

LAPORAN KERJA PRAKTIK

**Pengembangan Aplikasi Mobile Pencatatan Inventory pada CV
Wijaya Lesmana Sejahtera**

**Diajukan sebagai syarat untuk menyelesaikan
Mata Kuliah Kerja Praktik**

Oleh:

**Fernando 2125250014
Dep' Niel Sinaga 2125250058**



Program Studi Informatika

**Fakultas Ilmu Komputer dan Rekayasa
Universitas Multi Data Palembang
2024**

LEMBAR PENGESAHAN

LEMBAR PENGESAHAN LAPORAN KERJA PRAKTIK

Pengembangan Aplikasi Mobile Pencatatan Inventory pada CV
Wijaya Lesmana Sejahtera

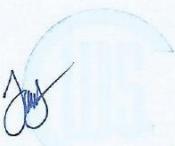
Oleh:

Fernando 2125250014
Dep' Niel Sinaga 2125250058

Palembang, 26 Juni 2024
Menyetujui:

Penyelia Perusahaan

Dosen Pembimbing


Indra Wijaya, S.T.
Direktur


M. Ezar Al Rivan, M.Kom.
NIK. 131092

Mengetahui,
Ketua Program Studi Informatika



Derry Alamsyah, S.Si., M.Kom., M.Pd
NIK. 111069

PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER DAN REKAYASA
UNIVERSITAS MULTI DATA PALEMBANG
PALEMBANG
2024

**Fakultas Ilmu Komputer dan Rekayasa
Universitas Multi Data Palembang**

Program Studi Informatika
Laporan Kerja Praktik
Semester Gasal/Genap* Tahun 20XX/20XX

**Pengembangan Aplikasi Mobile Pencatatan Inventory pada CV
Wijaya Lesmana Sejahtera**

Fernando 2125250014
Dep' Niel Sinaga 2125250058

Abstrak

CV Wijaya Lesmana Sejahtera adalah perusahaan yang memanfaatkan teknologi aplikasi untuk manajemen inventaris. Pencatatan data barang, barang masuk, dan barang keluar sangat penting, namun sering terjadi kesalahan manual. Untuk mengatasi ini, dikembangkan aplikasi InventoryKu yang meningkatkan efisiensi dan akurasi manajemen inventaris. Aplikasi InventoryKu mempermudah input data barang, barang masuk, dan barang keluar bagi admin dan karyawan. Pengajuan oleh karyawan menjadi lebih terstruktur dengan riwayat pengajuan, dan pembuatan laporan harian, mingguan, dan bulanan menjadi lebih cepat. Proses penambahan klien juga lebih mudah dengan menu khusus. Aplikasi ini dikembangkan menggunakan metodologi System Development Life Cycle (SDLC) model Waterfall. Dengan fitur-fitur yang disediakan, aplikasi ini mampu memastikan keakuratan dan efisiensi manajemen inventaris perusahaan.

Kata kunci: Aplikasi, Klien, Laporan, Manajemen, Pengajuan, Perusahaan, Stok Barang, Waterfall.

Faculty of Computer Science and Engineering
Universitas Multi Data Palembang

Undergraduate Program in Informatics
Internship Report
2nd Semester of Academic Year 2023/2024

**Development of a Mobile Application for Inventory Recording at CV Wijaya
Lesmana Sejahtera**

Fernando 2125250014
Dep' Niel Sinaga 2125250058

Abstract

CV Wijaya Lesmana Sejahtera is a company that leverages application technology for inventory management. Recording data on goods, incoming goods, and outgoing goods is crucial, but manual errors often occur. To address this, the InventoryKu application was developed to enhance the efficiency and accuracy of inventory management. The InventoryKu application facilitates the input of goods data, incoming goods, and outgoing goods for both administrators and employees. Employee submissions become more structured with a submission history, and the creation of daily, weekly, and monthly reports becomes faster. The client addition process is also easier with a dedicated menu. This application was developed using the System Development Life Cycle (SDLC) methodology with the Waterfall model. With its provided features, this application ensures the accuracy and efficiency of the company's inventory management.

Keywords: Application, Client, Report, Management, Submission, Company, Inventory, Waterfall.

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat, karunia, serta hidayah-Nya yang selalu memberikan kemudahan dan keberkahan dalam setiap langkah kami dalam menyelesaikan Laporan Kerja Praktik ini yang berjudul "Pengembangan Aplikasi Mobile Pencatatan Inventory pada CV Wijaya Lesmana Sejahtera".

Selama proses penulisan laporan ini, penulis mendapat berbagai ilmu, bantuan serta informasi dari berbagai pihak, baik secara langsung ataupun tidak langsung. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih kepada pihak yang telah membantu dalam pembuatan laporan ini, yaitu sebagai berikut:

1. Bapak Johannes Petrus, S.Kom., M.T.I., selaku Rektor Universitas Multi Data Palembang yang telah memberikan kesempatan untuk pelaksanaan kerja praktik ini.
2. Ibu Yulistia, S.Kom, M.TI, selaku Wakil Rektor I Universitas Multi Data Palembang.
3. Ibu Kathryn Sugara, S.E., M.Si., selaku Wakil Rektor II Universitas Multi Data Palembang.
4. Bapak Dedy Hermanto, S.Kom., M.T.I, selaku Wakil Rektor III Universitas Multi Data Palembang.
5. Charisma Ayu P, M.HRM, selaku Wakil Rektor IV Universitas Multi Data Palembang.

6. Bapak Dr. Wijang Widhiarso, M.Kom., selaku Dekan Fakultas Ilmu Komputer dan Rekayasa Universitas Multi Data Palembang.
7. Ibu Yoannita, M.Kom., selaku Wakil Dekan Fakultas Ilmu Komputer dan Rekayasa Universitas Multi Data Palembang.
8. Bapak Derry Alamsyah, S.Si, M.Kom., M.Pd., selaku Ketua Program Studi Informatika yang telah memberikan kesempatan dan persetujuan untuk pelaksanaan kerja praktik ini.
9. Bapak M. Ezar Al Rivan, M.Kom, sebagai pembimbing yang telah banyak membimbing penulis dalam menulis laporan kerja praktik ini.
10. Orang Tua dan keluarga penulis yang telah memberikan dukungan dan semangat kepada penulis.
11. Bapak Indra Wijaya, S.T., selaku Direktur dan Penyelia CV Wijaya Lesmana Sejahtera
12. Segenap Dosen Universitas Multi Data Palembang yang telah memberikan bimbingan akademik kepada penulis.
13. Staf Administrasi Universitas Multi Data Palembang yang telah membantu penulis dalam menyelesaikan urusan akademis.
14. Staf Perpustakaan Universitas Multi Data Palembang yang telah membantu penulis memeriksa kesalahan penulisan laporan kerja praktik ini.
15. Semua pihak yang telah terlibat dalam proses pembuatan laporan ini yang tidak bisa disebutkan satu-persatu.

Penulis menyadari bahwa dalam penyusunan Laporan Kerja Praktik ini, masih banyak kekurangan dan kelemahan yang perlu penulis perbaiki kedepannya. Oleh karena itu, kritik dan saran yang membangun dari pembaca sangat penulis harapkan guna meningkatkan kualitas Laporan Kerja Praktik ini.

Palembang, 28 Juni 2024

Dep` Niel & Fernando

DAFTAR ISI

LEMBAR PENGESAHAN	iv
Abstrak	v
Abstract	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	x
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Profil Lokasi Kerja Praktik	3
1.3.1 Denah Lokasi Kerja Praktik	3
1.3.2 Visi Misi	4
1.3.3 Struktur Organisasi	5
1.3.4 Tugas dan Wewenang	5
1.4 Tujuan dan Manfaat Kerja Praktik	7
1.5 Ruang Lingkup	7
1.6 Metodologi	8
1.7 Sistematika Penulisan.....	10
BAB 2 PELAKSANAAN KERJA PRAKTIK.....	11
2.1 Tinjauan Pustaka	11
2.1.1 Android.....	11
2.1.2 Java.....	12
2.1.3 Android Studio	13
2.1.4 Firebase	13
2.2 Teknologi yang Telah Dimanfaatkan.....	15
2.3 Analisis Permasalahan.....	15
2.4 Analisis Kebutuhan	16
2.5 Deskripsi Pekerjaan	17
2.6 Kesesuaian Ilmu Teoritis dan Praktis	21
BAB 3 HASIL DAN PEMBAHASAN.....	22
3.1 Perancangan Sistem.....	22
3.1.1 Use Case Diagram.....	22
3.1.2 Tampilan Aplikasi	24
3.2 Pengujian Program	40
3.3 Hasil Kuesioner	47
BAB 4 PENUTUP.....	50
4.1 Kesimpulan.....	50
4.2 Saran	50

DAFTAR PUSTAKA	51
LAMPIRAN.....	52
Surat Pengantar KP dari Kampus	52
Surat Balasan KP dari Perusahaan	53
Surat Selesai KP dari Perusahaan.....	54
Lampiran Foto	55
Kerangka Acuan Kerja Praktik.....	56
Log Harian Dep` Niel Sinaga.....	66
Log Harian Fernando	76
Kuesioner	86
Notulen Seminar Kerja Praktik	92
Kode Program	93

DAFTAR TABEL

Tabel 2. 1 Jadwal Kerja Praktik	17
Tabel 3. 1 Uji Coba <i>Form Login</i>	40
Tabel 3. 2 Uji Coba Fitur Lupa Password.....	41
Tabel 3. 3 Uji Coba Fitur Stock Barang.....	42
Tabel 3. 4 Uji Coba Fitur Client.....	43
Tabel 3. 5 Uji Coba Fitur Barang Masuk	43
Tabel 3. 6 Uji Coba Fitur Barang Keluar	44
Tabel 3. 7 Uji Coba Fitur Pengajuan.....	45
Tabel 3. 8 Uji Coba Fitur Riwayat Pengajuan	45
Tabel 3. 9 Uji Coba Fitur Laporan	46
Tabel 3. 10 Uji Coba Fitur admin	47
Tabel 3. 11 Hasil Kuesioner InventoryKu	47
Tabel 3. 12 Persentase Kriteria Skor Likert	48
Tabel 3. 13 Kriteria Skor Likert Kuesioner	49

DAFTAR GAMBAR

Gambar 1. 1 Denah Lokasi Kerja Praktik CV Wijaya Lesmana Sejahtera	4
Gambar 1. 2 Struktur Organisasi.....	5
Gambar 1. 3 Tahapan Metode <i>Waterfall</i>	8
Gambar 3. 1 Use Case Diagram Karyawan	22
Gambar 3. 2 Use Case Diagram Admin.....	23
Gambar 3. 3 Tampilan Splash Screen.....	24
Gambar 3. 4 Tampilan Halaman Login	25
Gambar 3. 5 Tampilan Lupa Password	26
Gambar 3. 6 Tampilan Email dari Lupa Password	26
Gambar 3. 7 Tampilan Main Activity Karyawan.....	27
Gambar 3. 8 Tampilan Main Activity Admin.....	28
Gambar 3. 9 Tampilan Barang Masuk	29
Gambar 3. 10 Tampilan Barang Keluar	30
Gambar 3. 11 Tampilan Pengajuan.....	31
Gambar 3. 12 Tampilan Riwayat Pengajuan.....	32
Gambar 3. 13 Tampilan Client.....	34
Gambar 3. 14 Tampilan Stock Barang	35
Gambar 3. 15 Tampilan Laporan	36
Gambar 3. 16 Tampilan Laporan Barang Masuk	37
Gambar 3. 17 Tampilan Laporan Barang Keluar.....	38
Gambar 3. 18 Tampilan Admin	39
Gambar 3. 19 Tampilan Tambah User	39
Gambar 3. 20 Tampilan Kelola User	40

BAB 1

PENDAHULUAN

Pada Bab 1 akan membahas Latar Belakang, Rumusan Masalah, Profil CV Wijaya Lesmana Sejahtera yang mencakup Denah Lokasi, Visi dan Misi, Struktur Organisasi, Tugas dan Wewenang dari CV Wijaya Lesmana Sejahtera, Tujuan dan Manfaat Kerja Praktik, Ruang Lingkup Kerja Praktik, dan Metodologi yang berisikan metode yang penulis gunakan dalam dalam membangun Aplikasi.

1.1 Latar Belakang

Aplikasi merupakan hal yang penting dalam suatu organisasi atau perusahaan (Alda, 2021). Dalam era digital yang semakin berkembang pesat, penggunaan teknologi aplikasi telah menjadi kunci dalam meningkatkan efisiensi, produktivitas, dan akurasi dalam berbagai aspek bisnis, termasuk manajemen inventaris. Dengan memanfaatkan aplikasi yang tepat, perusahaan dapat mengoptimalkan proses pencatatan, pemantauan, dan manajemen inventaris secara efisien, serta meminimalkan kesalahan manusia yang mungkin terjadi dalam proses manual. Oleh karena itu, pengembangan aplikasi mobile pencatatan inventaris menjadi sebuah langkah strategis bagi CV Wijaya Lesmana Sejahtera untuk meningkatkan daya saing dan efektivitas operasionalnya di tengah persaingan bisnis yang semakin ketat.

CV Wijaya Lesmana Sejahtera bergerak di bidang jasa sebagai Kontraktor Interior bagi orang-orang yang ingin mendesain interior di bangunannya. Kontraktor Interior bertanggung jawab dalam mendesain, merencanakan, mengatur, dan membuat berbagai jenis kebutuhan interior yang ada. Perusahaan akan bekerja sama dengan pihak yang terkait dalam perancangan desain interior yang nanti akan dibuat.

CV Wijaya Lesmana Sejahtera dihadapkan pada berbagai kendala dalam pengelolaan pergudangan. Salah satu permasalahan utama yang dihadapi adalah kecenderungan melakukan kesalahan dalam input atau pencatatan data yang akurat terkait dengan barang-barang yang masuk dan keluar dari pergudangan. Ketidakakuratan informasi terkait inventaris dapat mengganggu efisiensi operasional dan manajemen stok barang secara efektif di perusahaan.

Perusahaan ini kesulitan mengetahui secara pasti informasi tentang barang masuk dan keluar akibat dari kurangnya disiplin dalam memperbarui data yang ada menyebabkan informasi mengenai status barang seringkali tidak akurat atau tidak terkini (*real time*). Hal ini dapat mengganggu efisiensi operasional serta menghambat proses pengambilan keputusan yang berkaitan dengan manajemen pergudangan.

Selain itu, dengan adanya beberapa proyek interior yang berlangsung secara bersamaan, proses pencatatan barang di pergudangan CV Wijaya Lesmana Sejahtera menjadi semakin rumit dan kompleks. Proyek-proyek tersebut memerlukan manajemen stok yang cermat dan terkoordinasi, namun pengelolaan data yang tidak terstruktur menyulitkan upaya tersebut. Keterbatasan dalam proses

pencatatan barang juga menyebabkan pengambilan barang menjadi tidak dapat diawasi, sehingga menghambat perencanaan dan eksekusi proyek secara efisien di CV Wijaya Lesmana Sejahtera.

1.2 Rumusan Masalah

CV Wijaya Lesmana Sejahtera memiliki kendala dalam pengelolaan pergudangan. Salah satu permasalahan utama yang dihadapi adalah kecenderungan melakukan kesalahan dalam input atau pencatatan data yang akurat terkait dengan barang-barang yang masuk dan keluar dari pergudangan. Ketidakakuratan informasi terkait inventaris dapat mengganggu efisiensi operasional dan manajemen stok barang secara efektif di perusahaan.

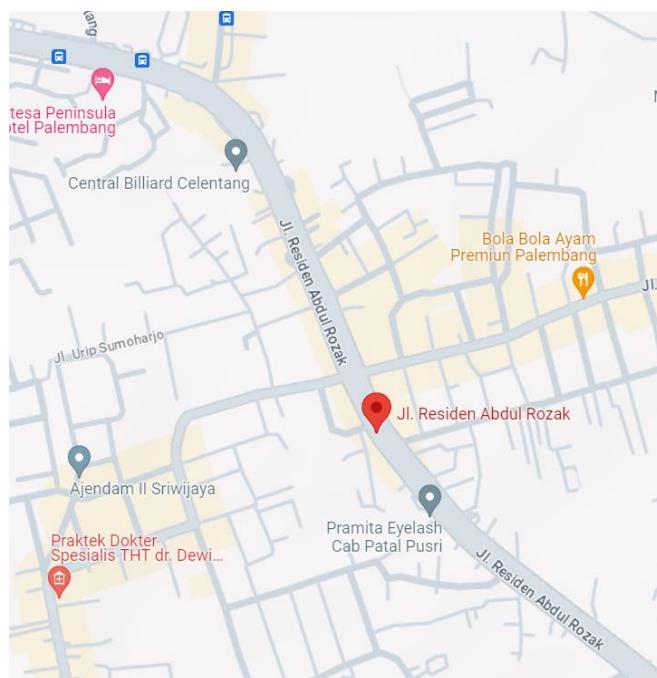
1.3 Profil Lokasi Kerja Praktik

CV Wijaya Lesmana Sejahtera bergerak di bidang jasa sebagai Kontraktor Interior bagi orang-orang yang ingin mendesain interior di bangunannya. Kontraktor Interior bertanggung jawab dalam mendesain, merencanakan, mengatur, dan membuat berbagai jenis kebutuhan interior yang ada. Perusahaan akan bekerja sama dengan pihak yang terkait dalam perancangan desain interior yang nanti akan dibuat.

1.3.1 Denah Lokasi Kerja Praktik

CV Wijaya Lesmana Sejahtera beralamat di Lr. Tembusan No.1828-A, Kel. Bukit Lama, Kec. Ilir Barat I, Kota Palembang, Provinsi Sumatera Selatan. CV Wijaya Lesmana Sejahtera berdiri pada tanggal 27 November 2015 dengan NIB Nomor 1208000220212. Perusahaan ini merupakan penyedia jasa layanan *interior*

contractor yang didirikan dan dikelola oleh Indra Wijaya sebagai pemilik sekaligus direktur dari CV Wijaya Lesmana Sejahtera. CV Wijaya Lesmana Sejahtera memiliki pusat penjualan dan perencanaan yang berada di Jalan Residen Haji Abdul Rozak, Desa/Kelurahan Bukit Sangkal, Kec. Kalidoni, Kota Palembang, Provinsi Sumatera Selatan. Denah lokasi pusat penjualan dan perencanaan diperlihatkan pada Gambar 1.1 yang dapat diakses melalui link <https://maps.app.goo.gl/jhm3kCU81nmH71Jf7>.



Gambar 1. 1 Denah Lokasi Kerja Praktik CV Wijaya Lesmana Sejahtera

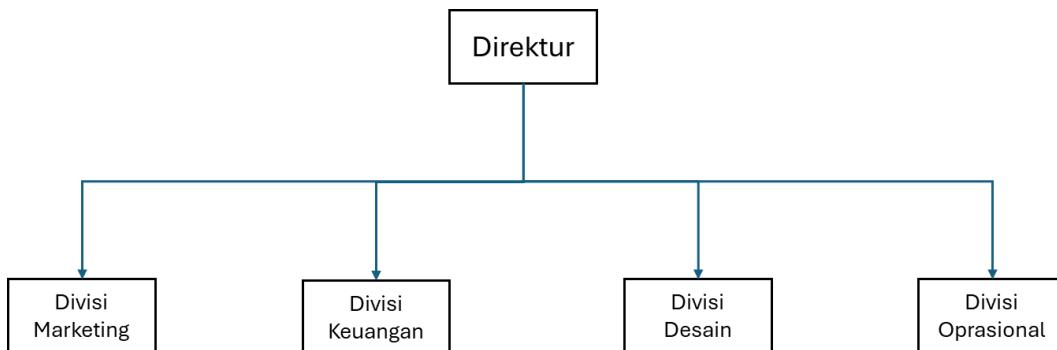
1.3.2 Visi Misi

CV Wijaya Lesmana Sejahtera mempunyai visi yakni menjadi sebuah perusahaan yang berkompeten di bidang Kontraktor Interior berbagai produk *Kitchen Set*, Plafon, Karaoke, Kamar Set, *Interior*, *Eksterior*, dll. Dengan meningkatkan kualitas dari produk yang ditawarkan serta Desain yang menarik. Sedangkan misi yang terus akan dikembangkan adalah menjalin hubungan yang



baik dengan pelanggan serta mengutamakan kepuasan pelanggan dan memberikan hasil yang memuaskan dan berkualitas pada konsumen.

1.3.3 Struktur Organisasi



Gambar 1. 2 Struktur Organisasi

CV Wijaya Lesmana Sejahtera ini juga memiliki struktur organisasi pegawai yang digunakan untuk mengetahui bagian-bagian apa saja yang merupakan tanggung jawab masing-masing pekerjaan. CV Wijaya Lesmana Sejahtera memiliki beberapa karyawan dengan 4 (empat) divisi, yaitu Divisi Keuangan, Divisi Desain, Divisi *Marketing*, dan Divisi Operasional. Dimana Direktur sebagai pemimpin dari keempat divisi.

1.3.4 Tugas dan Wewenang

Sebagai pemimpin perusahaan, Direktur bertanggung jawab atas pengambilan keputusan strategis dan pengawasan seluruh operasional perusahaan. Direktur memastikan setiap divisi bekerja sesuai dengan visi dan misi perusahaan, serta menjalin hubungan baik dengan klien dan mitra bisnis seperti bertemu dengan klien, membuat kontrak dengan klien untuk proyek yang akan dikerjakan. Selain itu, Direktur bertugas untuk memantau perkembangan proyek, dan mengarahkan kebijakan perusahaan untuk mencapai tujuan jangka panjang.

Divisi Keuangan bertanggung jawab untuk mengelola semua aspek keuangan perusahaan, termasuk akuntansi, pengelolaan anggaran, dan penyusunan laporan keuangan. Mereka memastikan semua transaksi keuangan dicatat dengan benar dan transparan. Selain itu, divisi ini bertugas untuk mengelola arus kas, memonitor pengeluaran dan pemasukan, serta memastikan ketersediaan dana yang cukup untuk operasional perusahaan. Divisi Keuangan juga bertanggung jawab untuk mengembangkan strategi keuangan yang mendukung pertumbuhan perusahaan.

Divisi Desain memiliki tugas utama dalam perancangan dan pengembangan konsep interior. Mereka bekerja mulai dari membuat desain awal berdasarkan kebutuhan dan preferensi klien, hingga mempresentasikan konsep kepada klien untuk mendapatkan persetujuan. Setelah desain disetujui, divisi ini mengembangkan desain detail yang siap untuk diimplementasikan. Mereka juga berkoordinasi dengan divisi operasional untuk memastikan bahwa desain yang telah dibuat dapat direalisasikan sesuai dengan rencana dan standar kualitas yang ditetapkan.

Divisi Marketing bertanggung jawab untuk mempromosikan layanan perusahaan dan mengembangkan strategi pemasaran yang efektif. Tugas mereka meliputi riset pasar, serta mengelola kampanye pemasaran dan promosi. Divisi ini juga mengurus branding perusahaan, menjalin dan memelihara hubungan dengan klien, serta mengelola komunikasi publik dan media sosial. Tujuannya adalah untuk meningkatkan visibilitas dan reputasi perusahaan di pasar, serta menarik lebih banyak klien.

Divisi Operasional mengatur dan mengelola pelaksanaan proyek desain interior di lapangan. Tugas mereka mencakup koordinasi dengan tim desain untuk memahami detail proyek, pengawasan tenaga kerja, dan pengelolaan logistik bahan dan peralatan yang dibutuhkan. Divisi ini memastikan bahwa setiap proyek berjalan sesuai dengan jadwal, dan standar kualitas yang ditetapkan. Mereka juga bertanggung jawab untuk menangani masalah yang muncul selama proses implementasi dan memastikan bahwa proyek selesai dengan memuaskan bagi klien.

1.4 Tujuan dan Manfaat Kerja Praktik

Tujuan Kerja Praktik ini adalah untuk mengembangkan aplikasi mobile demi meningkatkan efisiensi dan efektivitas pengelolaan pergudangan di CV Wijaya Lesmana Sejahtera. Manfaat dari Kerja Praktik ini adalah sebagai berikut:

1. Memberikan pengalaman berharga dalam mengembangkan aplikasi yang berkualitas dan berorientasi pada solusi bagi penyelia.
2. Memperluas dan memperdalam keterampilan dalam mengembangkan aplikasi mobile, manajemen proyek, dan komunikasi dengan penyelia.

1.5 Ruang Lingkup

Dalam Kerja Praktik ini, penulis memberikan beberapa batasan masalah agar pengembangan aplikasi sesuai tujuan, antara lain sebagai berikut:

1. Aplikasi dapat melakukan proses pengelolaan barang, seperti input data barang, hapus data barang, edit data barang, pencarian data barang, dan pembuatan laporan stok barang.

2. Aplikasi dapat melakukan proses pengelolaan data klien seperti input klien, edit klien, hapus klien dan pencarian data klien.
3. Aplikasi dapat mengontrol stok barang, dan melakukan pengecekan stok barang yang akan habis. Aplikasi dapat mengedit data pembelian barang, menghapus data pembelian barang, dan melihat laporan pemakaian barang berdasarkan tempat/klien, data barang masuk, dan data barang keluar berdasarkan periode tertentu, *supplier*, dan jenis barang.
4. Aplikasi dapat menyimpan data klien tetap, mengedit data klien, menghapus data klien, mencari data klien, dan membuat laporan daftar klien tetap, serta laporan pemakaian barang berdasarkan klien masing-masing.

1.6 Metodologi

Metodologi yang digunakan dalam Kerja Praktik ini adalah. *System Development Life Cycle* (SDLC), yaitu model *Waterfall*. Metode waterfall merupakan pengembangan perangkat lunak dengan penyelesaian satu tahap secara lengkap sebelum melangkah ke tahap berikutnya secara sistematis. Metode *waterfall* ini harus dilakukan secara berurutan sesuai dengan tahap yang ada pada Gambar 1.3.



Gambar 1. 3 Tahapan Metode Waterfall

Adapun beberapa tahapan di dalam *Waterfall* sebagai berikut :

1. Tahap *Analysis*

Pada tahap ini, wawancara dilakukan untuk mengumpulkan informasi kebutuhan apa saja yang diperlukan untuk pengembangan aplikasi dan permasalahan yang ada.

2. Tahap *Design*

Pada tahap ini dilakukan pembuatan *Use Case Diagram* sebagai desain perangkat lunak yang mencakup keseluruhan sistem, desain *database*, desain antarmuka pengguna, dan juga spesifikasi teknis.

3. Tahap *Development*

Pada tahap ini dilakukan implementasi perangkat lunak berdasarkan desain yang telah ditetapkan dengan mengubah konsep desain menjadi kode yang dapat dieksekusi oleh komputer.

4. Tahap *Testing*

Setelah tahap pengembangan selesai, perangkat lunak diuji untuk memastikan bahwa setiap komponen yang ada berfungsi sesuai yang diharapkan. Hal yang dapat diuji yakni pengujian fungsionalitas, integrasi, kinerja, dan keamanan.

5. Tahap *Deployment*

Setelah pengujian selesai dilakukan dan tidak ada kendala yang muncul maka perangkat lunak akan diterapkan ke Pengguna akhir. Dalam proses ini dapat dilakukan instalasi perangkat lunak, migrasi data, dan juga pelatihan bagi pengguna.

1.7 Sistematika Penulisan

Laporan kerja praktik ini disusun secara sistematik dari bab pendahuluan sampai dengan bab akhir. Adapun penjelasan terkait dari laporan kerja praktik ini adalah :

BAB 1 PENDAHULUAN

Pada bab ini dijelaskan mengenai latar belakang masalah dari pihak penyelia, profil tempat kerja praktik, tujuan dan manfaat kerja praktik, ruang lingkup, metodologi pengembangan sistem informasi yang digunakan, dan sistematika penulisan laporan kerja praktik.

BAB 2 PELAKSANAAN KERJA PRAKTIK

Pada bab ini menjelaskan tentang tinjauan pustaka mengenai elemen-elemen apa saja yang digunakan selama pelaksanaan kerja praktik. Selain itu, terdapat juga analisis kebutuhan, deskripsi pekerjaan, serta kesesuaian antara ilmu teoritis dan praktis.

BAB 3 PELAKSANAAN KERJA PRAKTIK

Pada bab ini membahas hasil dari pelaksanaan kerja praktik, yang terdiri dari perancangan sistem, pengujian program, dan hasil .

BAB 4 PENUTUP

Pada bab ini menjelaskan tentang kesimpulan yang dapat diambil dari hasil kerja praktik yang telah terlaksana dan saran.

BAB 2

PELAKSANAAN KERJA PRAKTIK

Kerja praktik adalah salah satu mata kuliah wajib bagi mahasiswa di Universitas MDP. Tujuan dari kerja praktik tersebut diantaranya adalah dapat mengembangkan ilmu yang telah didapatkan dalam proses perkuliahan, dan tentunya memberikan pengalaman kerja bagi mahasiswa agar tujuan dari pelaksanaan kerja praktik ini dapat memberikan wawasan serta kemampuan dalam dunia kerja bidang teknologi informasi. Pada bab ini akan dibahas mengenai tinjauan pustaka, analisis kebutuhan, dan deskripsi pekerjaan, serta kesesuaian ilmu teoritis dan praktis.

2.1 Tinjauan Pustaka

Pada bab ini akan dibahas berbagai literatur yang relevan dengan pembuatan aplikasi. Tinjauan pustaka ini bertujuan untuk memberikan dasar teoritis yang kuat mengenai elemen-elemen yang dipakai dalam pembuatan aplikasi kerja praktik ini. adapun elemen-elemen tersebut yaitu:

2.1.1 Android

Android (sistem operasi) –OS Android –Merupakan sebuah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet (Deybi et al., 2015). Keunggulan utama Android terletak pada sifatnya yang terbuka dan fleksibel. Sebagai sistem operasi open-source, Android memberikan kebebasan kepada pengembang untuk memodifikasi dan menyesuaikan sistem sesuai dengan kebutuhan mereka. Ini memungkinkan kompatibilitas yang luas dengan berbagai jenis perangkat keras

dari berbagai produsen. Android juga terintegrasi dengan berbagai layanan Google seperti Gmail, Google Maps, dan Google Drive, yang menambah nilai fungsionalitasnya. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak (Novita dan Zahri, 2018).

2.1.2 Java

Java adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990 (Budiyanto dan Sultonuddin, 2021). Java adalah selaku teknologi yang baik dibanding hanya sebuah bahasa pemrograman konvensional yang lain, karena java lebih lengkap. Java dirancang untuk memiliki portabilitas tinggi, memungkinkan kode yang ditulis di satu platform untuk dijalankan di berbagai platform lainnya tanpa perubahan berarti. Bahasa ini memiliki sintaks yang mirip dengan C++ namun lebih sederhana, dengan fokus pada menghilangkan beberapa fitur kompleks dan berpotensi membingungkan dari C++, seperti operator overloading dan pointer aritmetik. Java juga memiliki manajemen memori otomatis melalui Garbage Collection, yang membantu mengelola alokasi dan dealokasi memori secara otomatis, sehingga mengurangi risiko kebocoran memori. Dengan pustaka standar yang kaya dan ekosistem yang luas, Java banyak digunakan dalam berbagai aplikasi mulai dari pengembangan aplikasi Android, aplikasi web, sistem perusahaan, hingga aplikasi embedded. Kelebihan lainnya adalah keamanannya yang tinggi dan kemampuannya untuk diintegrasikan dengan teknologi lain, menjadikannya pilihan yang populer di kalangan pengembang dan perusahaan di seluruh dunia.

2.1.3 Android Studio

Android Studio merupakan lingkungan pengembangan perangkat lunak terpadu

– Integrated Development Environment (IDE) untuk pengembangan aplikasi android, berdasarkan IntelliJ IDEA (Budiyanto dan Sultonuddin, 2021). Fitur utamanya meliputi editor kode yang cerdas dengan kemampuan pelengkapan otomatis dan analisis kode, desain layout visual yang memungkinkan pengembang merancang antarmuka pengguna secara intuitif, serta sistem build berbasis Gradle yang kuat untuk mengelola dependensi dan otomatisasi build. Android Studio juga dilengkapi dengan emulator Android yang dapat digunakan untuk menguji aplikasi pada berbagai perangkat virtual dan konfigurasi. Selain itu, integrasi yang mendalam dengan berbagai layanan Google seperti Firebase memungkinkan pengembang menambahkan fitur backend dengan mudah, seperti autentikasi pengguna, penyimpanan data real-time, dan analitik.

2.1.4 Firebase

Firebase adalah platform pengembangan aplikasi yang menyediakan berbagai layanan untuk membantu pengembang membangun, meningkatkan, dan mengelola aplikasi dengan lebih mudah. Firebase dapat menyimpan data dalam format JavaScript Object Notation (JSON) yang tidak menggunakan query untuk memasukan, memperbarui, menghapus, atau menambah data ke dalamnya (Evandi et al., 2020).

2.1.4.1 Realtime Database

Realtime Database Adalah salah satu fasilitas yang menyimpan data ke database dan mengambil data darinya dengan sangat cepat tetapi firebase bukan hanya

realtime database, jauh lebih dari itu (Payara dan Tanone, 2018). Dirancang untuk menyimpan dan menyinkronkan data antara pengguna secara real-time, layanan ini memungkinkan setiap perubahan yang dilakukan pada data di salah satu klien langsung diperbarui di semua klien lain yang terhubung tanpa penundaan. Fitur ini sangat berguna untuk aplikasi yang memerlukan pembaruan data langsung, seperti aplikasi obrolan, kolaborasi dokumen, atau pelacakan lokasi secara real-time. Selain itu, Firebase Realtime Database memiliki kemampuan offline, di mana SDK-nya menyimpan data di perangkat pengguna sehingga tetap dapat diakses meskipun perangkat offline. Ketika perangkat kembali online, perubahan data yang dilakukan saat offline akan disinkronkan secara otomatis dengan basis data utama.

2.1.4.2 Firebase Authentication

Firebase Authentication, atau Firebase Auth, adalah layanan yang disediakan oleh platform Firebase untuk mengelola autentikasi pengguna dalam aplikasi. Layanan ini menyediakan berbagai metode autentikasi, termasuk email dan kata sandi, nomor telepon, serta autentikasi berbasis penyedia pihak ketiga seperti Google, Facebook, Twitter, dan GitHub. Ini juga mencakup sistem manajemen pengguna di mana pengembang dapat mengaktifkan otentikasi pengguna dengan email dan login kata sandi yang disimpan dengan Firebase (Khawas, 2018). Firebase Auth membantu pengembang mengelola autentikasi pengguna dengan aman dan efisien, memastikan aplikasi mereka aman dan memberikan pengalaman pengguna yang baik.

2.1.4.3 Firebase Storage

Firebase Storage adalah layanan penyimpanan file berbasis cloud yang disediakan oleh platform Firebase. Layanan ini dirancang untuk membantu pengembang

menyimpan dan mengelola konten buatan pengguna seperti foto, video, dan file lainnya dengan mudah dan aman. Ini memfasilitasi transfer file yang mudah dan aman terlepas dari kualitas jaringan untuk aplikasi Firebase (Khawas, 2018). Firebase Storage juga menyediakan SDK untuk berbagai platform termasuk Android, iOS, dan web, memudahkan pengembang untuk mengunggah dan mengunduh file langsung dari aplikasi mereka. Dengan kemampuan untuk menangani unggahan dan unduhan file besar secara efisien, Firebase Storage mendukung fitur seperti pengunggahan latar belakang, melanjutkan unggahan yang terhenti, dan monitoring progres unggahan.

2.2 Teknologi yang Telah Dimanfaatkan

CV Wijaya Lesmana Sejahtera menggunakan Microsoft Excel yang berada pada komputer kantor untuk menyimpan data barang masuk, barang keluar dan data klien. Dimana proses penginputan data-data tersebut dilakukan oleh admin saja. Microsoft Excel adalah sebuah aplikasi yang digunakan oleh bisnis dalam kebutuhan mereka akan grafik, grafik, perhitungan data statistik, dan pembuatan formula dan diluncurkan oleh Microsoft pada tahun 1985 (Hermawati dan Primasetya, 2021).

2.3 Analisis Permasalahan

CV Wijaya Lesmana Sejahtera memiliki kendala dalam pengelolaan pergudangan. Salah satu permasalahan utama yang dihadapi adalah kecenderungan lupa dalam melakukan input atau pencatatan data barang terkait dengan barang-barang yang masuk dan keluar dari pergudangan, para pegawai seringkali lupa melapor ke Admin bahwa mereka mengeluarkan barang. Ketidakakuratan informasi terkait inventaris

dapat mengganggu efisiensi operasional dan manajemen stok barang secara efektif di perusahaan. Sehingga dapat menghambat pekerjaan yang ditangani oleh CV Wijaya Lesmana.

2.4 Analisis Kebutuhan

CV Wijaya Lesmana Sejahtera menghadapi kendala dalam pengelolaan pergudangan, terutama dalam hal pencatatan dan input data barang yang masuk dan keluar karena Seringkali para pegawai lupa melapor ke Admin bahwa mereka memakai barang, sehingga menyebabkan Ketidakakuratan informasi inventaris mengganggu efisiensi operasional dan manajemen stok barang. Untuk mengatasi masalah ini, disarankan pengembangan aplikasi yang akan digunakan oleh pegawai dan admin sehingga ketika karyawan menggunakan atau mengambil barang untuk keperluan projek mereka bisa menginputnya sendiri barang apa saja yang mereka butuhkan. aplikasi ini memiliki dua role, yaitu Karyawan dan Admin.

Karyawan memiliki kemampuan terbatas karena untuk menjaga keamanan dan integritas data. pada role karyawan ini terdapat menu input barang masuk dimana barang tersebut telah terdaftar dimiliki pada stok barang, dan terdapat juga menu input barang keluar dimana menu ini berfungsi untuk menginput barang apa saja yang telah digunakan atau dikeluarkan dari gudang, ini butuh pengajuan barang keluar yang membutuhkan persetujuan dari Admin sebelum dikirim ke klien.

