berhasil
Ketepatan Deskripsi	Berhasil
Ketepatan Gambar	Berhasil

### 5.1.1.2 Pengujian Ketepatan Data Pada Pesanan Jasa Tukang

Pengujian ini untuk menguji apakah data yang ditampilkan pada fitur ini sesuai dengan yang ada di *database* atau tidak tanpa ada kekurangan atau kesalahan. Berikut adalah data yang akan ditampilkan pada fitur pesanan jasa tukang pada menu pesanan saya :

**Tabel 5.4** Atribut data untuk di tampilkan di fitur cari latih tanding

No	Atribut data
1	Nama Tukang

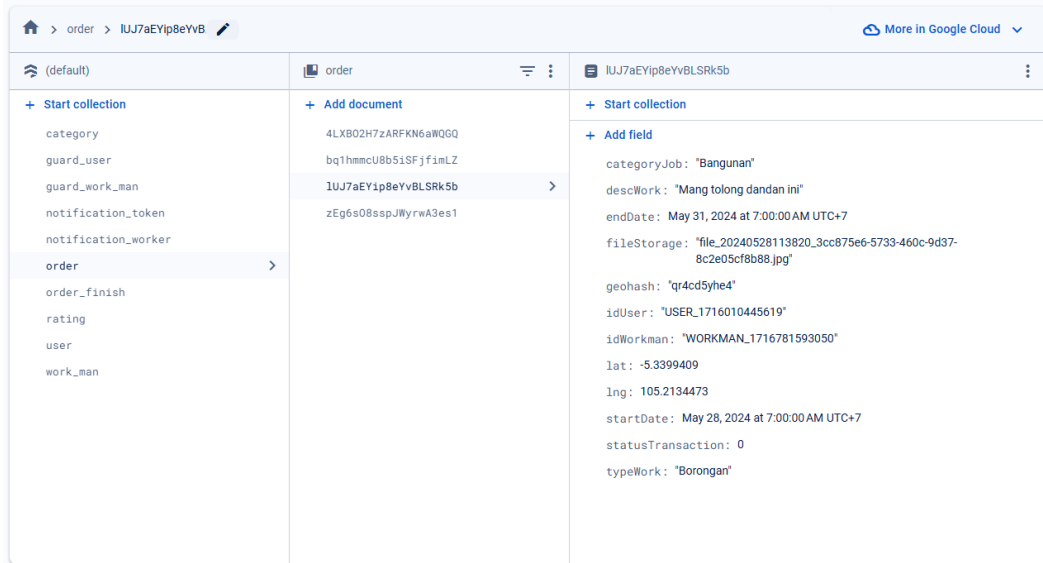
2	Tipe Pengerjaan
3	Jadwal
4	Lokasi
5	Deskripsi
6	Gambar

Berikut adalah hasil pengujian ketepatan data menampilkan data yang berada di *database firebase* kedalam aplikasi :



**Gambar 5.3** Pengujian pada Aplikasi





**Gambar 5.4** Hasil Pengujian *database*

**Tabel 5.5** Hasil Pengujian Fitur pesanan jasa tukang

Atribut Data	Hasil	
Ketepatan Nama Tukang	Sesuai	Tidak sesuai
Ketepatan Tipe Pengerjaan	Sesuai	Tidak sesuai
Ketepatan Jadwal	Sesuai	Tidak sesuai
Ketepatan Lokasi	Sesuai	Tidak sesuai
Ketepatan Deskripsi	Sesuai	Tidak sesuai
Ketepatan Gambar	Sesuai	Tidak sesuai

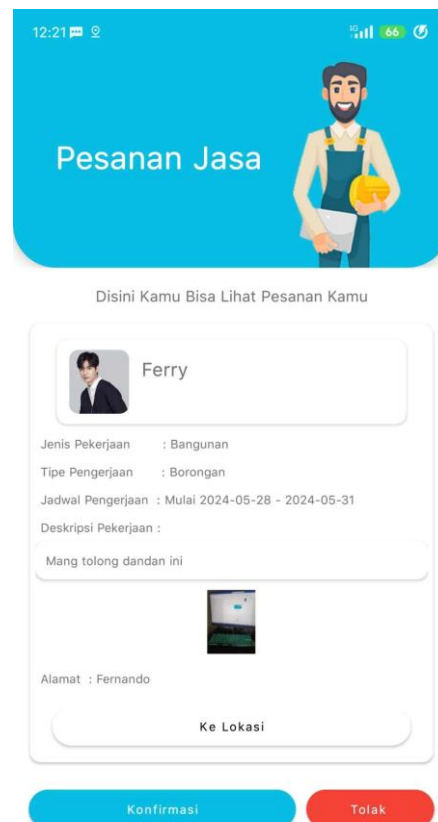
### 5.1.1.3 Pengujian Ketepatan Data Pada Pesanan Jasa Pada Tukang

