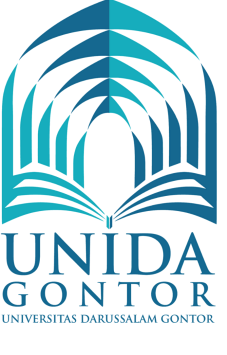
**IMPLEMENTASI ANDROID ARCHITECTURE MODEL – VIEW – VIEWMODEL (MVVM) DAN JETPACK COMPOSE PADA APLIKASI HUKUM BERBASIS ANDROID**



ABDULLAH HIKMATYAR PRIANDIKA

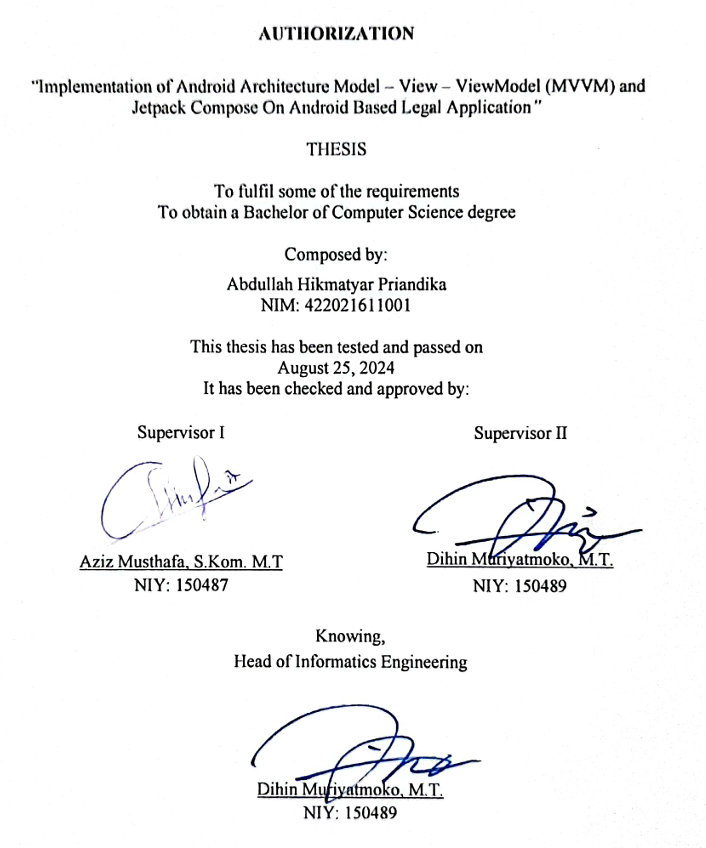
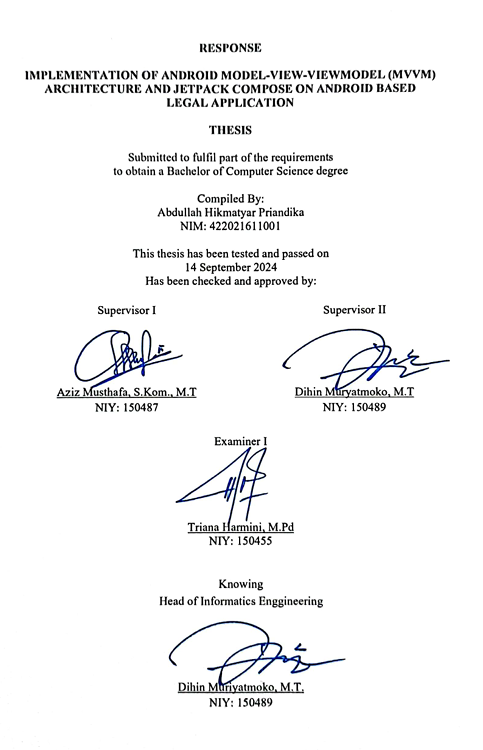
NIM: 422021611001

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS DARUSSALAM GONTOR**

**2024**



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

KATA PENGANTAR

Puji dan syukur kita panjatkan kehadirat Allah SWT. karena dengan nikmat, rahmat dan karunia-Nya yang senantiasa tercurah kepada kita semua sebagai hamba-Nya sehingga kita dapat menimba ilmu di tempat yang mulia ini. Shalawat serta salam kita haturkan kepada nabi besar kita Nabi Muhammad SAW yang telah membawa umatnya dari jaman kegelapan menuju jalan yang benar dan diridhoi oleh Allah SWT.