Pada role Admin, Admin memiliki kontrol penuh, mencakup manajemen pengguna, manajemen stok barang, persetujuan pengajuan barang keluar, riwayat persetujuan, manajemen klein, dan laporan barang keluar dari gudang dan diantar ke

klien. Fitur untuk Admin meliputi manajemen pengguna (tambah user, pilih role, aktif/nonaktifkan user), manajemen stok barang (tambah barang baru, nonaktifkan barang habis/tidak diproduksi lagi), pengajuan barang keluar (ACC/tolak pengajuan), riwayat persetujuan, manajemen klien (input/edit data klien), dan laporan (lihat riwayat barang keluar).

Alur kerja aplikasi dimulai dengan Karyawan yang menginput data barang masuk dan mengajukan permintaan barang keluar. Admin yang nantinya Admin akan meninjau, dan memutuskan untuk menyetujui atau menolak pengajuan tersebut. Jika disetujui, barang keluar dapat digunakan atau dikirim ke klien. Dengan aplikasi ini, CV Wijaya Lesmana Sejahtera dapat meningkatkan akurasi data inventaris, sehingga efisiensi operasional dan manajemen stok barang menjadi lebih efektif.

2.5 Deskripsi Pekerjaan

Pelaksanaan kerja praktik dilakukan selama 10 minggu, dimulai dari tanggal 8 Maret 2024 sampai 17 Mei 2024 dengan total durasi 280 jam. Jadwal pelaksanaan kegiatan kerja praktik adalah setiap hari Senin sampai Sabtu (Tabel 2.1).

Tabel 2. 1 Jadwal Kerja Praktik

Hari Pelaksanaan Kerja Praktik	Fernando (2125250014)	Dep'Niel Sinaga (2125250058)
Senin	13:00 – 15:00	13:00 – 15:00
Selasa	16:00 – 19:00	16:00 – 19:00

Rabu	09:00 – 12:00	09:00 – 12:00
Kamis	13:00 – 16:00	13:00 – 16:00
Jumat	16:00 – 19:00	16:00 – 19:00
Sabtu	13:00 – 15:00	13:00 – 15:00

Pada minggu pertama pelaksanaan kerja praktik di CV Wijaya Lesmana tanggal 26 Februari 2024 sampai 2 Maret 2024. Di hari pertama dilakukan wawancara mengenai permasalahan dari tempat kerja praktik, serta dilakukan pengenalan lingkungan kerja praktik oleh penyelia, Bapak Indra Wijaya selaku Direktur CV Wijaya Lesmana. Dilanjuti dengan penyusunan KAKP dan analisis kebutuhan aplikasi yang akan dibuat, dan membuat desain awal menggunakan figma. Pada minggu ini semua kegiatan berjalan sesuai dengan KAKP.

Minggu kedua kerja praktik dilaksanakan dari tanggal 4 Maret 2024 sampai 9 Maret 2024. Kegiatan yang dilakukan adalah melanjutkan pembuatan desain awal menggunakan figma serta melanjutkan penyusunan KAKP dan melakukan revisi KAKP pada bimbingan kedua serta menyiapkan Android Studio sebagai IDE untuk pembuatan aplikasi. Selain itu, dilakukan juga persiapan DBMS, yaitu Firebase dengan menyiapkan *database* dan integrasi dengan aplikasi. Pada minggu ini semua kegiatan berjalan sesuai dengan KAKP.

Minggu ketiga kerja praktik dilaksanakan dari tanggal 11 Maret 2024 sampai 16 Maret 2024. Kegiatan yang dilakukan adalah menyelesaikan KAKP dan

sistem *login*, *register*, dan lupa password. pada minggu ketiga ini juga dilakukan pembuatan sistem input klien dan serta melakukan desain pada tampilan aplikasi. Pembuatan sistem juga dihubungkan dengan *realtime database* dari Firebase. Pada minggu ini semua kegiatan berjalan lebih cepat dari KAKP.

Minggu keempat kerja praktik dilaksanakan dari tanggal 18 Maret 2024 sampai 23 Maret 2024. Kegiatan yang dilakukan adalah memperbaiki error code dan menyelesaikan input klien serta membuat fitur edit, hapus klien, membuat fitur input stock barang dan melakukan desain pada tampilan aplikasi. Pada minggu ini beberapa kegiatan berjalan lebih cepat dari KAKP.

Minggu kelima kerja praktik dilaksanakan dari tanggal 25 Maret 2024 sampai 30 Maret 2024. Kegiatan yang dilakukan pada minggu ini memperbaiki error. membuat dialog tampil stock, status barang berupa enable atau disable, menambahkan search bar pada stock barang dan client, membuat fitur input barang masuk, barang keluar dan serta mendesain tampilan aplikasi. Pada minggu ini beberapa kegiatan berjalan lebih cepat dari KAKP.

Minggu keenam kerja praktik dilaksanakan dari tanggal 1 April 2024 sampai 6 April 2024. Kegiatan yang dilakukan adalah memperbaiki error, membuat pengajuan barang masuk dan barang keluar untuk role karyawan, dan mendesain tampilan aplikasi. Pada minggu beberapa kegiatan tidak sesuai dengan KAKP dikarenakan penambahan fitur dan kebutuhan aplikasi.

Minggu ketujuh kerja praktik dilaksanakan dari tanggal 15 April 2024 sampai 20 April 2024. Kegiatan yang dilakukan adalah membuat laporan harian,

mingguan dan bulanan pada laporan barang masuk dan laporan barang keluar, dan memperbaiki error pada laporan barang masuk dan laporan barang keluar Pada minggu ini, beberapa kegiatan tidak sesuai KAKP dikarenakan terdapat banyaknya error dan penambahan fitur kebutuhan aplikasi.

Minggu kedelapan kerja praktik dilaksanakan dari tanggal 22 April 2024 sampai 27 April 2024. Kegiatan yang dilakukan adalah melanjutkan pembuatan laporan barang keluar, memperbaiki error, menambahkan role karyawan dan admin, memperbarui halaman register fitur kirim email untuk mengganti password ketika melakukan aktivitas lupa password. Pembuatan laporan akhir kerja praktik juga dilanjutkan. Pada minggu, kegiatan tidak sesuai dengan KAKP dikarenakan penambahan fitur dan kebutuhan aplikasi.

Minggu kesembilan kerja praktik dilaksanakan dari tanggal 29 April 2024 sampai 4 Mei 2024. Kegiatan yang dilakukan adalah bertemu dengan penyelia untuk menyampaikan progres, menambahkan fitur kelola user, mendesain tampilan Pada minggu, beberapa kegiatan tidak berjalan sesuai dengan KAKP dikarenakan penambahan fitur dan kebutuhan aplikasi.

Minggu kesepuluh kerja praktik dilaksanakan dari tanggal 6 Mei 2024 sampai 11 Mei 2024. Kegiatan yang dilakukan adalah penyerahan aplikasi ke pihak penyelia, melakukan survei aplikasi dan penyelesaian laporan akhir kerja praktik. Pada minggu ini semua kegiatan berjalan sesuai dengan KAKP.

2.6 Kesesuaian Ilmu Teoritis dan Praktis

Selama melaksanakan kerja praktik di CV Wijaya Lesmana Sejahtera, penulis mendapatkan kesesuaian ilmu teoritis dan praktis yaitu:

1. Telah diajarkan langkah – langkah pembuatan laporan pada mata kuliah Bahasa Indonesia.
2. Dasar – dasar dari bahasa pemrograman Java telah diajarkan di lingkungan akademik, sehingga mempermudah dalam pembuatan Aplikasi Presensi Karyawan pada CV Wijaya Lesmana Sejahtera.
3. Pada mata kuliah Pemrograman Aplikasi Bergerak 1 dan 2 yang diajarkan di lingkungan akademik, yaitu dengan menggunakan DBMS dan API dapat mempermudah penulis dalam pembuatan database aplikasi dari CV Wijaya Lesmana Sejahtera.
4. Dalam menghubungkan relasi antar tabel pada database aplikasi ini, diterapkan teori akademis dari mata kuliah Pemrograman Aplikasi Bergerak 1 dan 2 yang telah diajarkan.

BAB 3

HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai rancangan sistem berupa *use case diagram* dan tampilan program, serta hasil kuesioner dari karyawan terhadap aplikasi yang telah dikembangkan.

3.1 Perancangan Sistem

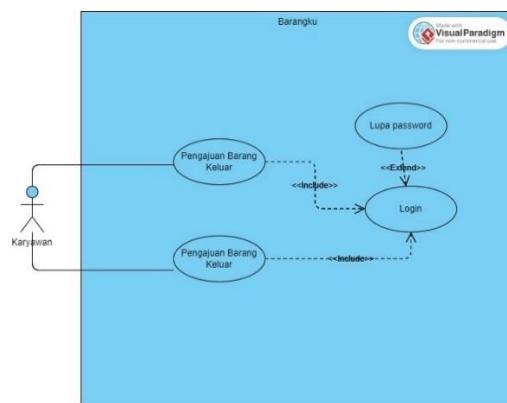
Perancangan sistem untuk CV Wijaya Lesmana Sejahtera akan dijelaskan mengenai sistem informasi yang dikembangkan dalam ruang lingkup CV Wijaya Lesmana Sejahtera.

3.1.1 Use Case Diagram

Interaksi antara pengguna dan sistem digambarkan dalam bentuk *use case diagram* dengan memiliki 2 sisi, yaitu admin dan karyawan. Interaksi dari Sisi Karyawan dengan sistem memiliki 2 menu utama, yaitu:

1. Barang Masuk yang memiliki tanggal dan data barang masuk.
2. Barang Keluar yang memiliki tanggal dan data barang keluar.

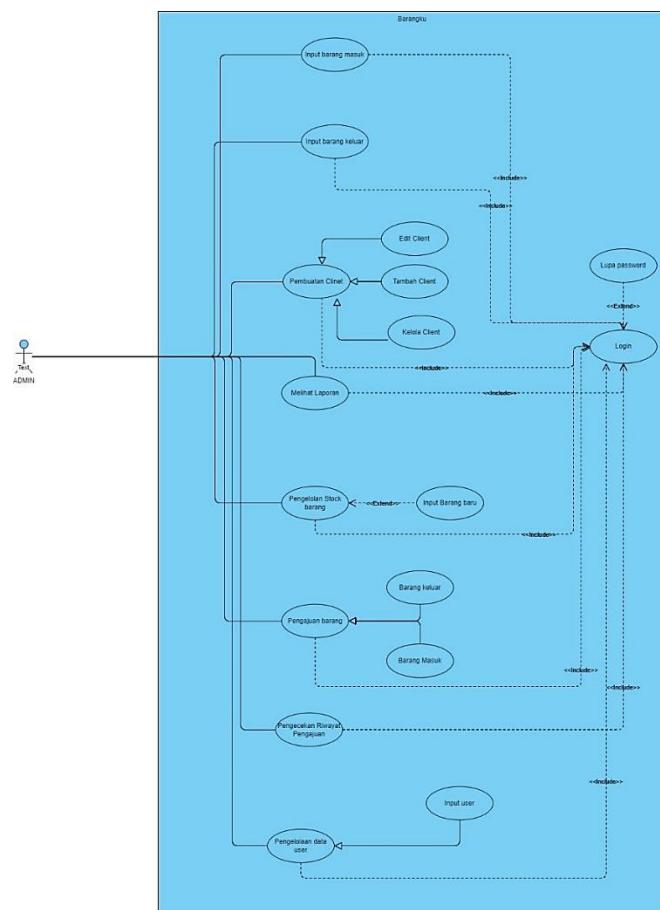
Interaksi karyawan dengan sistem yang dikembangkan dalam bentuk *use case diagram* dapat dilihat pada gambar 3.1.



Gambar 3. 1 Use Case Diagram Karyawan

Dan untuk interaksi dari sisi Admin dengan sistem memiliki 8 menu utama, yaitu:

1. Stock Barang, yang memiliki pencarian barang, list barang dan jumlah barang keseluruhan.
2. Admin, yang memiliki pendaftaran karyawan dan pengelolaan user.
3. Client, yang memiliki pencarian klien dan penambahan klien.
4. Barang Masuk yang memiliki tanggal dan data barang masuk.
5. Barang Keluar yang memiliki tanggal dan data barang keluar.



Gambar 3. 2 Use Case Diagram Admin

3.1.2 Tampilan Aplikasi

Aplikasi *Inventoryku* untuk CV Wijaya Lesmana memiliki dua tampilan utama, yaitu karyawan dan admin. Dari sisi karyawan, aplikasi menyediakan fitur untuk melakukan pengajuan barang masuk dan pengajuan barang keluar. Pada tampilan admin memiliki tampilan yang berbeda dengan karyawan dengan tambahan beberapa menu seperti pengajuan, riwayat pengajuan, klien, laporan, stock barang, dan admin.

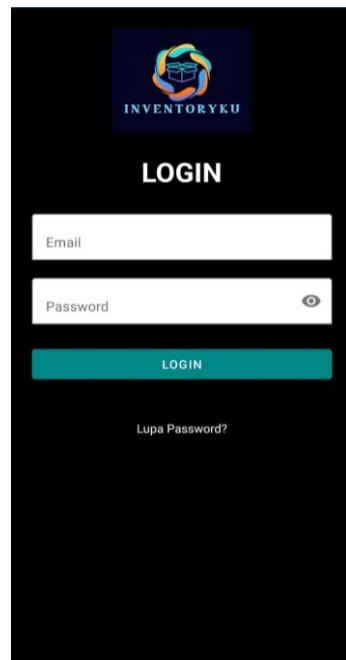
Pada bagian Splash Screen Activity muncul saat aplikasi dibuka. Splash Screen memiliki logo aplikasi yang dapat dilihat pada Gambar 3.3.



Gambar 3. 3 Tampilan Splash Screen

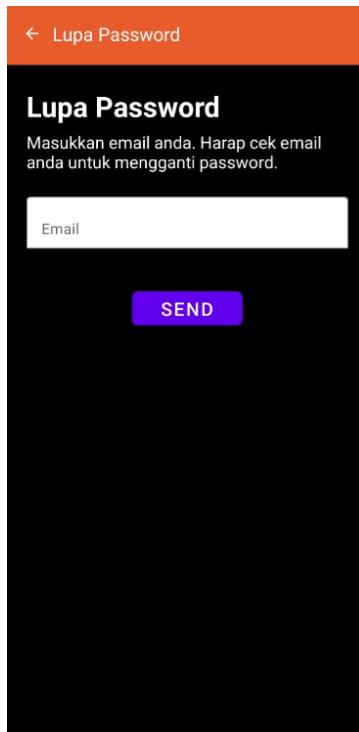
Kemudian pada tampilan Login memiliki dua input yaitu input email dan password serta tombol Login untuk mengakses aplikasi setelah mengisi email dan

password dengan benar maka akan mengakses menu yang ada di aplikasi sesuai dengan rolenya masing masing, serta terdapat Lupa password jika user lupa password untuk login aplikasi. Tampilan login dapat dilihat pada Gambar 3.4.

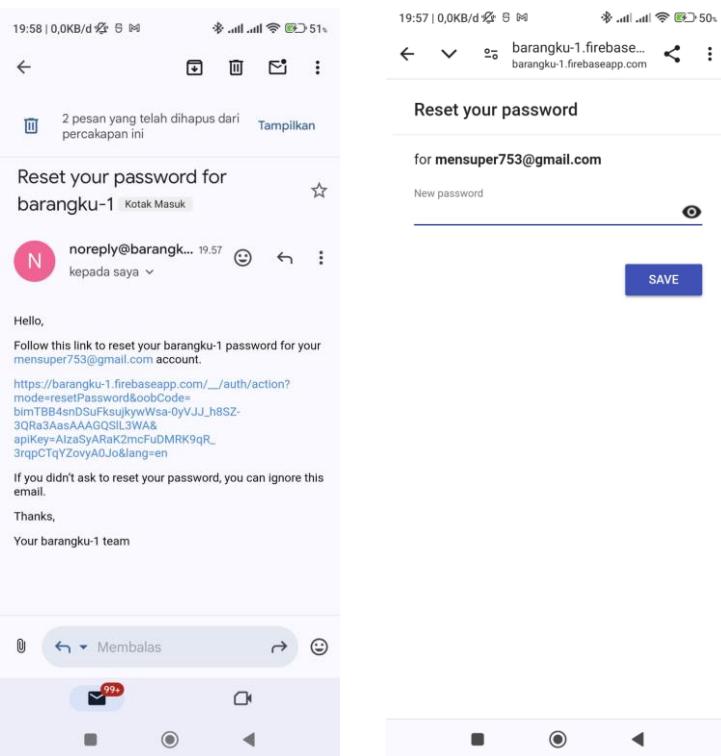


Gambar 3. 4 Tampilan Halaman Login

Kemudian pada bagian Lupa Password terdapat input email yang telah terdaftar ini berfungsi untuk mereset password, dan terdapat tombol Send yang berfungsi untuk mengirimkan email yang berisikan link yang dapat digunakan untuk mereset password dari email yang telah terdaftar pada aplikasi ini. Tampilan Lupa Password dapat dilihat seperti gambar 3.5.



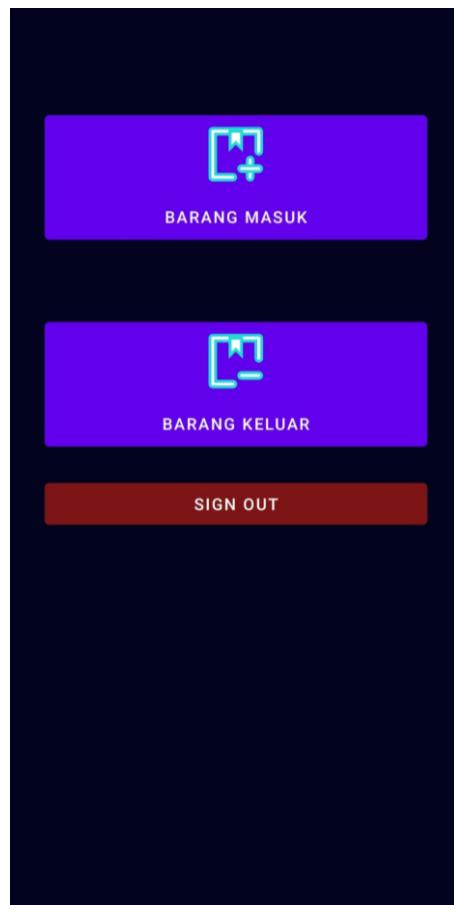
Gambar 3. 5 Tampilan Lupa Password



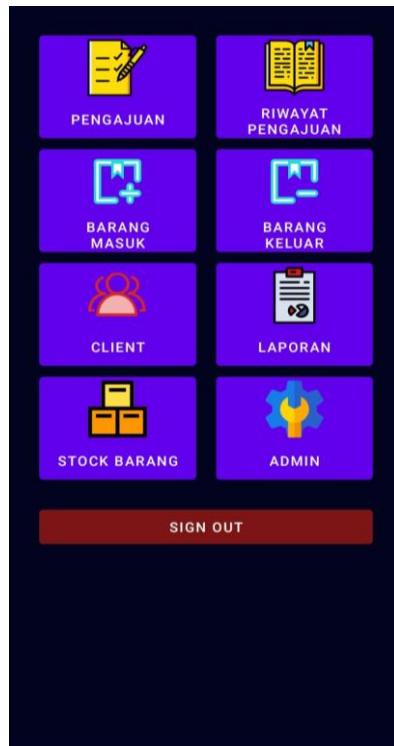
Gambar 3. 6 Tampilan Email dari Lupa Password

Tampilan Main Activity memiliki 2 tampilan yaitu tampilan dari sisi karyawan dan dari sisi admin dimana pada tampilan dari sisi karyawan hanya

terdapat 2 menu saja dan tombol sign out, dan pada tampilan admin memiliki 8 menu dan 1 tombol sign out. Tampilan untuk Main Activity karyawan dan admin dapat dilihat pada gambar 3.7 dan gambar 3.8.

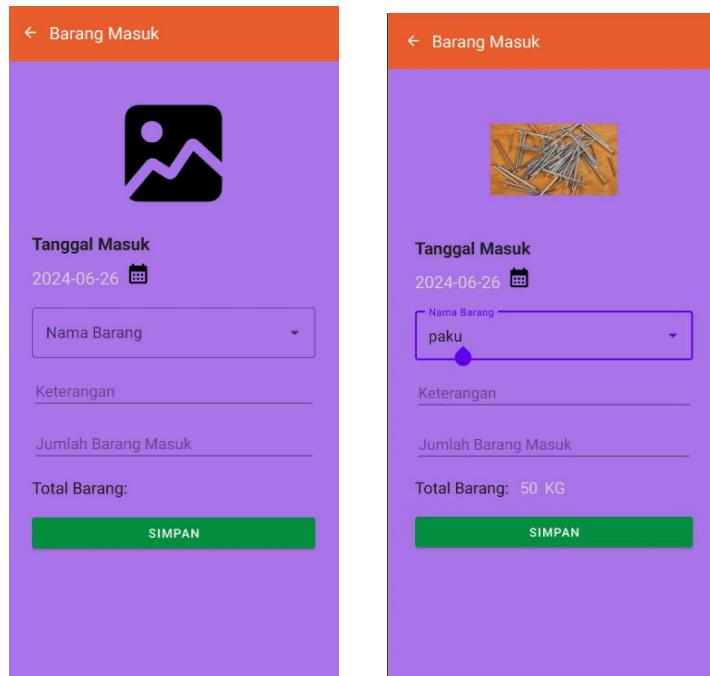


Gambar 3. 7 Tampilan Main Activity Karyawan



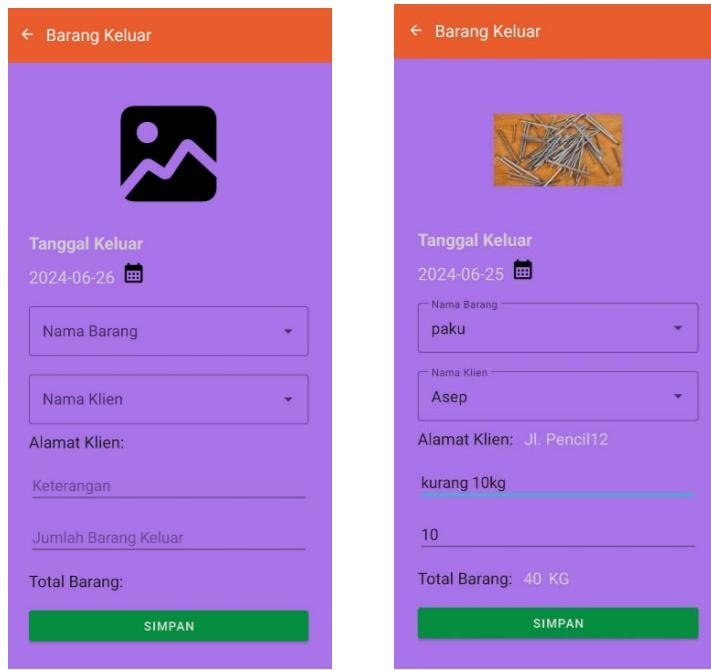
Gambar 3. 8 Tampilan Main Activity Admin

Pada tampilan Barang Masuk terdapat input nama barang, keterangan dan jumlah barang. pada tampilan ini dapat melakukan input nama barang dapat melakukan input barang yang masuk ke gudang dengan melakukan menginput nama barang yang masuk ke gudang, keterangan barang tersebut dan jumlah barang tersebut, dimana tanggal yang ada pada tampilan ini adalah tanggal melakukan input barang masuk mengikuti tanggal terkini dan tidak dapat diubah, ketika pemilihan barang dan pemilihan client maka pada saat memilih salah satu barang maupun klien maka akan menampilkan jumlah barang yang berada di stock barang dan klien akan menampilkan alamat yang sesuai dengan nama client tersebut. Dan terdapat tombol Simpan dimana ketika menekan tombol Simpan inputan barang masuk ini akan mengirimkan pengajuan barang tersebut ke admin dimana nantinya admin akan melakukan validasi untuk memastikan benar atau tidaknya barang tersebut dipesan. sedangkan bila di input langsung oleh admin maka tidak membutuhkan pengajuan seperti yang seperti karyawan ketika admin melakukan input barang masuk dan menekan tombol Simpan maka stok barang tersebut akan bertambah. Tampilan barang masuk dapat dilihat pada Gambar 3.9.



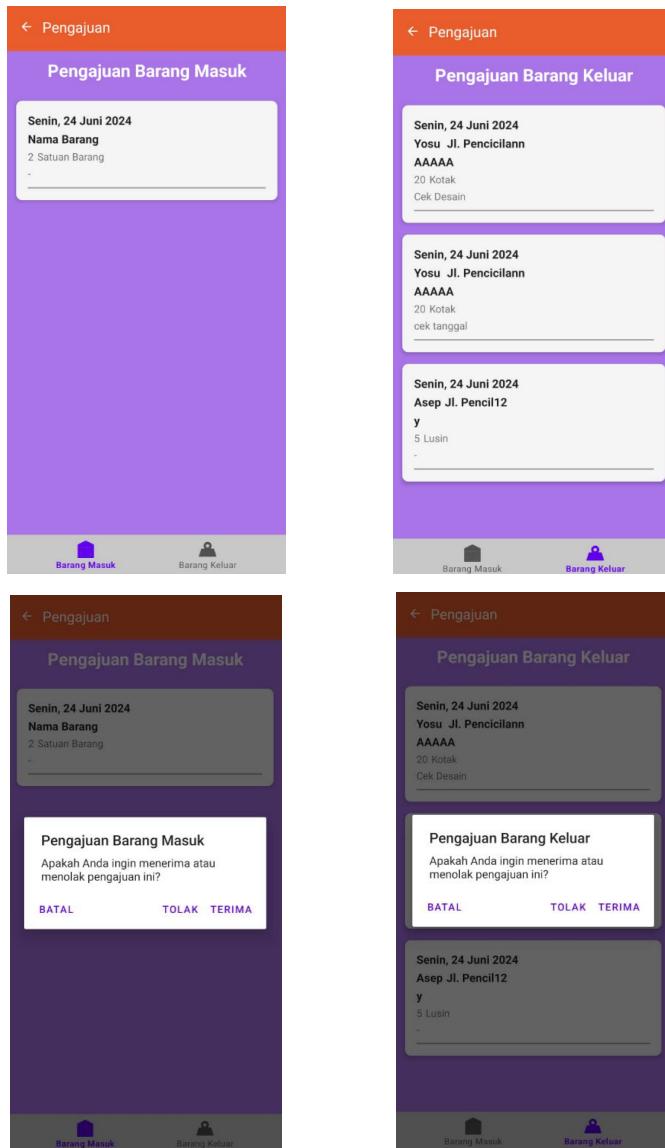
Gambar 3. 9 Tampilan Barang Masuk

Tampilan barang keluar terdapat input nama barang, nama klien, keterangan dan jumlah barang. pada tampilan ini dapat melakukan input barang yang akan keluar dari gudang, dengan menginput nama barang yang keluar dari gudang, nama klien sebagai tempat barang tersebut dikirim, alamat klien yang akan terisi otomatis sesuai dengan data klien, keterangan yang harus diisi guna untuk memberikan keterangan barang tersebut digunakan untuk apa, dan menampilkan jumlah barang yang tersedia dan satuan barang tersebut, dan terdapat tombol Simpan dimana ketika menekan tombol Simpan inputan barang keluar ini akan mengirimkan pengajuan barang tersebut ke admin dimana nantinya admin akan memberikan izin mengeluarkan barang tersebut. sedangkan bila di input langsung oleh admin maka tidak membutuhkan pengajuan seperti yang seperti karyawan ketika admin melakukan input barang masuk dan menekan tombol Simpan maka stok barang tersebut akan bertambah. Tampilan barang keluar dapat dilihat pada gambar 3.10.



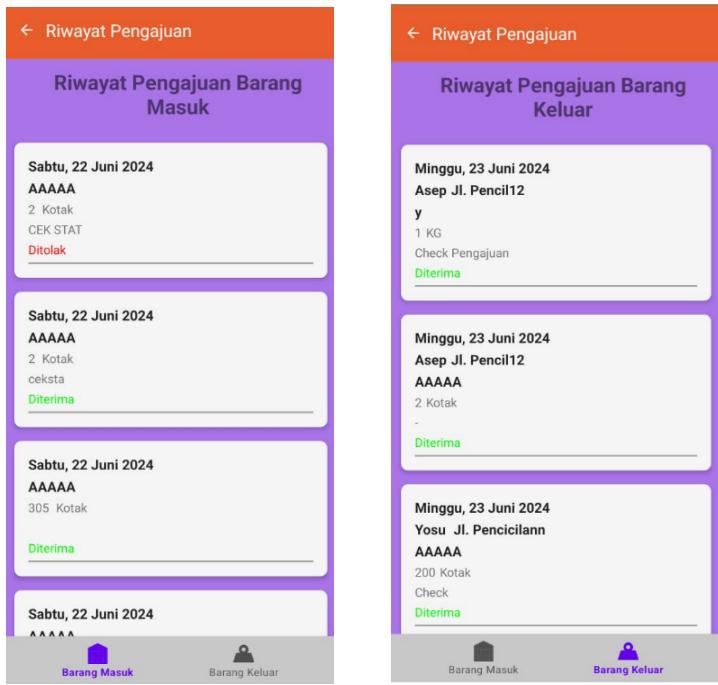
Gambar 3. 10 Tampilan Barang Keluar

Pada tampilan pengajuan terdapat 2 navbar di bawah untuk menampilkan pengajuan barang masuk dan barang keluar. Recyclerview yang menampilkan tanggal pengajuan, nama barang yang diajukan , jumlah barang yang diajukan, satuan barang, dan keterangan. saat recyclerview tersebut ditekan maka akan keluar toast untuk menerima atau menolak pengajuan barang tersebut saat pengajuan barang tersebut diterima maka stok barang yang ada akan berubah mengikuti transaksi yang dilakukan sedangkan bila ditolak maka stok barang tersebut tidak akan berubah. Tampilan Pengajuan dapat dilihat pada gambar 3.11.



Gambar 3. 11 Tampilan Pengajuan

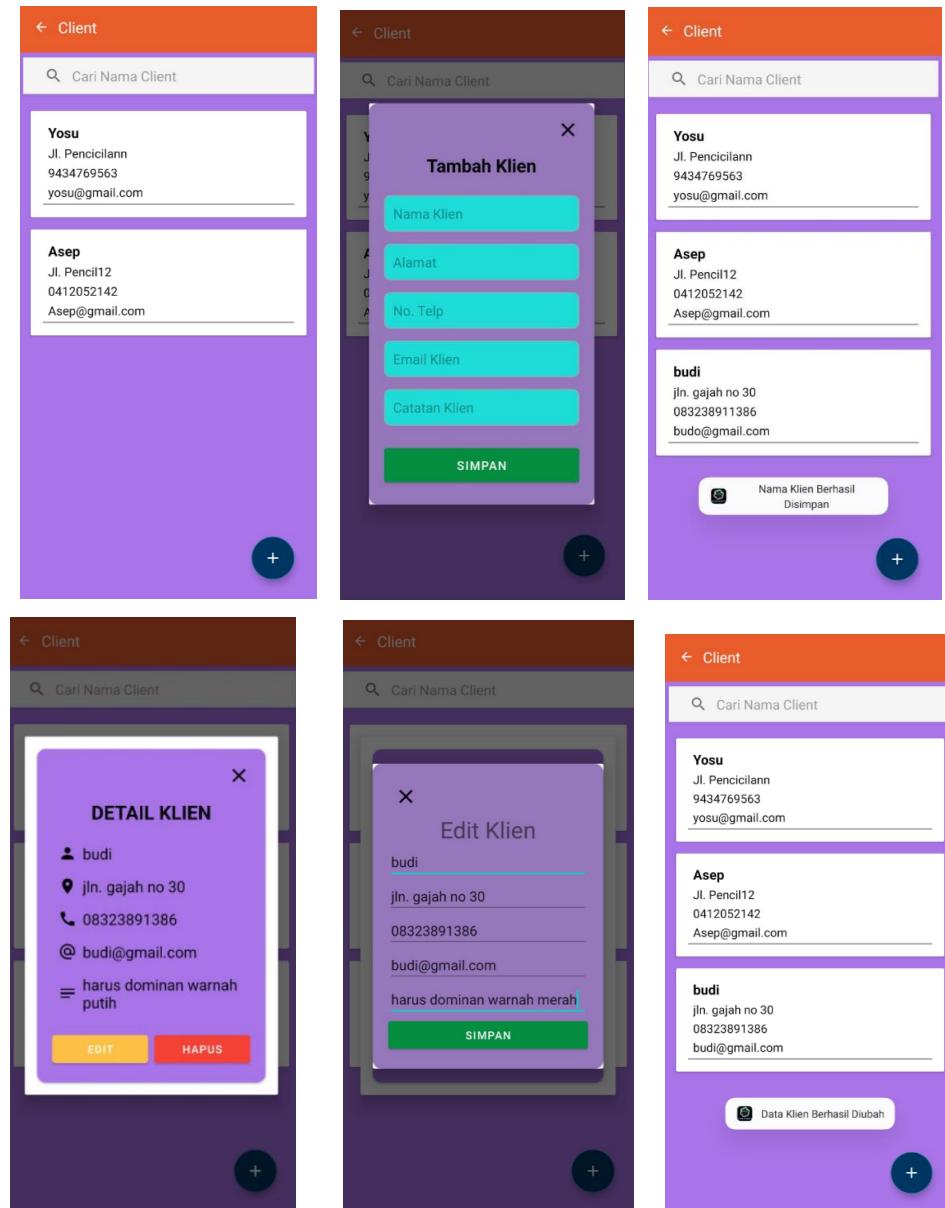
Pada tampilan Riwayat pengajuan terdapat.recyclerview yang menampilkan tanggal pengajuan barang masuk maupun barang keluar nama barang, jumlah barang, satuan barang, keterangan dan, status pengajuan barang diterima atau ditolak pada bagian riwayat pengajuan terdapat 2 navbar di bagian bawah yang berfungsi untuk menunjukan riwayat pengajuan barang masuk dan riwayat pengajuan barang keluar

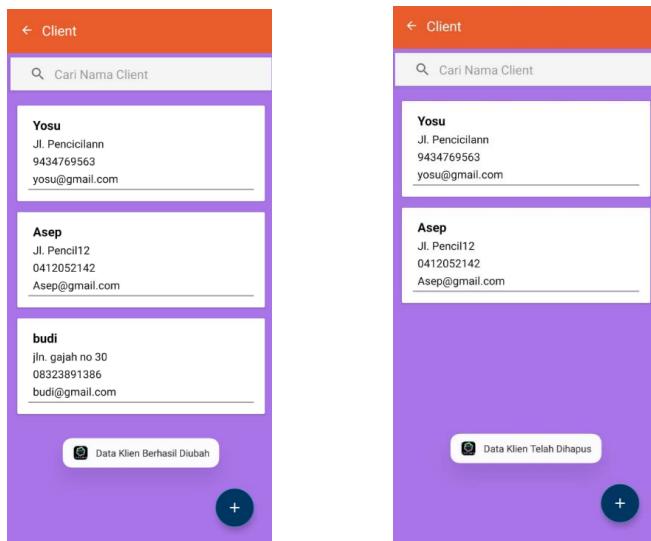


Gambar 3. 12 Tampilan Riwayat Pengajuan

Pada tampilan Client terdapat recyclerview, search bar dan floating action button. pada recyclerview menampilkan nama klien, alamat klien, nomor telepon klien. Ketika recyclerview ditekan akan menampilkan dialog yang menampilkan nama klien, alamat klien, nomor telepon klien, email klien dan catatan khusus dari klien dan terdapat 2 tombol yaitu tombol edit dan hapus. ketika tombol edit di tekan maka akan keluar dialog lain yang berisikan input nama klien, alamat klien, nomor telepon klien, email klien dan catatan khusus dan terdapat tombol simpan pada dialog tersebut yang berfungsi untuk menyimpan perubahan data sehingga menungkinkan untuk dapat melakukan perubahan data sewaktu-waktu bila terjadi kesalahan informasi pada data klien. dan fungsi tombol hapus adalah untuk menghapus data klien jika dirasa data tersebut tidak perlu digunakan lagi atau ketika kerjasama antar klien tersebut telah selesai. pada tampilan awal menu terdapat search bar yang berfungsi untuk mencari klien berdasarkan nama klien tersebut. Floating action button pada tampilan awal berfungsi untuk menampilkan dialog yang berguna untuk menambahkan data klien baru dengan memasukan beberapa input seperti nama klien, alamat klien, nomor telepon klien, email klien dan catatan khusus, dan terdapat juga tombol Simpan pada dialog ini berguna untuk menyimpan data klien baru. pada tampilan ini akan memberikan toast untuk kegiatan tambah

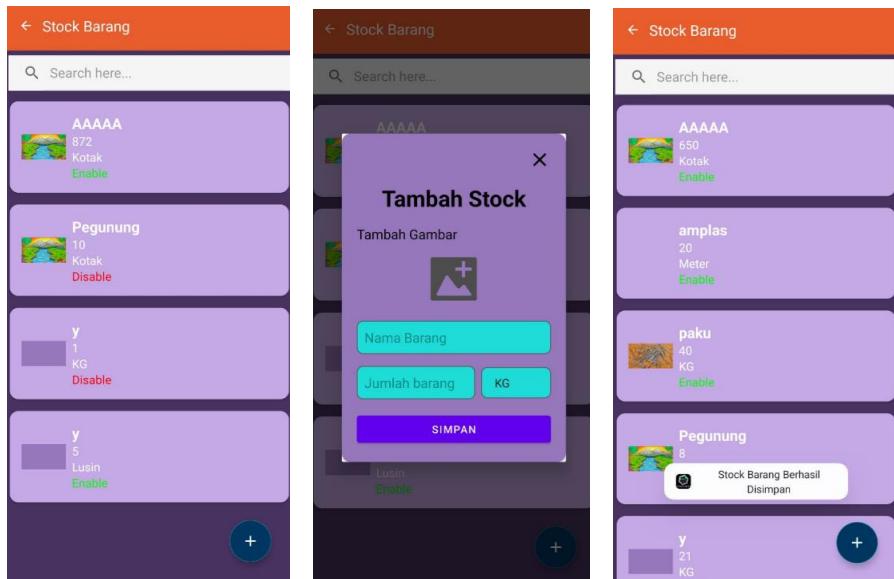
klien, edit klien dan hapus klien bila berhasil. Tampilan Client dapat dilihat pada gambar 3.11.