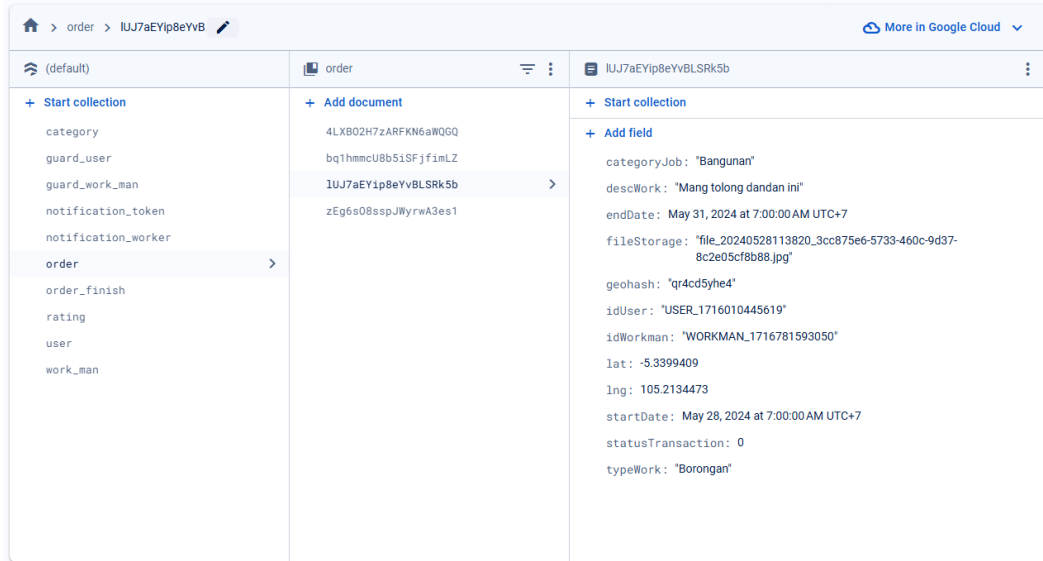
Pengujian ini untuk menguji apakah data yang ditampilkan pada fitur ini sesuai dengan yang ada di *database* atau tidak tanpa ada kekurangan atau kesalahan. Berikut adalah data yang akan ditampilkan pada fitur pesanan jasa tukang pada menu tukang :

**Tabel 5.6** Data pengujian pesanan jasa pada tukang

No	Atribut data
1	Nama Tukang
2	Tipe Pengerjaan
3	Jadwal
4	Lokasi
5	Deskripsi
6	Gambar

Berikut adalah hasil pengujian ketepatan data menampilkan data yang berada di *database firebase* kedalam aplikasi :

**Gambar 5.5** Hasil Tampilan pada aplikasi



**Gambar 5.6** hasil pengujian pada *database*

**Tabel 5.6** Hasil Pengujian Pesanan jasa pada Tukang

Atribut Data	Hasil	
Ketepatan Nama Tukang	Sesuai	Tidak sesuai
Ketepatan Tipe Pengerjaan	Sesuai	Tidak sesuai
Ketepatan Jadwal	Sesuai	Tidak sesuai
Ketepatan Lokasi	Sesuai	Tidak sesuai
Ketepatan Deskripsi	Sesuai	Tidak sesuai
Ketepatan Gambar	Sesuai	Tidak sesuai

### 5.1.2 Pengujian Algoritma

Pengujian ini dilakukan untuk mengetahui apakah algoritma dapat memberikan rekomendasi list tukang ke pengguna atau tidak untuk memberikan nilai prediksi rekomendasi menggunakan pendekatan *collaborative filtering* dan *Slope one*. Berikut adalah tabel hasil pengujian algoritma *Slope One* dengan mengambil sampel data 6 pengguna dan 6 tukang pada aplikasi Tulen :

**Tabel 5.7** Rating enam Tukang oleh enam orang Pengguna

Pengguna	Tukang					
	Dais (A)	Kaka(B)	Jamaludin(C)	Mang(D)	Wahyu(E)	Agung(F)
Ferry	0	5	4	3	0	0
Karim	0	0	3	2	4	1
Amir	0	3	0	0	0	3
Demas	4	0	0	1	0	0
Ilham	0	2	2	4	0	5
Deni	?	5	?	4	?	?

Tabel 5.7 adalah tabel yang berisikan rating dari enam pengguna untuk enam tukang. Nilai rating adalah 1 sampai 5. Berdasarkan tabel 5.7 akan dicari urutan rekomendasi untuk pengguna DENI dari empat tukang yang belum diratingnya.