Tak lupa penulis mengucapkan terima kasih kepada seluruh dosen yang telah membantu proses penulisan skripsi, terutama dosen pembimbing yang telah membimbing pengerjaan dan penulisan skripsi hingga selesai. Dengan segala kerendahan hati, penulis menyadari bahwa tahapan pengerjaan skripsi ini tidak lepas dari peran pihak-pihak yang telah membantu, memberikan arahan, dan juga menyemangati penulis dalam pengerjaan skripsi. Penulis juga ingin menyampaikan ucapan terima kasih kepada:

1. Orang tua penulis, Ibu, Bapak, dan keluarga yang selalu memberikan dukungan, doa kepada penulis serta selalu sabar, tegar dan kuat dalam mendidik dan membesarkan penulis.
2. Prof. Dr. KH. Hamid Fahmy Zarkasyi, M.A.Ed., M.Phil selaku rektor, segenap dosen Universitas Darussalam Gontor beserta seluruh jajarannya yang telah memberikan banyak ilmu serta mendidik penulis.
3. Haris Setyaningrum, M.Sc. selaku Dekan Fakultas Sains dan Teknologi.
4. Al-Ustadz Dihin Muriyatmoko, M.T. selaku Ketua Program Studi Teknik Informatika beserta dosen dan staf, yang telah memberikan bimbingan dalam penelitian ini.
5. Al-Ustadz Aziz Musthafa, S.Kom, M.T. dan Al-Ustadz Dihin Muriyatmoko, M.T. selaku dosen pembimbing penulis yang selalu memberikan bimbingan dan arahan kepada penulis dalam mengerjakan proses skripsi ini.
6. Ustadz Muhammad Taqiyuddin, S.H.I., M.Ag selaku dosen dan pembimbing Islamisasi ilmu pengetahuan yang telah bersedia membantu penulis dengan membimbing proses penulisan tesis.
7. Teman-teman dari program studi Teknik Informatika, dan teman-teman lainnya yang telah bersedia membantu penulis dalam menyelesaikan penelitian ini.

ABSTRAK

Dikutip dari jurnal Geraldo A. Lambonan berjudul “Rancang Bangun Aplikasi Ensiklopedia Hukum Indonesia Berbasis Android”, penggunaan buku-buku yang berisi istilah dan undang-undang hukum di Indonesia masih banyak digunakan sebagai sumber informasi. Namun, metode ini memiliki kelemahan dari segi kepraktisan. Untuk mengatasi masalah tersebut, penelitian ini mengembangkan sebuah aplikasi hukum berbasis Android yang membantu pengguna mendapatkan bantuan hukum dengan 2 fitur utama yaitu *Chat Bot* peraturan undang – undang dan daftar peraturan undang - undang. Aplikasi ini menggunakan pola arsitektur Model-View-ViewModel (MVVM) yang memisahkan tampilan antarmuka pengguna (UI) dari logika bisnis, sehingga memudahkan pengembangan dan pemeliharaan. Selain itu, Jetpack Compose digunakan sebagai toolkit UI untuk merancang antarmuka pengguna yang interaktif dan responsif. Hasil pengujian aplikasi menggunakan metode Black Box menunjukkan bahwa aplikasi berjalan dengan lancar tanpa error. Uji coba dengan penyebaran kuesioner kepada pengguna dan ahli materi juga menunjukkan hasil yang baik. Pengujian oleh ahli materi memperoleh nilai rata-rata sebesar 86% dari total perhitungan skala Likert (99 poin), sedangkan pengujian oleh pengguna mencapai nilai rata-rata 87,2% berdasarkan skala Likert dari setiap pertanyaan. Dan hasil akhir dari penelitian ini adalah berhasil mengembangkan aplikasi hukum berbasis android dengan mengimplemtasikan pola *model-view-viewmodel* (MVVM) sebagai arsitektur android dan android *Jetpack Compose* sebagai *UI Toolkit* desain aplikasi*.*

Kata kunci: android, MVVM, jetpack compose, hukum, chat bot

ABSTRACT

Quoted from Geraldo A. Lambonan's journal titled "Design and Development of the Indonesian Law Encyclopedia Application Based on Android," law books containing legal terms and regulations in Indonesia are still widely used as information sources, but they lack practicality. To address this, the research developed an Android-based law application with two main features: a regulation chatbot and a list of laws. The app uses the Model-View-ViewModel (MVVM) architecture, separating the user interface from business logic for easier development and maintenance. Jetpack Compose is used as a UI toolkit to design an interactive and responsive interface. Black Box testing showed that the app runs smoothly without errors. Questionnaire feedback from users and subject matter experts revealed positive results, with experts rating it at 86% and users at 87.2% on the Likert scale. The final result of this research successfully developed a law application with MVVM architecture and Jetpack Compose as the UI toolkit.