Gambar 3. 13 Tampilan Client

Padat Tampilan Stock Barang terdapat Search bar yang berfungsi untuk mencari barang dengan menggunakan nama barang. Floating action button yang berfungsi untuk menambahkan barang baru. Recyclerview yang berfungsi menampilkan barang-barang yang ada. ketika.recyclerview ditekan maka akan akan keluar dialog yang menampilkan foto barang, nama barang, jumlah barang dan satuan barang serta terdapat tombol enable dan disable yang digunakan sebagai status dari barang tersebut. Tombol Enable berfungsi untuk memberitahu bahwa barang tersebut masih tersedia atau bisa dipakai sedangkan Tombol Disable berfungsi untuk memberi tahu bahwa barang tersebut tidak bisa digunakan keduanya ini berpengaruh pada proses barang masuk dan keluar. barang yang bisa gunakan hanya barang yang memiliki status enable dan pada yang memiliki status disable barang tersebut tidak bisa dilakukan transaksi pada barang masuk dan barang keluar. Tampilan Stock Barang dapat dilihat pada gambar 3.14.



Gambar 3. 14 Tampilan Stock Barang

Pada Tampilan Laporan terdapat 2 tombol yaitu tombol Laporan Barang Masuk dan Laporan barang keluar. Tombol Laporan Barang Masuk berfungsi untuk menampilkan laporan barang masuk yang terdapat laporan barang masuk harian, laporan barang masuk mingguan dan laporan barang masuk bulanan. laporan barang masuk harian akan menampilkan laporan barang-barang yang masuk pada hari itu, pada laporan barang masuk mingguan akan menampilkan barang-barang yang masuk pada minggu tersebut dan pada laporan bulanan dapat memilih rentang waktu untuk menampilkan laporan tersebut dan ketika menekan Tampilkan Laporan barulah laporan tersebut akan muncul sesuai dengan rentang waktu yang dipilih. Tombol Laporan Barang Keluar berfungsi untuk menampilkan laporan barang masuk yang terdapat laporan barang keluar harian, laporan barang keluar mingguan dan laporan barang keluar bulanan. laporan barang keluar harian akan menampilkan laporan barang-barang yang keluar pada hari itu, pada laporan barang keluar mingguan akan menampilkan barang-barang yang keluar pada minggu tersebut dan pada laporan bulanan dapat memilih rentang waktu untuk menampilkan laporan tersebut dan ketika menekan Tampilkan Laporan barulah laporan tersebut akan muncul sesuai dengan rentang waktu yang dipilih. Tampilan Laporan, Laporan Barang Masuk dan Tampilan Barang keluar dapat dilihat pada gambar 3.15, gambar 3.16 dan gambar 3.17.



Gambar 3. 15 Tampilan Laporan

Two side-by-side screenshots of the mobile application. The left screenshot shows a list of reporting options under "Laporan Barang Masuk": "Laporan Harian", "Laporan Mingguan", and "Laporan Bulanan". The right screenshot shows a detailed view of a daily report titled "Laporan Barang Masuk Harian" for June 26, 2024. It lists an entry for "paku" (picks) with a quantity of 20, noting a check was performed. Another entry for "amplas" (clips) with a quantity of 20 is also listed, with a note to check the data.

← Barang Masuk Mingguan		← Barang Masuk Bulanan			
Laporan Barang Masuk					
Laporan Barang Masuk Mingguan					
Jumat, 21 Juni 2024					
Jumat, 21 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: pen2		
Jumat, 21 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: PEN2		
Jumat, 21 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: PEN2		
Jumat, 21 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: PEN2		
Jumat, 21 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: PEN2		
Sabtu, 22 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: COBA Catch		
Sabtu, 22 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: Cek indeks out of bond		
Sabtu, 22 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: 2		
MULAI: 2024-06-05 AKHIR: 2024-06-26					
TAMPILKAN LAPORAN					
Jumat, 21 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: pen2		
Jumat, 21 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: PEN2		
Jumat, 21 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: PEN2		
Jumat, 21 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: PEN2		
Jumat, 21 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: PEN4		
Sabtu, 22 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: COBA Catch		
Sabtu, 22 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: Cek indeks out of bond		
Sabtu, 22 Juni 2024	Nama Barang: AAAAAA	Jumlah Masuk: 2	Keterangan: 2		
Sabtu, 22 Juni 2024 Nama Barang: AAAAAA Jumlah Masuk: 2 Keterangan: 2					

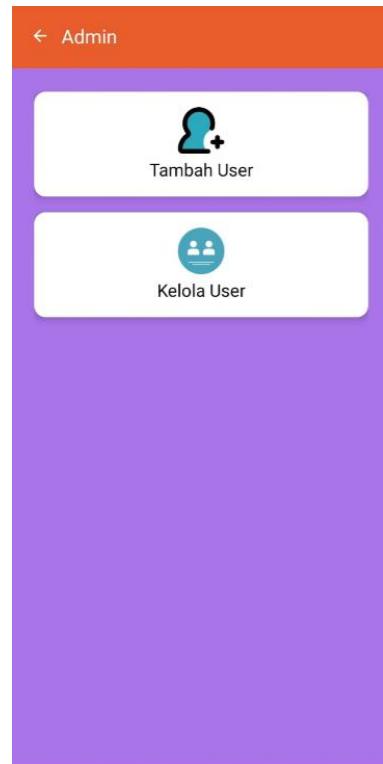
Gambar 3. 16 Tampilan Laporan Barang Masuk

← Laporan Barang Keluar		← Barang Keluar Harian			
Laporan Barang Keluar					
Laporan Barang Keluar Harian					
Laporan Harian	Rabu, 26 Juni 2024	Client: Yosu	Alamat Client: Jl. Pencicillann		
Laporan Mingguan	Rabu, 26 Juni 2024	Nama Barang: paku	Jumlah Keluar: 10		
Laporan Bulanan	Rabu, 26 Juni 2024	Client: Yosu	Alamat Client: Jl. Pencicillann		
	Rabu, 26 Juni 2024	Nama Barang: amplas	Jumlah Keluar: 10		

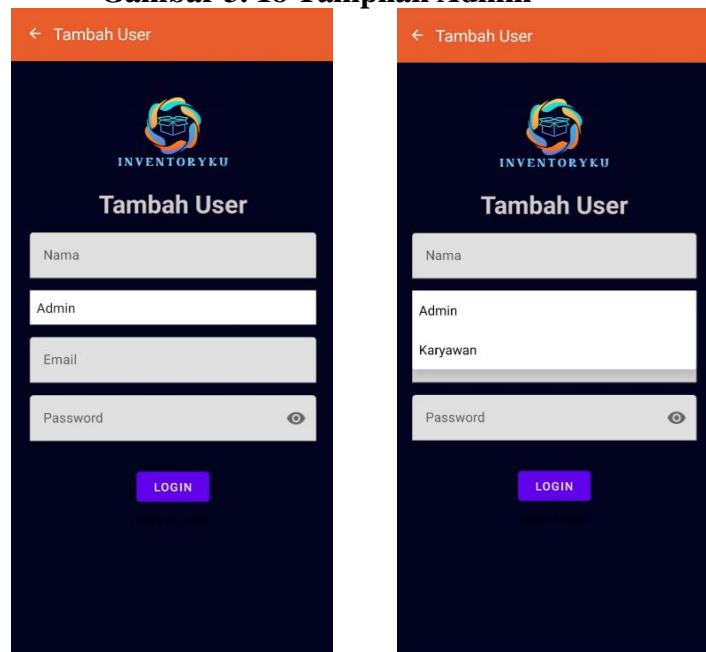
← Barang Keluar Mingguan	
Laporan Barang Keluar Mingguan	
Rabu, 19 Juni 2024	
Client: Yosu	
Alamat Client: Jl. Pencicilann	
Nama Barang: Asep	
Jumlah Keluar: 10	
Kamis, 20 Juni 2024	
Client: Yosu	
Alamat Client: Jl. Pencicilann	
Nama Barang: Daji	
Jumlah Keluar: 20	
Minggu, 23 Juni 2024	
Client: Asep	
Alamat Client: Jl. Pencil12	
Nama Barang: y	
Jumlah Keluar: 1	
Minggu, 23 Juni 2024	
Client: Asep	
Alamat Client: Jl. Pencil12	
Nama Barang: AAAAA	
Jumlah Keluar: 2	
Minggu, 23 Juni 2024	
Client: Yosu	
Alamat Client: Jl. Pencicilann	
Nama Barang: AAAAA	
Jumlah Keluar: 200	
← Barang Keluar Bulanan	
Laporan Barang Keluar Bulanan	
MULAI: 2024-06-05	AKHIR: 2024-06-26
TAMPILKAN LAPORAN	
Rabu, 12 Juni 2024	
Client: Dono	
Alamat Client: Jl. Donooo	
Nama Barang: Aspal	
Jumlah Keluar: 20	
Rabu, 12 Juni 2024	
Client: Yosu	
Alamat Client: Jl. Pencicilann	
Nama Barang: Aspal	
Jumlah Keluar: 20	
Rabu, 12 Juni 2024	
Client: Yosu	
Alamat Client: Jl. Pencicilann	
Nama Barang: Kayu	
Jumlah Keluar: 2	
Sabtu, 15 Juni 2024	
Client: Yosu	
Alamat Client: Jl. Pencicilann	
Nama Barang: Daji	
Jumlah Keluar: 1	
Rabu, 19 Juni 2024	
Client: Yosu	
Alamat Client: Jl. Pencicilann	
Nama Barang: Asep	
Jumlah Keluar: 10	
Kamis, 20 Juni 2024	

Gambar 3. 17 Tampilan Laporan Barang Keluar

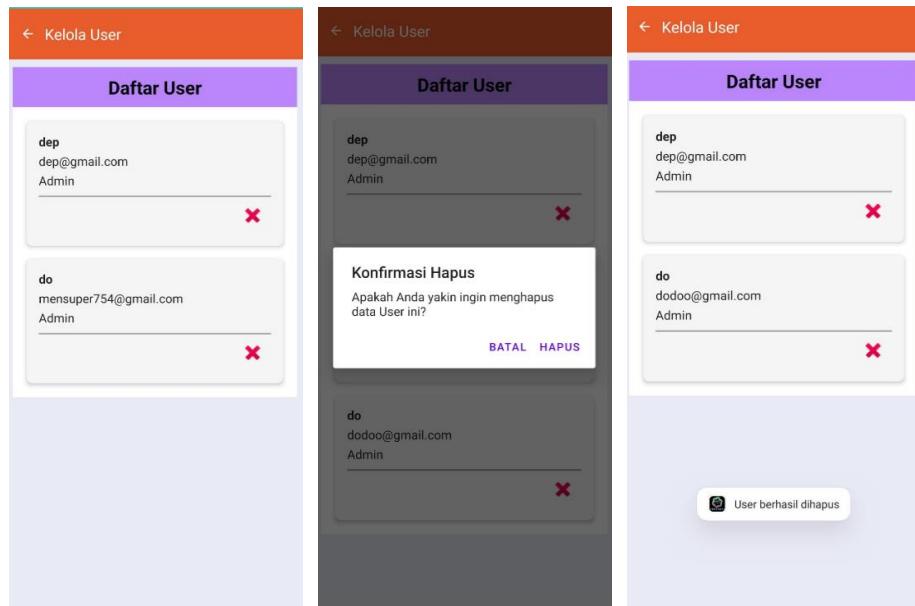
Pada tampilan Admin terdapat 2 menu yaitu menu Tambah user dan Kelola user. Saat Tambah user ditekan maka terdapat tampilan Tambah User yang berfungsi untuk menambahkan user. pada saat register harus mengisikan nama, role yang berupa karyawan atau admin, email dan password. Saat tampilan kelola user pada tampilan admin ditekan maka akan keluar daftar user dan tanda x yang berwarna merah untuk menghapus user saat user berhasil dihapus maka akan keluar toast yang bertuliskan user berhasil dihapus. Tampilan Admin, tampilan Tambah User, dan Kelola User dapat dilihat pada gambar 3.17, gambar 3.18, gambar 3.19.



Gambar 3. 18 Tampilan Admin



Gambar 3. 19 Tampilan Tambah User



Gambar 3. 20 Tampilan Kelola User

3.2 Pengujian Program

Hasil pengujian aplikasi *InventoryKu* di CV Wijaya Lesmana Sejahtera akan ditunjukkan dalam bentuk tabel. Pengujian Login Activity meliputi pengisian data, *error* pada pengisian data, dan fungsi tombol yang dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Uji Coba Form Login

No	Kasus	Hasil yang Diharapkan	Hasil Uji Coba
1.	Pengguna memasukkan <i>email</i> dan <i>password</i> yang salah lalu menekan tombol login	Muncul <i>messagebox</i> yang memberitahukan bahwa Login Gagal	Berhasil
2.	Pengguna tidak memasukkan email.	Aplikasi menampilkan error pada field email dengan pesan "Silakan Masukan Email".	Berhasil
3.	Pengguna tidak memasukkan password	Aplikasi menampilkan error pada field password dengan pesan "Silakan Masukan Password".	Berhasil

No	Kasus	Hasil yang Diharapkan	Hasil Uji Coba
4.	Masukkan <i>username</i> dan <i>password</i> benar dan menekan tombol <i>login</i>	Muncul <i>messagebox</i> bahwa <i>login</i> berhasil dan masuk ke menu utama beserta <i>messagebox</i> status jabatan	Berhasil
5.	Pengguna menekan tombol Lupa <i>Password</i> .	Aplikasi menampilkan halaman Lupa <i>Password</i> .	Berhasil
6.	Pengguna menginput email untuk mengirimkan notifikasi email secara otomatis sesuai dengan email yang diinput.	Aplikasi menampilkan pesan “Reset Password aplikasi telah dikirimkan melalui email anda”	Berhasil
7.	Pengguna sudah login dan membuka aplikasi kembali.	Aplikasi langsung mengarahkan pengguna ke halaman MainActivity tanpa perlu login ulang.	Berhasil

Pengujian Lupa Password Activity meliputi pengisian email, error pengisian email, dan fungsi tombol yang dapat dilihat pada Tabel 3.2

Tabel 3. 2 Uji Coba Fitur Lupa Password

No	Kasus	Hasil yang Diharapkan	Hasil Uji Coba
1.	Pengguna memasukkan email yang valid.	Aplikasi menampilkan pesan: Reset password sudah terkirim ke email anda	Berhasil
2.	Pengguna memasukkan email yang tidak valid.	Error message "Alamat email tidak valid" muncul pada field email.	Berhasil
3.	Pengguna tidak memasukkan email.	Aplikasi menampilkan Pesan Toast "Masukkan Email Anda" muncul.	Berhasil
4.	Pengguna menekan tombol back (panah kembali).	Progress bar tampil. Sistem mengirimkan email reset password Pesan Toast "Reset password sudah terkirim ke email anda" muncul.	Berhasil

No	Kasus	Hasil yang Diharapkan	Hasil Uji Coba
5.	Pengguna memasukkan email yang tidak valid.	Progress bar tampil dan kemudian hilang. Pesan Toast "Error: (pesan error Firebase)".	Berhasil
6.	Pengguna mulai mengetik di field email.	Validasi real-time memeriksa format email.	Berhasil

Pengujian Stock Barang Activity meliputi pengisian data barang, error pengisian data barang, pencarian barang, status barang dan fungsi tombol yang dapat dilihat pada Tabel 3.3.

Tabel 3. 3 Uji Coba Fitur Stock Barang

No	Kasus	Hasil yang Diharapkan	Hasil Uji Coba
1.	Pengguna menambahkan barang baru.	Barang baru disimpan ke Firebase dan muncul di daftar stok.	Berhasil
2.	Pengguna mencari barang melalui SearchView.	Daftar stok difilter sesuai query; tampilkan pesan jika tidak ada barang yang cocok.	Berhasil
3.	Pengguna melihat detail barang dengan mengklik item di daftar stok.	Dialog detail barang muncul dengan informasi lengkap.	Berhasil
4.	Pengguna mengubah status barang.	Status barang di Firebase diperbarui sesuai tombol yang ditekan.	Berhasil
5.	Kesalahan saat menyimpan barang ke Firebase.	Menampilkan pesan error yang sesuai dan ProgressDialog ditutup setelah proses selesai.	Berhasil

Pengujian Client Activity meliputi pengisian nama klien, Alamat klien, error pengisian klien, mengedit klien, menghapus klien dan fungsi tombol yang dapat dilihat pada Tabel 3.4.

Tabel 3. 4 Uji Coba Fitur Client

No	Kasus	Hasil yang Diharapkan	Hasil Uji Coba
1.	Pengguna menambahkan klien baru.	Klien baru disimpan ke Firebase dengan ID unik dan ditampilkan di daftar klien.	Berhasil
2.	Pengguna mencari klien melalui SearchView.	Daftar klien difilter sesuai query dan menampilkan pesan jika tidak ada klien yang cocok.	Berhasil
3.	Pengguna mengklik item di daftar klien.	Dialog detail klien muncul dengan informasi lengkap klien.	Berhasil
4.	Pengguna mengedit data klien.	Data klien di Firebase diperbarui dan daftar klien diperbarui dengan data yang telah diubah.	Berhasil
5.	Pengguna menghapus klien dari daftar.	Klien dihapus dari Firebase dan daftar klien diperbarui.	Berhasil
6.	Pengguna mengisi formulir penambahan atau perubahan klien.	Validasi dilakukan pada setiap field dan menampilkan pesan error jika ada field yang tidak valid.	Berhasil

Pengujian Barang Masuk Activity meliputi pengisian barang masuk, error pengisian data barang masuk, dan fungsi tombol yang dapat dilihat pada Tabel 3.5.

Tabel 3. 5 Uji Coba Fitur Barang Masuk

No	Kasus	Hasil yang Diharapkan	Hasil Uji Coba
1.	Pengguna menginput form pengisian barang masuk.	Data barang masuk di simpan ke dalam database	Berhasil
2.	Pengguna mengklik simpan tanpa mengisi nama barang yang masuk.	Error message "Nama barang harus diisi" muncul pada field Nama Barang.	Berhasil

No	Kasus	Hasil yang Diharapkan	Hasil Uji Coba
3.	Pengguna mengklik simpan tanpa mengisi jumlah barang yang masuk.	Error message "Jumlah barang harus diisi" muncul pada field jumlah barang masuk.	Berhasil
4.	Pengguna memilih Barang yang tersedia.	Mengambil data dari table Stock dan menampilkan dalam form.	Berhasil

Pengujian Barang Keluar Activity meliputi pengisian data barang keluar, error pengisian data barang, dan fungsi tombol yang dapat dilihat pada Tabel 3.6

Tabel 3. 6 Uji Coba Fitur Barang Keluar

No	Kasus	Hasil yang Diharapkan	Hasil Uji Coba
1.	Pengguna menginput form pengisian barang keluar.	Data Barang Keluar disimpan dalam database	Berhasil
2.	Pengguna mengosongkan form barang keluar dan mengklik untuk menyimpan.	Pesan “pastikan semua data terisi dengan benar”	Berhasil
3.	Pengguna memilih barang atau mengetikkan nama barang dan memilih barang.	Mengambil data dari database tabel stock barang dan menampilkan data yang dipanggil	Berhasil
4.	Pengguna memilih atau mengetikkan nama klien dan memilih klien.	Mengambil data dari database table client dan menampilkan data yang dipanggil	Berhasil
5.	Pengguna menginputkan jumlah barang keluar melebihi stok barang yang tersedia	Menampilkan pesan “ jumlah barang keluar melebihi stok yang ada”	Berhasil

Pengujian Pengajuan Activity meliputi penampilan data pengajuan, penerimaan pengajuan, penolakan pengajuan, dan fungsi tombol yang dapat dilihat pada Tabel 3.7.

Tabel 3. 7 Uji Coba Fitur Pengajuan

No	Kasus	Hasil yang Diharapkan	Hasil Uji Coba
1.	Admin menginputkan data Barang Masuk dan Barang Keluar	Data yang di input admin tidak mengirimkan pengajuan kehalaman pengajuan	Berhasil
2.	Karyawan Menginputkan data barang masuk dan barang keluar	Pengajuan yang berasal dari karyawan berhasil ditampilkan dalam halaman pengajuan.	Berhasil
3.	Admin Menyetujui Pengajuan barang masuk atau barang keluar yang diajukan karyawan	Update Data Stok barangmasuk atau barang keluar	Berhasil
4.	Admin menolak Pengajuan barang masuk atau barang keluar yang diajukan karyawan.	Data Stok tidak di update	Berhasil

Pengujian Riwayat Activity meliputi penampilan Riwayat pengajuan, dan fungsi tombol yang dapat dilihat pada Tabel 3.8.

Tabel 3. 8 Uji Coba Fitur Riwayat Pengajuan

No	Kasus	Hasil yang Diharapkan	Hasil Uji Coba
1.	Admin Menyetujui Pengajuan Barang Masuk atau Barang Keluar yang diajukan oleh karyawan	Halaman Riwayat menampilkan data pengajuan yang diterima oleh admin dengan status “Diterima”. Dan update stok barang	Berhasil
2.	Admin Menolak Pengajuan Barang Masuk atau Barang Keluar yang diajukan oleh karyawan	Halaman Riwayat menampilkan data pengajuan yang ditolak oleh admin dengan status “ditolak”	Berhasil

Pengujian Laporan Activity meliputi perekapan laporan harian, laporan mingguan, laporan Bulanan, fungsi tombol yang dapat dilihat pada Tabel 3.9.

Tabel 3.9 Uji Coba Fitur Laporan

Tabel 3. 9 Uji Coba Fitur Laporan

No	Kasus	Hasil yang Diharapkan	Hasil Uji Coba
1.	Pengguna menekan tombol Barang Masuk lalu menekan tombol Laporan Harian	Memunculkan semua daftar Barang masuk yang dilakukan pada hari ini	Berhasil
2.	Pengguna menekan tombol Barang Masuk lalu menekan tombol Laporan Mingguan	Memunculkan semua daftar barang masuk yang telah dilakukan minggu sebelumnya	Berhasil
3.	Pengguna menekan tombol Barang Masuk lalu menekan tombol Laporan Bulanan dan memilih rentang dari tanggal yang diinginkan	Memunculkan semua daftar Barang masuk yang dilakukan pada tanggal yang telah di filter	Berhasil
4.	Pengguna menekan tombol Barang Keluar lalu menekan tombol Laporan Harian	Memunculkan semua daftar Barang keluar yang dilakukan pada hari ini	Berhasil
5.	Pengguna menekan tombol Barang Keluar lalu menekan tombol Laporan Mingguan	Memunculkan semua daftar Barang masuk yang dilakukan pada Minggu Sebelumnya	Berhasil
6.	Pengguna menekan tombol Barang keluar lalu menekan tombol Laporan Bulanan dan memilih rentang dari tanggal yang diinginkan	Memunculkan semua daftar Barang Keluar yang dilakukan pada tanggal yang telah di filter	Berhasil

Pengujian Admin Activity meliputi pendaftaran email kayawan, pengelolaan user, dan fungsi tombol yang dapat dilihat pada Tabel 3.10.

Tabel 3. 10 Uji Coba Fitur admin

No	Kasus	Hasil yang Diharapkan	Hasil Uji Coba
1.	Admin mendaftarkan karyawan perusahaan.	Karyawan terdaftar dan data karyawan di buat.	Berhasil
2.	Pengguna dapat melihat list karyawan dalam Kelola User	Semua nama karyawan yang di daftarkan admin, muncul dalam list	Berhasil
3.	Admin dapat menghapus Karyawan yang ada	Karyawan yang dihapus hilang dari list karyawan	Berhasil

3.3 Hasil Kuesioner

Pengujian kepuasan aplikasi menggunakan kuesioner berupa Google Form yang ditujukan untuk seluruh karyawan di perusahaan. Kuesioner ini ditujukan kepada 9 karyawan dan ditujukan kepada 3 Admin. Kuesioner dilakukan pada tanggal 10 Mei 2024. Kuesioner ini menggunakan skala Likert untuk menilai berbagai aspek kepuasan, dengan keterangan skor sebagai berikut:

1 = Sangat Tidak Setuju (STS)

2 = Tidak Setuju (TS)

3 = Netral (N)

4 = Setuju (S)

5 = Sangat Setuju (SS)

Hasil kuesioner menunjukkan bahwa aplikasi beserta semua fiturnya dapat diterima dengan baik yang ditunjukkan pada Tabel 3.11

Tabel 3. 11 Hasil Kuesioner InventoryKu

No	Pernyataan	Skor Likert				
		SS	S	N	TS	STS
1	Aplikasi mudah dipahami dan digunakan	8	4	1	0	0
2	Proses login ke dalam aplikasi ini mudah.	7	5	0	0	0
3	Fitur lupa password dapat digunakan dengan mudah	7	3	2	0	0
4	Pengelolaan data user di aplikasi ini mudah dilakukan.	9	2	1	0	0

5	Pendaftaran pengguna baru memerlukan waktu yang singkat.	9	3	0	0	0
6	Fitur pengajuan mudah diakses dan digunakan.	8	4	0	0	0
7	Fitur riwayat pengajuan barang mudah diakses dan digunakan.	5	7	0	0	0
8	Fitur Barang Masuk mudah diakses dan digunakan.	7	4	1	0	0
9	Fitur Barang Keluar mudah diakses dan digunakan.	8	2	2	0	0
10	Fitur laporan mudah diakses dan digunakan.	3	4	4	1	0
11	Fitur kelola client mudah diakses dan digunakan.	11	0	1	0	0
12	Fitur pengelolaan stok barang mudah diakses dan digunakan.	12	0	0	0	0
13	Kinerja aplikasi sangat memuaskan.	10	2	0	0	0
14	Aplikasi ini jarang mengalami crash atau bug.	11	0	1	0	0
15	Aplikasi ini memenuhi kebutuhan CV wijaya Lesmana	11	1	0	0	0
16	Tampilan aplikasi sangat enak dipandang	3	5	4	0	0

Hasil kuesioner akan dihitung menggunakan skala likert. Perhitungan skala likert berupa persentase interval untuk menentukan kriteria skor yang terdiri dari 5 kriteria skor yang dapat dilihat pada Tabel 3.12.

Tabel 3. 12 Persentase Kriteria Skor Likert

No	Presentase Skala Likert	Kriteria Skor
1	80% - 100%	Sangat Setuju
2	60% - 80%	Setuju
3	40% - 60%	Netral
4	20% - 40%	Tidak Setuju
5	0% - 20%	Sangat Tidak Setuju

Perhitungan skala likert dihitung berdasarkan total responden dengan skor likert

pada setiap pernyataan dari kuesioner dengan menggunakan persamaan (3.1).

$$Skala\ Likert = Total\ Responden \times Skor\ Likert \quad (3.1)$$

Untuk mendapatkan kriteria skor dari skala likert, maka diperlukan persamaan (3.2) untuk mendapatkan persentase skala likert.

$$Presentase\ Skala\ Likert = \frac{Total\ Skala\ Likert}{Total\ Responden \times Skor\ Likert\ Tertinggi} \times 100\% \quad (3.2)$$

Berdasarkan perhitungan skala likert dari hasil kuesioner pada Tabel 3.11, maka didapatkan kriteria skor untuk masing-masing pernyataan ditunjukkan pada Tabel 3.13

Tabel 3. 13 Kriteria Skor Likert Kuesioner

No	Pernyataan	Presentase Skala Likert	Kriteria Skor
1	Aplikasi mudah dipahami dan digunakan	98	Sangat Setuju
2	Proses login ke dalam aplikasi ini mudah.	92	Sangat Setuju
3	Fitur lupa password dapat digunakan dengan mudah	88	Sangat Setuju
4	Pengelolaan data user di aplikasi ini mudah dilakukan.	93	Sangat Setuju
5	Pendaftaran pengguna baru memerlukan waktu yang singkat.	95	Sangat Setuju
6	Fitur pengajuan mudah diakses dan digunakan.	93	Sangat Setuju
7	Fitur riwayat pengajuan barang mudah diakses dan digunakan.	88	Sangat Setuju
8	Fitur Barang Masuk mudah diakses dan digunakan.	90	Sangat Setuju
9	Fitur Barang Keluar mudah diakses dan digunakan.	90	Sangat Setuju
10	Fitur laporan mudah diakses dan digunakan.	75	Setuju
11	Fitur kelola client mudah diakses dan digunakan.	97	Sangat Setuju
12	Fitur pengelolaan stok barang mudah diakses dan digunakan.	100	Sangat Setuju
13	Kinerja aplikasi sangat memuaskan.	97	Sangat Setuju
14	Aplikasi ini jarang mengalami crash atau bug.	97	Sangat Setuju
15	Aplikasi ini memenuhi kebutuhan CV wijaya Lesmana	98	Sangat Setuju
16	Tampilan aplikasi sangat enak dipandang	78	Setuju

BAB 4

PENUTUP

4.1 Kesimpulan

Berdasarkan laporan kerja praktik dengan judul “Pengembangan Aplikasi Mobile Pencatatan Inventory pada CV Wijaya Lesmana Sejahtera” yang telah dilaksanakan, maka dapat disimpulkan bahwa:

1. Proses penginputan data barang, barang masuk, dan barang keluar baik admin maupun karyawan sebagai pengguna aplikasi sudah dapat dilakukan lebih mudah.
2. Proses pengajuan yang dilakukan oleh karyawan ke admin, menjadi lebih mudah dan terstruktur disertai riwayat pengajuan yang tercantum dalam aplikasi.
3. Proses pembuatan laporan harian, mingguan dan bulanan menjadi lebih cepat karena adanya menu Barang Masuk dan Barang Keluar.
4. Proses penambahan klien lebih cepat dan mudah karena adanya menu untuk menambahkan klien.

4.2 Saran

Berdasarkan hasil kerja praktik dari proses pembuatan hingga testing aplikasi, didapatkan sara yang disampaikan untuk hasil yang lebih untuk kedepannya terhadap “Pengembangan Aplikasi Mobile Pencatatan Inventory pada CV Wijaya Lesmana Sejahtera” yaitu:

1. Diharapkan agar penstrukturkan karyawan menjadi lebih terfokus dan dibutuhkan penambahan untuk lebih banyak admin kedepannya.
2. Perlu diadakan pelatihan terhadap karyawan agar lebih memahami secara mendalam mengenai aplikasi *Inventoryku* CV Wijaya Lesmana.

DAFTAR PUSTAKA

- Sede, D. W. E., Sinsuw, A. A. E., & Najoan, X. B. N. (2015). Rancang Bangun Aplikasi Pemesanan Tiket Online Kapal Laut Berbasis Android. *E-jurnal Teknik Informatika*, 6(1). ISSN: 2301-8364.
- Novita, R., & Zahri, H. (2018). Rancang Bangun Aplikasi Katalog Kerja Praktek dan Tugas Akhir Berbasis Android. *Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi*, 4(2), 209-215. e-ISSN 2502-8995, p-ISSN 2460-8181
- Budiyanto, N. E., & Sultonuddin, R. (2021). Penerapan QR Code pada Presensi Seminar Kerja Praktek Teknik Informatika Unwahas Berbasis Android. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 3(2), 148-155. ISSN: 2656-2855. e-ISSN: 2685-5518.
- Alda, M. (2021). Pemanfaatan Barcode Scanner Pada Aplikasi Manajemen Inventory Barang Berbasis Android. *SISFOKOM (Sistem Informasi dan Komputer)*, 10(3), 368-375.
- Payara, G. R., & Tanone, R. (2018). Penerapan Firebase Realtime Database pada prototype aplikasi pemesanan makanan berbasis Android. *Jurnal Teknik Informatika dan Sistem Informasi*, 4(3), 397.
- Khawas, C. (2018). Application of Firebase in Android App Development-A Study. *International Journal of Computer Applications*, 179(46), 49.
- Hermawati, F. A., & Primasetya, A. (2021). Pemanfaatan Microsoft Excel untuk aplikasi data pelanggan pada usaha kecil jasa pembersihan sepatu. INAMISIA: Jurnal Pengabdian Kepada Masyarakat, 5(4), 850-859.

LAMPIRAN

Surat Pengantar KP dari Kampus



UNIVERSITAS
MDP
FAKULTAS
ILMU KOMPUTER
DAN REKAYASA

Jl. Rajawali No. 14 Palembang 30113,
Sumatera Selatan, Indonesia.
Telp. 0711 376400
www.mdp.ac.id

Palembang, 22 Februari 2024

Nomor : 42/UMDP/II/M/2024
Lampiran : -
Perihal : Pengantar Surat Kerja Praktek

Kepada Yth.

Pimpinan CV WIJAYA LESMANA SEJAHTERA

Jalan Residen Haji Abdul Rozak, Desa/Kelurahan Bukitsangkal, Kec. Kalidoni, Kota Palembang, Provinsi Sumatera Selatan
PALEMBANG

Dengan hormat,

Sesuai dengan kurikulum pada Fakultas Ilmu Komputer dan Rekayasa Universitas Multi Data Palembang, kerja praktik merupakan mata kuliah wajib yang harus ditempuh bagi mahasiswa. Sehubungan dengan ini, kami mengharapkan kesediaan Bapak/Ibu untuk memberikan kesempatan dan izin kepada mahasiswa kami yang akan mengadakan survei/pengambilan data dalam lingkungan perusahaan/instansi yang Bapak/Ibu pimpin. Berikut data mahasiswa yang akan melaksanakan Kerja Praktik :

No	Nama	NPM	Program Studi
1	Dep` Niel Sinaga	2125250058	Informatika (S1)
2	Fernando	2125250014	Informatika (S1)

Kepada mahasiswa yang melakukan Kerja Praktik, diharapkan disamping mendapatkan pengalaman praktis, juga sedapat mungkin memberikan kontribusinya sesuai dengan bidang ilmu yang dimiliki.

Atas kesediaan, dukungan dan kerjasamanya diucapkan terima kasih

Dekan
Ilmu Komputer dan Rekayasa
Universitas Multi Data Palembang

Wijang Widhiarso, Dr., M.Kom.
NIK 021018



Tembusan :
1. Ketua Prog. Studi Informatika (S1)
2. Arsip

Catatan :

- Silakan scan QRcode untuk melihat keaslian surat

Surat Balasan KP dari Perusahaan



CV. WIJAYA LESMANA SEJAHTERA INTERIOR CONTRACTOR

Kitchen Set | Plafon | Karaoke | Kamar Set | Interior/Eksterior, dll
Telp: 0819-5865-3945 Email: wijayalesmanasejahtera@gmail.com

Palembang, 28 Februari 2024

Nomor : 055/WLS/III/24
Perihal : Persetujuan Kerja Praktik

Yang bertanda tangan di bawah ini:

Nama : Indra Wijaya, S.T.
Jabatan : Direktur CV Wijaya Lesmana Sejahtera

Dengan ini menerangkan bahwa :

No	Nama	NPM	Program Studi
1.	Dep Niel Sinaga	2125250058	Informatika (S1)
2.	Fernando	2125250014	Informatika (S1)

Diizinkan untuk Kerja Praktik di CV Wijaya Lesmana Sejahtera pada semester genap tahun akademik 2023/2024 mulai tanggal 01 Maret 2024 s.d. 25 Juni 2024.

Demikian surat keterangan ini dibuat untuk digunakan sebagaimana mestinya.

Palembang, 28 Februari 2024

INDRA WIJAYA, S.T.
DIREKTUR CV. WLS

Surat Selesai KP dari Perusahaan



CV. WIJAYA LESMANA SEJAHTERA INTERIOR CONTRACTOR

Kitchen Set | Plafon | Karaoke | Kamar Set | Interior/Eksterior, dll
Telp: 0819-5865-3945 Email: wijayalesmanasejahtera@gmail.com

Palembang, 25 Juni 2024

Nomor : 067/WLS/VI/30

Perihal : Kerja Praktik

Yang bertanda tangan di bawah ini:

Nama : Indra Wijaya, S.T.

Jabatan : Direktur CV Wijaya Lesmana Sejahtera

Dengan ini menerangkan bahwa :

No	Nama	NPM	Program Studi
1.	Dep Niel Sinaga	2125250058	Informatika (S1)
2.	Fernando	2125250014	Informatika (S1)

Telah menyelesaikan Kerja Praktik di CV Wijaya Lesmana Sejahtera pada semester genap tahun akademik 2023/2024 mulai tanggal 01 Maret 2024 s.d. 25 Juni 2024.

Demikian surat keterangan ini dibuat untuk digunakan sebagaimana mestinya.

Palembang, 25 Juni 2024



INDRA WIJAYA, S.T.
DIREKTUR CV. WLS

Lampiran Foto
Foto Bersama dengan Penyelia



Foto Ruangan Kerja Praktek



Kerangka Acuan Kerja Praktik

**Pengembangan Aplikasi Mobile Pencatatan Inventory pada CV
Wijaya Lesmana Sejahtera**

oleh :

**Fernando 2125250014
Dep' Niel Sinaga 2125250058**



**Program Studi Informatika
Fakultas Ilmu Komputer dan Rekayasa
Universitas Multi Data Palembang Palembang
2024**

1. Pelaksanaan Kerja Praktik

Berdasarkan Surat Keterangan No. 055/WLS/III/2024, yang dikeluarkan oleh CV Wijaya Lesmana Sejahtera Palembang pada tanggal 28 Februari 2024, mahasiswa program studi Informatika Universitas Multi Data Palembang yang bernama Fernando dengan NPM 2125250014 dan Dep'Niel Sinaga dengan NPM 2125250058 dinyatakan diterima melakukan Kerja Praktik di CV Wijaya Lesmana Sejahtera Palembang.