Langkah pertama adalah dengan menghitung rata-rata selisih rating antar tukang berdasarkan data rating dari pengguna di tabel 5.7 menggunakan persamaan ke 1. Berikut adalah contoh penghitungan rata-rata selisih rating antar tukang Kaka(B) dan tukang Mang(D) dengan rumus :

$$\text{dev}_{j,i} = \sum_{u \in S_{j,i}(\chi)} \frac{u_j - u_i}{\text{card}(S_{j,i}(\chi))} \quad (1)$$

Dimana  $u$  adalah pengguna yang merating tukang B dan Tukang D

$u_j$  = rating untuk Kaka

$u_i$  = rating untuk Mang

$\text{card}(S_{j,i}(x))$  = jumlah pelanggan yang memberi rating pada tukang B dan D yaitu 3 orang (ferry,ilham,deni)

Nilai rata-rata selisih ranting antar tukang B dan tukang D adalah :

$$\text{dev}_{j,i} = \frac{5-3}{3} + \frac{2-4}{3} + \frac{5-4}{3} = 0,67 - 0,67 + 0,33 = 0,33$$

Dengan contoh perhitungan diatas, maka didapat lah tabel Nilai rata-rata rating antar tukang yang ditunjukkan pada tabel 5.8 di bawah ini :

**Tabel 5.8** Nilai rata-rata selisih rating antar tukang

Tukang	Dais(A)	Kaka(B)	Jamaludin(C)	Mang(D)	Wahyu(E)	Agung(F)
Dais (A)	0	0	0	-3	0	0
Kaka (B)	0	0	-0,5	-0,33	0	1,5
Jamaludin (C)	0	0,5	0	0	1	0,5
Mang (D)	3	0,33	0	0	2	0
Wahyu (E)	0	0	-1	-2	0	-3
Agung (F)	0	-1,5	-0,5	0	3	0

Tabel 5.8 memperlihatkan nilai rata-rata selisih rating antar tukang berdasarkan nilai rating yang diberikan oleh pengguna. Nilai pada tabel 5.8 menunjukkan semakin besar nilai secara absolut berarti perbedaan rating antar user semakin berbeda. Langkah selanjutnya adalah menghitung prediksi berdasarkan data rata-rata selisih rating pada tabel 5.8 dan tabel 5.7 menggunakan persamaan ke 2 dibawah ini :

$$P(u)_j = \frac{1}{card(R_j)} \sum_{i \in R_j} (dev_{j,i} + u_i) \quad (2)$$

Nilai prediksi yang akan dicari adalah nilai prediksi tukang-tukang yang akan direkomendasikan kepada pelanggan Deni, yaitu nilai prediksi tukang A, tukang C, tukang E dan tukang F. Sehingga dari nilai prediksi ini dapat dibuat rekomendasi untuk pelanggan Deni

- a. Nilai prediksi tukang A, dibantu oleh pengguna Dema yang telah memberi rating tukang D

$$P(A) = 1/1 * (3+4) = 7$$

- b. Nilai Prediksi tukang C, dibantu oleh pengguna Ferry dan Ilham yang telah memberi rating pada tukang B dan D. Pengguna Karim yang telah memberi rating tukang D.

$$P(C) = 1/5 * ((-0,5+5)+(0+4)+(-0,5+5)+(0+4)+(0+4)) = 1/5 * 22 = 4.4$$

- c. Nilai prediksi tukang E, dibantu oleh pengguna Karim yang telah memberi rating tukang D.

$$P(E) = 1/1 * (2+4) = 6$$

- d. Nilai prediksi tukang F, dibantu oleh pengguna Ilham yang telah memberi rating tukang B dan tukang D. Pengguna Amir yang telah memberi rating tukang B. Pengguna Karim yang telah memberi rating tukang D.

$$P(F) = 1/4 * ((1,5+5)+(0+4)+(1.5+5)+(0+4)) = 1/4 * 21 = 5.25$$

Sehingga mendapatkan Tabel nilai prediksi yang bisa dilihat pada Tabel 5.9 dibawah ini :

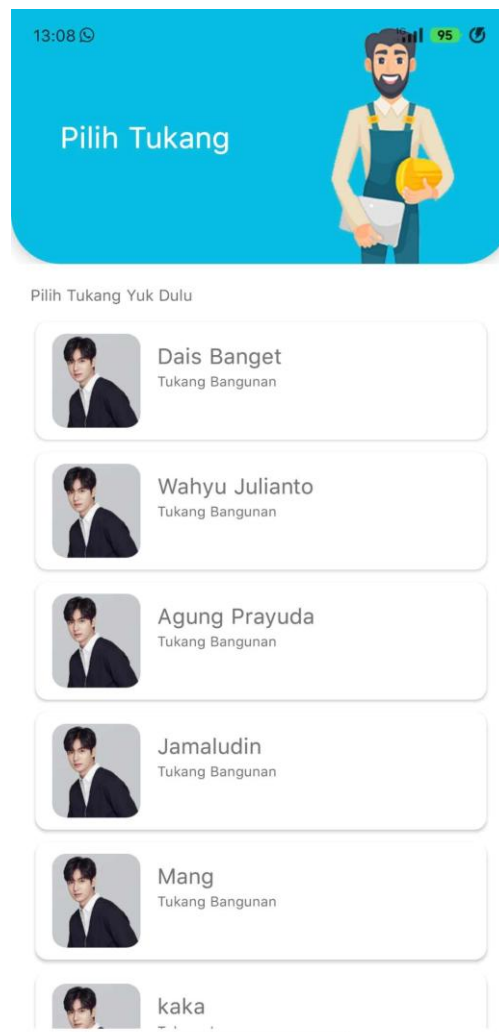
**Tabel 5.9** Nilai prediksi tukang A,C,E,F