Keywords: android, MVVM, jetpack compose, law, chat bot

**Daftar Isi**

[PERNYATAAN ORISINALITAS iii](#_Toc177552332)

[KATA PENGANTAR iv](#_Toc177552333)

[ABSTRAK v](#_Toc177552334)

[ABSTRACT vi](#_Toc177552335)

[Daftar Isi vii](#_Toc177552336)

[Daftar Tabel x](#_Toc177552337)

[Daftar Gambar xi](#_Toc177552338)

[Bab 1 Pendahuluan 1](#_Toc177552339)

[1.1 Latar Belakang 1](#_Toc177552340)

[1.2 Rumusan Masalah 3](#_Toc177552341)

[1.3 Tujuan Penelitian 3](#_Toc177552342)

[1.4 Batasan Masalah 4](#_Toc177552343)

[1.5 Manfaat Penelitian 4](#_Toc177552344)

[Bab 2 Tinjauan Pustaka 5](#_Toc177552345)

[2.1 Penelitian Terdahulu 5](#_Toc177552346)

[2.1.1 Muhammad Syakir Arif, Aziz Musthafa, Dihin Muryatmoko (2019) 5](#_Toc177552347)

[2.1.2 Galuh Wilujeng Saraswati, Nanang Febrianto (2021) 5](#_Toc177552348)

[2.1.3 Putu Reditya Permana Putra, Herman Tolle (2023) 6](#_Toc177552349)

[2.1.4 Irya Muhammad Riyadhi (2023) 6](#_Toc177552350)

[2.1.5 Cahya Diantoni, Oman Komarudin, Adhi Rizal (2024) 6](#_Toc177552351)

[2.2 Landasan Teori 9](#_Toc177552352)

[2.2.1 Aplikasi Mobile 9](#_Toc177552353)

[2.2.2 Android 9](#_Toc177552354)

[2.2.3 Website Hukum (peraturan.co.id) 9](#_Toc177552355)

[2.2.4 Model – View – ViewModel (MVVM) 10](#_Toc177552356)

[2.2.5 Jetpack Compose 11](#_Toc177552357)

[Bab 3 Metode Penelitian 12](#_Toc177552358)

[3.1 Rencana Kegiatan 12](#_Toc177552359)

[3.2 Alat Dan Bahan 12](#_Toc177552360)

[3.2.1 Perangkat Keras 12](#_Toc177552361)

[3.2.2 Perangkat Lunak 13](#_Toc177552362)

[3.3 Tahapan Penelitian 13](#_Toc177552363)

[3.3.1 Analisis Kebutuhan (Requirements) 14](#_Toc177552364)

[3.3.2 Desain Aplikasi (Design) 14](#_Toc177552365)

[3.3.3 Implementation (Development) 24](#_Toc177552366)

[3.3.4 Pengujian (Testing) 25](#_Toc177552367)

[3.3.5 Pemeliharaan (Maintenance) 26](#_Toc177552368)

[Bab 4 Hasil dan Pembahasan 27](#_Toc177552369)

[4.1 Rest Api Pada Postman 27](#_Toc177552370)

[4.2 Implementasi Android Jetpack Compose 27](#_Toc177552371)

[4.2.1 Implementasi Android Jetpack Compose Pada Halaman Login 28](#_Toc177552372)

[4.2.2 Implementasi Android Jetpack Compose Pada Halaman register 28](#_Toc177552373)

[4.2.3 Implementasi Android Jetpack Compose Pada Halaman Home 29](#_Toc177552374)

[4.2.4 Implementasi Android Jetpack Compose Pada Halaman Chat bot 30](#_Toc177552375)

[4.2.5 Implementasi Android Jetpack Compose Pada Halaman Peraturan 31](#_Toc177552376)

[4.2.6 Implementasi Android Jetpack Compose Pada Halaman Detail Peraturan 31](#_Toc177552377)

[4.3 Implementasi Model – View – ViewModel (MVVM) 32](#_Toc177552378)

[4.3.1 Implementasi MVVM Untuk Menangani Logika Halaman Login 33](#_Toc177552379)

[4.3.2 Implementasi MVVM Untuk Menangani Logika Halaman Register 34](#_Toc177552380)

[4.3.3 Implementasi MVVM Untuk Menangani Logika Halaman Chat Bot 35](#_Toc177552381)

[4.3.4 Implementasi MVVM Untuk Menangani Logika Halaman Peraturan 36](#_Toc177552382)