1.1 Tempat, Waktu, dan Topik Kerja Praktik

Nama : CV Wijaya Lesmana Sejahtera
Alamat : Jalan Residen Haji Abdul Rozak, Desa/Kelurahan
Bukit Sangkal, Kec. Kalidoni, Kota Palembang,
Provinsi Sumatera Selatan
Waktu Pelaksanaan : 8 Maret 2024 - 17 Mei 2024
Hari Kerja : Rabu dan Jumat
Waktu Kerja : 13.30 WIB – 15.00 WIB
Topik : Pengembangan Aplikasi Mobile Pencatatan
Inventory pada CV Wijaya Lesmana Sejahtera

1.2 Penyelia Kerja Praktik

Nama : Indra Wijaya, S.T.
Jabatan : Direktur CV Wijaya Lesmana Sejahtera
E-mail : liongkimho@gmail.com
Telepon : 081958653945

1.3 Pelaksana Kerja Praktik

Nama : Fernando
NPM : 2125250014
Program Studi : Informatika
E-mail : fernand0753@mhs.mdp.ac.id
Telepon : 08991305452

Nama : Dep` Niel Sinaga
NPM : 2125250058
Program Studi : Informatika
E-mail : depniel08@mhs.mdp.ac.id
Telepon : 081392644433

1.4 Pembimbing Kerja Praktik

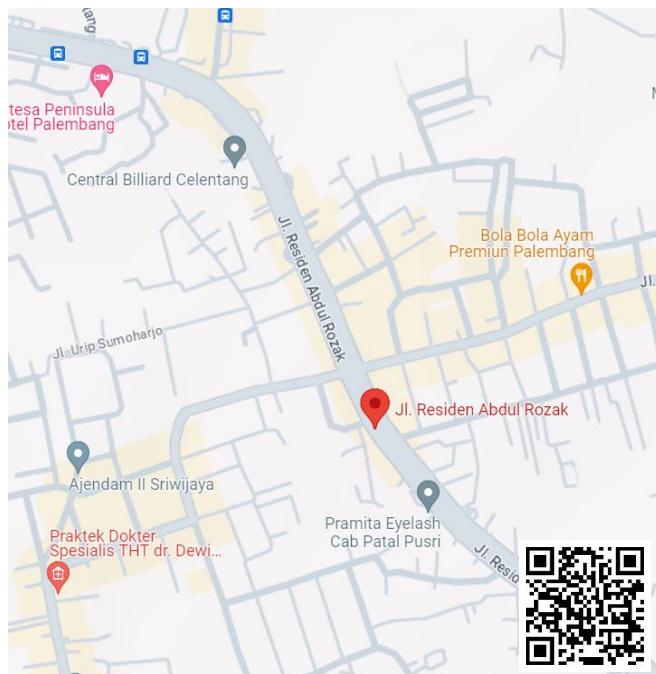
Nama : M. Ezar Al Rivan, M.Kom.
Jabatan : Dosen Tetap Program Studi Informatika
Universitas Multi Data Palembang
E-mail : meedzhar@mdp.ac.id
Telepon : 085797451919

2. Profil Singkat Tempat Kerja Praktik

CV Wijaya Lesmana Sejahtera adalah sebuah perusahaan jasa layanan yang bergerak di bidang *interior contractor*, yang dimana perusahaan akan mengorganisir suatu pembuatan barang sesuai dengan desain keinginan klien. Perusahaan ini didirikan oleh sekelompok anak muda yang memiliki semangat juang yang tinggi, kreatif, dan kebersamaan antar team dengan satu tujuan yaitu menjalankan setiap tugas dengan konsep yang inovatif dan *budget* yang sesuai tanpa menurunkan kualitas. CV Wijaya Lesmana Sejahtera dibagi menjadi beberapa divisi yang memiliki tanggung jawab masing-masing yang selalu berkomitmen menjalankan, serta mewujudkan visi misi perusahaan.

CV Wijaya Lesmana Sejahtera beralamat di Lr. Tembusan No.1828-A, Kel. Bukit Lama, Kec. Ilir Barat I, Kota Palembang, Provinsi Sumatera Selatan. CV Wijaya Lesmana Sejahtera berdiri pada tanggal 27 November 2015 dengan NIB Nomor 1208000220212. Perusahaan ini merupakan penyedia jasa layanan *interior contractor* yang didirikan dan dikelola oleh Indra Wijaya sebagai pemilik sekaligus direktur dari CV Wijaya Lesmana Sejahtera. CV Wijaya Lesmana Sejahtera memiliki pusat penjualan dan perencanaan yang berada di Jalan Residen Haji Abdul Rozak, Desa/Kelurahan Bukit Sangkal, Kec. Kalidoni,

Kota Palembang, Provinsi Sumatera Selatan. Denah lokasi pusat penjualan dan perencanaan diperlihatkan pada Gambar 1 yang dapat diakses melalui link <https://maps.app.goo.gl/jhm3kCU81nmH71Jf7>.



Gambar 1. Denah Lokasi Kerja Praktik CV Wijaya Lesmana Sejahtera

CV Wijaya Lesmana Sejahtera mempunyai visi yakni menjadi sebuah perusahaan yang berkompeten di bidang Kontraktor Interior berbagai produk *Kitchen Set*, Plafon, Karaoke, Kamar Set, *Interior*, *Eksterior*, dll. Dengan meningkatkan kualitas dari produk yang ditawarkan serta Desain yang menarik. Sedangkan misi yang terus akan dikembangkan adalah menjalin hubungan yang baik dengan pelanggan serta mengutamakan kepuasan pelanggan dan memberikan hasil yang memuaskan dan berkualitas pada konsumen.

CV Wijaya Lesmana Sejahtera ini juga memiliki struktur organisasi pegawai yang digunakan untuk mengetahui bagian-bagian apa saja yang merupakan tanggung jawab masing-masing pekerjaan. CV Wijaya Lesmana Sejahtera memiliki beberapa karyawan dengan 4 (empat) divisi, yaitu Sektor Keuangan, Sektor Desain, Sektor *Marketing*, dan Sektor Operasional.

3. Latar Belakang Masalah

CV Wijaya Lesmana Sejahtera bergerak di bidang jasa sebagai Kontraktor Interior bagi orang-orang yang ingin mendesain interior di bangunannya. Kontraktor Interior bertanggung jawab dalam mendesain, merencanakan, mengatur, dan membuat berbagai jenis kebutuhan interior yang ada. Perusahaan akan bekerja sama dengan pihak yang terkait dalam perancangan desain interior yang nanti akan dibuat.

CV Wijaya Lesmana Sejahtera dihadapkan pada berbagai kendala dalam pengelolaan pergudangan. Salah satu permasalahan utama yang dihadapi adalah kecenderungan melakukan kesalahan dalam input atau pencatatan data yang akurat terkait dengan barang-barang yang masuk dan keluar dari pergudangan. Ketidakakuratan informasi terkait inventaris dapat mengganggu efisiensi operasional dan manajemen stok barang secara efektif di perusahaan.

Perusahaan ini kesulitan mengetahui secara pasti informasi tentang barang masuk dan keluar akibat dari kurangnya disiplin dalam memperbarui data yang ada menyebabkan informasi mengenai status barang seringkali tidak akurat atau tidak terkini (*real time*). Hal ini dapat mengganggu efisiensi operasional serta menghambat proses pengambilan keputusan yang berkaitan dengan manajemen pergudangan.

Selain itu, dengan adanya beberapa proyek interior yang berlangsung secara bersamaan, proses pencatatan barang di pergudangan CV Wijaya Lesmana Sejahtera menjadi semakin rumit dan kompleks. Proyek-proyek tersebut memerlukan manajemen stok yang cermat dan terkoordinasi, namun pengelolaan data yang tidak terstruktur menyulitkan upaya tersebut. Keterbatasan dalam proses pencatatan barang juga menyebabkan pengambilan barang menjadi tidak dapat diawasi, sehingga menghambat perencanaan dan eksekusi proyek secara efisien di CV Wijaya Lesmana Sejahtera.

4. Tujuan dan Manfaat Kerja Praktik

4.1 Tujuan

Tujuan Kerja Praktik ini adalah untuk mengembangkan aplikasi mobile demi meningkatkan efisiensi dan efektivitas pengelolaan pergudangan di CV Wijaya Lesmana Sejahtera.

4.2 Manfaat

Manfaat dari Kerja Praktik ini adalah sebagai berikut:

1. Memberikan pengalaman berharga dalam mengembangkan aplikasi yang berkualitas dan berorientasi pada solusi bagi penyelia.
2. Memperluas dan memperdalam keterampilan dalam pengembangan aplikasi mobile, manajemen proyek, dan komunikasi dengan penyelia.

5. Ruang Lingkup Kerja Praktik

Dalam Kerja Praktik ini, penulis memberikan beberapa batasan masalah agar pengembangan aplikasi sesuai tujuan, antara lain sebagai berikut:

1. Aplikasi dapat melakukan proses pengelolaan barang, seperti input data barang, hapus data barang, edit data barang, pencarian data barang, dan pembuatan laporan stok barang
2. Aplikasi dapat melakukan proses pengelolaan data klien seperti input klien, edit klien, hapus klien dan pencarian data klien.
3. Aplikasi dapat mengontrol stok barang, dan melakukan pengecekan stok barang yang akan habis. Aplikasi dapat mengedit data pembelian barang, menghapus data pembelian barang, dan melihat laporan pemakaian barang berdasarkan tempat/klien, data barang masuk, dan data barang keluar berdasarkan periode tertentu, *supplier*, dan jenis barang.
4. Aplikasi dapat menyimpan data klien tetap, mengedit data klien, menghapus data klien, mencari data klien, dan membuat laporan daftar klien tetap, serta laporan pemakaian barang berdasarkan klien masing-masing.

6. Metodologi Pengembangan Aplikasi

Metodologi yang digunakan dalam Kerja Praktik ini adalah. *System Development Life Cycle* (SDLC), yaitu model *Waterfall*. Metode waterfall merupakan pengembangan perangkat lunak dengan penyelesaian satu tahap secara lengkap sebelum melangkah ke tahap berikutnya secara sistematis. Metode *waterfall* ini harus dilakukan secara berurutan sesuai dengan tahap yang ada pada Gambar 2.



Gambar 2. Tahapan Metode *Waterfall*

Adapun beberapa tahapan di dalam *Waterfall* sebagai berikut :

1. Tahap *Analysis*

Pada tahap ini, wawancara dilakukan untuk mengumpulkan informasi kebutuhan apa saja yang diperlukan untuk pengembangan aplikasi dan permasalahan yang ada

2. Tahap *Design*

Pada tahap ini dilakukan pembuatan *Use Case Diagram* sebagai desain perangkat lunak yang mencakup keseluruhan sistem, desain *database*, desain antarmuka pengguna, dan juga spesifikasi teknis.

3. Tahap *Development*

Pada tahap ini dilakukan implementasi perangkat lunak berdasarkan desain yang telah ditetapkan dengan mengubah konsep desain menjadi kode yang dapat dieksekusi oleh komputer.

4. Tahap *Testing*

Setelah tahap pengembangan selesai, perangkat lunak diuji untuk memastikan bahwa setiap komponen yang ada berfungsi sesuai yang diharapkan. Hal yang dapat diuji yakni pengujian fungsionalitas, integrasi, kinerja, dan keamanan.

5. Tahap *Deployment*

Setelah pengujian selesai dilakukan dan tidak ada kendala yang muncul maka perangkat lunak akan diterapkan ke Pengguna akhir. Dalam proses ini dapat dilakukan instalasi perangkat lunak, migrasi data, dan juga pelatihan bagi pengguna.

7. Jadwal Kegiatan

Diperkirakan Kerja Praktik ini dapat diselesaikan dalam kurun waktu 280 jam atau sekitar 10 (Sepuluh) minggu. Rincian kegiatan Kerja Praktik akan dilakukan selama 10 (Sepuluh) minggu yaitu sebagai berikut:

Tabel 1. Rencana Jadwal Kegiatan Kerja Praktik

Waktu	Kegiatan
Minggu ke-1	<ul style="list-style-type: none"> - Mengumpulkan kebutuhan apa saja yang dibutuhkan oleh pihak penyelia untuk aplikasi yang akan dibuat. - Mengidentifikasi permasalahan utama yang dialami oleh pihak penyelia mengenai sistem pengelolaan gudang yang sudah pernah diterapkan.
Minggu ke-2	<ul style="list-style-type: none"> - Pembuatan <i>Use Case Diagram</i> mengenai fitur-fitur yang diperlukan didalam aplikasi. - Persiapan <i>DBMS</i> dan <i>IDE</i> yang akan digunakan.
Minggu ke-3	<ul style="list-style-type: none"> - Perancangan tampilan awal <i>Login Register</i> - Pembuatan <i>database</i> dengan menggunakan <i>Firebase</i> secara <i>Realtime Database</i>
Minggu ke-4	<ul style="list-style-type: none"> - Perancangan sistem untuk <i>Login</i> dan <i>Register</i>. - Perancangan <i>UI</i> dan sistem pengelolaan pergudangan.
Minggu ke-5	<ul style="list-style-type: none"> - Menghubungkan <i>database</i> dengan sistem pengelolaan gudang dan memperbaiki <i>error</i> atau <i>bug</i>. - Perancangan <i>UI</i> dan data pelanggan
Minggu ke-6	<ul style="list-style-type: none"> - Perancangan <i>UI</i> dan sistem pemasukan dan pengeluaran barang dan menghubungkannya ke <i>DBMS</i>. - Memperbaiki <i>error code</i> atau <i>bug</i>. - Perancangan untuk membuat laporan data yang telah di <i>input</i>
Minggu ke-7	<ul style="list-style-type: none"> - Perancangan <i>UI</i> dan relasi setiap kode barang - Pemindahan data barang
Minggu ke-8	<ul style="list-style-type: none"> - Mengintegrasikan setiap komponen-komponen aplikasi dengan <i>DBMS</i>

Waktu	Kegiatan
	<ul style="list-style-type: none"> - Memperbaiki <i>error</i> atau <i>bug</i>
Minggu ke-9	<ul style="list-style-type: none"> - Melakukan pengujian untuk mengetahui performa dari aplikasi - Memberikan aplikasi kepada pihak penyelia
Minggu ke-10	<ul style="list-style-type: none"> - Menyelesaikan Laporan Akhir Kerja Praktik

Yang bertanda-tangan di bawah ini menyatakan telah membaca dan memahami isi dari Kerangka Acuan Kerja Praktik.

Palembang, 09 Maret 2024

Pelaksana Kerja Praktik


Fernando
NPM 2125250014


Dep Niel Sinaga
NPM 2125250058

Menyetujui,
Penyelia


Indra Wijaya, S.T.
Direktur


M. Ezar Al Rivan, M.Kom
NIK. 131092

Mengetahui,
Kaprodi Informatika


Derry Alamsyah, S.Si., M.Kom., M.Pd
NIK. 111069

Log Harian Dep` Niel Sinaga

 UNIVERSITAS MDP	Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-1
--	--	--

NPM : 2125250058
 Nama : Dep` Niel Sinaga
 Lokasi KP : CV Wijaya Lesmana Sejahtera
 Penyelia : Indra Wijaya, S.T.
 Dosen Pembimbing : M. Ezar Al Rivan, M.Kom.

URAJAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Sabtu, 24-02-2024	14:00-18:00	4	Kerjanya ketemu Penyelia KP, Melakukan Uji Hasil Coba, Mengumpulkan Data.
2.	Selasa 27-02-2024	15:00-19:00	4	Mengerjakan Rangkuman KP, Pengembangan Projek KP menggunakan Android Studio dan Firebase,
3.	Rabu, 28-02-2024	09:00-13:00	4	Pembuatan Use Case Diagram, Identifikasi Masalah.
4.	Kamis, 29-02-2024	09:00-13:00	4	Mendesain Aplikasi dengan menggunakan Figma.
5.	Jumat, 01-03-2024	15:00-19:00	4	Melanjutkan Desain pembuatan fitur Aplikasi dengan menggunakan Figma.
6.	Sabtu, 02-03-2024	13:00-18:00	5	Memaparkan Desain Aplikasi Sementara ke pihak Penyelia KP.
Total Waktu KP (jam)			28	...

Palembang, 02 Maret 2024

Mengatahi,

Mahasiswa Pelaksana KP,

Penyelia,

Dosen Pembimbing,

Tanda tangan



Dep` Niel Sinaga

Tanda tangan & CAP



Indra Wijaya, S.T.

Tanda tangan



M. Ezar Al Rivan, M.Kom.

 UNIVERSITAS MDP	Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-2
--	--	--

NPM : 2125250058
 Nama : Dep' Niel Sinaga
 Lokasi KP : CV Wijaya Lesmana Sejahtera
 Penyelia : Indra Wijaya, S.T.
 Dosen Pembimbing : M. Ezar Al Rivan, M.Kom.

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Sabtu, 04-03- 2024	13:00 - 17:00	4	Melanjutkan Desain Pembuatan Fitur Aplikasi dengan menggunakan Figma
2.	Selasa, 05-03- 2024	15:00 - 19:00	4	Persiapan DBMS untuk Database Aplikasi menggunakan Firebase
3.	Rabu, 06-03- 2024	09:00 - 12:00	3	Revisi Desain Aplikasi Figma
4.	Jumat, 07-03- 2024	09:00 - 13:00	4	Menyusun Kerangka Acuan Tesis Praktik
5.	Jumat, 08-03- 2024	15:00 - 18:00	4	Menyusun Kerangka Acuan Tesis Praktik
6.	Sabtu, 09-03- 2024	09:00 - 15:00	6	Revisi Kerangka Acuan Tesis Praktik
Total Waktu KP (jam)			25	

Palembang, 09 Maret 2024

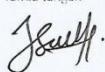
Mengetahui,

Mahasiswa Pelaksana KP,

Penyelia,

Dosen Pembimbing,

Tanda tangan



Dep' Niel Sinaga

Tanda tangan & CAP



CV WIJAYA LESMANA
SEJAHTERA
Indra Wijaya, S.T.

Tanda tangan



M. Ezar Al Rivan, M.Kom.



Fakultas
Ilmu Komputer dan Rekayasa
Program Studi
Informatika

**FORMULIR LOG HARIAN
KERJA PRAKTIK
MINGGU KE-3**

NPM 2125250058
Nama Dep' Niel Sinaga
Lokasi KP CV Wijaya Lesmara Sejahtera
Penyelia Indra Wijaya, S.T.
Dosen Pembimbing M. Ezar Al Rivai, M.Kom.

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Senin, 11-03- 2024	13:00- 16:00	3	Penyelesaian Laporan JCA SCP
2.	Selasa, 12-03- 2024	15:00- 18:00	3	Membuat Tampilan Splash Screen, Register, Login, lupa Password
3.	Rabu, 13-03- 2024	10:00- 13:00	3	Melanjutkan Pembuatan lupa Password dan Register, dan memperbaiki Bug
4.	Kamis, 14-03- 2024	10:00- 14:00	4	Mendesain tampilan Aplikasi Awal
5.	Jumat, 15-03- 2024	15:00- 18:00	3	Melanjutkan Desain Aplikasi Awal dan Membuat Tampilan Menu
6.	Sabtu, 16-03- 2024	09:00- 13:00	4	Membuat Tampilan Menu Client dan Input Client
Total Waktu KP (jam)		20	...	

Palembang, 16 Maret 2024

Mengetahui

Mahasiswa Pelaksana KP

Penyelia

Rosen Pembimpinan

Tardigrades

Jerry

Dep Niel Sinaga

Fundamentals

Tanda tangan
& CAP

Tanda tangan

M. Far Aghayev, M. Kurnaz

Program Studi Informatika, Fakultas Ilmu Komputer dan Rekayasa
Universitas Multi Data Palembang

	Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-4
---	--	---

NPM 2125250058
 Nama Dep' Niel Sinaga
 Lokasi KP CV Wijaya Lesmana Sejahtera
 Penyelia Indra Wijaya, S.T.
 Dosen Pembimbing M. Ezar Al Rivan, M.Kom.

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Sabtu, 10-03-2024	13:00 - 17:00	4	Memperbaiki bug pada Input Client, Mendesain menu Client
2.	Selasa, 13-03-2024	15:00 - 19:00	4	Menampilkan data Client, Membuat Filter edit dan Hapus Client
3.	Rabu 20-03-2024	09:00 - 12:00	3	Memperbaiki Error pada Filter edit dan hapus Client
4.	Kamis 21-03-2024	10:00 - 14:00	4	Membuat Menu Stock Barang, Input Data Barang.
5.	Jumat, 22-03-2024	15:00 - 19:00	4	Memperbaiki error pada Stock Barang dan mendesain menu Stock Barang
6.	Sabtu, 23-03-2024	09:00 - 11:00	2	Mendesain menu Stock Barang
Total Waktu KP (jam)			21 ...	

Palembang, 23 Maret 2024

Mengetahui,

Mahasiswa Pelaksana KP,

Penyelia,

Dosen Pembimbing,

Tanda tangan



Dep' Niel Sinaga

Tanda tangan
& CAP



INDRA WIJAYA, S.T.

Tanda tangan



M. Ezar Al Rivan, M.Kom

	Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE- 5
---	--	--

NPM : 2125250058
 Nama : Dep' Niel Sinaga
 Lokasi KP : CV Wijaya Lesmana Sejahtera
 Penyelia : Indra Wijaya, S.T.
 Dosen Pembimbing : M. Ezar Al Rivan, M.Kom.

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Senin, 25-03-2024	12:00 - 16:00	4	Membuat dialog tampil Stock Beserta Status Penggunaan Barang
2.	Selasa, 26-03-2024	15:00 - 18:00	3	Memperbaiki Error dan mendesain dialog tampil Stock
3.	Rabu, 27-03-2024	09:00 - 14:00	5	Membuat komponen Pencarian Barang dan Memperbaiki logic dan Mendesain Komponen
4.	Kamis, 28-03-2024	09:00 - 15:00	6	Membuat Menu Barang Masuk, Input, dan Pencarian Barang di Stock Barang .
5.	Jumat, 29-03-2024	15:00 - 19:00	4	Memperbaiki kesalahan dalam penginputan dan Membuat Rules pada saat Penginputan Barang masuk
6.	Sabtu, 30-03-2024	09:00 - 14:00	5	Mendesain menu Barang Masuk dan membuat tampilan awal Barang luar
Total Waktu KP (jam)				26

Palernbang, 30 Maret 2024

Mengetahui,

Mahasiswa Pelaksana KP,

Penyelia,

Dosen Pembimbing,

Tanda tangan



Dep' Niel Sinaga

Tanda tangan
& CAP



Indra Wijaya, S.T.

Tanda tangan



M. Ezar Al Rivan, M.Kom.

	Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-6
---	--	---

NPM : 2125250058
 Nama : Dep' Niel Sinaga
 Lokasi KP : CV Wijaya Lesmana Sejahtera
 Penyelia : Indra Wijaya, S.T.
 Dosen Pembimbing : M. Ezar Al Rivan, M.Kom.

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Senin, 01-04- 2024	12:00 - 18:00	6	Membuat Menu Barang Keluar, Input, dan Pencarian Barang di Stock Barang
2.	Selasa, 02-04- 2024	15:00 - 20:00	5	Memperbaiki Error Retrieve Barang dan Memambahkan Retrieve Client pada menu Client, Meresain Barang Keluar
3.	Rabu, 03-04- 2024	09:00 - 15:00	6	Membuat Menu Pengajuan Barang Masuk, Meretrieve data dari menu Barang Masuk, Meresain Tampilan Pengajuan
4.	Kamis, 04-04- 2024	09:00 - 15:00	6	Membuat Menu Pengajuan Barang Keluar, Memperbaiki Bug Pada Menu Barang Pengajuan Barang Masuk
5.	Jumat, 05-04- 2024	15:00 - 20:00	5	Memperbaiki Error pada saat retrieve data dalam menu Barang Keluar, Meresain Tampilan Pengajuan Barang Keluar
6.	Sabtu, 06-04- 2024	10:00 - 14:00	4	Pengerjaan Desain Akhir Tampilan Pengajuan Barang Masuk dan Pengajuan Barang Keluar
Total Waktu KP (jam)			32	...

Palembang, 06 April 2024

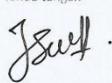
Mengetahui,

Mahasiswa Pelaksana KP,

Penyelia,

Dosen Pembimbing,

Tanda tangan



Dep' Niel Sinaga

Tanda tangan & CAP



Indra Wijaya, S.T.

Tanda tangan



M. Ezar Al Rivan, M.Kom.

 UNIVERSITAS MDP	Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-7
---	--	--

NPM : 2125250058
 Nama : Dep' Niel Sinaga
 Lokasi KP : CV Wijaya Lesmana Sejahtera
 Penyelia : Indra Wijaya, S.T.
 Dosen Pembimbing : M. Ezar Al Rivan, M.Kom.

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Senin, 15-04-2024	11:00 - 17:00	6	Membuat laporan Barang Masuk ; laporan Masukan dan Mengeluarkan .
2.	Selasa, 16-04-2024	15:00 - 19:00	4	Memperbaiki error pada saat retrieve data Barang Masuk
3.	Rabu, 17-04-2024	09:00 - 15:00	6	Membuat laporan Tambahan ; laporan Bulanan, dan Memperbaiki error pada saat retrieve data dan filter Tanggal , Bulan Tahun
4.	Kamis, 18-04-2024	09:00 - 15:00	6	Membuat laporan Barang Keluar ; laporan Masukan dan Mengeluarkan
5.	Jumat, 19-04-2024	15:00 - 20:00	5	Memperbaiki error pada saat retrieve data Barang Keluar
6.	Sabtu, 20-04-2024	09:00 - 15:00	6	Membuat laporan Barang Keluar Bulanan , Memperbaiki error dan Filter Tanggal
Total Waktu KP (jam)				33

Palembang, 20 April 2024

Mengetahui,

Mahasiswa Pelaksana KP,

Penyelia,

Dosen Pembimbing,

Tanda tangan



Dep' Niel Sinaga

Tanda tangan & CAP



Indra Wijaya, S.T.

Tanda tangan



M. Ezar Al Rivan, M.Kom

 UNIVERSITAS MDP	Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-8
---	--	--

NPM 2125250058
 Nama Dep' Niel Sinaga
 Lokasi KP CV Wijaya Lesmana Sejahtera
 Penyelia Indra Wijaya, S.T.
 Dosen Pembimbing M. Ezar Al Rivan, M.Kom.

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Sabtu, 22-04-2024	12:00-17:00	5	Meresam Laporan Barang Masuk, Laporan Barang Keluar
2.	Selasa, 23-04-2024	11:00-17:00	6	Menambahkan File Jobber dengan fitur kategori baru Adipati dan User/Maryam
3.	Rabu, 24-04-2024	09:00-15:00	6	Membuat Tokenisasi Untuk Senap Pengguna Aplikasi dan memperbaiki error
4.	Kamis, 25-04-2024	09:00-15:00	6	Mendesain Tampilan menu Aplikasi
5.	Jumat, 26-04-2024	15:00 - 18:00	3	Memperbarui Input Register, menambahkan Validasi pada form Inputan.
6.	Sabtu, 27-04-2024	09:00-15:00	6	Memperbaiki Bug pada Register dan melanjutkan Verifikasi email untuk lupa password
Total Waktu KP (jam)			32	...

Palembang, 27 April 2024

Mengetahui,

Mahasiswa Pelaksana KP,

Penyelia,

Dosen Pembimbing

Tanda tangan



Dep' Niel Sinaga

Tanda tangan
& CAP



Indra Wijaya, S.T.

Tanda tangan



M. Ezar Al Rivan, M.Kom

 UNIVERSITAS MDP	Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-9
---	--	--

NPM 2125250058
 Nama Dep' Niel Sinaga
 Lokasi KP CV Wijaya Lesmana Sejahtera
 Penyelia Indra Wijaya, S.T.
 Dosen Pembimbing M. Ezar Al Rivan, M.Kom.

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Senin, 29-04- 2024	14:00 - 15:00	1	Mengungsi pihak Penyelia dan mempertemu error
2.	Selasa, 30-04- 2024	-	-	-
3.	Rabu, 01-05- 2024	08:00 - 14:00	6	Pembuatan filter Mengikuti User dan memperbaiki Bug pada pembuatan
4.	Kamis, 02-05- 2024	09:00 - 12:00	3	Mendesain Menu Admin
5.	Jumat, 03-05- 2024	-	-	-
6.	Sabtu, 04-05- 2024	10:00 - 15:00	5	Mendesain Ulang Font, Warna, Background Aplikasi
Total Waktu KP (jam)			18

Palembang, 04 Mei 2024

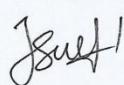
Mengetahui

Mahasiswa Pelaksana KP,

Penyelia,

Dosen Pembimbing,

Tanda tangan



Dep' Niel Sinaga

Tanda tangan
& CAP



Indra Wijaya, S.T.

Tanda tangan



M. Ezar Al Rivan, M.Kom.

 UNIVERSITAS MDP	Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-10
---	--	---

NPM 2125250058
 Nama Dep' Niel Sinaga
 Lokasi KP CV Wijaya Lesmana Sejahtera
 Penyelia Indra Wijaya, S.T.
 Dosen Pembimbing M. Ezar Al Rivan, M.Kom.

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Sabtu, 06-05-2024	12:00 - 15:00	3	Bertemu Penyelia dan memperbaiki Progress
2.	Selasa, 07-05-2024	-	-	
3.	Paias, 08-05-2024	09:00 - 12:00	3	Pengstalan Admin pada perangkat Penyediaan Para karyawan CV Wijaya Lesmana Sejahtera
4.	Kamis, 09-05-2024	09:00 - 12:00	3	Pengstalan Aplikasi pada perangkat Penyelia / Admin . Berserta penyalinan caratanya fitur Aplikasi
5.	Jumat, 10-05-2024	-	-	-
6.	Sabtu 10-05-2024	09:00 - 14:00	5	Melakukan Survey dan Penyerahan Aplikasi Berpersi Reheu Penyelia
Total Waktu KP (jam)		

Palembang, 10 Mei 2024

Mengetahui,

Mahasiswa Pelaksana KP,

Penyelia,

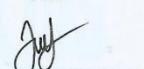
Dosen Pembimbing,

Tanda tangan



Dep' Niel Sinaga

Tanda tangan
& CAP



Indra Wijaya, S.T.

Tanda tangan



M. Ezar Al Rivan, M.Kom.

Log Harian Fernando

 Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-1
---	--

NPM Nama Lokasi KP Penyelia Dosen Pembimbing	2125250014 Fernando CV Wijaya Desainra Indra Wijaya M. Ezar Al Rivan, M.Kom			
URAIAN KEGIATAN				
No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Senin 26-02- 2024	08.00 - 10.00	2	Kunjungan ke tempat penyelenggaraan, mendeklarasikan naskah dan menyampaikan dana
2.	Selasa 27-02- 2024	08.00 - 10.00	2	Mengikuti Rapat kerja KP, pengembangan prajek KP, menerima arahan dari dosen pembimbing
3.	Rabu 28-02- 2024	09.00 - 13.00	4	Pembuatan use case program dan identifikasi masalah
4.	Kamis 29-02- 2024	09.00 - 13.00	4	menulis aplikasi dengan menggunakan Python
5.	Jumat 01-03- 2024	08.00 - 10.00	2	melanjutkan Desain mengoptimalkan aplikasi
6.	Sabtu 02-03- 2024	08.00 - 10.00	2	Menyelesaikan Desain aplikasi, persentasi rencana penyelenggaraan, mobilization (begin)
		Total Waktu KP (jam)	25	...

Palembang, 02 Maret 2024

Mengetahui,

Mahasiswa Pelaksana KP,

Penyelia,

Dosen Pembimbing,

Tanda tangan



Fernando



Indra Wijaya, S.T

Tanda tangan



M. Ezar Al Rivan, M.Kom.

	Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-7
--	--	---

NPM
Nama
Lokasi KP
Penyelia
Dosen Pembimbing

2175290019
Fernando
Evriwijaya Lesmana
Indra Wijaya
M. Ezar Al Rivan, M.Kom

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Sabtu 04-03-2024	11.00 - 15.00	4	menyelesaikan desain aplikasi & plesiasi dengan menyajikan kryma
2.	Sabtu 05-03-2024	13.00 - 17.00	4	menyiapkan DBMS untuk Database Aplikasi menggunakan Firebase
3.	Rabu 06-03-2024	11.00 - 14.00	3	Rewisasi Desain Aplikasi
4.	Kamis 07-03-2024	11.00 - 13.00	2	menyelesaikan KAKP
5.	Jumat 08-03-2024	13.00 - 17.00	4	Menyelesaikan menyelesaikan KAKP
6.	Sabtu 09-03-2024	10.00 - 15.00	5	Melakukan Revisi KAKP
Total Waktu KP (jam)			24

Palembang, 09 Maret 2024

Mengetahui,

Mahasiswa Pelaksana KP,

Penyelia,

Dosen Pembimbing,

Tanda tangan

Fernando



Tanda tangan
& CAP
Indra Wijaya, S.T

Tanda tangan

M. Ezar Al Rivan, M.Kom.

 <p>UNIVERSITAS MDP</p>	Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-3
---	--	--

NPM 2127250014
 Nama Fernando
 Lokasi KP CV UWIJAYA LEMBARA
 Penyelia Indra Wijaya
 Dosen Pembimbing M. Ezar Al Rivan. M.Kom

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Sabtu 11-03-2024	12.00 - 15.00	3	mengelusukan KTP
2.	Sabtu 12-03-2024	15.00 - 18.00	3	membuat tampilan splash screen, login, lupa password dan register
3.	Rabu 13-03-2024	09.00 - 13.00	4	melanjutkan pembuatan lupa password, dan register; memperbaiki bug
4.	Kamis 14-03-2024	09.00 - 14.00	5	memperbaiki bug dan men desain tampilan aplikasi
5.	Jumat 15-03-2024	15.00 - 19.00	4	melanjutkan desain aplikasi dan membuat tampilan beranda/ menu
6.	Sabtu 16-03-2024	13.00 - 16.00	3	membuat tampilan untuk menu Client, input Client
Total Waktu KP (jam)			22

Palembang, 16-03-2024

Mengetahui,

Mahasiswa Pelaksana KP,

Tanda tangan



Fernando

Penyelia,



Indra Wijaya, S.T

Dosen Pembimbing,-

Tanda tangan



M. Ezar Al Rivan, M.Kom.

 UNIVERSITAS MDP	Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-4
---	--	--

NPM 2128250014
 Nama Fernando
 Lokasi KP CV Wijaya Lesmana
 Penyelia Indra Wijaya
 Dosen Pembimbing M. Ezar Al Rivan, M.Kom

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Sabtu 10-03-2024	12.00 - 16.00	4	memperbaiki bug pada input Client dan mendesain menu Client
2.	Sabtu 17-03-2024	15.00 - 19.00	4	membuat Fatur Edit data Client dan hapus data Client dan menampilkan data Client
3.	Rabu 20-03-2024	09.00 - 12.00	3	memperbaiki bug dan mendesain aplikasi
4.	Kamis 21-03-2024	09.00 - 17.00	4	membuat Fatur Stok barang dan input data barang dan memperbaiki bug
5.	Jumat 22-03-2024	15.00 - 19.00	4	memperbaiki bug dan mendesain menu stok barang
6.	Sabtu 23-03-2024	10.00 - 12.00	2	menDESAIN tampilan Stok barang
Total Waktu KP (jam)			21

Palembang, 23 Maret 2024

Mengetahui,

Mahasiswa Pelaksana KP,

Penyelia,

Dosen Pembimbing, -

Tanda tangan



Fernando



Tanda tangan & CAP
Indra Wijaya, S.T

Tanda tangan



M. Ezar Al Rivan, M.Kom.

 <p>UNIVERSITAS MDP</p>	<p>Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika</p>	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-5
---	--	--

NPM 2125255014
 Nama Penulis Ibu Wijaya Lesdina
 Lokasi KP CV Wijaya Lesdina
 Penelia Ibu Eni Wijaya
 Dosen Pembimbing M. Ezar Al Rivan, M.Kom

URAIAN KEGIATAN				
No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Sabtu 25-03 2024	12.00 - 16.00	4	membuat daftar tampil stock besar di seutas barang active / tidak active
2.	Sabtu 26-03 2024	15.00 - 18.00	3	memperbaiki bug dan menambahkan tampilan Stock
3.	Rabu 27-03 2024	09.00 - 14.00	5	memperbaiki search bar dan memperbaiki bug, dan mendesain
4.	Kamis 28-03 2024	10.00 - 16.00	6	memperbaiki menu barang masuk dan memambahkan pencarian barang di stock baran dan perbaikan namp Cara pendaftaran client
5.	Jumat 29-03 2024	15.00 - 19.00	4	memperbaiki bug pada input barang masuk dan menambahkan rule validasi, fitur menutup barang
6.	Sabtu 30-03 2024	09.00 15.00	6	mendesain menu barang masuk dan membuat menu barang keluar
Total Waktu KP (jam)			28

Palembang, 30 maret 2024

Mengetahui,

Mahasiswa Pelaksana KP,

Penelia,

Dosen Pembimbing,

Tanda tangan



Fernando



Indra Wijaya, S.T

Tanda tangan



M. Ezar Al Rivan, M.Kom.

 <p>UNIVERSITAS MDP</p>	<p>Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika</p>	<p>FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-6</p>
---	--	---

NPM 212525004
 Nama Fernando
 Lokasi KP CV Wijaya Lestana
 Penyelia Indra Wijaya
 Dosen Pembimbing M. Ezar Al Rivan M.Kom

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Sabtu 1-04 2024	12.00 - 17.00	5	membatasi bug dan membaik desain barang keluar
2.	Selasa 2-04 2024	15.00 - 19.00	4	membatasi bug dan men men desain templat
3.	Rabu 3-04 2024	09.00 - 15.00	6	membatasi pengajuan barang masuk dan memperbaiki memperbaiki bug
4.	Kamis 4-04 2024	09.00 - 15.00	6	men perbaiki bug pengajuan barang masuk dan memperbaiki membuat pengajuan barang keluar
5.	Jumat 5-04 2024	14.00 - 19.00	5	men desain templat dan pengajuan dan memperbaiki bug barang keluar
6.	Sabtu 6-04 2024	10.00 - 14.00	4	mendesain templat barang keluar dan barang masuk, memperbaiki bug
Total Waktu KP (jam)			30

Palembang, 6 April 2024

Mengetahui,

Mahasiswa Pelaksana KP,

Tanda tangan



Fernando



Penyelia,

Dosen Pembimbing,

Tanda tangan



M. Ezar Al Rivan, M.Kom.

 UNIVERSITAS MDP	Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-7
--------------------------------------	--	--

NPM	2175250024			
Nama	Fernando			
Lokasi KP	CV Wijaya Lesmana			
Penyelia	Indra Wijaya			
Dosen Pembimbing	M. Ezar Al Rivan, M.Kom.			
URAIAN KEGIATAN				
No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Senin 15-04 2024	12.00 13.00	6	Membuat laporan barang masuk, laporan kurir dan mengolah
2.	Selasa 16-04 2024	13.00 14.00	1	Memperbaiki bug pada laporan barang masuk dan mengolah
3.	Rabu 17-04 2024	09.00 15.00	6	Membuat laporan barang masuk, buktikan laporan dokumen dan memperbaiki error
4.	Kamis 18-04 2024	09.00 15.00	6	Membuat Filter terhadap catatan pada laporan dokumen dan membuat laporan barang keluar harian
5.	Jumat 19-04 2024	13.00 20.00	5	memperbaiki bug dan membuat laporan Mingguan
6.	Jumat 20-04 2024	09.00 15.00	6	memperbaiki bug pada laporan barang keluar mengolah dan membuat laporan barang keluar bulanan
Total Waktu KP (jam)		33	...	