	<b>Nilai Prediksi</b>
Tukang A (Dais)	7
Tukang C (Jamaludin)	4.4
Tukang E (Wahyu)	6
Tukang F (Agung)	5,25

Dari nilai prediksi pada tabel 5.9, kita dapat membuat urutan rekomendasi tukang untuk pengguna Deni, yaitu :

1. Tukang A (Dais)
2. Tukang E (Wahyu)
3. Tukang F (Agung) dan
4. Tukang C (Jamaludin)

Berikut adalah hasil dari nilai prediksi tersebut yang sudah di implementasikan pada aplikasi Tulen, sehingga mendapatkan Rekomendasi seperti pada gambar dibawah ini :



**Gambar 5.7** Hasil Implementasi algoritma pada aplikasi

### 5.1.2.1 Keefektifan Metode dan Algoritma pada penelitian ini

Pada penelitian ini menggunakan teknik metode *collaborative filtering* dan dipadukan dengan algoritma *slope one* untuk menghitung nilai prediksi rating yang kemudian akan dijadikan sebagai rekomendasi untuk pengguna. Namun rekomendasi yang diberikan kurang efektif mengingat cara kerja algoritma ini adalah :

1. Tujuan utama metode dan algoritma ini adalah memprediksi rating tukang yang belum dirating oleh pengguna, dan kemudian hasil dari perhitungan ini akan dijadikan sebagai rekomendasi untuk pengguna.
2. Mengambil hasil rating dari pengguna lain dan pengguna itu sendiri untuk digunakan sebagai acuan perhitungan prediksi.
3. Algoritma ini tidak berjalan jika pengguna tidak pernah melakukan rating ke tukang seperti poin (2).

Untuk penelitian pada “Jasa Tukang”, penulis menemukan ketidakcocokan algoritma ini sebagai sistem rekomendasi jasa tukang karena :

1. Algoritma tidak berjalan pada pengguna baru (belum pernah merating)
2. Algoritma ini tidak akan pernah memberikan rekomendasi pada tukang yang sudah pernah dirating oleh pengguna karena poin (1) pada cara kerja algoritma ini.
3. Jika seorang pengguna telah memberikan rating ke seluruh tukang yang ada pada sistem, maka algoritma tidak bisa berjalan dikarenakan sudah tidak ada tukang yang akan diprediksi ratingnya.

Maka dari itu, penulis berkesimpulan metode dan algoritma ini tidak efektif untuk jasa tukang. Terlepas dari itu, aplikasi ini sangat membantu masyarakat.



### 5.1.3 Pengujian Sistem Aplikasi

Pengujian dilakukan untuk mencari kesalahan atau *error* serta kekurangan pada aplikasi yang dibangun, sehingga dapat diketahui apakah aplikasi memenuhi kriteria sesuai dengan tujuan pembuatan aplikasi atau tujuan penelitian. Pengujian pada aplikasi ini menggunakan pengujian ISO 25010 dengan menggunakan dua (2) aspek pengujian, yaitu aspek *functional* dan aspek *Usability*. Pengujian ini dilakukan oleh tukang dan pengguna tukang dengan responden berjumlah 20 orang. Jumlah pertanyaan pada aspek *Functionlity* sebanyak 10 pertanyaan dan *usability* sebanyak 16 pertanyaan

#### 5.1.3.1 Pengujian *Functional Suitability*

Pengujian yang dilakukan pada aspek *functional suitability* ini memiliki tujuan untuk melakukan pengecekan terhadap fungsi-fungsi yang terdapat dalam sistem yang telah dibuat berdasarkan desain. Berikut pengujian pada aspek *functional suitability* :

**Tabel 5.10** Bobot jawaban *functional suitability*

Jawaban	Ya	Tidak
<i>Bobot</i>	<i>1</i>	<i>0</i>

Sumber : Sugiyono (2019)

Berikut ini adalah hasil pengujian *functionability* :

**Tabel 5.11** Hasil Pengujian

Pertanyaan	Ya	Tidak	Skor
<b><i>Functional Completness</i></b>			
Apakah aplikasi ini dapat terkoneksi ke database?	20		20
Apakah aplikasi dapat menampilkan form pesan tukang?	20		20

Apakah aplikasi dapat memesan tukang?	20		20
Apakah pesanan sampai ke tukang?	20		20
<b>Functional Correctness</b>			
Apakah aplikasi ini dapat diakses dengan mudah?	20		20
Apakah aplikasi ini dapat menampilkan list rekomendasi tukang?	20		20
Apakah aplikasi ini dapat memilih tukang?	20		20
<b>Functional Appropriateness</b>			
Apakah aplikasi ini dibuat sesuai dengan kebutuhan?	20		20
Apakah dengan dibuatkannya aplikasi ini dapat mempermudah tukang, pengguna tukang dalam mencari dan mendapatkan pesanan pekerjaan?	20		20
Apakah aplikasi ini menyimpan data sesuai dengan fungsinya?	20		20
<b>Total</b>			<b>200</b>