[4.3.5 Implementasi MVVM Untuk Menangani Logika Halaman Daftar Peraturan 37](#_Toc177552383)

[4.3.6 Implementasi MVVM Untuk Menangani Logika Halaman Detail Peraturan 38](#_Toc177552384)

[4.4 Hasil Perancangan Aplikasi 39](#_Toc177552385)

[4.5 Pengujian 40](#_Toc177552386)

[4.5.1 Pengujian Black Box 40](#_Toc177552387)

[4.5.2 Pengujian Hardware 41](#_Toc177552388)

[4.5.3 Pengujian Ahli Materi 42](#_Toc177552389)

[4.5.4 Pengujian Ahli Pengguna 43](#_Toc177552390)

[4.6 Maintenance 44](#_Toc177552391)

[Bab 5 Penutup 46](#_Toc177552392)

[5.1 Kesimpulan 46](#_Toc177552393)

[5.2 Saran 46](#_Toc177552394)

[Daftar Pustaka 47](#_Toc177552395)

[Lampiran 49](#_Toc177552396)

[1. Formulir Pengajuan Judul Skripsi 49](#_Toc177552397)

[2. Bukti Observasi dengan Kuesioner 50](#_Toc177552398)

[3. Kuesioner Uji Pengguna 50](#_Toc177552399)

[4. Kuesioner Uji Ahli Materi 54](#_Toc177552400)

Daftar Tabel

[Tabel 2. 1 Tabel Penelitian Terdahulu 7](#_Toc177552404)

[Tabel 3. 1 Tabel Rencana Kegiatan 12](#_Toc177433241)

[Tabel 3. 2 Rancangan Aplikasi 14](#_Toc177433242)

[Tabel 4. 1 Tabel Pengujian Black Box 40](#_Toc177552405)

[Tabel 4. 2 Tabel Pengujian Hardware 41](#_Toc177552406)

[Tabel 4. 3 Tabel Pengujian Ahli Materi 42](#_Toc177552407)

[Tabel 4. 4 Pengujian Pengguna 43](#_Toc177552408)

[Tabel 4. 5 Saran untuk maintenance aplikasi. 44](#_Toc177552409)

Daftar Gambar

[Gambar 2. 1 Pola Arsitektur MVVM 10](#_Toc177559105)

[Gambar 2. 2 Arah Flow Data Jetpack Compose 11](#_Toc177559106)

[Gambar 3. 1 Tahapan Penelitian dengan Metode Waterfall 13](#_Toc177552418)

[Gambar 3. 2 Flowchart Aplikasi 15](#_Toc177552419)

[Gambar 3. 3 Use Case Diagram 16](#_Toc177552420)

[Gambar 3. 4 Activity Diagram 17](#_Toc177552421)

[Gambar 3. 5 Sequence Diagram 19](#_Toc177552422)

[Gambar 3. 6 Halaman Login 20](#_Toc177552423)

[Gambar 3. 7 Halaman Register 20](#_Toc177552424)

[Gambar 3. 8 Halaman Home 21](#_Toc177552425)

[Gambar 3. 9 Halaman Peraturan 22](#_Toc177552426)

[Gambar 3. 10 Halaman List Peraturan 23](#_Toc177552427)

[Gambar 3. 11 Halaman Detail Peraturan 23](#_Toc177552428)

[Gambar 3. 12 Halaman Chat Bot 24](#_Toc177552429)

[Gambar 4. 1 Rest Api 27](#_Toc177552410)

[Gambar 4. 2 Implementasi Android Jetpack Compose pada Halaman Login 28](#_Toc177552411)

[Gambar 4. 3 Implementasi Android Jetpack Compose pada Halaman Register 29](#_Toc177552412)

[Gambar 4. 4 Implementasi Android Jetpack Compose pada Halaman Home 30](#_Toc177552413)

[Gambar 4. 5 Implementasi Android Jetpack Compose pada Halaman Chat Bot 30](#_Toc177552414)

[Gambar 4. 6 Implementasi Android Jetpack Compose pada Halaman Peraturan 31](#_Toc177552415)

[Gambar 4. 7 Implementasi Android Jetpack Compose pada Halaman Detail Peraturan 32](#_Toc177552416)

[Gambar 4. 9 Hasil Perancangan Halaman Peraturan (a) Jenis Peraturan (b) List Peraturan dari Jenis yang dipilih (c) Detail Peraturan 39](#_Toc177552417)