Palembang, 20 April 2024

Mengetahui,

Mahasiswa Pelaksana KP,

Penyelia,

Dosen Pembimbing,

Tanda tangan

Fernando



Tanda tangan

M. Ezar Al Rivan, M.Kom.

 UNIVERSITAS MDP	Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE- 8
--------------------------------	--	---

NPM *2125250014*
 Nama *Fernando*
 Lokasi KP *CV Wijaya Lesmana*
 Penyelia *Indra Wijaya*
 Dosen Pembimbing *M. Ezar Al Rivan*

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Sabtu 22-04 2024	12.00 - 17.00	5	mempersiapkan bag. Parkir Surat barang dan kendar kularan dan memambahkan feature resep resepsi resepsi
2.	Sabtu 23-04 2024	11.00 - 17.00	6	memambahkan role pada aplikasi role Admin dan role karyawan/User
3.	Rabu 24-04 2024	09.00 - 15.00	6	membuat teknisasi untuk sektor Pengumpulan dan memperbaiki error
4.	Kamis 25-04 2024	08.00 - 16.00	7	mempersiapkan prot dan modul dalam aplikasi
5.	Jumat 26-04 2024	15.00 - 19.00	4	mempersiapkan halaman resister dengan memambahkan role saat pendaftaran dan memperbaiki bug
6.	Sabtu 27-04 2024	09.00 - 15.00	6	mempersiapkan bug dan memambahkan feature mengirim email ketika melakukan login login dengan password
Total Waktu KP (jam)			34...	...

Palembang,

Mengetahui,

Mahasiswa Pelaksana KP,

Penyelia,

Dosen Pembimbing,

Tanda tangan

Fernando



Indra Wijaya, S.T

Tanda tangan

M. Ezar Al Rivan, M.Kom.

 <p>UNIVERSITAS MDP</p>	<p>Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika</p>	<p>FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-9</p>
---	--	---

NPM *21252500161*
 Nama *Fernando*
 Lokasi KP *CV Wijaya Lesmana*
 Penyelia *Indra Wijaya*
 Dosen Pembimbing *M. Ezar Al Rivan, M.Kom.*

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	<i>29-04 2024</i>	<i>11.00 15.00</i>	<i>4</i>	<i>bertemu dengan penyelia dan menyampaikan perseroan</i>
2.	<i>30-04 2024</i>	<i>-</i>	<i>-</i>	<i>-</i>
3.	<i>Rabu 1-05 2024</i>	<i>09.00 13.00</i>	<i>6</i>	<i>Perbaiki pembuatan Entitas Kelola G/Sar dan menambahkan entitas</i>
4.	<i>Rabu 2-05 2024</i>	<i>09.00 12.00</i>	<i>3</i>	<i>Mendesain aplikasi menu admin</i>
5.	<i>Jumat 3-05 2024</i>	<i>-</i>	<i>-</i>	<i>-</i>
6.	<i>Sabtu 4-05 2024</i>	<i>09.00 14.00</i>	<i>5</i>	<i>menambahkan desain, jenis font, ukuran, warna</i>
Total Waktu KP (jam)			15	...

Palembang, 4 Mei 2024

Mengetahui,

Mahasiswa Pelaksana KP,

Penyelia,

Dosen Pembimbing, -

Tanda tangan



Fernando



CV WIJAYA LESMANA
Indra Wijaya, S.T

Tanda tangan



M. Ezar Al Rivan, M.Kom.

	Fakultas Ilmu Komputer dan Rekayasa Program Studi Informatika	FORMULIR LOG HARIAN KERJA PRAKTIK MINGGU KE-10
---	--	---

NPM *2122350404*
 Nama *Perrangga*
 Lokasi KP *CV Wijaya Lemang*
 Penyelia *Indra Wijaya*
 Dosen Pembimbing *M. Ezar Al Rivan, M.Kom.*

URAIAN KEGIATAN

No.	Hari, Tanggal	Waktu	Durasi (jam)	Keterangan
1.	Sabtu 6-05 2024	12.00 (15.00)	3	berlatih penyedia dan menampilkannya program apk seolah selesai
2.	Selasa 7-05 2024	-	-	-
3.	Rabu 8-05 2024	09.00 (2.00)	3	menyisip aplikasi pada handphone pegawai CV Wijaya Lemang dan membenarkan pemasaran
4.	Kamis 9-05 2024	09.00 - 12.00	3	menyisip aplikasi pada handphone admin CV Wijaya Lemang dan memberikan kredit pada pegawai
5.	Jumat 10-05 2024	-	-	-
6.	Sabtu 11-05 2024	09.00 - 14.00	5	melakukan survei dan penyampaian aplikasi kepada penyedia
Total Waktu KP (jam)				64 ...

Palembang, 1 Mei 2024

Mengetahui,

Mahasiswa Pelaksana KP,

Tanda tangan



Fernando

Penyelia,

Dosen Pembimbing,

Tanda tangan



Indra Wijaya, SIT, MM

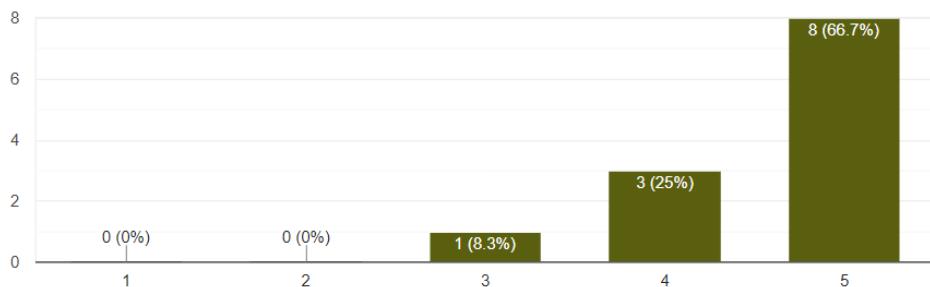
M. Ezar Al Rivan, M.Kom.

Kuesioner

Aplikasi mudah dipahami dan digunakan

 Copy

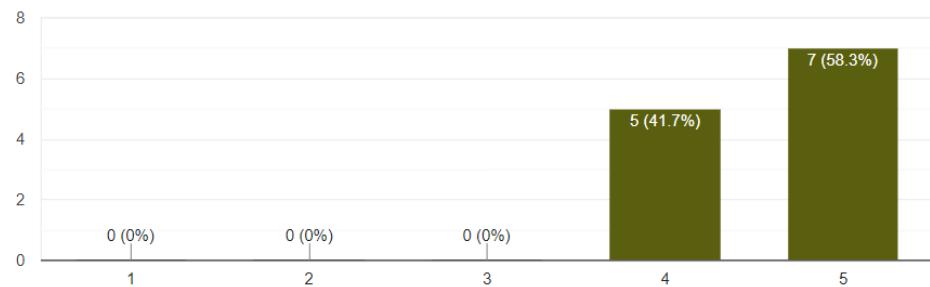
12 responses



Proses login ke dalam aplikasi ini mudah.

 Copy

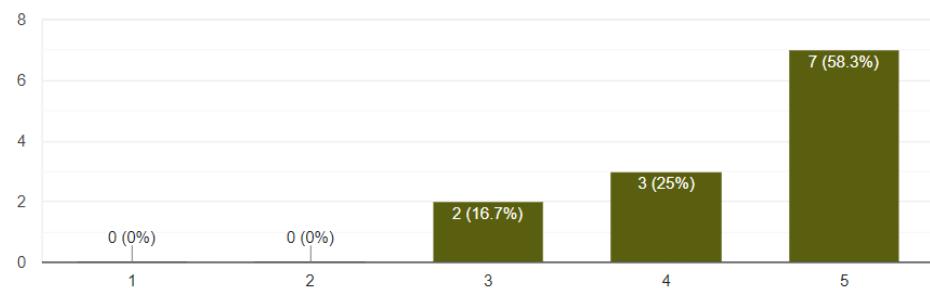
12 responses



Fitur lupa password dapat digunakan dengan mudah

 Copy

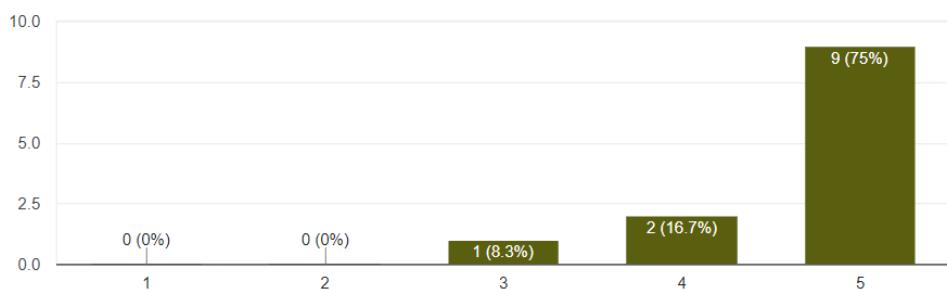
12 responses



Pengelolaan data user di aplikasi ini mudah dilakukan.

 Copy

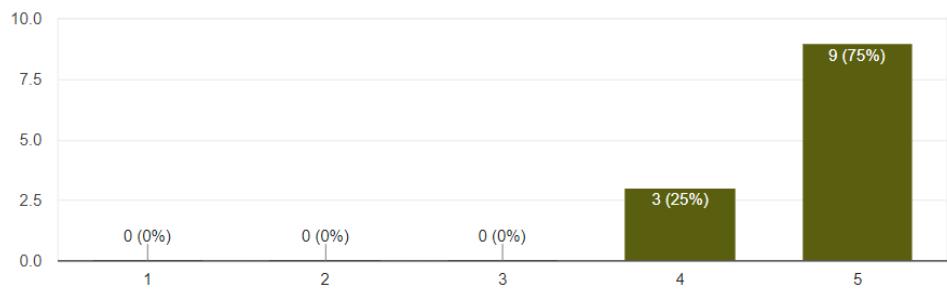
12 responses



Pendaftaran pengguna baru memerlukan waktu yang singkat.

 Copy

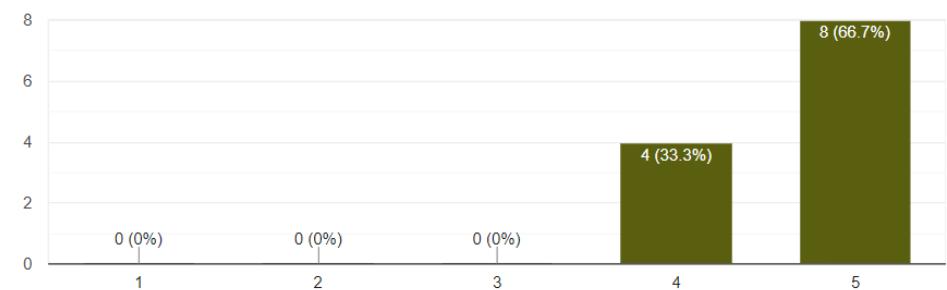
12 responses



Fitur pengajuan mudah diakses dan digunakan.

 Copy

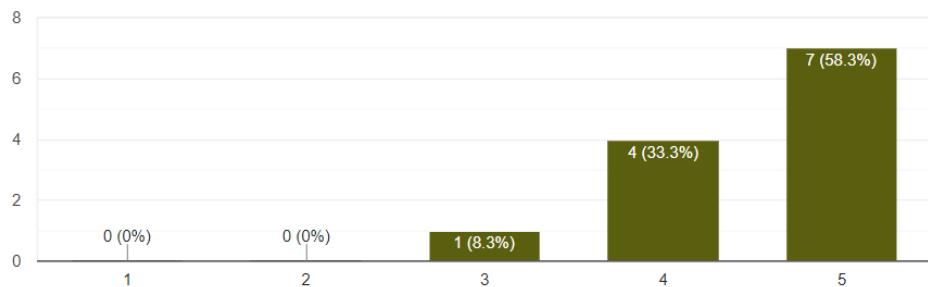
12 responses



Fitur Barang Masuk mudah diakses dan digunakan.

 Copy

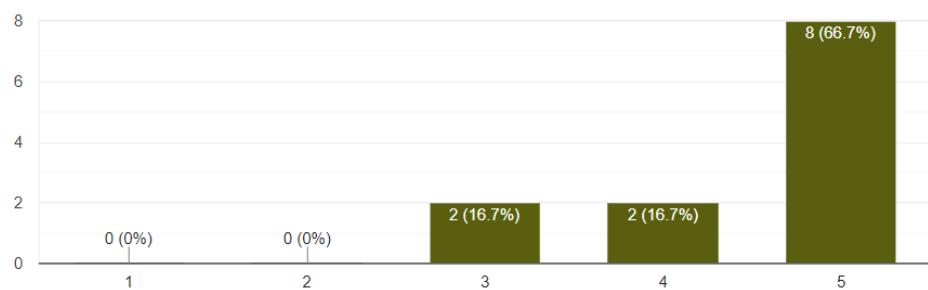
12 responses



Fitur Barang Keluar mudah diakses dan digunakan.

 Copy

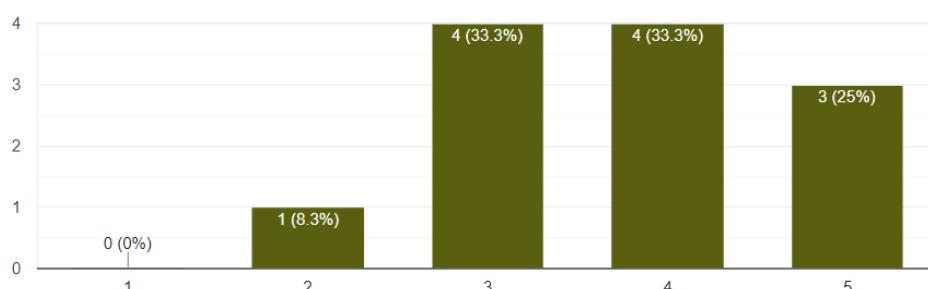
12 responses



Fitur laporan mudah diakses dan digunakan.

 Copy

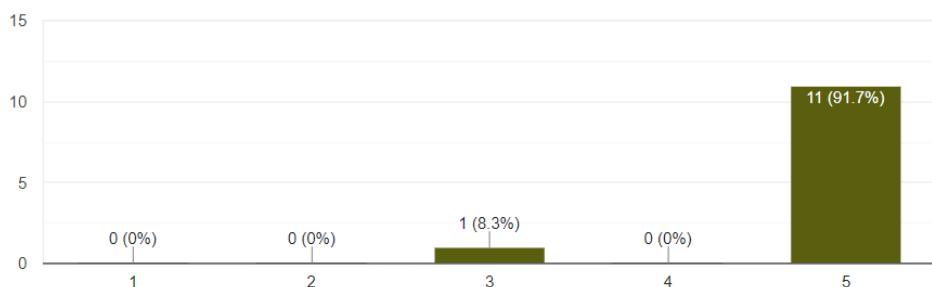
12 responses



Fitur kelola client mudah diakses dan digunakan.

 Copy

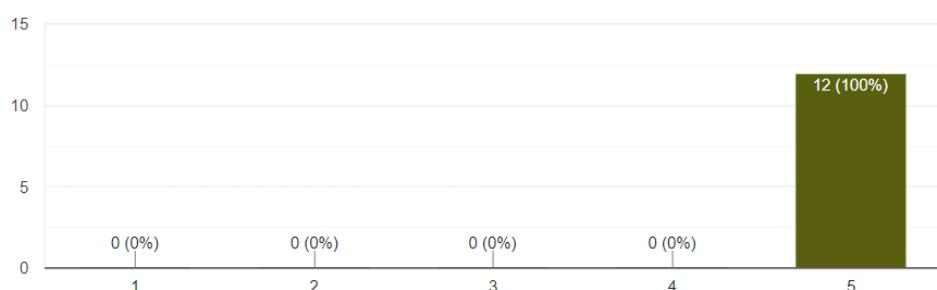
12 responses



Fitur pengelolaan stok barang mudah diakses dan digunakan.

 Copy

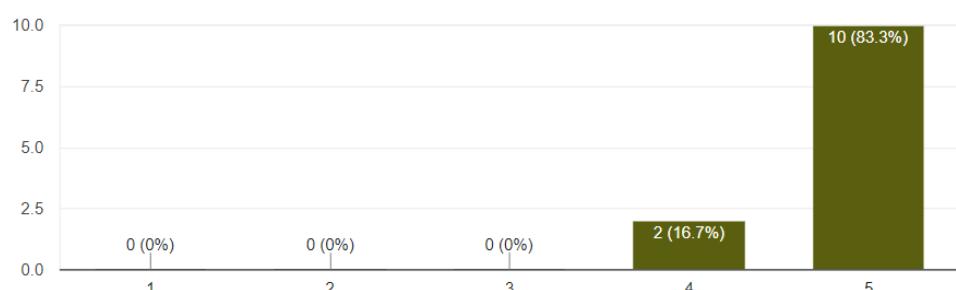
12 responses



Kinerja aplikasi sangat memuaskan

 Copy

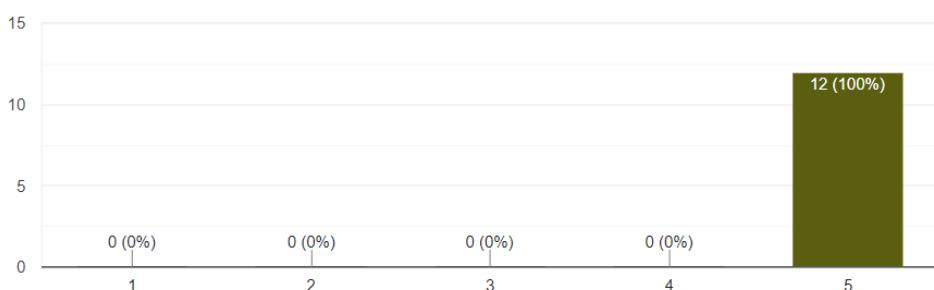
12 responses



Fitur pengelolaan stok barang mudah diakses dan digunakan.

 Copy

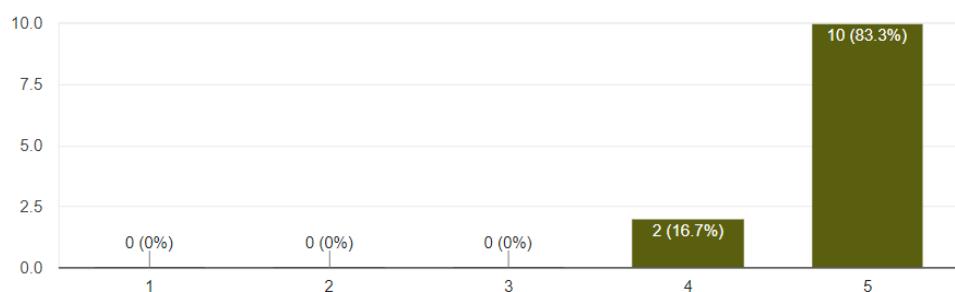
12 responses



Kinerja aplikasi sangat memuaskan

 Copy

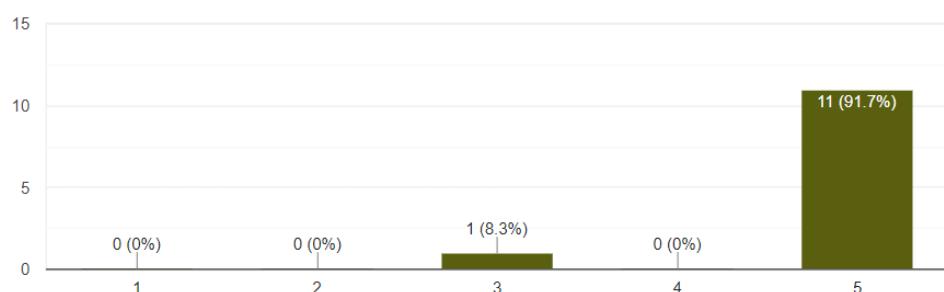
12 responses



Aplikasi ini jarang mengalami crash atau bug.

 Copy

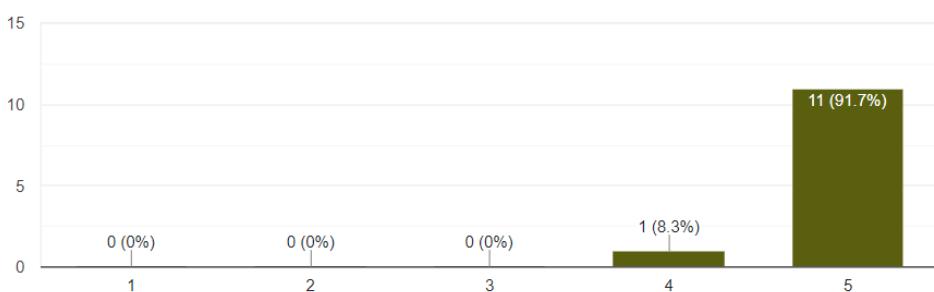
12 responses



Aplikasi ini memenuhi kebutuhan CV wijaya Lesmana

 Copy

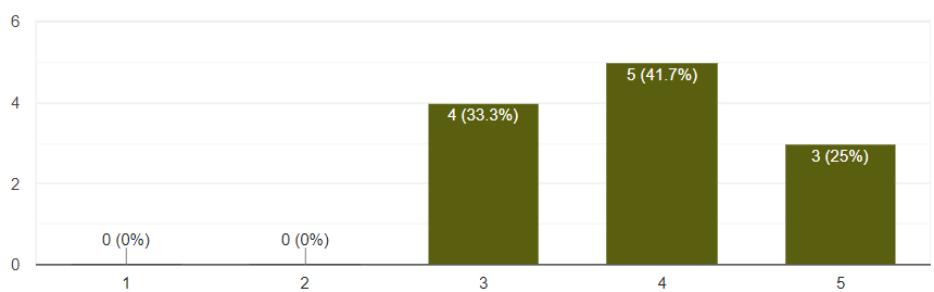
12 responses



Tampilan aplikasi sangat enak dipandang

 Copy

12 responses



Notulen Seminar Kerja Praktik

Kode Program

Kode Program Halaman SplashScreen.java

```
1. package com.example.barangku.activity.user_activity;
2.
3. import androidx.appcompat.app.AppCompatActivity;
4.
5. import android.content.Intent;
6. import android.os.Bundle;
7. import android.os.Handler;
8.
9. import com.example.barangku.R;
10.
11. public class SplashScreen extends AppCompatActivity {
12.
13.     @Override
14.     protected void onCreate(Bundle savedInstanceState) {
15.         super.onCreate(savedInstanceState);
16.         setContentView(R.layout.activity_splash_screen);
17.         new Handler().postDelayed(new Runnable() {
18.             @Override
19.             public void run() {
20.                 Intent mainIntent = new Intent(SplashScreen.this, Login.class);
21.                 startActivity(mainIntent);
22.                 finish();
23.             }
24.         }, 3000); // 3000 milliseconds = 3 seconds
25.     }
26. }
```

Kode Program Halaman Login.java

```
1. package com.example.barangku.activity.user_activity;
2.
3. import androidx.annotation.NonNull;
4. import androidx.appcompat.app.AppCompatActivity;
5.
6. import android.content.Intent;
7. import android.os.Bundle;
8. import android.text.TextUtils;
9. import android.view.View;
10. import android.widget.Button;
11. import android.widget.TextView;
12. import android.widget.Toast;
13.
14. import com.example.barangku.R;
15.
16. import com.google.android.gms.tasks.OnCompleteListener;
17. import com.google.android.gms.tasks.Task;
18. import com.google.android.material.textfield.TextInputEditText;
19. import com.google.firebase.auth.AuthResult;
20. import com.google.firebase.auth.FirebaseAuth;
21. import com.google.firebase.auth.FirebaseUser;
22.
23. public class Login extends AppCompatActivity {
24.     private TextInputEditText etEmail_login, etPassword_login;
```

```

25.     private Button btn_login;
26.     private TextView tv_register,tv_lupapassword;
27.     private FirebaseAuth mAuth;
28.     private String email, password;
29.     @Override
30.     public void onStart() {
31.         super.onStart();
32.         FirebaseUser currentUser = mAuth.getCurrentUser();
33.         if(currentUser != null ){
34.             Intent intent = new Intent(getApplicationContext(),
MainActivity.class);
35.             startActivity(intent);
36.
37.         }
38.     }
39.
40.     @Override
41.     protected void onCreate(Bundle savedInstanceState) {
42.         super.onCreate(savedInstanceState);
43.         setContentView(R.layout.activity_login);
44.         tv_register=findViewById(R.id.tv_register);
45.         mAuth=FirebaseAuth.getInstance();
46.         etEmail_login= findViewById(R.id.et_email_login);
47.         etPassword_login=findViewById(R.id.et_password_login);
48.         btn_login=findViewById(R.id.btn_login_button);
49.         tv_lupapassword=findViewById(R.id.tv_lupaPassword);
50.
51.     tv_lupapassword.setOnClickListener(new View.OnClickListener() {
52.         @Override
53.         public void onClick(View v) {
54.             Intent intent = new Intent(Login.this, LupaPassword.class);
55.             startActivity(intent);
56.             finish();
57.         }
58.     });
59.
60.     btn_login.setOnClickListener(new View.OnClickListener() {
61.         @Override
62.         public void onClick(View v) {
63.             email = etEmail_login.getText().toString();
64.             password= etPassword_login.getText().toString();
65.
66.             if(email.trim().isEmpty()){
67.                 etEmail_login.setError("Silakan Masukan Email ");
68.             }
69.             if(password.trim().isEmpty()){
70.                 etPassword_login.setError("Silakan Masukan Password ");
71.             }
72.             String email = etEmail_login.getText().toString().trim();
73.             String password = etPassword_login.getText().toString().trim();
74.
75.             if (!TextUtils.isEmpty(email) && !TextUtils.isEmpty(password)) {
76.
77.                 mAuth.signInWithEmailAndPassword(email, password)
78.                     .addOnCompleteListener( new
OnCompleteListener<AuthResult>() {
79.                         @Override
80.                         public void onComplete(@NonNull Task<AuthResult>
task) {
81.                             if (task.isSuccessful()) {
82.                                 Toast.makeText(Login.this, "Login berhasil",
Toast.LENGTH_SHORT).show();
83.                                 FirebaseAuth user = mAuth.getCurrentUser();

```

```

84.                         Intent intent = new Intent(Login.this,
85.                           MainActivity.class);
86.                         startActivity(intent);
87.                         finish();
88.                     } else {
89.                         Toast.makeText(Login.this, "login Gagal.",
90.                                         Toast.LENGTH_SHORT).show();
91.                     }
92.                 });
93.             }
94.         );
95.     });
96.     tv_register.setOnClickListener(new View.OnClickListener() {
97.         @Override
98.         public void onClick(View v) {
99.             Intent intent = new Intent(Login.this, Register.class);
100.            startActivity(intent);
101.            finish();
102.        }
103.    });
104. }
105. }
106.

```

Kode Program Halaman LupaPassword.java

```

1. package com.example.barangku.activity.user_activity;
2.
3. import androidx.annotation.NonNull;
4. import androidx.appcompat.app.AppCompatActivity;
5.
6. import android.content.Intent;
7. import android.os.Bundle;
8. import android.text.Editable;
9. import android.text.TextUtils;
10. import android.text.TextWatcher;
11. import android.view.View;
12. import android.widget.ImageView;
13. import android.widget.ProgressBar;
14. import android.widget.TextView;
15. import android.widget.Toast;
16.
17. import com.example.barangku.R;
18. import com.google.android.gms.tasks.OnCompleteListener;
19. import com.google.android.gms.tasks.OnFailureListener;
20. import com.google.android.gms.tasks.OnSuccessListener;
21. import com.google.android.gms.tasks.Task;
22. import com.google.android.material.button.MaterialButton;
23. import com.google.android.material.textfield.TextInputEditText;
24. import com.google.android.material.textfield.TextInputLayout;
25. import com.google.firebase.auth.FirebaseAuth;
26. import com.google.firebase.auth.SignInMethodQueryResult;
27.
28. import java.util.regex.Matcher;
29. import java.util.regex.Pattern;
30.
31. public class LupaPassword extends AppCompatActivity {
32.
33.     private MaterialButton btnResetPassword;

```

```

34.     private TextInputEditText etEmail;
35.     private TextInputLayout ilEmail;
36.     private ProgressBar progressBar;
37.     private FirebaseAuth mAuth;
38.     private String email;
39.     private TextView tv_toolbar;
40.     private ImageView ivback;
41.
42.     @Override
43.     protected void onCreate(Bundle savedInstanceState) {
44.         super.onCreate(savedInstanceState);
45.         setContentView(R.layout.activity_lupa_password);
46.
47.         btnResetPassword = findViewById(R.id.btn_resetpassword);
48.         etEmail = findViewById(R.id.et_email);
49.         ilEmail = findViewById(R.id.il_email);
50.         ivback = findViewById(R.id.iv_back);
51.         progressBar = findViewById(R.id.progressBar);
52.         mAuth = FirebaseAuth.getInstance();
53.
54.         tv_toolbar = findViewById(R.id.tv_judul);
55.         tv_toolbar.setText("Lupa Password");
56.
57.         ivback.setOnClickListener(new View.OnClickListener() {
58.             @Override
59.             public void onClick(View v) {
60.                 Intent intent = new Intent(LupaPassword.this, Login.class);
61.                 startActivity(intent);
62.             }
63.         });
64.
65.         etEmail.addTextChangedListener(new TextWatcher() {
66.             @Override
67.             public void beforeTextChanged(CharSequence s, int start, int count,
int after) {
68.
69.             }
70.
71.             @Override
72.             public void onTextChanged(CharSequence s, int start, int before,
int count) {
73.                 String email = s.toString();
74.                 Pattern pattern = Pattern.compile("^[\w-\\.]+@[\\w-\\.]+[\\w-]{2,4}$");
75.                 Matcher matcher = pattern.matcher(email);
76.                 boolean isValid = matcher.matches();
77.                 if (isValid) {
78.                     ilEmail.setError("");
79.                 } else {
80.                     ilEmail.setError("Alamat email tidak valid");
81.                 }
82.             }
83.
84.             @Override
85.             public void afterTextChanged(Editable s) {
86.                 String email = s.toString();
87.                 if (email.isEmpty()) {
88.                     ilEmail.setError("");
89.                 }
90.             }
91.         });
92.
93.         btnResetPassword.setOnClickListener(new View.OnClickListener() {
94.             @Override

```

```

95.         public void onClick(View v) {
96.             progressBar.setVisibility(View.VISIBLE);
97.
98.             email = String.valueOf(etEmail.getText());
99.             if (TextUtils.isEmpty(email)) {
100.                 Toast.makeText(LupaPassword.this, "Masukkan Email Anda",
101.                             Toast.LENGTH_SHORT).show();
102.                 progressBar.setVisibility(View.GONE);
103.                 return;
104.             }
105.
mAuth.fetchSignInMethodsForEmail(email).addOnCompleteListener(new
OnCompleteListener<SignInMethodQueryResult>() {
106.                 @Override
107.                 public void onComplete(@NonNull
Task<SignInMethodQueryResult> task) {
108.                     if (task.isSuccessful()) {
109.                         boolean isRegistered
= !task.getResult().getSignInMethods().isEmpty();
110.                         if (isRegistered) {
111.
mAuth.sendPasswordResetEmail(email).addOnSuccessListener(new
OnSuccessListener<Void>() {
112.                         @Override
113.                         public void onSuccess(Void unused) {
114.                             Toast.makeText(LupaPassword.this,
"Reset password sudah terkirim ke email anda",
Toast.LENGTH_SHORT).show();
115.                             Intent intent = new
Intent(getApplicationContext(), Login.class);
116.                             startActivity(intent);
117.                             finish();
118.                         }
119.                     }).addOnFailureListener(new OnFailureListener()
{
120.                         @Override
121.                         public void onFailure(@NonNull Exception e)
{
122.                             Toast.makeText(LupaPassword.this,
"Error : " + e.getMessage(),
Toast.LENGTH_SHORT).show();
123.                             progressBar.setVisibility(View.GONE);
124.                         }
125.                     } else {
126.                         Toast.makeText(LupaPassword.this, "Email tidak
terdaftar",
Toast.LENGTH_SHORT).show();
127.                         progressBar.setVisibility(View.GONE);
128.                     }
129.                 } else {
130.                     Toast.makeText(LupaPassword.this, "Error : " +
task.getException().getMessage(),
Toast.LENGTH_SHORT).show();
131.                     progressBar.setVisibility(View.GONE);
132.                 }
133.             }
134.         });
135.     });
136. }
137. });
138. }
139.
140. @Override
141. public void onBackPressed() {
142.     super.onBackPressed();
143.     Intent intent = new Intent(this, Login.class);

```

```

144.         intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |  

Intent.FLAG_ACTIVITY_NEW_TASK);  

145.         startActivity(intent);  

146.         finish();  

147.     }  

148. }  

149.