Berdasarkan dari total skor yang sudah dihitung pada tabel diatas, lalu seluruhnya dihitung menggunakan skala likert, sebagai berikut :

$$\text{Klasifikasi Persentase} = \frac{\text{Bobot Jawaban}}{\text{Bobot jawaban maksimal}} \times 100\%$$

$$\text{Persentase nilai, Ya} = \frac{1}{1} \times 100\% = 100\%$$

$$\text{Persentase nilai, Tidak} = \frac{0}{1} \times 100\% = 0\%$$

**Hasil Pengujian *Functional suitability* :**

$$\text{Klasifikasi Persentase} = \frac{200}{200} \times 100\% = 100\%$$

$$\text{Klasifikasi Persentase} = 100\%$$

Berdasarkan dengan kriteria persentase hasil uji, berikut Tabelnya :

**Tabel 5.12** Kriteria persentase hasil uji

Jumlah Skor (%)	Kriteria
0-49	Gagal
50-100	Sukses

Sumber : Sugiyono (2019)

**Tabel 5.13** Hasil Pengujian ISO 25010 pada kriteria *Functional Suitability*

Aspek	Skor Aktual	Skor Ideal	% Skor Aktual	Kriteria
<i>Functional Suitability</i>	200	200	100%	Sukses
Total perhitungan			100%	Sukses

Dari perhitungan hasil uji fungsional, menghasilkan persentase sebesar 100%, dapat disimpulkan bahwa aspek pada *Functional Suitability* yang dinilai oleh responden memiliki kriteria “Sukses”, karena memenuhi kriteria persentase hasil uji.

### 5.1.3.2 Pengujian Aspek Usability

Pengujian pada aspek ini bertujuan untuk menilai seberapa mudah sistem digunakan pengguna, dilakukan dengan menggunakan kuisioner, yang diberikan kepada tukang dan pengguna tukang dengan mencoba terlebih dahulu sistem yang telah dibuat sebelum mengisi kuisionernya. Pengujian ini terdapat 5 katagori jawaban yang berbeda dengan bobot yang berbeda juga, masing-masing memiliki jawaban sebagai berikut ini :

1. Sangat setuju = 5
2. Setuju = 4
3. Ragu-ragu = 3
4. Tidak setuju = 2
5. Sangat tidak setuju = 1

**Tabel 5.14** Hasil Pengujian *Usability*

No	Instrumen	SS (5)	S (4)	RG (3)	TS (2)	STS (1)	Skor
<b><i>Appropriateness Recognisability</i></b>							
1	Sistem ini membantu saya dalam mencari tukang / pekerjaan	20					100
2	Sistem ini sudah sesuai dengan kebutuhan saya	19	1				99
3	Sistem ini berjalan sesuai yang saya harapkan	18	2				98
<b><i>Operability</i></b>							
4	Sistem ini mudah dipakai	17	3				97
5	Tidak terdapat kesulitan dalam menggunakan aplikasi ini	18	2				98
6	Saya dapat melihat list pilihan rekomendasi tukang	20					100
7	Saya berhasil menggunakan aplikasi ini berjalan sesuai dengan kemampuan setiap kali digunakan	17	3				97
<b><i>Learnability</i></b>							
8	Saya belajar menggunakan aplikasi ini dengan cepat dan mudah	20					100
9	Saya mudah mengingat cara	16	4				96

	menggunakan sistem ini						
10	Sistem ini mudah untuk dipelajari	18	2				98
<b><i>User Interface Aesthetics</i></b>							
11	Saya puas dengan aplikasi ini	16	4				96
12	Saya sangat merekomendasikan aplikasi ini bagi tukang dan pengguna jasa tukang	20					100
<b><i>User error Protection</i></b>							
13	Jika frmulir pesanan ataupun inputan tidak lengkap, akan ada peringatan	20					100
<b><i>Accesibility</i></b>							
14	Sistem ini dapat digunakan oleh warga sindang sari	20					100
15	Sistem ini dapat digunakan untuk jangka waktu yang panjang	17	2	1			96
16	Kemudahan dalam aplikasi ini akan membuat kemudahan pada tukang dan pengguna tukang dalam mencari dan mendapatkan pekerjaan	20					100
<b>Total Skor</b>							<b>1575</b>

Berdasarkan dari hasil kuisisioner pada pengujian *usability* yang telah dilakukan, dapat dihitung dengan menggunakan skala *ikert* . menurut (Sugiyono, 2019) skor yang telah diperoleh akan dibagi dengan nilai tertinggi, jika kelima responden menjawab “sangat setuju” yang bernilai 5, maka hasilnya  $5 \times 5 = 25$ , lalu kalikan dengan jumlah pertanyaan sebanyak 450 dengan menghitung :

$$\text{Hasil} = \frac{\text{Skor Diperoleh}}{\text{Skormaksimal}} \times 100\%$$

$$\text{Hasil} = \frac{1575}{1600} \times 100\%$$

$$\text{Hasil} = 98.43\%$$

Dari hasil persentase diatas yang telah dihitung dapat dikatagorikan menggunakan tabel hasil uji sistem pada aspek *Usability* dalam pengujian ISO 25010, seperti dibawah ini :