Bab 1 Pendahuluan

1. Latar Belakang

Menurut UUD 1945 pasal 1 ayat 3, dinyatakan bahwa “Negara Indonesia adalah negara hukum”. Hal ini mengartikan bahwa Indonesia menegakkan supremasi hukum untuk menjamin kebenaran dan keadilan. Secara umum, dalam setiap negara yang menganut prinsip negara hukum, terdapat tiga prinsip dasar yang harus dipatuhi: pertama, supremasi hukum *(supremacy of law)*; kedua, kesetaraan di hadapan hukum *(equality before the law)*; dan ketiga, penegakan hukum yang tidak bertentangan dengan hukum itu sendiri *(due process of law).*[[1]](#footnote-1) Hukum mencakup semua aturan dan norma yang berlaku dalam kehidupan bersama yang mengatur perilaku, serta dapat diberlakukan dengan adanya sanksi. Hukum mengatur hubungan hukum yang mencakup ikatan antara individu dengan masyarakat dan antar individu itu sendiri, yang terwujud dalam hak dan kewajiban. Dalam perannya mengatur, hukum berupaya menyeimbangkan kepentingan individu dengan kepentingan masyarakat secara optimal. Karena masyarakat terdiri dari individu-individu yang berinteraksi, selalu ada potensi konflik atau ketegangan antara kepentingan individu dan kepentingan masyarakat.[[2]](#footnote-2)

Pada saat ini kemajuan teknologi informasi dan komunikasi telah menciptakan model layanan yang dapat diakses melalui aplikasi dan *web browser*. Layanan yang sebelumnya dilakukan secara manual kini dapat digantikan dengan aplikasi, sehingga meningkatkan efisiensi dibandingkan sistem manual. Teknologi *mobile*, terutama *smartphone Android*, saat ini sedang mengalami perkembangan yang sangat pesat. Tidak dapat dipungkiri bahwa perkembangan aplikasi pada *smartphone*, khususnya aplikasi *Android*, menjadi salah satu elemen penting dalam membuka wawasan baru dengan mengembangkan dan menerapkan penggunaan aplikasi di berbagai institusi dan perusahaan. Dengan memanfaatkan aplikasi ini, diharapkan dapat meningkatkan kualitas pelayanan dan bantuan hukum, sehingga memberikan manfaat yang signifikan bagi para pengguna.[[3]](#footnote-3)

Salah satu pengembangan aplikasi yang kompleks yaitu harus mempertimbangkan arsitektur yang tepat supaya aplikasi dapat berjalan dengan baik, efisien, dan dapat diuji dengan baik. Salah satu arsitektur yang dapat digunakan dalam pengembangan aplikasi ini adalah pola arsitektur *Model-View-ViewModel* (MVVM). Bedasarkan hasil penelitian yang dilakukan Irya Muhammad Riyadhi pola arsitektur MVVM memisahkan antara tampilan (*View*), logika bisnis (*ViewModel*), dan model data (*Model*) pada aplikasi. Hal ini dapat memudahkan pengembangan dan perawatan aplikasi karena setiap bagian aplikasi memiliki tanggung jawab yang jelas dan terpisah. Selain itu, pola arsitektur MVVM juga dapat meningkatkan pengalaman pengguna, kinerja aplikasi, dan efisiensi pengembangan.[[4]](#footnote-4)

Dan bedasarkan penelitian Muhammad Syakir Arif, *Model-View-ViewModel* (MVVM) adalah salah satu pola arsitektur yang mengedepankan pemisahan tanggung jawab. Pola ini memungkinkan kita untuk memisahkan antara tampilan antarmuka grafis (*graphical user interface*) dengan logika bisnis atau logika belakang (*back-end logic*). Dengan pemisahan ini, pengembangan dan pemeliharaan aplikasi menjadi lebih mudah dan terstruktur. Antarmuka grafis bisa dikembangkan secara *independent* dari logika bisnis, sehingga perubahan pada salah satu bagian tidak langsung mempengaruhi bagian lainnya. Selain itu, MVVM juga mendukung pengujian unit yang lebih baik dan mempermudah proses *debugging*, karena masing-masing komponen memiliki tanggung jawab yang jelas dan terpisah. Pemisahan ini juga memungkinkan pengembang untuk menggunakan kembali komponen logika bisnis di berbagai tampilan, sehingga meningkatkan efisiensi dan konsistensi dalam pengembangan aplikasi.[[5]](#footnote-5)