```

Kode Program MainActivity.java

```

1. package com.example.barangku.activity.user_activity;  

2.  

3. import androidx.annotation.NonNull;  

4. import androidx.appcompat.app.AlertDialog;  

5. import androidx.appcompat.app.AppCompatActivity;  

6.  

7. import android.app.Activity;  

8. import android.content.Intent;  

9. import android.os.Bundle;  

10. import android.view.View;  

11. import android.widget.Button;  

12. import android.widget.GridLayout;  

13. import android.widget.Toast;  

14.  

15. import com.example.barangku.R;  

16. import com.example.barangku.activity.user_activity.laporan.Laporan;  

17. import com.google.firebase.auth.FirebaseAuth;  

18. import com.google.firebase.auth.FirebaseUser;  

19. import com.google.firebaseio.database.DataSnapshot;  

20. import com.google.firebaseio.database.DatabaseError;  

21. import com.google.firebaseio.database DatabaseReference;  

22. import com.google.firebaseio.database FirebaseDatabase;  

23. import com.google.firebaseio.database ValueEventListener;  

24.  

25. public class MainActivity extends AppCompatActivity {  

26.     GridLayout menuUser, menuAdmin;  

27.     FirebaseAuth mAuth = FirebaseAuth.getInstance();  

28.     FirebaseUser mUser = mAuth.getCurrentUser();  

29.     private boolean toastShown = false;  

30.     DatabaseReference reference =  

FirebaseDatabase.getInstance().getReference("User");  

31.  

32.     private Button btnSignout, btnBarangMasukUser,  

btnBarangKeluarUser,btnAdmin, btnPengajuan, btnRiwayatPengajuan;  

33.     private Button btnBarangMasukAdmin, btnBarangKeluarAdmin, btnLaporanAdmin,  

btnClientAdmin, btnStockAdmin;  

34.  

35.     @Override  

36.     protected void onCreate(Bundle savedInstanceState) {  

37.         super.onCreate(savedInstanceState);  

38.         setContentView(R.layout.activity_main);  

39.  

40.         menuUser = findViewById(R.id.menu_user);  

41.         menuAdmin = findViewById(R.id.menu_admin);  

42.  

43.         btnPengajuan = findViewById(R.id.btn_pengajuan);  

44.         btnRiwayatPengajuan = findViewById(R.id.btn_riwayat_pengajuan);  

45.  

46.         btnBarangMasukUser = findViewById(R.id.btn_Barang_Masuk_User);  

47.         btnBarangKeluarUser = findViewById(R.id.btn_Barang_Keluar_User);

```

```

48.
49.     btnBarangMasukAdmin = findViewById(R.id.btn_barang_masuk_admin);
50.     btnBarangKeluarAdmin = findViewById(R.id.btn_barang_keluar_admin);
51.     btnClientAdmin = findViewById(R.id.btn_client_admin);
52.     btnLaporanAdmin = findViewById(R.id.btn_laporan_admin);
53.     btnStockAdmin = findViewById(R.id.btn_stok_barang_admin);
54.     btnAdmin = findViewById(R.id.btn_admin);
55.
56.     btnSignout = findViewById(R.id.btn_Signout);
57.
58.     reference.addValueEventListener(new ValueEventListener() {
59.         @Override
60.         public void onDataChange(@NonNull DataSnapshot snapshot) {
61.             if (toastShown) return;
62.             for (DataSnapshot userSnapshot : snapshot.getChildren()) {
63.                 if
(userSnapshot.getKey().equals(mAuth.getCurrentUser().getUid())) {
64.                     String jabatan =
userSnapshot.child("jabatan").getValue(String.class);
65.                     if (jabatan.equals("Admin")) {
66.                         menuAdmin.setVisibility(View.VISIBLE);
67.                         menuUser.setVisibility(View.GONE);
68.                         Toast.makeText(MainActivity.this, "Anda Login
Sebagai Admin", Toast.LENGTH_SHORT).show();
69.                         btnSignout.setVisibility(View.VISIBLE);
70.                     } else {
71.                         menuUser.setVisibility(View.VISIBLE);
72.                         menuAdmin.setVisibility(View.GONE);
73.                         Toast.makeText(MainActivity.this, "Anda Login
Sebagai Karyawan", Toast.LENGTH_SHORT).show();
74.                         btnSignout.setVisibility(View.VISIBLE);
75.                     }
76.                     toastShown = true;
77.                     break; // Exit the loop once the user is found
78.                 }
79.             }
80.         }
81.
82.         @Override
83.         public void onCancelled(@NonNull DatabaseError error) {
84.             // Handle database error
85.             Toast.makeText(MainActivity.this, "Database error: " +
error.getMessage(), Toast.LENGTH_SHORT).show();
86.         }
87.     });
88.
89.     setupButtonListeners();
90. }
91.
92.     private void setupButtonListeners() {
93.         btnPengajuan.setOnClickListener(v -> pindahHalaman(MainActivity.this,
Pengajuan.class));
94.         btnRiwayatPengajuan.setOnClickListener(v ->
pindahHalaman(MainActivity.this, RiwayatPengajuan.class));
95.         btnBarangMasukUser.setOnClickListener(v ->
pindahHalaman(MainActivity.this, BarangMasuk.class));
96.         btnBarangKeluarUser.setOnClickListener(v ->
pindahHalaman(MainActivity.this, BarangKeluar.class));
97.         btnBarangMasukAdmin.setOnClickListener(v ->
pindahHalaman(MainActivity.this, BarangMasuk.class));
98.         btnBarangKeluarAdmin.setOnClickListener(v ->
pindahHalaman(MainActivity.this, BarangKeluar.class));
99.         btnClientAdmin.setOnClickListener(v -> pindahHalaman(MainActivity.this,
Client.class));

```

```

100.         btnLaporanAdmin.setOnClickListener(v ->
101.             pindahHalaman(MainActivity.this, Laporan.class));
102.         btnStockAdmin.setOnClickListener(v -> pindahHalaman(MainActivity.this,
103.             StockBarang.class));
104.         btnAdmin.setOnClickListener(v -> pindahHalaman(MainActivity.this,
105.             Admin.class));
106.         btnSignout.setOnClickListener(v -> {
107.             FirebaseAuth.getInstance().signOut();
108.             Intent intent = new Intent(MainActivity.this, Login.class);
109.             startActivity(intent);
110.             finish();
111.         });
112.     }
113. 
114.     private void pindahHalaman(Activity activityawal, Class activitytujuan) {
115.         Intent intent = new Intent(activityawal, activitytujuan);
116.         intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
117.         startActivity(intent);
118.     }
119. 
120.     @Override
121.     public void onBackPressed() {
122.         if (getSupportFragmentManager().getBackStackEntryCount() == 0) {
123.             new AlertDialog.Builder(this)
124.                 .setMessage("Are you sure you want to exit the app?")
125.                 .setCancelable(false)
126.                 .setPositiveButton("Yes", (dialog, id) -> {
127.                     moveTaskToBack(true);
128.                     android.os.Process.killProcess(android.os.Process.myPid());
129.                     System.exit(1);
130.                 })
131.                 .setNegativeButton("No", (dialog, id) -> dialog.cancel())
132.                 .create()
133.                 .show();
134.         } else {
135.             super.onBackPressed();
136.         }
137.     }

```

Kode Program Menu BarangMasuk.java

```

1. package com.example.barangku.activity.user_activity;
2.
3. import androidx.annotation.NonNull;
4. import androidx.annotation.Nullable;
5. import androidx.appcompat.app.AppCompatActivity;
6.
7. import android.content.Intent;
8. import android.net.Uri;
9. import android.os.Bundle;
10. import android.view.View;
11. import android.widget.AdapterView;
12. import android.widget.ArrayAdapter;
13. import android.widget.AutoCompleteTextView;
14. import android.widget.Button;
15. import android.widget.EditText;
16. import android.widget.ImageView;

```

```

17. import android.widget.TextView;
18. import android.widget.Toast;
19.
20. import com.bumptech.glide.Glide;
21. import com.example.barangku.R;
22. import com.example.barangku.activity.model.ModelBarangMasuk;
23. import com.example.barangku.activity.model.ModelStock;
24. import com.github.dhaval2404.imagepicker.ImagePicker;
25. import com.google.firebase.auth.FirebaseAuth;
26. import com.google.firebaseio.database.DataSnapshot;
27. import com.google.firebaseio.database.DatabaseError;
28. import com.google.firebaseio.database.DatabaseReference;
29. import com.google.firebaseio.database.FirebaseDatabase;
30. import com.google.firebaseio.database.ValueEventListener;
31.
32. import java.time.LocalDate;
33. import java.util.ArrayList;
34. import java.util.List;
35.
36. public class BarangMasuk extends AppCompatActivity {
37.     private TextView tvToolbar, tvTanggalMasuk, tvNamaBarang, tvSatuan,
tvJumlahItem;
38.     private AutoCompleteTextView etNamaBarang;
39.     private EditText etJumlah, etKeterangan;
40.     private Button btnSimpan;
41.     private ImageView ivback, ivKalender, ivGambar;
42.     private Uri gambarBarang;
43.     private String simpan, namaBarang, satuan, keterangan, tanggalMasuk;
44.     private int jumlahMasuk = 0;
45.     private String userRole;
46.     private DatabaseReference reference =
FirebaseDatabase.getInstance().getReference("StockBarang");
47.     private DatabaseReference barangMasukref =
FirebaseDatabase.getInstance().getReference("BarangMasuk");
48.     private DatabaseReference userRef =
FirebaseDatabase.getInstance().getReference("User");
49.     private List<String> namaBarangList = new ArrayList<>();
50.     private List<ModelStock> stocklist = new ArrayList<>();
51.
52.     private ModelStock currentStockItem;
53.     private FirebaseAuth mAuth;
54.
55.     @Override
56.     protected void onCreate(Bundle savedInstanceState) {
57.         super.onCreate(savedInstanceState);
58.         setContentView(R.layout.activity_barang_masuk);
59.
60.         tvToolbar = findViewById(R.id.tv_judul);
61.         tvToolbar.setText("Barang Masuk");
62.
63.         ivback = findViewById(R.id.iv_back);
64.         etNamaBarang = findViewById(R.id.et_nama_barang);
65.         etJumlah = findViewById(R.id.et_jumlah_barang);
66.         etKeterangan = findViewById(R.id.et_keterangan);
67.
68.         tvNamaBarang = findViewById(R.id.tv_nama_barang);
69.         tvJumlahItem = findViewById(R.id.tv_jumlah_item);
70.         tvSatuan = findViewById(R.id.tv_satuan);
71.
72.         tvTanggalMasuk = findViewById(R.id.tv_tanggal_masuk);
73.         ivKalender = findViewById(R.id.iv_kalender);
74.         ivGambar = findViewById(R.id.iv_gambar_barang);
75.
76.         mAuth = FirebaseAuth.getInstance();

```

```

77.         String currentUserID = mAuth.getCurrentUser().getUid();
78.
79.         LocalDate localDate = LocalDate.now();
80.         int yearToday = localDate.getYear();
81.         int monthToday = localDate.getMonthValue();
82.         int dayOfMonthToday = localDate.getDayOfMonth();
83.
84.         tanggalMasuk = String.format("%04d-%02d-%02d", yearToday, monthToday,
dayOfMonthToday);
85.         tvTanggalMasuk.setText((tanggalMasuk));
86.
87.         btnSimpan = findViewById(R.id.btn_simpan);
88.
89.         ivback.setOnClickListener(new View.OnClickListener() {
90.             @Override
91.             public void onClick(View v) {
92.                 Intent intent = new Intent(BarangMasuk.this,
MainActivity.class);
93.                 startActivity(intent);
94.             }
95.         });
96.
97.         btnSimpan.setOnClickListener(new View.OnClickListener() {
98.             @Override
99.             public void onClick(View view) {
100.                 if (validateFields()) {
101.                     try {
102.                         namaBarang = tvNamaBarang.getText().toString();
103.                         keterangan = etKeterangan.getText().toString();
104.                         jumlahMasuk =
Integer.parseInt(etJumlah.getText().toString());
105.                         satuan = tvSatuan.getText().toString();
106.                     } catch (NumberFormatException e) {
107.                         Toast.makeText(BarangMasuk.this, "Jumlah harus berupa
angka", Toast.LENGTH_SHORT).show();
108.                         return;
109.                     }
110.
111.                     gambarBarang = ivGambar.getDrawable() != null ?
gambarBarang : Uri.EMPTY;
112.                     if (namaBarang.trim().isEmpty() || jumlahMasuk == 0) {
113.                         Toast.makeText(BarangMasuk.this, "Pastikan semua data
terisi dengan benar", Toast.LENGTH_SHORT).show();
114.                     } else {
115.                         simpanBarangMasuk();
116.                         // Reset fields after saving
117.                         resetFields();
118.                     }
119.                 }
120.             }
121.         });
122.
123.         etNamaBarang.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
124.             @Override
125.             public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
126.                 String searchNamaBarang =
parent.getItemAtPosition(position).toString();
127.                 cariBarang(searchNamaBarang);
128.             }
129.         });
130.
131.         retrieveStockData();

```

```

132.        checkUserRole(currentUserId);
133.    } // Akhir On Create
134.
135.    private boolean validateFields() {
136.        if (etNamaBarang.getText().toString().trim().isEmpty()) {
137.            etNamaBarang.setError("Nama barang harus diisi");
138.            return false;
139.        }
140.        if (etJumlah.getText().toString().trim().isEmpty()) {
141.            etJumlah.setError("Jumlah barang harus diisi");
142.            return false;
143.        }
144.        if (tvNamaBarang.getText().toString().trim().isEmpty()) {
145.            Toast.makeText(this, "Nama barang tidak boleh kosong",
146.                Toast.LENGTH_SHORT).show();
147.            return false;
148.        }
149.        if (tvSatuan.getText().toString().trim().isEmpty()) {
150.            Toast.makeText(this, "Satuan barang tidak boleh kosong",
151.                Toast.LENGTH_SHORT).show();
152.            return false;
153.        }
154.
155.        private void resetFields() {
156.            etKeterangan.setText("");
157.            etNamaBarang.setText("");
158.            etJumlah.setText("");
159.            tvSatuan.setText("");
160.            tvJumlahItem.setText("");
161.
162.        ivGambar.setImageDrawable(getResources().getDrawable(R.drawable.insert_image));
163.
164.        private void checkUserRole(String userId) {
165.            userRef.child(userId).addListenerForSingleValueEvent(new
166.                ValueEventListener() {
167.                    @Override
168.                    public void onDataChange(@NonNull DataSnapshot snapshot) {
169.                        if (snapshot.exists()) {
170.                            UserRole =
171.                                snapshot.child("jabatan").getValue(String.class);
172.                            updateUIBasedOnRole();
173.                        } else {
174.                            Toast.makeText(BarangMasuk.this, "Gagal memuat data
175. pengguna", Toast.LENGTH_SHORT).show();
176.                        }
177.                    }
178.                    @Override
179.                    public void onCancelled(@NonNull DatabaseError error) {
180.                        Toast.makeText(BarangMasuk.this, "Gagal memuat data pengguna",
181.                            Toast.LENGTH_SHORT).show();
182.                    });
183.
184.        private void updateUIBasedOnRole() {
185.            if ("Admin".equals(userRole)) {
186.                // Admin can see all functionalities
187.            } else {
188.                // User (non-admin) has limited functionalities

```

```

188.         ivKalender.setVisibility(View.GONE); // Hide the calendar icon for
users
189.     }
190. }
191.
192.     private void simpanBarangMasuk() {
193.         String id = barangMasukref.push().getKey();
194.         if (id == null) {
195.             Toast.makeText(this, "Gagal mendapatkan ID unik",
Toast.LENGTH_SHORT).show();
196.             return;
197.         }
198.
199.         ModelBarangMasuk bm = new ModelBarangMasuk(id, namaBarang, satuan,
keterangan, tanggalMasuk, jumlahMasuk);
200.
201.         if ("Admin".equals(userRole)) {
202.             // Admin saves directly
203.             barangMasukref.child(id).setValue(bm).addOnCompleteListener(task ->
{
204.                 if (task.isSuccessful()) {
205.                     updateStockBarang(currentStockItem, jumlahMasuk);
206.                     Toast.makeText(BarangMasuk.this, "Data berhasil disimpan",
Toast.LENGTH_SHORT).show();
207.                     resetFields();
208.                 } else {
209.                     Toast.makeText(BarangMasuk.this, "Gagal menyimpan data",
Toast.LENGTH_SHORT).show();
210.                 }
211.             });
212.         } else {
213.             // User sends a request for approval
214.             kirimPengajuanKeAdmin(bm);
215.         }
216.     }
217.
218.     private void kirimPengajuanKeAdmin(ModelBarangMasuk bm) {
219.         DatabaseReference notifRef =
FirebaseDatabase.getInstance().getReference("PengajuanBarangMasuk");
220.         String notifId = notifRef.push().getKey();
221.         if (notifId == null) {
222.             Toast.makeText(this, "Gagal mengirim pengajuan",
Toast.LENGTH_SHORT).show();
223.             return;
224.         }
225.         notifRef.child(notifId).setValue(bm).addOnCompleteListener(task -> {
226.             if (task.isSuccessful()) {
227.                 Toast.makeText(BarangMasuk.this, "Pengajuan dikirim ke admin",
Toast.LENGTH_SHORT).show();
228.                 resetFields();
229.             } else {
230.                 Toast.makeText(BarangMasuk.this, "Gagal mengirim pengajuan ke
admin", Toast.LENGTH_SHORT).show();
231.             }
232.         });
233.     }
234.
235.     private void updateStockBarang(ModelStock stockItem, int jumlahMasuk) {
236.         int jumlahBaru =
Integer.parseInt(String.valueOf(Integer.parseInt(stockItem.getJumlahBarang()) +
jumlahMasuk));
237.         stockItem.setJumlahBarang(String.valueOf(jumlahBaru));
238.

```

```

239.     reference.child(stockItem.getId()).setValue(stockItem).addOnCompleteListener(task ->
240.     {
241.         if (task.isSuccessful()) {
242.             Toast.makeText(BarangMasuk.this, "Stok barang berhasil diperbarui", Toast.LENGTH_SHORT).show();
243.         } else {
244.             Toast.makeText(BarangMasuk.this, "Gagal memperbarui stok barang", Toast.LENGTH_SHORT).show();
245.         }
246.     });
247.
248.     private void cariBarang(String searchNamaBarang) {
249.         boolean found = false;
250.         for (ModelStock ms : stockList) {
251.             if (ms.getNamaBarang().equalsIgnoreCase(searchNamaBarang)) {
252.                 currentStockItem = ms;
253.                 tvNamaBarang.setText(ms.getNamaBarang());
254.                 tvSatuan.setText(ms.getSatuan());
255.                 tvJumlahItem.setText(String.valueOf(ms.getJumlahBarang()));
256.                 Glide.with(BarangMasuk.this)
257.                     .load(ms.getGambar())
258.                     .into(ivGambar);
259.                 found = true;
260.                 break;
261.             }
262.         }
263.         if (!found) {
264.             Toast.makeText(BarangMasuk.this, "Barang tidak ditemukan", Toast.LENGTH_SHORT).show();
265.         }
266.     }
267.
268.     private void retrieveStockData() {
269.
reference.orderByChild("enable").equalTo(true).addValueEventListener(new
ValueEventListener() {
270.         public void onDataChange(@NonNull DataSnapshot snapshot) {
271.             namaBarangList.clear();
272.             stockList.clear();
273.
274.             for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
275.                 ModelStock ms = dataSnapshot.getValue(ModelStock.class);
276.                 if (ms != null) {
277.                     namaBarangList.add(ms.getNamaBarang());
278.                     stockList.add(ms);
279.                 }
280.             }
281.             setupAutoCompleteTextView();
282.         }
283.
284.         @Override
285.         public void onCancelled(@NonNull DatabaseError error) {
286.             Toast.makeText(BarangMasuk.this, "Failed to load data", Toast.LENGTH_SHORT).show();
287.         }
288.     });
289. }
290.
291.     @Override
292.     protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
293.         super.onActivityResult(requestCode, resultCode, data);

```

```

294.         if (data != null && data.getData() != null) {
295.             gambarBarang = data.getData();
296.             ivGambar.setImageURI(gambarBarang);
297.         }
298.     }
299.
300.     private void setupAutoCompleteTextView() {
301.         ArrayAdapter<String> adapterBarang = new ArrayAdapter<>(this,
R.layout.list_nama_barang, namaBarangList);
302.         etNamaBarang.setAdapter(adapterBarang);
303.         etNamaBarang.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
304.             @Override
305.             public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
306.                 Toast.makeText(BarangMasuk.this,
parent.getItemAtPosition(position).toString(), Toast.LENGTH_SHORT).show();
307.                 cariBarang(parent.getItemAtPosition(position).toString());
308.             }
309.         });
310.     }
311.
312. }
313.

```

Kode Program Menu BarangKeluar.java

```

1. package com.example.barangku.activity.user_activity;
2.
3. import androidx.annotation.NonNull;
4. import androidx.appcompat.app.AppCompatActivity;
5.
6. import android.content.Intent;
7. import android.net.Uri;
8. import android.os.Bundle;
9. import android.util.Log;
10. import android.view.View;
11. import android.widget.ArrayAdapter;
12. import android.widget.AutoCompleteTextView;
13. import android.widget.Button;
14. import android.widget.EditText;
15. import android.widget.ImageView;
16. import android.widget.TextView;
17. import android.widget.Toast;
18.
19. import com.bumptech.glide.Glide;
20. import com.example.barangku.R;
21. import com.example.barangku.activity.model.ModelBarangKeluar;
22. import com.example.barangku.activity.model.ModelClient;
23. import com.example.barangku.activity.model.ModelStock;
24. import com.google.firebase.auth.FirebaseAuth;
25. import com.google.firebaseio.database.DataSnapshot;
26. import com.google.firebaseio.database.DatabaseError;
27. import com.google.firebaseio.database.DatabaseReference;
28. import com.google.firebaseio.database.FirebaseDatabase;
29. import com.google.firebaseio.database.ValueEventListener;
30.
31. import java.time.LocalDate;
32. import java.util.ArrayList;
33. import java.util.List;

```

```

34.
35. public class BarangKeluar extends AppCompatActivity {
36.     private TextView tv_toolbar, tvTanggalKeluar, tvNamaBarang, tvSatuan,
tvJumlahItem, tvAlamatKlien;
37.     private AutoCompleteTextView etNamaBarang, etNamaKlien;
38.     private EditText etJumlah, etKeterangan;
39.     private Button btnSimpan, btnCari;
40.     private ImageView ivback, ivKalender, ivGambar;
41.     private Uri gambarBarang;
42.     private String namaBarang, satuan, keterangan, tanggalKeluar, namaKlien,
alamatKlien, userRole;
43.     private long jumlahKeluar = 0;
44.     private DatabaseReference stockReference, barangKeluarRef, clientReference,
userRef;
45.     private List<String> namaBarangList, namaClientList;
46.     private List<ModelStock> stockList;
47.     private List<ModelClient> clientList;
48.     private ModelStock currentStockItem;
49.
50.     private FirebaseAuth mAuth;
51.
52.     @Override
53.     protected void onCreate(Bundle savedInstanceState) {
54.         super.onCreate(savedInstanceState);
55.         setContentView(R.layout.activity_barang_keluar);
56.
57.
58.         initViews();
59.         initFirebaseReferences();
60.         setInitialDate();
61.         setAdapters();
62.         retrieveStockData();
63.         retrieveClientData();
64.         mAuth = FirebaseAuth.getInstance();
65.         String currentUserId = mAuth.getCurrentUser().getUid();
66.         checkUserRole(currentUserId); // Ganti dengan user ID yang sesuai
67.
68.         ivback.setOnClickListener(v -> navigateToMainActivity());
69.         btnCari.setOnClickListener(v -> searchBarang());
70.         btnSimpan.setOnClickListener(v -> saveBarangKeluar());
71.     }
72.
73.     private void initViews() {
74.         tv_toolbar = findViewById(R.id.tv_judul);
75.         tv_toolbar.setText("Barang Keluar");
76.
77.         ivback = findViewById(R.id.iv_back);
78.         etNamaBarang = findViewById(R.id.et_nama_barang);
79.         etNamaKlien = findViewById(R.id.et_nama_klien);
80.         etJumlah = findViewById(R.id.et_jumlah_barang);
81.         etKeterangan = findViewById(R.id.et_keterangan);
82.
83.         tvNamaBarang = findViewById(R.id.tv_nama_barang);
84.         tvJumlahItem = findViewById(R.id.tv_jumlah_item);
85.         tvSatuan = findViewById(R.id.tv_satuan);
86.         tvAlamatKlien = findViewById(R.id.tv_alamat_client);
87.
88.         tvTanggalKeluar = findViewById(R.id.tv_tanggal_keluar);
89.         ivKalender = findViewById(R.id.iv_kalender);
90.         ivGambar = findViewById(R.id.iv_gambar_barang);
91.
92.         btnCari = findViewById(R.id.btn_cari_barang);
93.         btnSimpan = findViewById(R.id.btn_simpan);
94.

```

```

95.         namaBarangList = new ArrayList<>();
96.         namaClientList = new ArrayList<>();
97.         stockList = new ArrayList<>();
98.         clientList = new ArrayList<>();
99.     }
100.
101.    private void initFirebaseReferences() {
102.        stockReference =
103.            FirebaseDatabase.getInstance().getReference("StockBarang");
104.        barangKeluarRef =
105.            FirebaseDatabase.getInstance().getReference("BarangKeluar");
106.        clientReference =
107.            FirebaseDatabase.getInstance().getReference("Client");
108.        userRef = FirebaseDatabase.getInstance().getReference("User");
109.    }
110.
111.    private void setInitialDate() {
112.        LocalDate localDate = LocalDate.now();
113.        tanggalKeluar = String.format("%04d-%02d-%02d", localDate.getYear(),
114.            localDate.getMonthValue(), localDate.getDayOfMonth());
115.        tvTanggalKeluar.setText(tanggalKeluar);
116.    }
117.
118.    private void setAdapters() {
119.        ArrayAdapter<String> adapterBarang = new ArrayAdapter<>(this,
120.            R.layout.list_nama_barang, namaBarangList);
121.        etNamaBarang.setAdapter(adapterBarang);
122.        etNamaBarang.setOnItemSelectedListener((parent, view, position, id) ->
123.            updateSelectedBarang(parent.getItemAtPosition(position).toString()));
124.        ArrayAdapter<String> adapterClient = new ArrayAdapter<>(this,
125.            R.layout.list_nama_klien, namaClientList);
126.        etNamaKlien.setAdapter(adapterClient);
127.        etNamaKlien.setOnItemSelectedListener((parent, view, position, id) ->
128.            updateSelectedClient(parent.getItemAtPosition(position).toString()));
129.
130.    private void updateSelectedBarang(String selectedBarang) {
131.        Toast.makeText(this, selectedBarang, Toast.LENGTH_SHORT).show();
132.        cariBarang(selectedBarang);
133.    }
134.
135.    private void updateSelectedClient(String selectedClient) {
136.        for (ModelClient client : clientList) {
137.            if (client.getNama().equals(selectedClient)) {
138.                tvAlamatKlien.setText(client.getAlamat());
139.                break;
140.            }
141.        }
142.
143.    private void searchBarang() {
144.        String searchNamaBarang = etNamaBarang.getText().toString();
145.        String searchNamaClient = etNamaKlien.getText().toString();
146.        if (!searchNamaBarang.isEmpty() && !searchNamaClient.isEmpty()) {
147.            cariBarang(searchNamaBarang);
148.            cariKlien(searchNamaClient);
149.        } else {

```

```

150.             Toast.makeText(this, "Masukkan nama barang dan klien",
151.                         Toast.LENGTH_SHORT).show();
152.         }
153.
154.     private void saveBarangKeluar() {
155.         if (validateFields()) {
156.             namaBarang = tvNamaBarang.getText().toString();
157.             namaKlien = etNamaKlien.getText().toString();
158.             keterangan = etKeterangan.getText().toString();
159.             try {
160.                 jumlahKeluar = Long.parseLong(etJumlah.getText().toString());
161.             } catch (NumberFormatException e) {
162.                 Toast.makeText(this, "Jumlah harus berupa angka",
163.                             Toast.LENGTH_SHORT).show();
164.             return;
165.         }
166.         satuan = tvSatuan.getText().toString();
167.         alamatKlien = tvAlamatKlien.getText().toString();
168.         gambarBarang = ivGambar.getDrawable() != null ? gambarBarang :
Uri.EMPTY;
169.         if (currentStockItem != null) {
170.             long stokTersedia =
171.                 Long.parseLong(currentStockItem.getJumlahBarang());
172.             if (jumlahKeluar > stokTersedia) {
173.                 Toast.makeText(this, "Jumlah barang keluar melebihi stok
yang tersedia!!", Toast.LENGTH_SHORT).show();
174.             } else {
175.                 simpanBarangKeluar();
176.             }
177.         } else {
178.             Toast.makeText(this, "Barang tidak ditemukan!!",
179.                         Toast.LENGTH_SHORT).show();
180.         }
181.     }
182. }
183.
184. private boolean validateFields() {
185.     if (tvNamaBarang.getText().toString().trim().isEmpty() ||
186.         etNamaKlien.getText().toString().trim().isEmpty() ||
187.         etKeterangan.getText().toString().trim().isEmpty() ||
188.         etJumlah.getText().toString().trim().isEmpty() ||
189.         tvSatuan.getText().toString().trim().isEmpty() ||
190.         tvAlamatKlien.getText().toString().trim().isEmpty()) {
191.         return false;
192.     }
193.     return true;
194. }
195.
196. private void simpanBarangKeluar() {
197.     String id = barangKeluarRef.push().getKey();
198.     if (id == null) {
199.         Toast.makeText(this, "Gagal mendapatkan ID unik",
200.                         Toast.LENGTH_SHORT).show();
201.         return;
202.     }
203.     ModelBarangKeluar bk = new ModelBarangKeluar(id, namaBarang, satuan,
keterangan, tanggalKeluar, namaKlien, alamatKlien, jumlahKeluar);
204.

```

```

205.        if ("Admin".equals(userRole)) {
206.            barangKeluarRef.child(id).setValue(bk).addOnCompleteListener(task
-> {
207.                if (task.isSuccessful()) {
208.                    updateStockBarang(currentStockItem, jumlahKeluar);
209.                    Toast.makeText(BarangKeluar.this, "Data berhasil disimpan",
Toast.LENGTH_SHORT).show();
210.                    resetFields();
211.                } else {
212.                    Toast.makeText(BarangKeluar.this, "Gagal menyimpan data",
Toast.LENGTH_SHORT).show();
213.                }
214.            });
215.        } else {
216.            // User sends a request for approval
217.            kirimPengajuanKeAdmin(bk);
218.        }
219.    }
220.
221.    private void kirimPengajuanKeAdmin(ModelBarangKeluar bk) {
222.        DatabaseReference pengajuanRef =
FirebaseDatabase.getInstance().getReference("PengajuanBarangKeluar");
223.        String pengajuanId = pengajuanRef.push().getKey();
224.        if (pengajuanId != null) {
225.
pengajuanRef.child(pengajuanId).setValue(bk).addOnCompleteListener(task -> {
226.                if (task.isSuccessful()) {
227.                    Toast.makeText(this, "Pengajuan barang keluar berhasil
dikirim", Toast.LENGTH_SHORT).show();
228.                    resetFields();
229.                } else {
230.                    Toast.makeText(this, "Pengajuan barang keluar gagal
dikirim", Toast.LENGTH_SHORT).show();
231.                }
232.            });
233.        } else {
234.            Toast.makeText(this, "Gagal membuat pengajuan",
Toast.LENGTH_SHORT).show();
235.        }
236.    }
237.
238.    private void updateStockBarang(ModelStock stock, long jumlahKeluar) {
239.        long stokBaru = Long.parseLong(stock.getJumlahBarang()) - jumlahKeluar;
240.
stockReference.child(stock.getId()).child("jumlahBarang").setValue(String.valueOf(st
okBaru));
241.    }
242.
243.    private void resetFields() {
244.        etNamaBarang.setText("");
245.        etNamaKlien.setText("");
246.        etJumlah.setText("");
247.        etKeterangan.setText("");
248.        tvNamaBarang.setText("");
249.        tvSatuan.setText("");
250.        tvJumlahItem.setText("");
251.        tvAlamatKlien.setText("");
252.        ivGambar.setImageResource(R.drawable.insert_image);
253.        currentStockItem = null;
254.    }
255.
256.    private void retrieveStockData() {

```

```

257.     stockReference.orderByChild("enable").equalTo(true).addValueEventListener(new
258.         @Override
259.         public void onDataChange(@NonNull DataSnapshot snapshot) {
260.             namaBarangList.clear();
261.             stockList.clear();
262.             for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
263.                 ModelStock stock = dataSnapshot.getValue(ModelStock.class);
264.                 if (stock != null) {
265.                     namaBarangList.add(stock.getNamaBarang());
266.                     stockList.add(stock);
267.                 }
268.             }
269.             setAdapters();
270.         }
271.
272.         @Override
273.         public void onCancelled(@NonNull DatabaseError error) {
274.             Log.e("BarangKeluar", "Gagal mengambil data stok: " +
error.getMessage());
275.         }
276.     });
277. }
278.
279. private void retrieveClientData() {
280.     clientReference.addValueEventListener(new ValueEventListener() {
281.         @Override
282.         public void onDataChange(@NonNull DataSnapshot snapshot) {
283.             namaClientList.clear();
284.             clientList.clear();
285.             for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
286.                 ModelClient client =
dataSnapshot.getValue(ModelClient.class);
287.                 if (client != null) {
288.                     namaClientList.add(client.getNama());
289.                     clientList.add(client);
290.                 }
291.             }
292.             setAdapters();
293.         }
294.
295.         @Override
296.         public void onCancelled(@NonNull DatabaseError error) {
297.             Log.e("BarangKeluar", "Gagal mengambil data klien: " +
error.getMessage());
298.         }
299.     });
300. }
301.
302. private void retrieveUserRole() {
303.     // Assume you have a way to get the user role, e.g., from Firebase
304.     // Authentication or Realtime Database
305.     DatabaseReference userRoleRef =
306.         FirebaseDatabase.getInstance().getReference("UserRole");
307.     String userId = "currentUserId"; // Replace with the actual current
308.     user ID
309.     userRoleRef.child(userId).addListenerForSingleValueEvent(new
310.         ValueEventListener() {
311.             @Override
312.             public void onDataChange(@NonNull DataSnapshot snapshot) {
313.                 if (snapshot.exists()) {
314.                     UserRole = snapshot.getValue(String.class);
315.                 }
316.             }
317.         });
318. }
319.
320. 
```

```

312.         }
313.     }
314.
315.     @Override
316.     public void onCancelled(@NonNull DatabaseError error) {
317.         Log.e("BarangKeluar", "Gagal mendapatkan peran pengguna: " +
error.getMessage());
318.     }
319. });
320. }
321.
322. private void cariBarang(String namaBarang) {
323.     for (ModelStock stock : stockList) {
324.         if (stock.getNamaBarang().equals(namaBarang)) {
325.             tvNamaBarang.setText(stock.getNamaBarang());
326.             tvJumlahItem.setText(stock.getJumlahBarang());
327.             tvSatuan.setText(stock.getSatuan());
328.             currentStockItem = stock;
329.             loadImage(stock.getGambar());
330.             break;
331.         }
332.     }
333. }
334.
335. private void cariKlien(String namaKlien) {
336.     for (ModelClient client : clientList) {
337.         if (client.getNama().equals(namaKlien)) {
338.             tvAlamatKlien.setText(client.getAlamat());
339.             break;
340.         }
341.     }
342. }
343.
344. private void loadImage(String imageUrl) {
345.     if (!imageUrl.isEmpty()) {
346.         Glide.with(this)
347.             .load(imageUrl)
348.             .into(ivGambar);
349.     } else {
350.         ivGambar.setImageResource(R.drawable.insert_image);
351.     }
352. }
353.
354. private void checkUserRole(String userId) {
355.     userRef.child(userId).addListenerForSingleValueEvent(new
ValueEventListener() {
356.         @Override
357.         public void onDataChange(@NonNull DataSnapshot snapshot) {
358.             if (snapshot.exists()) {
359.                 UserRole =
snapshot.child("jabatan").getValue(String.class);
360.                 updateUIBasedOnRole();
361.             } else {
362.                 Toast.makeText(BarangKeluar.this, "Gagal memuat data
pengguna", Toast.LENGTH_SHORT).show();
363.             }
364.         }
365.         @Override
366.         public void onCancelled(@NonNull DatabaseError error) {
367.             Toast.makeText(BarangKeluar.this, "Gagal memuat data pengguna",
Toast.LENGTH_SHORT).show();
368.         }
369.     });
370. }

```

```

371.
372.     private void updateUIBasedOnRole() {
373.         if ("Admin".equals(userRole)) {
374.             // Admin can see all functionalities
375.         } else {
376.             // User (non-admin) has limited functionalities
377.             ivKalender.setVisibility(View.GONE); // Hide the calendar icon for
users
378.         }
379.     }
380. }
381.