**Tabel 5.15** Hasil Pengukuran Persentase

No	Nilai	Hasil
1	80%-100%	Sangat Layak
2	60%-79%	Layak
3	40%-59%	Netral
4	20%-39%	Tidak Layak
5	0%-19%	Sangat tidak Layak

Sumber : Sugiyono (2019)

**Tabel 5.16** Hasil Pengujian ISO 25010 pada kriteria *Usability*

Aspek	Skor Aktual	Skor Ideal	%Skor Aktual	Kriteria
<i>Usability</i>	1575	1600	98,43%	Sangat Layak

Dari hasil persentase skor diatas, diperoleh skor sebesar 98,43% yang dapat disimpulkan “Sangat Layak” bahwa sistem tersebut dibuat dengan sesuai.

## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Penelitian ini berhasil merancang dan membangun sistem rekomendasi jasa tukang menggunakan metode *Collaborative Filtering* dan algoritma *Slope One* berbasis Android. Studi kasus dilakukan di Desa Sindang Sari, Kecamatan Tanjung Bintang, Kabupaten Lampung Selatan, yang mengidentifikasi dua masalah utama: kesulitan penyedia jasa tukang dalam mendapatkan pesanan yang sesuai dengan keahlian mereka, dan kesulitan pengguna jasa dalam mencari tukang.

Sistem yang dikembangkan mampu memberikan rekomendasi berbasis rating dan preferensi pengguna sebelumnya. Namun rekomendasi yang diberikan tidak efektif pada “Jasa Tukang” dikarenakan cara kerja algoritma *slope one* yang mencari prediksi tukang yang belum dirating. Hal ini menyebabkan tukang yang telah dirating oleh user tidak akan menjadi rekomendasi lagi oleh user yang telah memberi rating kepada tukang tersebut.

Penggunaan teknologi seperti Android Studio, Firebase, dan metode pengembangan Extreme Programming memungkinkan aplikasi ini berjalan dengan baik pada platform Android.

Hasil pengujian usability menunjukkan bahwa aplikasi ini memiliki tingkat kegunaan yang baik dengan skor total 98,43%, yang dikategorikan sebagai "Sangat Layak". Ini menunjukkan bahwa aplikasi ini efektif dalam membantu pengguna mencari jasa tukang dan membantu penyedia jasa tukang mendapatkan pesanan yang sesuai.

## 6.2 Saran

Mengingat berbagai keterbatasan penulis, maka saran yang bisa diberikan untuk pengembangan atau penelitian selanjutnya adalah :

1. menambahkan katagori tukang
2. menambah foto profil setiap tukang
3. menambah *rate* harga tukang
4. memberikan *history* pesanan yang dikerjakan pada tukang
5. memberikan sistem rekomendasi yang cocok untuk jasa tukang
6. memperluas jangkauan studi kasus dan
7. memperbaiki tampilan *user interface* agar lebih menarik.



## Daftar Pustaka

- Arief, A. (2019). Rancang Bangun Sistem Rekomendasi Pariwisata Mobile dengan Menggunakan Metode Collaborative Filtering dan Location Based Filtering. *Jurnal Nasional Teknik Informatika*.
- Fadillah, A. (2019). Pemanfaatan Algoritma Slope One dengan Metode Collaborative Filtering Dalam Mengetahui Makanan Populer. 20-30.
- Janis, J. (2020). Rancang Bangun Aplikasi Online Sistem Pemesanan Jasa Tukang Bangunan Berbasis Lokasi. *Jurnal Teknik Informatika Vol 15*, 1-12.
- Kamilia, V. Z. (2019). Sistem Perekomendasi dengan Metode Item-Based Collaborative Filtering Pada Aplikasi Mobile Berbasis Android untuk Wisata Kuliner Kota Surakarta. 22-32.
- Kusnadi, A., & Daniel, D. (2021). Rancang Bangun Sistem Rekomendasi Resep Masakan Khas Indonesia Menggunakan Metode Collaboration Collective Intelligence dan Slope One. *Jurnal Teknik Informatika*.
- Nafis, R. M., & Setiawan, E. B. (2019). Aplikasi Pemesanan Jasa Tukang Menggunakan Teknologi Webhook dan Google Event Calendar. *Jurnal Ilmiah Komputer dan Informatika*.
- Pratama, D. (2022). Aplikasi Rekomendasi Tempat Makan Menggunakan Algoritma Slope One pada Platform Android. *Indonesian Journal of Computing and Cybernetics Systems (IJCCS)*, 2-6.
- Rojikin, A. (2019). Sistem Rekomendasi Pariwisata Lampung dengan Menggunakan Algoritma Slope One. *Jurnal Teknik Informatika*.
- Setiawan, Y., & Widodo, T. (2019). Aplikasi Pemilihan Jasa Tukang Menggunakan Metode Weight Product Berbasis Web. *Jurnal & Penelitian Teknik Informatika*, 3-6.
- Sitedu, S. Y. (2019). Sistem Rekomendasi Wisata Bernilai Sejarah di Jawa Barat Dengan Metode Collaborative Filtering Menggunakan Algoritma Slope One.
- Wijaksono, A. E. (2019). Aplikasi "Golek Tukang" untuk Pencarian Jasa Perbaikan Rumah di Daerah Surabaya Berbasis Android. *Jurnal Infra*, 2.
- Wijaya, A. E. (2020). Sistem Rekomendasi Laptop Menggunakan Collaborative Filtering dan Content-Based Filtering. *Jurnal Computech & Bisnis Vol 12 No 1*, 11-27.