Selain mempertimbangkan arsitektur aplikasi yang baik dan efisien, pengembangan ini menggunakan *Android Jetpack Compose* untuk mendesain *layout UI* yang keren dengan lebih cepat dan mudah. *Jetpack Compose* adalah sebuah *toolkit UI* modern yang dirancang untuk membuat antarmuka pengguna (UI) pada aplikasi *Android Native*. *Jetpack Compose* mempermudah dan mempercepat proses pengembangan UI di platform Android dengan menyederhanakan banyak aspek yang sebelumnya kompleks. Dengan *Jetpack Compose*, pengembang dapat menciptakan antarmuka yang interaktif dan responsif dengan lebih sedikit baris kode dibandingkan dengan pendekatan tradisional. Alat ini memungkinkan aplikasi untuk lebih cepat dihidupkan dan dijalankan, sehingga meningkatkan efisiensi pengembangan. Selain itu, *Jetpack Compose* mendukung fitur-fitur terbaru dalam desain UI, seperti animasi yang halus dan responsif, serta integrasi yang mudah dengan komponen Android lainnya. Dengan pendekatan deklaratif, pengembang bisa lebih fokus pada desain dan fungsionalitas, tanpa perlu khawatir tentang detail implementasi yang rumit.[[6]](#footnote-6)

Dan penelitian ini berfokus pada masalah pengetahuan dan akses terhadap bantuan hukum. Dikutip dari jurnal Geraldo A. Lambonan dengan judul “Rancang Bangun Aplikasi Ensiklopedia Hukum Indonesia Berbasis Android” penggunaan buku-buku yang berisi istilah dan undang-undang tentang hukum di Indonesia masih banyak digunakan untuk mendapatkan informasi, tetapi dengan demikian menimbulkan kelemahan dari sisi kepraktisan.[[7]](#footnote-7) Dan sesuai dari perbandingan dengan aplikasi hukum lain adalah belum adanya konsultasi dengan menggunakan *Chat Bot* pada aplikasi lain seperti contohnya aplikasi “Jago Hukum”, aplikasi “Perqara”, dan aplikasi “Hukumku”. *Tujuan* utama dari penelitian ini adalah untuk mengembangkan sebuah aplikasi hukum yang memanfaatkan pola arsitektur *Model-View-ViewModel* (MVVM) dan menggunakan *Jetpack Compose* sebagai *toolkit UI* untuk merancang tampilan antarmuka. Aplikasi ini dirancang untuk membantu dalam memperoleh bantuan hukum yang mereka butuhkan. Dengan aplikasi ini, diharapkan user dapat mengakses bantuan hukum secara lebih efektif dan fleksibel, dibandingkan dengan metode tradisional. Selain itu, aplikasi ini akan menjadi alternatif yang praktis bagi mereka yang membutuhkan konsultasi atau bantuan hukum, sehingga mempermudah proses pencarian informasi dan bantuan hukum secara cepat dan efisien. Dengan adanya aplikasi ini, diharapkan dapat meningkatkan pemahaman hukum masyarakat dan memberikan solusi praktis dalam mengatasi berbagai permasalahan hukum yang mereka hadapi.

1. Rumusan Masalah

Adapun rumusan masalah pada penilitian ini adalah sebagai berikut :

* Dikutip dari jurnal Geraldo A. Lambonan dengan judul “Rancang Bangun Aplikasi Ensiklopedia Hukum Indonesia Berbasis Android” Pada saat ini penggunaan buku-buku yang berisi istilah dan undang-undang tentang hukum di Indonesia masih banyak digunakan untuk mendapatkan informasi, tetapi dengan demikian menimbulkan kelemahan dari sisi kepraktisan.

1. Tujuan Penelitian

Dari rumusan masalah yang telah di tuliskan diatas, maka tujuan dari penelitian yang dilakukan adalah sebagai berikut :

* Mengembangkan sebuah aplikasi hukum berbasis Android dengan menerapkan pola arsitektur *Model-View-ViewModel* (MVVM) yang memisahkan tampilan antarmuka pengguna dari logika bisnis, sehingga memudahkan pengembangan dan pemeliharaan. Selain itu, aplikasi ini akan menggunakan *Jetpack Compose* sebagai *toolkit UI* untuk merancang antarmuka pengguna yang interaktif dan responsif.

1. Batasan Masalah

Untuk perumusan permasalahan yang khusus, maka Batasan masalah pada penelitian ini adalah sebagai berikut :

* Objek yang diteliti ini hanya sebatas menggunakan smartphone berbasis android.
* Aplikasi yang dikembangkan hanya akan dibangun untuk platform Android menggunakan *Jetpack Compose*, tidak mencakup platform lain seperti iOS atau web.
* Pengembangan aplikasi ini akan menggunakan pola arsitektur *Model-View-ViewModel* (MVVM) secara eksklusif, tanpa membandingkannya dengan pola arsitektur lainnya.
* Aplikasi ini fokus pada 2 fitur utama yaitu : fitur peraturan undang – undang dan fitur *Chat Bot* peraturan undang - undang*.*
* Data yang digunakan dalam aplikasi ini berasal dari situs resmi *peraturan.go.id*, dan akan selalu dipebarui setiap waktu.
* Evaluasi kinerja aplikasi akan dilakukan melalui simulasi dan uji coba terbatas, tanpa melibatkan uji coba langsung di lapangan secara luas.