```

Kode Program Menu Client.java

```

1. package com.example.barangku.activity.user_activity;
2.
3. import androidx.annotation.NonNull;
4. import androidx.appcompat.app.AppCompatActivity;
5. import androidx.appcompat.widget.SearchView;
6. import androidx.recyclerview.widget.LinearLayoutManager;
7. import androidx.recyclerview.widget.RecyclerView;
8.
9. import android.app.Dialog;
10. import android.app.ProgressDialog;
11. import android.content.Intent;
12. import android.os.Bundle;
13. import android.view.View;
14. import android.view.ViewGroup;
15. import android.widget.EditText;
16. import android.widget.ImageView;
17. import android.widget.TextView;
18. import android.widget.Toast;
19.
20. import com.example.barangku.R;
21. import com.example.barangku.activity.adapter.AdapterClient;
22. import com.example.barangku.activity.model.ModelClient;
23. import com.example.barangku.activity.user_activity.utils.ItemClickClient;
24. import com.google.android.material.floatingactionbutton.FloatingActionButton;
25. import com.google.firebase.database.DataSnapshot;
26. import com.google.firebaseio.database.DatabaseError;
27. import com.google.firebaseio.database.DatabaseReference;
28. import com.google.firebaseio.database.FirebaseDatabase;
29. import com.google.firebaseio.database.ValueEventListener;
30. import com.google.firebaseio.storage.FirebaseStorage;
31. import com.google.firebaseio.storage.StorageReference;
32.
33. import java.util.ArrayList;
34. import java.util.List;
35. import java.util.regex.Pattern;
36.
37. public class Client extends AppCompatActivity implements ItemClickClient {
38.     private TextView tvToolbar;
39.     private Dialog dialog, dialogClient, dialogEdit;
40.     private FloatingActionButton fabTambah;
41.     private String namaKlien, alamatKlien, noTelpKlien, emailKlien,
catatanKlien;
42.     private DatabaseReference reference =
FirebaseDatabase.getInstance().getReference("Client");

```

```

43.     DatabaseReference counterRef =
FirebaseDatabase.getInstance().getReference().child("counter").child("clientId");
44.     private StorageReference storageReference =
FirebaseStorage.getInstance().getReference();
45.     private SearchView searchView;
46.     private ImageView ivback;
47.     private ProgressDialog pd;
48.     RecyclerView rvClient;
49.     AdapterClient adapterClient;
50.     private List<ModelClient> list_client = new ArrayList<>();
51.
52.
53.     @Override
54.     protected void onCreate(Bundle savedInstanceState) {
55.         super.onCreate(savedInstanceState);
56.         setContentView(R.layout.activity_client);
57.
58.         tvToolbar = findViewById(R.id.tv_judul);
59.         tvToolbar.setText("Client");
60.
61.         fabTambah = findViewById(R.id.fab_tambah_klien);
62.         searchView = findViewById(R.id.search);
63.         rvClient = findViewById(R.id.rv_client);
64.
65.         ivback = findViewById(R.id.iv_back);
66.         ivback.setOnClickListener(new View.OnClickListener() {
67.             @Override
68.             public void onClick(View v) {
69.                 Intent intent = new Intent(Client.this, MainActivity.class);
70.                 startActivity(intent);
71.             }
72.         });
73.         pd = new ProgressDialog(this);
74.         pd.setMessage("Loading");
75.         pd.setCancelable(false);
76.         pd.show();
77.
78.
79.         searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener()
{
80.             @Override
81.             public boolean onQueryTextSubmit(String query) {
82.                 filterList(query);
83.                 searchView.clearFocus();
84.                 return true;
85.             }
86.
87.             @Override
88.             public boolean onQueryTextChange(String newText) {
89.                 filterList(newText);
90.                 return true;
91.             }
92.         });
93.
94.
95.         adapterClient = new AdapterClient(Client.this, list_client);
96.         adapterClient.setItemClickListener(this);
97.         LinearLayoutManager lm_client = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
98.         rvClient.setLayoutManager(lm_client);
99.         rvClient.setAdapter(adapterClient);
100.
101.        FirebaseDatabase database = FirebaseDatabase.getInstance();
102.        DatabaseReference reference = database.getReference("Client");

```

```

103.        reference.addValueEventListener(new ValueEventListener() {
104.            @Override
105.            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
106.                for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
107.                    ModelClient mc = snapshot.getValue(ModelClient.class);
108.                    list_client.add(mc);
109.                }
110.                adapterClient.notifyDataSetChanged();
111.                pd.dismiss();
112.            }
113.
114.            @Override
115.            public void onCancelled(@NonNull DatabaseError databaseError) {
116.            }
117.        });
118.
119.
120.
121.
122.        fabTambah.setOnClickListener(new View.OnClickListener() {
123.            @Override
124.            public void onClick(View view) {
125.                dialog = new Dialog(Client.this);
126.                dialog.setContentView(R.layout.dialog_tambah_client);
127.
dialog.getWindow().setLayout(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
128.                dialog.setCancelable(false);
129.                dialog.show();
130.
131.                EditText etNamaKlien = dialog.findViewById(R.id.et_nama_klien);
132.                EditText etAlamatKlien =
dialog.findViewById(R.id.et_alamat_klien);
133.                EditText etNoTelpKlien =
dialog.findViewById(R.id.et_telp_klien);
134.                EditText etEmailKlien =
dialog.findViewById(R.id.et_email_klien);
135.                EditText etCatatanKlien =
dialog.findViewById(R.id.et_catatan_klien);
136.
137.
138.                dialog.findViewById(R.id.iv_close).setOnClickListener(new
View.OnClickListener() {
139.                    @Override
140.                    public void onClick(View view) {
141.                        dialog.dismiss();
142.                    }
143.                });
144.                dialog.findViewById(R.id.btn_simpan).setOnClickListener(new
View.OnClickListener() {
145.                    @Override
146.                    public void onClick(View view) {
147.                        namaKlien = etNamaKlien.getText().toString();
148.                        alamatKlien = etAlamatKlien.getText().toString();
149.                        noTelpKlien = etNoTelpKlien.getText().toString();
150.                        emailKlien = etEmailKlien.getText().toString();
151.                        catatanKlien = etCatatanKlien.getText().toString();
152.
153.                        if (namaKlien.trim().isEmpty()) {
154.                            etNamaKlien.setError("Required");
155.                            return;
156.                        } else if (alamatKlien.trim().isEmpty()) {
157.                            etAlamatKlien.setError("Required");
158.                            return;

```

```

159.             } else if (noTelpKlien.trim().isEmpty()) {
160.                 etNoTelpKlien.setError("Required");
161.                 return;
162.             } else if (noTelpKlien.length() > 13) {
163.                 etNoTelpKlien.setError("Nomor Telpon Tidak Boleh
Melebihi 13 digit");
164.             return;
165.         } else if (emailKlien.trim().isEmpty()) {
166.             etEmailKlien.setError("Required");
167.             return;
168.         }
169.         else if (!isValidEmail(emailKlien)){
170.             etEmailKlien.setError("Format Email Tidak
Didukung");
171.             return;
172.         }
173.         else if (catatanKlien.trim().isEmpty()) {
174.             etCatatanKlien.setError("Required");
175.             return;
176.         } else {
177.             TambahClient(namaKlien, alamatKlien,noTelpKlien,
emailKlien, catatanKlien);
178.         }
179.
180.     }
181. );
182. );
183. );
184. }
185.
186. private void TambahClient(String namaKlien, String alamatKlien, String
noTelpKlien, String emailKlien, String catatanKlien) {
187.     counterRef.addListenerForSingleValueEvent(new ValueEventListener() {
188.         @Override
189.         public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
190.             long currentId = dataSnapshot.exists() ?
dataSnapshot.getValue(Long.class) : 0;
191.             long newId = currentId + 1;
192.
193.             String clientId = String.valueOf(newId);
194.             ModelClient mc = new ModelClient(clientId, namaKlien,
alamatKlien, noTelpKlien, emailKlien, catatanKlien);
195.             reference.child(clientId).setValue(mc);
196.             Toast.makeText(Client.this, "Nama Klien Berhasil Disimpan",
Toast.LENGTH_SHORT).show();
197.             dialog.dismiss();
198.
199.             counterRef.setValue(newId);
200.             refreshClientList();
201.         }
202.
203.         @Override
204.         public void onCancelled(@NonNull DatabaseError error) {
205.
206.     }
207. );
208. }
209.
210. private void refreshClientList() {
211.     reference.addListenerForSingleValueEvent(new ValueEventListener() {
212.         @Override
213.         public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
214.             list_client.clear();
215.             for (DataSnapshot snapshot : dataSnapshot.getChildren()) {

```

```

216.             ModelClient mc = snapshot.getValue(ModelClient.class);
217.             list_client.add(mc);
218.         }
219.         adapterClient.notifyDataSetChanged();
220.         pd.dismiss();
221.     }
222.
223.     @Override
224.     public void onCancelled(@NonNull DatabaseError databaseError) {
225.         // Handle database errors (optional)
226.     }
227. };
228. }
229.
230. private boolean isValidEmail(String emailKlien) {
231.     String regex = "^[\\w-_\\.]+@[\\w-]+\\.[\\w-]{2,}$";
232.     return Pattern.matches(regex, emailKlien);
233. }
234.
235.
236. private void filterList(String query) {
237.     List<ModelClient> filteredList = new ArrayList<>();
238.     String queryInLower = query.toLowerCase();
239.     for (ModelClient mc : list_client) {
240.         if (mc.getNama().toLowerCase().contains(queryInLower)) {
241.             filteredList.add(mc);
242.         }
243.     }
244.     if (filteredList.isEmpty()) {
245.         Toast.makeText(this, "No Clients Found",
Toast.LENGTH_SHORT).show();
246.     } else {
247.         adapterClient.setFilter(filteredList);
248.     }
249. }
250.
251. @Override
252. protected void onStart() {
253.     super.onStart();
254. }
255.
256. @Override
257. protected void onStop() {
258.     super.onStop();
259. }
260.
261. @Override
262. public void onItemClickListener(ModelClient data, int position) {
263.     dialogClient = new Dialog(Client.this);
264.     dialogClient.setContentView(R.layout.dialog_tampil_client);
265.     dialogClient.getWindow().setLayout(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
266.     dialogClient.setCancelable(false);
267.     dialogClient.show();
268.
269.     TextView tvNamaKlien = dialogClient.findViewById(R.id.tv_nama_klien);
270.     tvNamaKlien.setText(data.getNama());
271.     TextView tvAlamatKlien =
dialogClient.findViewById(R.id.tv_alamat_klien);
272.     tvAlamatKlien.setText(data.getAlamat());
273.     TextView tvTelpKlien = dialogClient.findViewById(R.id.tv_telp_klien);
274.     tvTelpKlien.setText(data.getNo_telp());
275.     TextView tvEmailKlien = dialogClient.findViewById(R.id.tv_email_klien);
276.     tvEmailKlien.setText(data.getEmail());

```

```

277.         TextView tvCatatanKlien =
dialogClient.findViewById(R.id.tv_catatan_klien);
278.         tvCatatanKlien.setText(data.getCatatan());
279.         dialogClient.findViewById(R.id.iv_close).setOnClickListener(view ->
{dialogClient.dismiss();});
280.
281.         dialogClient.findViewById(R.id.btn_edit_klien).setOnClickListener(new
View.OnClickListener() {
282.             @Override
283.             public void onClick(View view) {
284.                 dialogEdit = new Dialog(Client.this);
285.                 dialogEdit.setContentView(R.layout.dialog_edit_client);
286.
dialogEdit.getWindow().setLayout(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
287.                 dialogEdit.setCancelable(false);
288.                 dialogEdit.show();
289.
290.                 EditText etNamaKlien =
dialogEdit.findViewById(R.id.et_nama_klien);
291.                 etNamaKlien.setText(data.getNama());
292.                 EditText etAlamatKlien =
dialogEdit.findViewById(R.id.et_alamat_klien);
293.                 etAlamatKlien.setText(data.getAlamat());
294.                 EditText etNoTelpKlien =
dialogEdit.findViewById(R.id.et_telp_klien);
295.                 etNoTelpKlien.setText(data.getNo_telp());
296.                 EditText etEmailKlien =
dialogEdit.findViewById(R.id.et_email_klien);
297.                 etEmailKlien.setText(data.getEmail());
298.                 EditText etCatatanKlien =
dialogEdit.findViewById(R.id.et_catatan_klien);
299.                 etCatatanKlien.setText(data.getCatatan());
300.                 dialogEdit.findViewById(R.id.iv_close).setOnClickListener(new
View.OnClickListener() {
301.                     @Override
302.                     public void onClick(View view) {
303.                         dialogEdit.dismiss();
304.                     }
305.                 });
306.
307.
dialogEdit.findViewById(R.id.btn_simpan_edit).setOnClickListener(new
View.OnClickListener() {
308.             @Override
309.             public void onClick(View view) {
310.                 namaKlien = etNamaKlien.getText().toString();
311.                 alamatKlien = etAlamatKlien.getText().toString();
312.                 noTelpKlien = etNoTelpKlien.getText().toString();
313.                 emailKlien = etEmailKlien.getText().toString();
314.                 catatanKlien = etCatatanKlien.getText().toString();
315.
316.                 if (namaKlien.trim().isEmpty()) {
317.                     etNamaKlien.setError("Required");
318.                 } else if (alamatKlien.trim().isEmpty()) {
319.                     etAlamatKlien.setError("Required");
320.                 } else if (noTelpKlien.trim().isEmpty()) {
321.                     etNoTelpKlien.setError("Required");
322.                 } else if (noTelpKlien.length() > 13) {
323.                     etNoTelpKlien.setError("Nomor Telpon Tidak Boleh
Melebihi 13 digit");
324.                 } else if (emailKlien.trim().isEmpty()) {
325.                     etEmailKlien.setError("Required");
326.                 } else if (!isValidEmail(emailKlien)) {

```

```

327.                               etEmailKlien.setError("Format Email Tidak
Didukung");
328.                           } else if (catatanKlien.trim().isEmpty()) {
329.                               etCatatanKlien.setError("Required");
330.                           } else {
331.                               updateClient(data.getId(), namaKlien, alamatKlien,
noTelpKlien, emailKlien, catatanKlien);
332.                               dialogEdit.dismiss();
333.                               dialogClient.dismiss();
334.                           }
335.                       }
336.                   });
337.               });
338.
339.
340.           });
341.       });
342.
343.       dialogClient.findViewById(R.id.btn_hapus_klien).setOnClickListener(new
View.OnClickListener() {
344.           @Override
345.           public void onClick(View view) {
346.               String clientId = data.getId();
347.               reference.child(clientId).removeValue();
348.               Toast.makeText(Client.this, "Data Klien Telah Dihapus",
Toast.LENGTH_SHORT).show();
349.               refreshClientList();
350.               dialogClient.dismiss();
351.           }
352.       });
353.   }
354.   private void updateClient(String id, String nama, String alamat, String
noTelp, String email, String catatan) {
355.       ModelClient mc = new ModelClient(id, nama, alamat, noTelp, email,
catatan);
356.       reference.child(id).setValue(mc);
357.       Toast.makeText(Client.this, "Data Klien Berhasil Diubah",
Toast.LENGTH_SHORT).show();
358.       refreshClientList();
359.   }
360. }

```

Kode Program Menu KelolaUser.java

```

1. package com.example.barangku.activity.user_activity;
2.
3. import android.content.DialogInterface;
4. import android.content.Intent;
5. import android.os.Bundle;
6. import android.view.View;
7. import android.widget.ImageView;
8. import android.widget.TextView;
9. import android.widget.Toast;
10.
11. import androidx.annotation.NonNull;
12. import androidx.appcompat.app.AlertDialog;
13. import androidx.appcompat.app.AppCompatActivity;
14. import androidx.recyclerview.widget.LinearLayoutManager;
15. import androidx.recyclerview.widget.RecyclerView;
16.

```

```

17. import com.example.barangku.R;
18.
19. import com.example.barangku.activity.adapter.AdapterKaryawan;
20. import com.example.barangku.activity.model.ModelUser;
21. import com.google.android.gms.tasks.OnCompleteListener;
22. import com.google.android.gms.tasks.Task;
23. import com.google.firebaseio.database.DataSnapshot;
24. import com.google.firebaseio.database.DatabaseError;
25. import com.google.firebaseio.database.DatabaseReference;
26. import com.google.firebaseio.database.FirebaseDatabase;
27. import com.google.firebaseio.database.ValueEventListener;
28.
29. import java.util.ArrayList;
30. import java.util.List;
31.
32. public class KelolaUser extends AppCompatActivity {
33.     private Toolbar tvToolbar;
34.     private ImageView ivBack;
35.     private RecyclerView rvKaryawan;
36.     private DatabaseReference karyawanRef =
        FirebaseDatabase.getInstance().getReference("User");
37.     private List<ModelUser> listKaryawan;
38.     AdapterKaryawan adapter;
39.     @Override
40.     protected void onCreate(Bundle savedInstanceState) {
41.         super.onCreate(savedInstanceState);
42.         setContentView(R.layout.activity_karyawan);
43.         tvToolbar = findViewById(R.id.tv_judul);
44.         tvToolbar.setText("Kelola User");
45.
46.         ivBack = findViewById(R.id.iv_back);
47.         ivBack.setOnClickListener(new View.OnClickListener() {
48.             @Override
49.             public void onClick(View view) {
50.                 Intent intent = new Intent(KelolaUser.this, Admin.class);
51.                 startActivity(intent);
52.             }
53.         });
54.
55.         rvKaryawan = findViewById(R.id.rv_karyawan);
56.         rvKaryawan.setHasFixedSize(true);
57.         rvKaryawan.setLayoutManager(new LinearLayoutManager(this));
58.
59.         listKaryawan = new ArrayList<>();
60.         adapter = new AdapterKaryawan(this, listKaryawan);
61.         rvKaryawan.setAdapter(adapter);
62.
63.         karyawanRef.orderByChild("jabatan").addValueEventListener(new
ValueEventListener() {
64.             @Override
65.             public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
66.                 listKaryawan.clear();
67.                 for (DataSnapshot postSnapshot : dataSnapshot.getChildren()) {
68.                     ModelUser user = postSnapshot.getValue(ModelUser.class);
69.                     listKaryawan.add(user);
70.                 }
71.                 adapter.notifyDataSetChanged();
72.             }
73.
74.             @Override
75.             public void onCancelled(@NonNull DatabaseError databaseError) {
76.                 Toast.makeText(KelolaUser.this, "Failed to load employees",
Toast.LENGTH_SHORT).show();
77.             }

```

```

78.        });
79.        adapter.setOnItemDeleteListener(new
AdapterKaryawan.OnItemDeleteListener() {
80.            @Override
81.            public void onItemDelete(int position) {
82.                showDeleteConfirmationDialog(position);
83.            }
84.        });
85.    }
86.
87.    private void showDeleteConfirmationDialog(int position) {
88.        new AlertDialog.Builder(this)
89.            .setTitle("Konfirmasi Hapus")
90.            .setMessage("Apakah Anda yakin ingin menghapus data User ini?")
91.            .setPositiveButton("Hapus", new
DialogInterface.OnClickListener() {
92.                @Override
93.                public void onClick(DialogInterface dialogInterface, int i)
{
94.                    deleteKaryawan(position);
95.                }
96.            })
97.            .setNegativeButton("Batal", null)
98.            .show();
99.    }
100.
101.    private void deleteKaryawan(int position) {
102.        ModelUser user = listKaryawan.get(position);
103.        DatabaseReference userRef = karyawanRef.child(user.getId());
104.        userRef.removeValue().addOnCompleteListener(new
OnCompleteListener<Void>() {
105.            @Override
106.            public void onComplete(@NonNull Task<Void> task) {
107.                if (task.isSuccessful()) {
108.                    Toast.makeText(KelolaUser.this, "User berhasil dihapus",
Toast.LENGTH_SHORT).show();
109.                } else {
110.                    Toast.makeText(KelolaUser.this, "Gagal menghapus data
user", Toast.LENGTH_SHORT).show();
111.                }
112.            }
113.        });
114.    }
115.}
116.

```

Kode Program Menu Pengajuan.java

```

1. package com.example.barangku.activity.user_activity;
2.
3. import androidx.appcompat.app.AppCompatActivity;
4. import androidx.fragment.app.Fragment;
5. import androidx.fragment.app.FragmentManager;
6. import androidx.fragment.app.FragmentTransaction;
7.
8. import android.os.Bundle;
9.
10. import com.example.barangku.R;
11. import com.google.android.material.bottomnavigation.BottomNavigationView;
12.

```

```

13. public class Pengajuan extends AppCompatActivity {
14.     @Override
15.     protected void onCreate(Bundle savedInstanceState) {
16.         super.onCreate(savedInstanceState);
17.         setContentView(R.layout.activity_pengajuan);
18.         gantiFragment(new PengajuanBarangMasukFragment());
19.         BottomNavigationView navigationView = findViewById(R.id.navigation_bar);
20.         navigationView.setOnItemSelectedListener(item -> {
21.             if (item.getItemId() == R.id.nav_barang_masuk){
22.                 gantiFragment(new PengajuanBarangMasukFragment());
23.             }else if(item.getItemId() == R.id.nav_barang_keluar){
24.                 gantiFragment(new PengajuanBarangKeluarFragment());
25.             }
26.             return true;
27.         });
28.     }
29.     private void gantiFragment(Fragment fragment){
30.         FragmentManager fragmentManager = getSupportFragmentManager();
31.         FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
32.         fragmentTransaction.replace(R.id.frameLayout, fragment);
33.         fragmentTransaction.commit();
34.     }
35. }
36.

```

Kode Program Halaman PengajuanBarangKeluarFragment.java

```

1. package com.example.barangku.activity.user_activity;
2.
3. import android.app.AlertDialog;
4. import android.content.DialogInterface;
5. import android.content.Intent;
6. import android.os.Bundle;
7. import androidx.annotation.NonNull;
8. import androidx.annotation.Nullable;
9. import androidx.fragment.app.Fragment;
10. import androidx.recyclerview.widget.LinearLayoutManager;
11. import androidx.recyclerview.widget.RecyclerView;
12. import android.view.LayoutInflater;
13. import android.view.View;
14. import android.view.ViewGroup;
15. import android.widget.ImageView;
16. import android.widget.TextView;
17. import android.widget.Toast;
18. import com.example.barangku.R;
19. import com.example.barangku.activity.adapter.AdapterPengajuanBarangKeluar;
20. import com.example.barangku.activity.model.ModelPengajuanBarangKeluar;
21. import com.example.barangku.activity.model.ModelBarangKeluar;
22. import com.example.barangku.activity.model.ModelStock;
23. import
com.example.barangku.activity.user_activity.utils.ItemClickPengajuanBarangKeluar;
24. import com.google.firebase.database.DataSnapshot;
25. import com.google.firebase.database.DatabaseError;
26. import com.google.firebase.database.DatabaseReference;
27. import com.google.firebase.database.FirebaseDatabase;
28. import com.google.firebase.database.ValueEventListener;
29.
30. import java.text.ParseException;
31. import java.text.SimpleDateFormat;

```

```

32. import java.util.ArrayList;
33. import java.util.Date;
34. import java.util.List;
35. import java.util.Locale;
36.
37. public class PengajuanBarangKeluarFragment extends Fragment implements
ItemClickPengajuanBarangKeluar {
38.     private TextView tvToolbar;
39.     private ImageView ivBack;
40.
41.     private RecyclerView rvPengajuanBarangKeluar;
42.     private AdapterPengajuanBarangKeluar adapter;
43.     private List<ModelPengajuanBarangKeluar> listPengajuan = new ArrayList<>();
44.
45.     private DatabaseReference pengajuanRef =
FirebaseDatabase.getInstance().getReference("PengajuanBarangKeluar");
46.     private DatabaseReference riwayatPengajuanRef =
FirebaseDatabase.getInstance().getReference("RiwayatPengajuanBarangKeluar");
47.     DatabaseReference barangKeluarRef =
FirebaseDatabase.getInstance().getReference("BarangKeluar");
48.     DatabaseReference stokBarangRef =
FirebaseDatabase.getInstance().getReference("StockBarang");
49.
50.     public PengajuanBarangKeluarFragment() {
51.         // Required empty public constructor
52.     }
53.
54.     @Override
55.     public void onCreate(Bundle savedInstanceState) {
56.         super.onCreate(savedInstanceState);
57.     }
58.
59.     @Override
60.     public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {
61.         // Inflate the layout for this fragment
62.         return inflater.inflate(R.layout.fragment_pengajuan_barang_keluar,
container, false);
63.     }
64.
65.     @Override
66.     public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
67.         super.onViewCreated(view, savedInstanceState);
68.
69.         tvToolbar = view.findViewById(R.id.tv_judul);
70.         tvToolbar.setText("Pengajuan");
71.
72.         ivBack = view.findViewById(R.id.iv_back);
73.         ivBack.setOnClickListener(new View.OnClickListener() {
74.             @Override
75.             public void onClick(View view) {
76.                 Intent intent = new Intent(getActivity(),
MainActivity.class );
77.                 startActivity(intent);
78.             }
79.         });
80.         rvPengajuanBarangKeluar = view.findViewById(R.id.rv_pengajuan);
81.
82.         adapter = new AdapterPengajuanBarangKeluar(listPengajuan,
getView().getContext());
83.         adapter.setItemClickPengajuanBarangKeluar(this);
84.         LinearLayoutManager lmPengajuanBk = new
LinearLayoutManager(getView().getContext(), LinearLayoutManager.VERTICAL, false);

```

```

85.         rvPengajuanBarangKeluar.setLayoutManager(lmPengajuanBk);
86.         rvPengajuanBarangKeluar.setAdapter(adapter);
87.
88.         loadPengajuanData();
89.     }
90.
91.     private void loadPengajuanData() {
92.         pengajuanRef.addValueEventListener(new ValueEventListener() {
93.             @Override
94.             public void onDataChange(@NonNull DataSnapshot snapshot) {
95.                 listPengajuan.clear();
96.
97.                 for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
98.                     String id = dataSnapshot.getKey();
99.                     String namaBarang =
dataSnapshot.child("namaBarang").getValue(String.class);
100.                    String satuan =
dataSnapshot.child("satuan").getValue(String.class);
101.                    String keterangan =
dataSnapshot.child("keterangan").getValue(String.class);
102.                    String tanggalKeluar =
dataSnapshot.child("tanggalKeluar").getValue(String.class);
103.                    String formattedTanggalKeluar =
formatTanggal(tanggalKeluar);
104.                    String namaKlien =
dataSnapshot.child("namaKlien").getValue(String.class);
105.                    String alamatKlien =
dataSnapshot.child("alamatKlien").getValue(String.class);
106.                    String status =
dataSnapshot.child("status").getValue(String.class);
107.                    Long jumlahBarang =
dataSnapshot.child("jumlahBarang").getValue(Long.class);
108.
109.                    ModelPengajuanBarangKeluar mpbk = new
ModelPengajuanBarangKeluar(id, namaBarang, satuan, keterangan,
formattedTanggalKeluar, namaKlien, alamatKlien, status, jumlahBarang);
110.
111.                    if (mpbk != null) {
112.                        listPengajuan.add(mpbk);
113.                    }
114.                }
115.            adapter.notifyDataSetChanged();
116.        }
117.
118.        @Override
119.        public void onCancelled(@NonNull DatabaseError error) {
120.            Toast.makeText(requireActivity(), "Gagal memuat data
pengajuan", Toast.LENGTH_SHORT).show();
121.        }
122.    });
123. }
124. private String formatTanggal(String tanggal) {
125.     SimpleDateFormat sdfSource = new SimpleDateFormat("yyyy-MM-dd", new
Locale("id", "ID"));
126.     SimpleDateFormat sdfDestination = new SimpleDateFormat("EEEE, dd MMMM
yyyy", new Locale("id", "ID"));
127.     try {
128.         Date date = sdfSource.parse(tanggal);
129.         return sdfDestination.format(date);
130.     } catch (ParseException e) {
131.         e.printStackTrace();
132.         return tanggal;
133.     }
134. }

```

```

135.     private void approvePengajuan(ModelPengajuanBarangKeluar pengajuan, int
position) {
136.         String barangKeluarId = barangKeluarRef.push().getKey();
137.         ModelBarangKeluar barangKeluarBaru = new
ModelBarangKeluar(barangKeluarId, pengajuan.getNamaBarang(), pengajuan.getSatuan(),
pengajuan.getKeterangan(), pengajuan.getTanggalKeluar(), pengajuan.getNamaKlien(),
pengajuan.getAlamatKlien(), pengajuan.getJumlahBarang());
138.
barangKeluarRef.child(barangKeluarId).setValue(barangKeluarBaru).addOnCompleteListener(addTask -> {
139.             if (addTask.isSuccessful()) {
140.                 perbaruiStokBarang(pengajuan);
141.                 pengajuan.setStatus("Diterima");
142.                 simpanRiwayatPengajuan(pengajuan);
143.                 hapusPengajuanSetelahDisetujui(pengajuan, position);
144.             } else {
145.                 Toast.makeText(requireActivity(), "Gagal menambahkan data baru
ke BarangKeluar", Toast.LENGTH_SHORT).show();
146.             }
147.         });
148.     }
149.
150.     private void rejectPengajuan(ModelPengajuanBarangKeluar pengajuan) {
151.         pengajuan.setStatus("Ditolak");
152.         simpanRiwayatPengajuan(pengajuan);
153.
pengajuanRef.child(pengajuan.getId()).removeValue().addOnCompleteListener(removeTask -> {
154.             if (removeTask.isSuccessful()) {
155.                 Toast.makeText(requireActivity(), "Pengajuan berhasil ditolak
dan disimpan dalam riwayat", Toast.LENGTH_SHORT).show();
156.             } else {
157.                 Toast.makeText(requireActivity(), "Gagal menghapus pengajuan",
Toast.LENGTH_SHORT).show();
158.             }
159.         });
160.     }
161.
162.     private void perbaruiStokBarang(ModelPengajuanBarangKeluar pengajuan) {
163.
stokBarangRef.orderByChild("namaBarang").equalTo(pengajuan.getNamaBarang()).addListenerForSingleValueEvent(new ValueEventListener() {
164.         @Override
165.         public void onDataChange(@NonNull DataSnapshot snapshot) {
166.             boolean stokDitemukan = false;
167.             for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
168.                 ModelStock stockItem =
dataSnapshot.getValue(ModelStock.class);
169.                 if (stockItem != null) {
170.                     stokDitemukan = true;
171.                     long currentStok =
Long.parseLong(stockItem.getJumlahBarang());
172.                     long updatedStok = currentStok -
pengajuan.getJumlahBarang();
173.                     stockItem.setJumlahBarang(String.valueOf(updatedStok));
174.
175.                     dataSnapshot.getRef().setValue(stockItem).addOnCompleteListener(updateTask -> {
176.                         if (updateTask.isSuccessful()) {
177.                             Toast.makeText(requireActivity(), "Pengajuan
disetujui dan Jumlah stok diperbarui", Toast.LENGTH_SHORT).show();
178.                         } else {
179.                             Toast.makeText(requireActivity(), "Gagal
memperbarui stok barang", Toast.LENGTH_SHORT).show();

```

```

180.             }
181.         });
182.     }
183. }
184. if (!stokDitemukan) {
185.     Toast.makeText(requireActivity(), "Stok barang tidak
ditemukan", Toast.LENGTH_SHORT).show();
186. }
187. }
188. }
189.
190. @Override
191. public void onCancelled(@NonNull DatabaseError error) {
192.     Toast.makeText(requireActivity(), "Gagal memperbarui stok
barang", Toast.LENGTH_SHORT).show();
193. }
194. });
195. }
196.
197. private void simpanRiwayatPengajuan(ModelPengajuanBarangKeluar pengajuan) {
198.     String riwayatId = riwayatPengajuanRef.push().getKey();
199.
riwayatPengajuanRef.child(riwayatId).setValue(pengajuan).addOnCompleteListener(task
-> {
200.     if (task.isSuccessful()) {
201.         Toast.makeText(requireActivity(), "Pengajuan " +
pengajuan.getStatus() + " dan disimpan dalam riwayat", Toast.LENGTH_SHORT).show();
202.     } else {
203.         Toast.makeText(requireActivity(), "Gagal menyimpan pengajuan ke
dalam riwayat", Toast.LENGTH_SHORT).show();
204.     }
205. });
206. }
207.
208. private void hapusPengajuanSetelahDisetujui(ModelPengajuanBarangKeluar
pengajuan, int position) {
209.
pengajuanRef.child(pengajuan.getId()).removeValue().addOnCompleteListener(removeTask
-> {
210.     if (!removeTask.isSuccessful()) {
211.         Toast.makeText(requireActivity(), "Gagal menghapus pengajuan
setelah disetujui", Toast.LENGTH_SHORT).show();
212.     } else {
213.         if (position >= 0 && position < listPengajuan.size()) {
214.             listPengajuan.remove(position);
215.             adapter.notifyItemRemoved(position);
216.         }
217.     }
218. });
219. }
220.
221. @Override
222. public void onItemClickListener(ModelPengajuanBarangKeluar data, int
position) {
223.     if (position >= 0 && position < listPengajuan.size()) {
224.         AlertDialog.Builder builder = new
AlertDialog.Builder(requireActivity());
225.         builder.setTitle("Pengajuan Barang Keluar")
226.             .setMessage("Apakah Anda ingin menerima atau menolak
pengajuan ini?")
227.             .setPositiveButton("Terima", new
DialogInterface.OnClickListener() {
228.                 @Override

```

```

229.             public void onClick(DialogInterface dialog, int which)
230. {
231.         approvePengajuan(data, position);
232.     }
233. }
234.     .setNegativeButton("Tolak", new
DialogInterface.OnClickListener() {
235.         @Override
236.         public void onClick(DialogInterface dialog, int which)
237. {
238.         rejectPengajuan(data);
239.     }
240.     .setNeutralButton("Batal", null)
241.     .show();
242. } else {
243.     Toast.makeText(requireActivity(), "Pengajuan tidak valid",
Toast.LENGTH_SHORT).show();
244. }
245. }
246.

```

Kode Program Halaman PengajuanBarangMasukFragment.java

```

1. package com.example.barangku.activity.user_activity;
2.
3. import android.app.AlertDialog;
4. import android.content.DialogInterface;
5. import android.content.Intent;
6. import android.os.Bundle;
7. import androidx.annotation.NonNull;
8. import androidx.annotation.Nullable;
9. import androidx.fragment.app.Fragment;
10. import androidx.recyclerview.widget.LinearLayoutManager;
11. import androidx.recyclerview.widget.RecyclerView;
12. import android.view.LayoutInflater;
13. import android.view.View;
14. import android.view.ViewGroup;
15. import android.widget.ImageView;
16. import android.widget.TextView;
17. import android.widget.Toast;
18. import com.example.barangku.R;
19. import com.example.barangku.activity.adapter.AdapterPengajuanBarangMasuk;
20. import com.example.barangku.activity.model.ModelBarangMasuk;
21. import com.example.barangku.activity.model.ModelPengajuanBarangMasuk;
22. import com.example.barangku.activity.model.ModelStock;
23. import com.example.barangku.activity.user_activity.laporan.Laporan;
24. import
com.example.barangku.activity.user_activity.utils.ItemClickPengajuanBarangMasuk;
25. import com.google.firebaseio.database.DataSnapshot;
26. import com.google.firebaseio.database.DatabaseError;
27. import com.google.firebaseio.database.DatabaseReference;
28. import com.google.firebaseio.database.FirebaseDatabase;
29. import com.google.firebaseio.database.ValueEventListener;
30.
31. import java.text.ParseException;
32. import java.text.SimpleDateFormat;
33. import java.util.ArrayList;
34. import java.util.Date;

```

```

35. import java.util.List;
36. import java.util.Locale;
37.
38. public class PengajuanBarangMasukFragment extends Fragment implements
ItemClickPengajuanBarangMasuk {
39.     private TextView tvToolbar;
40.     private ImageView ivBack;
41.     private RecyclerView rvPengajuanBarangMasuk;
42.     private AdapterPengajuanBarangMasuk adapter;
43.     private List<ModelPengajuanBarangMasuk> pengajuanList = new ArrayList<>();
44.     private DatabaseReference pengajuanRef =
        FirebaseDatabase.getInstance().getReference("PengajuanBarangMasuk");
45.     private DatabaseReference riwayatPengajuanRef =
        FirebaseDatabase.getInstance().getReference("RiwayatPengajuanBarangMasuk");
46.     DatabaseReference barangMasukRef =
        FirebaseDatabase.getInstance().getReference("BarangMasuk");
47.     DatabaseReference stokBarangRef =
        FirebaseDatabase.getInstance().getReference("StockBarang");
48.
49.     public PengajuanBarangMasukFragment() {
50.         // Required empty public constructor
51.     }
52.
53.     @Override
54.     public void onCreate(Bundle savedInstanceState) {
55.         super.onCreate(savedInstanceState);
56.     }
57.
58.     @Override
59.     public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {
60.         // Inflate the layout for this fragment
61.         return inflater.inflate(R.layout.fragment_pengajuan_barang_masuk,
container, false);
62.     }
63.
64.     @Override
65.     public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
66.         super.onViewCreated(view, savedInstanceState);
67.
68.         tvToolbar = view.findViewById(R.id.tv_judul);
69.         tvToolbar.setText("Pengajuan");
70.
71.         ivBack = view.findViewById(R.id.iv_back);
72.         ivBack.setOnClickListener(new View.OnClickListener() {
73.             @Override
74.             public void onClick(View view) {
75.                 Intent intent = new Intent(getActivity(),
MainActivity.class) ;
76.                 startActivity(intent);
77.             }
78.         });
79.
80.         rvPengajuanBarangMasuk = view.findViewById(R.id.rv_pengajuan);
81.
82.         adapter = new AdapterPengajuanBarangMasuk(pengajuanList,
getView().getContext());
83.         adapter.setItemClickPengajuanBarangMasuk(this);
84.         LinearLayoutManager lmPengajuanBm = new
LinearLayoutManager(getView().getContext(), LinearLayoutManager.VERTICAL, false);
85.         rvPengajuanBarangMasuk.setLayoutManager(lmPengajuanBm);
86.         rvPengajuanBarangMasuk.setAdapter(adapter);
87.

```

```

88.         loadPengajuanData();
89.     }
90.
91.     private void loadPengajuanData() {
92.         pengajuanRef.addValueEventListener(new ValueEventListener() {
93.             @Override
94.             public void onDataChange(@NonNull DataSnapshot snapshot) {
95.                 pengajuanList.clear();
96.
97.                 for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
98.                     String id = dataSnapshot.getKey();
99.                     String namaBarang =
dataSnapshot.child("namaBarang").getValue(String.class);
100.                    String satuan =
dataSnapshot.child("satuan").getValue(String.class);
101.                    String keterangan =
dataSnapshot.child("keterangan").getValue(String.class);
102.                    String tanggalMasuk =
dataSnapshot.child("tanggalMasuk").getValue(String.class);
103.                    String formattedTanggalMasuk = formatTanggal(tanggalMasuk);
104.                    String status =
dataSnapshot.child("status").getValue(String.class);
105.                    int jumlahBarang =
dataSnapshot.child("jumlahBarang").getValue(Integer.class);
106.
107.                    ModelPengajuanBarangMasuk mpbm = new
ModelPengajuanBarangMasuk(id, namaBarang, satuan, keterangan, formattedTanggalMasuk,
status, jumlahBarang);
108.
109.                    if (mpbm != null) {
110.                        pengajuanList.add(mpbm);
111.                    }
112.                }
113.            adapter.notifyDataSetChanged();
114.        }
115.
116.        @Override
117.        public void onCancelled(@NonNull DatabaseError error) {
118.            Toast.makeText(requireActivity(), "Gagal memuat data
pengajuan", Toast.LENGTH_SHORT).show();
119.        }
120.    });
121. }
122.     private String formatTanggal(String tanggal) {
123.         SimpleDateFormat sdfSource = new SimpleDateFormat("yyyy-MM-dd", new
Locale("id", "ID"));
124.         SimpleDateFormat sdfDestination = new SimpleDateFormat("EEEE, dd MMMM
yyyy", new Locale("id", "ID"));
125.         try {
126.             Date date = sdfSource.parse(tanggal);
127.             return sdfDestination.format(date);
128.         } catch (ParseException e) {
129.             e.printStackTrace();
130.             return tanggal;
131.         }
132.     }
133.     private void approvePengajuan(ModelPengajuanBarangMasuk pengajuan, int
position) {
134.         String barangMasukId = barangMasukRef.push().getKey();
135.         ModelBarangMasuk barangMasukBaru = new ModelBarangMasuk(barangMasukId,
pengajuan.getNamaBarang(), pengajuan.getSatuan(), pengajuan.getKeterangan(),
pengajuan.getTanggalMasuk(), pengajuan.getJumlahBarang());

```

```

136.
barangMasukRef.child(barangMasukId).setValue(barangMasukBaru).addOnCompleteListener(
addTask -> {
137.        if (addTask.isSuccessful()) {
138.            perbaruiStokBarang(pengajuan);
139.            pengajuan.setStatus("Diterima");
140.            simpanRiwayatPengajuan(pengajuan);
141.            hapusPengajuanSetelahDisetujui(pengajuan, position);
142.        } else {
143.            Toast.makeText(requireActivity(), "Gagal menambahkan data baru
ke BarangMasuk", Toast.LENGTH_SHORT).show();
144.        }
145.    });
146. }
147. private void rejectPengajuan(ModelPengajuanBarangMasuk pengajuan) {
148.     pengajuan.setStatus("Ditolak");
149.     simpanRiwayatPengajuan(pengajuan);
150.
pengajuanRef.child(pengajuan.getId()).removeValue().addOnCompleteListener(removeTask
-> {
151.        if (removeTask.isSuccessful()) {
152.            Toast.makeText(requireActivity(), "Pengajuan berhasil ditolak
dan disimpan dalam riwayat", Toast.LENGTH_SHORT).show();
153.        } else {
154.            Toast.makeText(requireActivity(), "Gagal menghapus pengajuan",
Toast.LENGTH_SHORT).show();
155.        }
156.    });
157. }
158.
159. private void perbaruiStokBarang(ModelPengajuanBarangMasuk pengajuan) {
160.
stokBarangRef.orderByChild("namaBarang").equalTo(pengajuan.getNamaBarang()).addListe
nerForSingleValueEvent(new ValueEventListener() {
161.        @Override
162.        public void onDataChange(@NonNull DataSnapshot snapshot) {
163.            boolean stokDitemukan = false;
164.            for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
165.                ModelStock stockItem =
dataSnapshot.getValue(ModelStock.class);
166.                if (stockItem != null) {
167.                    stokDitemukan = true;
168.                    int currentStok =
Integer.parseInt(stockItem.getJumlahBarang());
169.                    int updatedStok = currentStok +
pengajuan.getJumlahBarang();
170.                    stockItem.setJumlahBarang(String.valueOf(updatedStok));
171.
172.
dataSnapshot.getRef().setValue(stockItem).addOnCompleteListener(updateTask -> {
173.                if (updateTask.isSuccessful()) {
174.                    Toast.makeText(requireActivity(), "Pengajuan
disetujui dan Jumlah stok diperbarui", Toast.LENGTH_SHORT).show();
175.                } else {
176.                    Toast.makeText(requireActivity(), "Gagal
memperbarui stok barang", Toast.LENGTH_SHORT).show();
177.                }
178.            });
179.            break;
180.        }
181.    }
182.    if (!stokDitemukan) {
183.        Toast.makeText(requireActivity(), "Stok barang tidak
ditemukan", Toast.LENGTH_SHORT).show();

```

```

184.         }
185.     }
186.
187.     @Override
188.     public void onCancelled(@NonNull DatabaseError error) {
189.         Toast.makeText(requireActivity(), "Gagal memperbarui stok
barang", Toast.LENGTH_SHORT).show();
190.     }
191. });
192. }
193.
194. private void simpanRiwayatPengajuan(ModelPengajuanBarangMasuk pengajuan) {
195.     String riwayatId = riwayatPengajuanRef.push().getKey();
196.
riwayatPengajuanRef.child(riwayatId).setValue(pengajuan).addOnCompleteListener(task
-> {
197.     if (task.isSuccessful()) {
198.         Toast.makeText(requireActivity(), "Pengajuan " +
pengajuan.getStatus() + " dan disimpan dalam riwayat", Toast.LENGTH_SHORT).show();
199.     } else {
200.         Toast.makeText(requireActivity(), "Gagal menyimpan pengajuan ke
dalam riwayat", Toast.LENGTH_SHORT).show();
201.     }
202. });
203. }
204.
205. private void hapusPengajuanSetelahDisetujui(ModelPengajuanBarangMasuk
pengajuan, int position) {
206.
pengajuanRef.child(pengajuan.getId()).removeValue().addOnCompleteListener(removeTask
-> {
207.     if (!removeTask.isSuccessful()) {
208.         Toast.makeText(requireActivity(), "Gagal menghapus pengajuan
setelah disetujui", Toast.LENGTH_SHORT).show();
209.     } else {
210.         if (position >= 0 && position < pengajuanList.size()) {
211.             pengajuanList.remove(position);
212.             adapter.notifyItemRemoved(position);
213.         }
214.     }
215. });
216. }
217.
218. @Override
219. public void onItemClickListener(ModelPengajuanBarangMasuk data, int
position) {
220.     if (position >= 0 && position < pengajuanList.size()) {
221.         AlertDialog.Builder builder = new
AlertDialog.Builder(requireActivity());
222.         builder.setTitle("Pengajuan Barang Masuk")
223.             .setMessage("Apakah Anda ingin menerima atau menolak
pengajuan ini?")
224.             .setPositiveButton("Terima", new
DialogInterface.OnClickListener() {
225.                 @Override
226.                 public void onClick(DialogInterface dialog, int which)
{
227.                     approvePengajuan(data, position);
228.                 }
229.             })
230.             .setNegativeButton("Tolak", new
DialogInterface.OnClickListener() {
231.                 @Override

```

```

232.             public void onClick(DialogInterface dialog, int which)
233. {
234.         rejectPengajuan(data);
235.     }
236. }
237. .setNeutralButton("Batal", null)
238. .show();
239. } else {
240.     Toast.makeText(requireActivity(), "Pengajuan tidak valid",
241.     Toast.LENGTH_SHORT).show();
242. }
243.