## **LAMPIRAN**

Surat izin penelitian



**UNIVERSITAS TEKNOKRAT INDONESIA**  
**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

Nomor : 246/UTI/FTIK-S1.I/B.7.12/V/2024

16 Mei 2024

Perihal : Permohonan Izin Penelitian

Yth.

**Kepala Desa Sindang Sari**

Jl. Sindang Sari, Kec. Tanjung Bintang, Kab. Lampung Selatan.

Dengan hormat,

Sehubungan dengan penulisan skripsi mahasiswa Program Studi S1 Informatika Fakultas Teknik dan Ilmu Komputer Universitas Teknokrat Indonesia, dengan ini kami mohon kesediaan Bapak/Ibu untuk memperkenalkan mahasiswa/i kami melakukan penelitian pada desa yang Bapak/Ibu pimpin.

Adapun mahasiswa/i kami yang melakukan penelitian adalah:

nama : Ferry Fernando

NPM : 1731202

judul penelitian : Sistem Rekomendasi Jasa Tukang Dengan Metode *Collaborative Filtering* Dan Algoritma *Slope One* Berbasis Android (Studi Kasus Desa Sindang Sari)

Demikian permohonan ini, atas izin yang diberikan kami ucapkan terima kasih.

Hormat kami,

a.n. Dekan,

Wakil Dekan Bidang Akademik,



*(Signature)*  
Dr.Sc. Dedi Darwis, S.Kom., M.Kom., CDSP.



PEMERINTAH KABUPATEN LAMPUNG SELATAN  
KECAMATAN TANJUNG BINTANG  
**DESA SINDANGSARI**

Alamat: Jl. Ontowiryo Dusun 02 Desa Sindangsari Kec. Tanjungbintang 35361

Nomor : 118/243/VII.02/13/2024  
Lampiran :  
Perihal : Balasan Permohonan Izin Penelitian

Sindangsari, 22 Mei 2024  
Kepada Yth  
Dekan  
Universitas Teknokrat Indonesia  
di-  
Tempat

Dengan Hormat,

Berdasarkan surat Nomor : 246/UTI/FTIK-S1.I/B.7.12/V/2024, Tanggal 16 Mei 2024, Perihal Permohonan izin penelitian tugas akhir di Desa Sindangsari Kecamatan Tanjung Bintang Kabupaten Lampung Selatan, adapun mahasiswa tersebut adalah :

Nama : Ferry Fernando  
Nomor Mahasiswa : 17312021  
Alamat : Sindangsari dusun 4B  
Waktu Pelaksanaan : 10 Mei - sd selesai

Bersama ini kami sampaikan bahwa permohonan izin tersebut dapat di terima untuk melaksanakan Penelitian tersebut di Desa Sindangsari Kecamatan Tanjungbintang Kabupaten Lampung Selatan

Demikian Surat balasan ini kami terbitkan dan dapat dipergunakan sebagaimana mestinya.

  
Kepala Desa Sindangsari  
**M. RAPKAN**



PENYERAHAN SURAT IZIN KE BAPAK KEPALA DESA SINDANG SARI

# DOKUMENTASI WAWANCARA PENELITIAN



## Contoh lampiran rekaman wawancara

### Rekaman Wawancara

Nama Narasumber : Joko  
Profesi : WorkMan  
Tanggal Wawancara : 14-04-24  
Tema Wawancara : Sistem Rekomendasi Jasa Tukang dengan Metode Collaborative Filtering dan Algoritma SlopeOne berbasis Andorid (studi kasus : desa Sindang Sari)

Materi Wawancara		
1	Peneliti	Baagimana bapak mendapatkan pesanan pekerjaan?
	Narasumber	Dari bos atau tetangga
2	Peneliti	Apakah bapak kesulitan untuk mendapatkan pesanan pekerjaan?
	Narasumber	Sulit
3	Peneliti	Bagaimana sistem pembayaran pekerjaan bapak?
	Narasumber	Borongan
4	Peneliti	Lama pengerjaan pesanan biasanya berapa lama?
	Narasumber	1 bulan (26 hari)
5	Peneliti	Apakah bapak setuju jika ada aplikasi yang dapat mempermudah anda untuk mendapatkan pesanan pekerjaan?
	Narasumber	Setuju

Peneliti : Ferry Fernando

Narasumber :