1. Manfaat Penelitian

* Aplikasi ini dirancang untuk memudahkan dalam mendapatkan bantuan hukum yang butuhkan user, sehingga memperbaiki aksesibilitas terhadap layanan hukum.
* Penggunaan pola arsitektur *Model – View – ViewModel* (MVVM) dan *Jetpack Compose* akan meningkatkan efisiensi pengembangan dan pemeliharaan aplikasi, sehingga layanan hukum dapat diberikan dengan lebih cepat dan efektif.

Bab 2 Tinjauan Pustaka

Bab ini akan menguraikan beberapa hal yang berkaitan dengan penelitian. Dimana hal tersebut akan memberikan gambaran serta pemahaman yang lebih mendalam mengenai penelitian yang dilakukan. Beberapa hal yang akan diuraikan adalah sebagai berikut :

1. Penelitian Terdahulu
2. Muhammad Syakir Arif, Aziz Musthafa, Dihin Muryatmoko (2019)

Peneletian pertama yang berjudul “Implementasi Pola Arsitektur *Model-View-ViewModel* (MVVM) pada Sistem Informasi Akademik Universitas Darussalam Gontor Berbasis Mobile” yang disusun oleh Muhammad Syakir Arif, Aziz Musthafa, Dihin Muriyatmoko. Penelitian ini menyoroti pentingnya ponsel pintar dalam akses informasi dan layanan data, dengan menyebutkan bahwa pada tahun 2019 pengguna ponsel pintar di Indonesia mencapai 92 juta. Android, sebagai sistem operasi ponsel pintar open source, telah mencapai 2 miliar pengguna aktif bulanan di seluruh dunia pada tahun 2017. Di Indonesia, pangsa pasar Android mencapai 88,37% pada Desember 2017. Dan penelitian ini bertujuan untuk mengatasi kompleksitas dan perubahan perangkat lunak yang semakin meningkat dengan menggunakan pola arsitektur *Model-View-ViewModel* (MVVM). MVVM memungkinkan pemisahan antara antarmuka grafis dan logika bisnis, meningkatkan efisiensi dan keteraturan kode.[[8]](#footnote-8)

1. Galuh Wilujeng Saraswati, Nanang Febrianto (2021)

Penelitian kedua yang berjudul “Penerapan Arsitektur MVVM Pada Aplikasi Tanamin Untuk Mendeteksi Penyakit Tanaman Berbasis Android” yang disusun oleh Galuh Wilujeng Saraswati dan Nanang Febrianto. Penelitian ini menyoroti pentingnya sektor pertanian bagi Indonesia, yang merupakan negara agraris dengan mayoritas penduduknya bermata pencaharian sebagai petani. Berdasarkan data Badan Pusat Statistik (BPS) tahun 2021, sekitar 29,59 persen atau sekitar 38,77 juta penduduk Indonesia bekerja sebagai petani. Indonesia juga menempati urutan ke-5 sebagai negara agraris terbesar di dunia berdasarkan volume ekspor hasil pertanian, kehutanan, dan perikanan. Untuk mendukung sektor pertanian, penelitian ini mengembangkan aplikasi berbasis Android dengan menerapkan arsitektur *Model-View-ViewModel* (MVVM). Arsitektur MVVM dipilih karena memiliki keunggulan dalam efisiensi dan kinerja aplikasi, memisahkan tampilan aplikasi (*View*) dari logika bisnis (*Model* dan *ViewModel*). Penelitian menunjukkan bahwa MVVM memiliki eksekusi kode yang lebih cepat dibandingkan dengan pola arsitektur lain seperti MVC dan MVP, sehingga meningkatkan kinerja aplikasi.[[9]](#footnote-9)