```

Kode Program Halaman Register.java

```

1. package com.example.barangku.activity.user_activity;
2.
3. import static android.content.ContentValues.TAG;
4. import androidx.annotation.NonNull;
5. import androidx.appcompat.app.AppCompatActivity;
6.
7. import android.content.Intent;
8. import android.os.Bundle;
9. import android.text.TextUtils;
10. import android.util.Log;
11. import android.view.View;
12. import android.widget.AdapterView;
13. import android.widget.ArrayAdapter;
14. import android.widget.Button;
15. import android.widget.ImageView;
16. import android.widget.Spinner;
17. import android.widget.TextView;
18. import android.widget.Toast;
19.
20. import com.example.barangku.R;
21. import com.example.barangku.activity.model.ModelUser;
22. import com.google.android.gms.tasks.OnCompleteListener;
23. import com.google.android.gms.tasks.OnFailureListener;
24. import com.google.android.gms.tasks.Task;
25. import com.google.android.material.textfield.TextInputEditText;
26. import com.google.firebase.auth.AuthResult;
27. import com.google.firebase.auth.FirebaseAuth;
28. import com.google.firebase.auth.FirebaseUser;
29. import com.google.firebaseio.database.DatabaseReference;
30. import com.google.firebaseio.database.FirebaseDatabase;
31. import com.google.firebaseio.messaging.FirebaseMessaging;
32.
33. public class Register extends AppCompatActivity {
34.     private TextInputEditText
etregister_nama,etregister_email,etregister_password;
35.     private Button btn_register;
36.     private TextView tv_login,tv_toolbar;
37.     private Spinner spJabatan;
38.     private String email, password;
39.     private String nama,jabatan, token;
40.     private FirebaseAuth mAuth;
41.     private ImageView ivback;
42.     DatabaseReference reference =
FirebaseDatabase.getInstance().getReference("User");

```

```

43.
44.     @Override
45.     public void onStart() {
46.         super.onStart();
47.         FirebaseAuth currentUser = mAuth.getCurrentUser();
48.         if(currentUser != null){
49.             }
50.         }
51.
52.     @Override
53.     protected void onCreate(Bundle savedInstanceState) {
54.         super.onCreate(savedInstanceState);
55.         setContentView(R.layout.activity_register);
56.
57.         mAuth = FirebaseAuth.getInstance();
58.
59.         etregister_nama = findViewById(R.id.et_nama_register);
60.         etregister_email = findViewById(R.id.et_email_register);
61.         etregister_password = findViewById(R.id.et_password_register);
62.
63.         tv_login = findViewById(R.id.tv_login);
64.         btn_register = findViewById(R.id.btn_register_button);
65.         ivback = findViewById(R.id.iv_back);
66.
67.         spJabatan = findViewById(R.id.sp_jabatan);
68.
69.         ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
70.             Register.this,
71.             R.array.Jabatan,
72.             android.R.layout.simple_spinner_item
73.         );
74.
75.
76.         adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
77.         spJabatan.setAdapter(adapter);
78.
79.         spJabatan.setOnItemSelectedListener(new
80.             AdapterView.OnItemSelectedListener() {
81.                 @Override
82.                 public void onItemSelected(AdapterView<?> adapterView, View view,
83.                     int pos, long l) {
84.                         jabatan = adapterView.getItemAtPosition(pos).toString();
85.                     }
86.
87.                 }
88.             );
89.
90.         tv_toolbar = findViewById(R.id.tv_judul);
91.         tv_toolbar.setText("Tambah User");
92.
93.         // Set up login button click listener
94.         tv_login.setOnClickListener(new View.OnClickListener() {
95.             @Override
96.             public void onClick(View v) {
97.                 Intent intent = new Intent(Register.this, Login.class);
98.                 startActivity(intent);
99.                 // Close Register activity when login button is clicked
100.                 finish();
101.             }
102.         });
103.

```

```

104.         ivback.setOnClickListener(new View.OnClickListener() {
105.             @Override
106.             public void onClick(View v) {
107.                 Intent intent = new Intent(Register.this, Login.class);
108.                 startActivity(intent);
109.             }
110.         });
111.
112.         // Set up register button click listener
113.         btn_register.setOnClickListener(new View.OnClickListener() {
114.             @Override
115.             public void onClick(View v) {
116.                 nama = etregister_nama.getText().toString();
117.                 email = etregister_email.getText().toString();
118.                 password = etregister_password.getText().toString();
119.
120.                 if (nama.trim().isEmpty()) {
121.                     etregister_nama.setError("Silakan Masukan Nama");
122.                     return;
123.                 }
124.                 if (email.trim().isEmpty()) {
125.                     etregister_email.setError("Silakan Masukan Email");
126.                     return;
127.                 }
128.                 if (password.trim().isEmpty()) {
129.                     etregister_password.setError("Silakan Masukan Password");
130.                     return;
131.                 }
132.
133.                 // Check if email and password fields are not empty
134.                 if (!TextUtils.isEmpty(email) && !TextUtils.isEmpty(password))
135.                 {
136.                     // Create a new user account with email and password
137.                     mAuth.createUserWithEmailAndPassword(email,
138.                         password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
139.                             @Override
140.                             public void onComplete(@NonNull Task<AuthResult> task)
141.                             {
142.                                 if (task.isSuccessful()) {
143.                                     FirebaseUser user = mAuth.getCurrentUser();
144.
145.                                     FirebaseMessaging.getInstance().getToken().addOnCompleteListener(new
146.                                         OnCompleteListener<String>() {
147.                                             @Override
148.                                             public void onComplete(@NonNull
149.                                                 Task<String> task) {
150.                                                 token = task.getResult();
151.                                                 String userId = user.getUid();
152.                                                 ModelUser mu = new ModelUser(userId,
153.                         email, nama, jabatan, token);
154.                                                 reference.child(userId).setValue(mu);
155.                                                 Toast.makeText(Register.this, "Berhasil
156. Daftar Akun", Toast.LENGTH_SHORT).show();
157.                                         }
158.                                     }).addOnFailureListener(new OnFailureListener()
159.                                     {
160.                                         @Override
161.                                         public void onFailure(@NonNull Exception e)
162.                                         {
163.                                             Toast.makeText(Register.this, "Gagal
164. Mendapatkan Token FCM: " + e.getMessage(), Toast.LENGTH_SHORT).show();
165.                                         }
166.                                     }
167.                                 }
168.                             }
169.                         }
170.                     }
171.                 }
172.             }
173.         }
174.     }
175. 
```

```

157.                                });
158.                            } else {
159.                                // Tambahkan lebih banyak detail tentang
kesalahan yang terjadi
160.                                String errorMessage = task.getException() !=
null ? task.getException().getMessage() : "Unknown error";
161.                                Toast.makeText(Register.this, "Gagal Membuat
Akun: " + errorMessage, Toast.LENGTH_SHORT).show();
162.                            }
163.                        }
164.                    });
165.                } else {
166.                    Toast.makeText(Register.this, "Masukan Email dan Password",
Toast.LENGTH_SHORT).show();
167.                }
168.            }
169.        });
170.
171.
172.
173.    }
174.
175. }
176.

```

Kode Program Halaman RiwayatPengajuan.java

```

1. package com.example.barangku.activity.user_activity;
2.
3. import androidx.appcompat.app.AppCompatActivity;
4. import androidx.fragment.app.Fragment;
5. import androidx.fragment.app.FragmentManager;
6. import androidx.fragment.app.FragmentTransaction;
7.
8. import android.os.Bundle;
9.
10. import com.example.barangku.R;
11. import com.google.android.material.bottomnavigation.BottomNavigationView;
12.
13. public class RiwayatPengajuan extends AppCompatActivity {
14.
15.     @Override
16.     protected void onCreate(Bundle savedInstanceState) {
17.         super.onCreate(savedInstanceState);
18.         setContentView(R.layout.activity_riwayat_pengajuan);
19.         gantiFragment(new RiwayatPengajuanBarangMasukFragment());
20.         BottomNavigationView navigationView =
findViewById(R.id.navigations_bar);
21.         navigationView.setOnItemSelectedListener(item -> {
22.             if (item.getItemId() == R.id.nav_barang_masuk){
23.                 gantiFragment(new RiwayatPengajuanBarangMasukFragment());
24.             }else if(item.getItemId() == R.id.nav_barang_keluar){
25.                 gantiFragment(new RiwayatPengajuanBarangKeluarFragment());
26.             }
27.             return true;
28.         });
29.     }
30.     private void gantiFragment(Fragment fragment){
31.         FragmentManager fragmentManager = getSupportFragmentManager();
32.         FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();

```

```

33.         fragmentTransaction.replace(R.id.frameLayout, fragment);
34.         fragmentTransaction.commit();
35.     }
36. }
37.

```

Kode Program Halaman RiwayatPengajuanBarangKeluarFragment.java

```

1. package com.example.barangku.activity.user_activity;
2.
3. import android.content.Intent;
4. import android.os.Bundle;
5.
6. import androidx.annotation.NonNull;
7. import androidx.annotation.Nullable;
8. import androidx.fragment.app.Fragment;
9. import androidx.recyclerview.widget.LinearLayoutManager;
10. import androidx.recyclerview.widget.RecyclerView;
11.
12. import android.view.LayoutInflater;
13. import android.view.View;
14. import android.view.ViewGroup;
15. import android.widget.ImageView;
16. import android.widget.TextView;
17. import android.widget.Toast;
18.
19. import com.example.barangku.R;
20. import
com.example.barangku.activity.adapter.AdapterRiwayatPengajuanBarangKeluar;
21. import
com.example.barangku.activity.adapter.AdapterRiwayatPengajuanBarangMasuk;
22. import com.example.barangku.activity.model.ModelRiwayatPengajuanBarangKeluar;
23. import com.example.barangku.activity.model.ModelRiwayatPengajuanBarangMasuk;
24. import com.google.firebaseio.database.DataSnapshot;
25. import com.google.firebaseio.database.DatabaseError;
26. import com.google.firebaseio.database.DatabaseReference;
27. import com.google.firebaseio.database.FirebaseDatabase;
28. import com.google.firebaseio.database.ValueEventListener;
29.
30. import java.text.ParseException;
31. import java.text.SimpleDateFormat;
32. import java.util.ArrayList;
33. import java.util.Date;
34. import java.util.List;
35. import java.util.Locale;
36.
37.
38. public class RiwayatPengajuanBarangKeluarFragment extends Fragment {
39.     private TextView tvToolbar;
40.     private ImageView ivBack;
41.     private RecyclerView rvRiwayatPengajuan;
42.     private AdapterRiwayatPengajuanBarangKeluar adapter;
43.     private List<ModelRiwayatPengajuanBarangKeluar> riwayatList = new
ArrayList<>();
44.     private DatabaseReference riwayatPengajuanRef =
FirebaseDatabase.getInstance().getReference("RiwayatPengajuanBarangKeluar");
45.

```

```

46.     public RiwayatPengajuanBarangKeluarFragment() {
47.         // Required empty public constructor
48.     }
49.
50.
51.     @Override
52.     public void onCreate(Bundle savedInstanceState) {
53.         super.onCreate(savedInstanceState);
54.     }
55.
56.     @Override
57.     public View onCreateView(LayoutInflater inflater, ViewGroup container,
58.                             Bundle savedInstanceState) {
59.         // Inflate the layout for this fragment
60.         return
inflater.inflate(R.layout.fragment_riwayat_pengajuan_barang_keluar, container,
false);
61.     }
62.
63.     @Override
64.     public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
65.         super.onViewCreated(view, savedInstanceState);
66.
67.         tvToolbar = view.findViewById(R.id.tv_judul);
68.         tvToolbar.setText("Riwayat Pengajuan");
69.
70.         ivBack = view.findViewById(R.id.iv_back);
71.         ivBack.setOnClickListener(new View.OnClickListener() {
72.             @Override
73.             public void onClick(View view) {
74.                 Intent intent = new Intent(getActivity(),
MainActivity.class );
75.                 startActivity(intent);
76.             }
77.         });
78.
79.         rvRiwayatPengajuan = view.findViewById(R.id.rv_riwayat_pengajuan);
80.
81.         adapter = new AdapterRiwayatPengajuanBarangKeluar(riwayatList,
getView().getContext());
82.         LinearLayoutManager lmRiwayat = new
LinearLayoutManager(getView().getContext(), LinearLayoutManager.VERTICAL, false);
83.         rvRiwayatPengajuan.setLayoutManager(lmRiwayat);
84.         rvRiwayatPengajuan.setAdapter(adapter);
85.
86.         loadRiwayatData();
87.     }
88.
89.     private void loadRiwayatData() {
90.         riwayatPengajuanRef.addValueEventListener(new ValueEventListener() {
91.             @Override
92.             public void onDataChange(@NonNull DataSnapshot snapshot) {
93.                 riwayatList.clear();
94.                 for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
95.                     ModelRiwayatPengajuanBarangKeluar riwayat =
dataSnapshot.getValue(ModelRiwayatPengajuanBarangKeluar.class);
96.                     if (riwayat != null) {
97.
98.                         riwayat.setTanggalKeluar(formatTanggal(riwayat.getTanggalKeluar()));
99.                         riwayatList.add(riwayat);
100.                    }
101.                }
102.            }
103.        }
104.    }
105. 
```

```

102.         }
103.
104.     @Override
105.     public void onCancelled(@NonNull DatabaseError error) {
106.         Toast.makeText(requireActivity(), "Gagal memuat data riwayat
pengajuan", Toast.LENGTH_SHORT).show();
107.     }
108. });
109. }
110. private String formatTanggal(String tanggal) {
111.     SimpleDateFormat sdfSource = new SimpleDateFormat("yyyy-MM-dd", new
Locale("id", "ID"));
112.     SimpleDateFormat sdfDestination = new SimpleDateFormat("EEEE, dd MMMM
yyyy", new Locale("id", "ID"));
113.     try {
114.         Date date = sdfSource.parse(tanggal);
115.         return sdfDestination.format(date);
116.     } catch (ParseException e) {
117.         e.printStackTrace();
118.         return tanggal;
119.     }
120. }
121. }
122.

```

Kode Program Halaman RiwayatPengajuanBarangMasukFragment.java

```

1. package com.example.barangku.activity.user_activity;
2.
3. import android.content.Intent;
4. import android.os.Bundle;
5. import androidx.annotation.NonNull;
6. import androidx.annotation.Nullable;
7. import androidx.fragment.app.Fragment;
8. import androidx.recyclerview.widget.LinearLayoutManager;
9. import androidx.recyclerview.widget.RecyclerView;
10. import android.view.LayoutInflater;
11. import android.view.View;
12. import android.view.ViewGroup;
13. import android.widget.ImageView;
14. import android.widget.TextView;
15. import android.widget.Toast;
16. import com.example.barangku.R;
17. import
com.example.barangku.activity.adapter.AdapterRiwayatPengajuanBarangMasuk;
18. import com.example.barangku.activity.model.ModelRiwayatPengajuanBarangMasuk;
19. import com.google.firebase.database.DataSnapshot;
20. import com.google.firebase.database.DatabaseError;
21. import com.google.firebaseio.database.DatabaseReference;
22. import com.google.firebaseio.database.FirebaseDatabase;
23. import com.google.firebaseio.database.ValueEventListener;
24.
25. import java.text.ParseException;
26. import java.text.SimpleDateFormat;
27. import java.util.ArrayList;
28. import java.util.Date;
29. import java.util.List;

```

```

30. import java.util.Locale;
31.
32. public class RiwayatPengajuanBarangMasukFragment extends Fragment {
33.
34.     private TextView tvToolbar;
35.     private ImageView ivBack;
36.     private RecyclerView rvRiwayatPengajuan;
37.     private AdapterRiwayatPengajuanBarangMasuk adapter;
38.     private List<ModelRiwayatPengajuanBarangMasuk> riwayatList = new
ArrayList<>();
39.     private DatabaseReference riwayatPengajuanRef =
FirebaseDatabase.getInstance().getReference("RiwayatPengajuanBarangMasuk");
40.
41.     public RiwayatPengajuanBarangMasukFragment() {
42.         // Required empty public constructor
43.     }
44.
45.     @Override
46.     public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {
47.         // Inflate the layout for this fragment
48.         return
inflater.inflate(R.layout.fragment_riwayat_pengajuan_barang_masuk, container,
false);
49.     }
50.
51.     @Override
52.     public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
53.         super.onViewCreated(view, savedInstanceState);
54.         tvToolbar = view.findViewById(R.id.tv_judul);
55.         tvToolbar.setText("Riwayat Pengajuan");
56.
57.         ivBack = view.findViewById(R.id.iv_back);
58.         ivBack.setOnClickListener(new View.OnClickListener() {
59.             @Override
60.             public void onClick(View view) {
61.                 Intent intent = new Intent(getActivity(),
MainActivity.class) ;
62.                 startActivity(intent);
63.             }
64.         });
65.
66.         rvRiwayatPengajuan = view.findViewById(R.id.rv_riwayat_pengajuan);
67.
68.         adapter = new AdapterRiwayatPengajuanBarangMasuk(riwayatList,
getView().getContext());
69.         LinearLayoutManager lmRiwayat = new
LinearLayoutManager(getView().getContext(), LinearLayoutManager.VERTICAL, false);
70.         rvRiwayatPengajuan.setLayoutManager(lmRiwayat);
71.         rvRiwayatPengajuan.setAdapter(adapter);
72.
73.         loadRiwayatData();
74.     }
75.
76.     private void loadRiwayatData() {
77.         riwayatPengajuanRef.addValueEventListener(new ValueEventListener() {
78.             @Override
79.             public void onDataChange(@NonNull DataSnapshot snapshot) {
80.                 riwayatList.clear();
81.                 for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
82.                     ModelRiwayatPengajuanBarangMasuk riwayat =
dataSnapshot.getValue(ModelRiwayatPengajuanBarangMasuk.class);
83.                     if (riwayat != null) {

```

```

84.    riwayat.setTanggalMasuk(formatTanggal(riwayat.getTanggalMasuk())));
85.        riwayatList.add(riwayat);
86.    }
87. }
88.     adapter.notifyDataSetChanged();
89. }
90.
91. @Override
92. public void onCancelled(@NonNull DatabaseError error) {
93.     Toast.makeText(requireActivity(), "Gagal memuat data riwayat
pengajuan", Toast.LENGTH_SHORT).show();
94. }
95. });
96. }
97. private String formatTanggal(String tanggal) {
98.     SimpleDateFormat sdfSource = new SimpleDateFormat("yyyy-MM-dd", new
Locale("id", "ID"));
99.     SimpleDateFormat sdfDestination = new SimpleDateFormat("EEEE, dd MMMM
yyyy", new Locale("id", "ID"));
100.    try {
101.        Date date = sdfSource.parse(tanggal);
102.        return sdfDestination.format(date);
103.    } catch (ParseException e) {
104.        e.printStackTrace();
105.        return tanggal;
106.    }
107. }
108. }
109.

```

Kode Program Halaman StockBarang.java

```

1. package com.example.barangku.activity.user_activity;
2.
3. import androidx.annotation.NonNull;
4. import androidx.annotation.Nullable;
5. import androidx.appcompat.app.AppCompatActivity;
6. import androidx.appcompat.widget.SearchView;
7. import androidx.recyclerview.widget.LinearLayoutManager;
8. import androidx.recyclerview.widget.RecyclerView;
9.
10. import android.app.Dialog;
11. import android.app.ProgressDialog;
12. import android.content.Intent;
13. import android.net.Uri;
14. import android.os.Bundle;
15. import android.view.View;
16. import android.view.ViewGroup;
17. import android.widget.AdapterView;
18. import android.widget.ArrayAdapter;
19. import android.widget.Button;
20. import android.widget.EditText;
21. import android.widget.ImageView;
22. import android.widget.Spinner;
23. import android.widget.TextView;
24. import android.widget.Toast;
25.
26. import com.bumptech.glide.Glide;

```

```

27. import com.example.barangku.R;
28. import com.example.barangku.activity.adapter.AdapterStock;
29. import com.example.barangku.activity.model.ModelStock;
30. import com.example.barangku.activity.user_activity.utils.ItemClickStock;
31. import com.github.dhaval2404.imagepicker.ImagePicker;
32. import com.google.android.gms.tasks.OnCompleteListener;
33. import com.google.android.gms.tasks.OnFailureListener;
34. import com.google.android.gms.tasks.OnSuccessListener;
35. import com.google.android.gms.tasks.Task;
36. import com.google.android.material.floatingactionbutton.FloatingActionButton;
37. import com.google.firebase.database.DataSnapshot;
38. import com.google.firebase.database.DatabaseError;
39. import com.google.firebase.database.DatabaseReference;
40. import com.google.firebase.database.FirebaseDatabase;
41. import com.google.firebaseio.database.ValueEventListener;
42. import com.google.firebaseio.storage.FirebaseStorage;
43. import com.google.firebaseio.storage.StorageReference;
44. import com.google.firebaseio.storage.UploadTask;
45.
46. import java.text.SimpleDateFormat;
47. import java.util.ArrayList;
48. import java.util.Date;
49. import java.util.HashMap;
50. import java.util.List;
51. import java.util.Locale;
52.
53. public class StockBarang extends AppCompatActivity implements ItemClickStock {
54. private TextView tvToolbar, tvNamaBarang, tvJumlahItem, tvSatuan;
55. private Dialog dialog, dialogStock, dialogEdit;
56. private ImageView ivStockBarang;
57. private String namabarang, jumlahbarang, satuan;
58. private Uri gambar ;
59. private DatabaseReference reference =
FirebaseDatabase.getInstance().getReference("StockBarang");
60. private StorageReference storageReference =
FirebaseStorage.getInstance().getReference();
61. private FloatingActionButton fabTambah;
62. private ProgressDialog pd;
63. private SearchView searchView;
64. RecyclerView rvStock;
65. private ImageView ivback;
66. private List<ModelStock> list_stock = new ArrayList<>();
67. private AdapterStock adapterStock;
68.     @Override
69.     protected void onCreate(Bundle savedInstanceState) {
70.         super.onCreate(savedInstanceState);
71.         setContentView(R.layout.activity_stock_barang);
72.
73.         tvToolbar =findViewById(R.id.tv_judul);
74.         tvToolbar.setText("Stock Barang");
75.         searchView = findViewById(R.id.search);
76.
77.         tvNamaBarang = findViewById(R.id.tv_nama_barang);
78.         tvJumlahItem = findViewById(R.id.tv_jumlah_item);
79.         tvSatuan = findViewById(R.id.tv_satuan);
80.
81.         ivback=findViewById(R.id.iv_back);
82.         ivback.setOnClickListener(new View.OnClickListener() {
83.             @Override
84.             public void onClick(View v) {
85.                 Intent intent = new Intent(StockBarang.this,
MainActivity.class);
86.                 startActivity(intent);
87.             }

```

```

88.        });
89.        pd = new ProgressDialog(this);
90.        pd.setMessage("Loading");
91.        pd.setCancelable(false);
92.        pd.show();
93.
94.        rvStock = findViewById(R.id.rv_stock);
95.        searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener()
{
96.            @Override
97.            public boolean onQueryTextSubmit(String query) {
98.                filterList(query);
99.                searchView.clearFocus();
100.               return true;
101.            }
102.
103.            @Override
104.            public boolean onQueryTextChange(String newText) {
105.                filterList(newText);
106.                return true;
107.            }
108.        });
109.
110.       adapterStock = new AdapterStock(StockBarang.this, list_stock);
111.       adapterStock.setItemClickStock(this);
112.       LinearLayoutManager lmStock = new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false);
113.       rvStock.setLayoutManager(lmStock);
114.       rvStock.setAdapter(adapterStock);//
115.
116.       reference.addValueEventListener(new ValueEventListener() {
117.           @Override
118.           public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
119.               list_stock.clear();
120.               for (DataSnapshot snapshot: dataSnapshot.getChildren()){
121.                   ModelStock ms = snapshot.getValue(ModelStock.class);
122.                   list_stock.add(ms);
123.               }
124.               pd.dismiss();
125.               adapterStock.notifyDataSetChanged();//
126.           }
127.           @Override
128.           public void onCancelled(@NonNull DatabaseError databaseError) {
129.               Toast.makeText(StockBarang.this, "Data Gagal Dimuat",
Toast.LENGTH_SHORT).show();
130.           }
131.       });
132.       fabTambah = findViewById(R.id.fab_tambah_barang);
133.       fabTambah.setOnClickListener(new View.OnClickListener() {
134.           @Override
135.           public void onClick(View view) {
136.               dialog = new Dialog(StockBarang.this);
137.               dialog.setContentView(R.layout.dialog_tambah_stock);
138.
dialog.getWindow().setLayout(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
139.               dialog.setCancelable(false);
140.               dialog.show();
141.               EditText etNamaBarang =
dialog.findViewById(R.id.et_nama_barang);
142.               EditText etJumlahBarang =
dialog.findViewById(R.id.et_jumlah_barang);
143.
144.               ivStockBarang = dialog.findViewById(R.id.iv_stock_barang);

```

```

145.
146.        Spinner sp = dialog.findViewById(R.id.sp_satuan);
147.        ArrayAdapter<CharSequence> adapter =
148.            ArrayAdapter.createFromResource(
149.                StockBarang.this,
150.                R.array.Satuan,
151.                android.R.layout.simple_spinner_item
152.            );
153.
154.        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
155.        sp.setAdapter(adapter);
156.        sp.setOnItemSelectedListener(new
157.            AdapterView.OnItemSelectedListener() {
158.                @Override
159.                public void onItemSelected(AdapterView<?> adapterView, View
160. view, int pos, long l) {
161.                    satuan = adapterView.getItemAtPosition(pos).toString();
162.                }
163.                @Override
164.                public void onNothingSelected(AdapterView<?> adapterView) {
165.                }
166.            });
167.
168.        ivStockBarang.setOnClickListener(new View.OnClickListener() {
169.            @Override
170.            public void onClick(View view) {
171.                ImagePicker.with(StockBarang.this)
172.                    .galleryMimeTypes(new String[]{"image/png",
173. "image/jpg", "image/jpeg"})
174.                    .crop(16f, 9f)
175.                    .compress(256)
176.                    .maxResultSize(1080, 1080)
177.                    .start();
178.            }
179.        });
180.
181.        dialog.findViewById(R.id.iv_close).setOnClickListener(new
182. View.OnClickListener() {
183.             @Override
184.             public void onClick(View view) {
185.                 dialog.dismiss();
186.             }
187.         });
188.         dialog.findViewById(R.id.btn_simpan).setOnClickListener(new
189. View.OnClickListener() {
190.             @Override
191.             public void onClick(View view) {
192.                 namabarang = etNamaBarang.getText().toString();
193.                 jumlahbarang = etJumlahBarang.getText().toString();
194.
195.                 if (namabarang.trim().isEmpty()) {
196.                     etNamaBarang.setError("Required");
197.                     return;
198.                 }
199.                 if (jumlahbarang.trim().isEmpty()) {
200.                     etJumlahBarang.setError("Required");
201.                     return;
202.                 }
203.                 if (gambar == null) {

```

```

201.           Toast.makeText(StockBarang.this, "Pilih Gambar",
202.           Toast.LENGTH_SHORT).show();
203.           return;
204.           }//
205.           else {
206.               saveFirebase();
207.           }
208.       });
209.
210.   });
211. }
212. }
213.
214. private void saveFirebase() {
215.     StorageReference imagereference =
storageReference.child("StockBarang").child(String.valueOf(System.currentTimeMillis()));
216.     imagereference.putFile(gambar).addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
217.         @Override
218.         public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
219.             imagereference.getDownloadUrl().addOnSuccessListener(new
OnSuccessListener<Uri>() {
220.                 @Override
221.                 public void onSuccess(Uri uri) {
222. //                     String uniqueId = reference.getKey();
223. //                     ModelStock ms = new ModelStock(uniqueId,namabarang,
jumlahbarang, satuan, uri.toString());
224. //                     reference.child(uniqueId).setValue(ms);
225.
226. //                     Generate ID Barang secara otomatis di Firebaase
227. //                     DatabaseReference newReference = reference.push();
228. //                     String id = newReference.getKey();
229. //                     ModelStock ms = new ModelStock(id, namabarang,
jumlahbarang, satuan, uri.toString());
230. //                     newReference.setValue(ms);
231.
232.                     String idBarang = generateIdbarang(namabarang);
233.
234.                     ModelStock ms = new ModelStock(idBarang, namabarang,
jumlahbarang, satuan, uri.toString(), true);
235.                     reference.child(idBarang).setValue(ms);
236.
237.                     Toast.makeText(StockBarang.this, "Stock Barang Berhasil
Disimpan", Toast.LENGTH_SHORT).show();
238.                     dialog.dismiss();
239.                     refreshStockList();
240.                 }
241.
242.             }).addOnFailureListener(new OnFailureListener() {
243.                 @Override
244.                 public void onFailure(@NonNull Exception e) {
245.                     Toast.makeText(StockBarang.this, "Error! " +
e.getMessage(), Toast.LENGTH_SHORT).show();
246.                 }
247.             });
248.         });
249.     });
250.
251. }
252.
253. private String generateIdbarang(String namabarang) {
254.     int maxLength = 3;

```

```

255.         String prefix = namabarang.length() >= maxLength ?
namabarang.substring(0, maxLength).toUpperCase() : namabarang.toUpperCase();
256.
257.         String timestamp = new SimpleDateFormat("yyyyMMddHHmmss",
Locale.getDefault()).format(new Date());
258.         String id = prefix.concat(timestamp.substring(timestamp.length() - (7 -
maxLength)));
259.         return id;
260.     }
261.
262.     private void refreshStockList() {
263.         reference.addListenerForSingleValueEvent(new ValueEventListener() {
264.             @Override
265.             public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
266.                 // Clear the list again to ensure only new data is fetched
list_stock.clear();
267.
268.                 for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
269.                     ModelStock ms = snapshot.getValue(ModelStock.class);
270.                     list_stock.add(ms);
271.
272.                 }
273.                 adapterStock.notifyDataSetChanged();
274.                 pd.dismiss();
275.             }
276.         }
277.
278.         @Override
279.         public void onCancelled(@NonNull DatabaseError databaseError) {
280.             // Handle database errors (optional)
281.             Toast.makeText(StockBarang.this, "Data Gagal Di Simpan",
Toast.LENGTH_SHORT).show();
282.         }
283.     });
284. }
285.
286. private void filterList(String query) {
287.     List<ModelStock> filteredList = new ArrayList<>();
288.     String queryInLower = query.toLowerCase();
289.     for (ModelStock ms : list_stock) {
290.         if (ms.getNamaBarang().toLowerCase().contains(queryInLower)) {
291.             filteredList.add(ms);
292.         } else if
(ms.getJumlahBarang().toLowerCase().contains(queryInLower)) {
293.             filteredList.add(ms);
294.         }
295.     }
296.     if (filteredList.isEmpty()) {
297.         Toast.makeText(this, "Tidak Ada Stock Yang Tersedia",
Toast.LENGTH_SHORT).show();
298.     } else {
299.         adapterStock.setFilter(filteredList);
300.     }
301. }
302.
303. @Override
304. protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
305.     super.onActivityResult(requestCode, resultCode, data);
306.     gambar = data.getData();
307.     ivStockBarang.setImageURI(gambar);
308. }
309.
310. @Override
311. public void onItemClickListener(ModelStock data, int position) {

```

```

312.         dialogStock = new Dialog(StockBarang.this);
313.         dialogStock.setContentView(R.layout.dialog_tampil_stock);
314.         dialogStock.getWindow().setLayout(ViewGroup.LayoutParams.WRAP_CONTENT,
315.             ViewGroup.LayoutParams.WRAP_CONTENT);
316.         dialogStock.setCancelable(false);
317.         dialogStock.show();
318.
319.         ImageView ivStockBarang =
320.             dialogStock.findViewById(R.id.iv_foto_barang);
321.             Glide.with(StockBarang.this)
322.                 .load(data.getGambar())
323.                 .into(ivStockBarang);
324.
325.         TextView tvNamaBarang = dialogStock.findViewById(R.id.tv_nama_barang);
326.         tvNamaBarang.setText(data.getNamaBarang());
327.         TextView tvJumlahItem = dialogStock.findViewById(R.id.tv_jumlah_item);
328.         tvJumlahItem.setText(data.getJumlahBarang());
329.         TextView tvSatuan = dialogStock.findViewById(R.id.tv_satuan);
330.         tvSatuan.setText(data.getSatuan());
331.
332.         Button btnEnable = dialogStock.findViewById(R.id.btn_enable);
333.         Button btnDisable = dialogStock.findViewById(R.id.btn_disable);
334.
335.         btnEnable.setOnClickListener(new View.OnClickListener() {
336.             @Override
337.             public void onClick(View view) {
338.                 HashMap<String, Object> updateData = new HashMap<>();
339.                 updateData.put("enable", true);
340.                 reference.child(data.getId()).updateChildren(updateData)
341.                     .addOnCompleteListener(new OnCompleteListener<Void>() {
342.                         @Override
343.                         public void onComplete(@NonNull Task<Void> task) {
344.                             if(task.isSuccessful()){
345.                                 Toast.makeText(StockBarang.this, "Status
Berhasil Diubah Ke Enable", Toast.LENGTH_SHORT).show();
346.                             }
347.                             else {
348.                                 Toast.makeText(StockBarang.this, "Gagal
Mengubah Status", Toast.LENGTH_SHORT).show();
349.                             }
350.                         });
351.                     }
352.             });
353.         btnDisable.setOnClickListener(new View.OnClickListener() {
354.             @Override
355.             public void onClick(View view) {
356.                 HashMap<String, Object> updateData = new HashMap<>();
357.                 updateData.put("enable", false);
358.                 reference.child(data.getId()).updateChildren(updateData)
359.                     .addOnCompleteListener(new OnCompleteListener<Void>() {
360.                         @Override
361.                         public void onComplete(@NonNull Task<Void> task) {
362.                             if(task.isSuccessful()){
363.                                 Toast.makeText(StockBarang.this, "Status
Berhasil Diubah Ke Disable", Toast.LENGTH_SHORT).show();
364.                             }
365.                             else {
366.                                 Toast.makeText(StockBarang.this, "Gagal
Mengubah Status", Toast.LENGTH_SHORT).show();si
367.                             }
368.                         });
369.                     }

```

```
370.                });
371.            }
372.        });
373.        dialogStock.findViewById(R.id.iv_close).setOnClickListener(view ->
374.    {dialogStock.dismiss();});
375.
376.    }
377.
378.
379. }
380.
381.
```