# Rekaman wawancara pada pengguna tukang

## Rekaman Wawancara

Nama Narasumber : Pian  
Profesi : Pelan  
Tanggal Wawancara : 02-05-24  
Tema Wawancara : Sistem Rekomendasi Jasa Tukang dengan Metode Collaborative Filtering dan Algoritma SlopeOne berbasis Andorid (studi kasus : desa Sindang Sari)

Materi Wawancara		
1	Peneliti	Bagaimana cara anda mencari tukang untuk memperbaiki atau membuat sesuatu?
	Narasumber	Cari lewat kenalan
2	Peneliti	Apakah anda kesulitan untuk memilih tukang ?
	Narasumber	Sedikit
3	Peneliti	Bagaimana sistem pembayaran pada tukang?
	Narasumber	Saya lebih suka ke borongan
4	Peneliti	Apa pendapat anda untuk harga jasa tukang dengan harga yang cukup mahal, namun hasil pengerjaannya bagus?
	Narasumber	Saya suka karena ada harga ada kualitas
5	Peneliti	Apakah anda setuju jika ada aplikasi yang dapat memilih dan memesan tukang untuk keperluan anda?
	Narasumber	Sangat setuju
6	Peneliti	Untuk keperluan penelitian, dari nilai 1-5 berapakah yang anda berikan untuk tukang terakhir yang anda sewa?
	Narasumber	3

Peneliti : Ferry Fernando  
Narasumber :



## Contoh lampiran pengujian fungsional

### Pengujian Sistem Aspek Fungsional (*Functional Suitability*)

Nama : *Dais*  
Profesi : *Kuli*  
Tanggal : *30 - 4 - 2020*  
Tema : Sistem Rekomendasi Jasa Tukang dengan Metode Collaborative Filtering dan Algoritma Slope One berbasis Android (studi kasus : desa Sindang Sari)

Pertanyaan	Ya	Tidak	Skor
<b><i>Functional Completeness</i></b>			
Apakah aplikasi ini dapat terkoneksi ke database?	✓		
Apakah aplikasi dapat menampilkan form pesan tukang?	✓		
Apakah aplikasi dapat memesan tukang?	✓		
Apakah pesanan sampai ke tukang?	✓		
<b><i>Functional Correctness</i></b>			
Apakah aplikasi ini dapat diakses dengan mudah?	✓		
Apakah aplikasi ini dapat menampilkan list rekomendasi tukang?	✓		
Apakah aplikasi ini dapat memilih tukang?	✓		
<b><i>Functional Appropriateness</i></b>			
Apakah aplikasi ini dibuat sesuai dengan kebutuhan?	✓		
Apakah dengan dibuatkannya aplikasi ini dapat mempermudah tukang, pengguna tukang dalam mencari dan mendapatkan pesanan pekerjaan?	✓		
Apakah aplikasi ini menyimpan data sesuai dengan fungsinya?	✓		

Peneliti

  
Ferry Fernando

Narasumber

  
Dais



## Contoh lampiran pengujian usability

### Pengujian Sistem Aspek Kemudahan Pengguna (*Usability*)

Nama : Dars  
 Profesi : Kuli  
 Tanggal : 01-05-2024  
 Tema : Sistem Rekomendasi Jasa Tukang dengan Metode Collaborative Filtering dan Algoritma Slope One berbasis Android (Studi kasus : desa Sindang Sari)

No	Instrumen	SS (5)	S (4)	RG (3)	TS (2)	STS (1)	Skor
<b>Appropriateness Recognisability</b>							
1	Sistem ini membantu saya dalam mencaari tukang / pekerjaan	✓					
2	Sisstem ini sudah sesuai dengan kebutuhan saya	✓					
3	Sistem ini berjalan sesuai yang saya harapkan	✓					
<b>Operability</b>							
4	Sistem ini mudah dipakai	✓					
5	Tidak terdapat kesulitan dalam menggunakan aplikasi ini	✓					
6	Saya daapat melihat list pilihan rekomendasi tukang	✓					
7	Saya berhasil menggunakan aplikasi ini berjalan sesuai dengan kemampuan setiap kali digunakan		✓				
<b>Learnability</b>							
8	Saya belajar menggunakan aplikasi ini dengan cepat dan mudah	✓					
9	Saya mudah mengingat cara menggunakan sistem ini	✓					

10	Sistem ini mudah untuk dipelajari	✓					
<b>User Interface Aesthetics</b>							
11	Saya puas dengan aplikasi ini	✓					
12	Saya sangat merekomendasikan aplikasi ini bagi tukang dan pengguna jasa tukang	✓					
<b>User error Protection</b>							
13	Jika formulir pesanan ataupun inputan tidak lengkap, akan ada peringatan	✓					
<b>Accessability</b>							
14	Sistem ini dapat digunakan oleh warga sindang sari	✓					
15	Sistem ini dapat digunakan untuk jangka waktu yang panjang		✓				
16	Kemudahan dalam aplikasi ini akan membuat kemudahan pada tukang dan pengguna tukang dalam mencari dan mendapatkan pekerjaan	✓					
<b>Total Skor</b>							

Peneliti



Ferry Fernando

Narasumber



Dais