1. Putu Reditya Permana Putra, Herman Tolle (2023)

Penelitian ketiga yang berjudul “Pengembangan Aplikasi Pembelajaran Bahasa Bali berbasis Android menggunakan MVVM *Architecture* dan *Jetpack Compose*” yang disusun oleh I Putu Reditya Permana Putra dan Herman Tolle. Penelitian ini menyoroti pentingnya bahasa Bali sebagai penopang budaya, adat istiadat, dan agama Hindu di Bali, serta kekhawatiran mengenai penurunan penggunaan bahasa Bali di kalangan generasi muda akibat meningkatnya penggunaan bahasa Indonesia dalam komunikasi rumah tangga. Untuk mengatasi potensi penurunan eksistensi bahasa Bali, langkah preventif seperti pengajaran bahasa Bali di sekolah-sekolah sebagai muatan lokal telah diambil. Penelitian ini menghasilkan aplikasi pembelajaran bahasa Bali yang dibangun menggunakan *Jetpack Compose* dan Arsitektur MVVM (*Model-View-ViewModel*). *Jetpack Compose* dipilih karena dapat membuat ukuran aplikasi lebih kecil dibandingkan dengan penggunaan XML, sementara Arsitektur MVVM dipilih karena lebih efisien dalam penggunaan CPU dan kecepatan eksekusi. Aplikasi ini diharapkan dapat mempermudah pengguna dalam mempelajari bahasa Bali melalui pembelajaran, latihan soal, dan kamus bahasa Bali.[[10]](#footnote-10)

1. Irya Muhammad Riyadhi (2023)

Penelitian keempat yang berjudul “Penerapan Pola Arsitektur MVVM Pada Perancangan Aplikasi Pengaduan Masyarakat Berbasis Android” yang disusun oleh Irya Muhammad Riyadhi. Penelitian ini bertujuan untuk merancang aplikasi pengaduan masyarakat yang memungkinkan koordinasi mudah antara masyarakat dan pemerintah desa terkait permasalahan kebersihan lingkungan. Aplikasi ini memungkinkan masyarakat untuk membuat laporan kebersihan yang akan ditindaklanjuti oleh pemerintah desa. Dan Perancangan aplikasi ini menggunakan pola arsitektur *Model-View-ViewModel* (MVVM) untuk memastikan kode aplikasi terstruktur, mudah dirawat, dan mempermudah akses data serta mempertahankan keadaan dari perubahan konfigurasi pada smartphone pengguna. MVVM memecah kode aplikasi menjadi tiga komponen utama: *Model*, *View*, dan *ViewModel*.[[11]](#footnote-11)

1. Cahya Diantoni, Oman Komarudin, Adhi Rizal (2024)

Penelitian kelima yang berjudul “Arsitektur MVVM dan Framework Jetpack Compose Pada Pengembangan Aplikasi Android” yang disusun oleh Cahya Diantoni, Oman Komarudin, Adhi Rizal. Penelitian ini berfokus pada pengembangan Sukacolab, sebuah platform berbasis aplikasi Android yang dirancang untuk mahasiswa Fasilkom Unsika. Sukacolab bertujuan untuk memfasilitasi diskusi, pencarian rekan kerja, dan kolaborasi dalam proyek-proyek TI. Dengan memilih Android sebagai platform, aplikasi ini memanfaatkan popularitas smartphone Android di kalangan mahasiswa, sehingga memberikan akses yang mudah dan pengalaman pengguna yang konsisten dan responsif. Salah satu aspek penting dalam pengembangan aplikasi Android adalah pemilihan pola arsitektur yang tepat. Penelitian ini memilih *Model-View-ViewModel* (MVVM) dan *Jetpack Compose* sebagai pendekatan utama. MVVM memisahkan logika bisnis dari antarmuka pengguna, sementara *Jetpack Compose* memungkinkan pengembang untuk mendefinisikan antarmuka pengguna secara dinamis dan menarik menggunakan bahasa pemrograman Kotlin.[[12]](#footnote-12)

Berikut adalah ringkasan penelitian terdahulu dan rencana penelitian yang akan saya lakukan yang tertera pada Tabel 2.1:

Tabel 2. 1 Tabel Penelitian Terdahulu

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Peneliti** | **Tahun** | **Judul** | **Tujuan Penelitian** | **Persamaan** | **Perbedaan** |
| 1 | Muhammad Syakir Arif, Aziz Musthafa, Dihin Muryatmoko | 2019 | Implementasi Pola Arsitektur Model-View-ViewModel (MVVM) pada Sistem Informasi Akademik Universitas Darussalam Gontor Berbasis Mobile | Tujuan penelitian ini adalah untuk mengatasi kompleksitas dan perubahan perangkat lunak pada ponsel pintar dengan menggunakan pola arsitektur *Model-View-ViewModel* (MVVM) guna meningkatkan efisiensi dan keteraturan kode, dengan fokus pada pengguna Android di Indonesia. | Menggunakan MVVM sebagai arsitektur android. | menggunakan xml sebagai UI toolkit. |
| 2 | Galuh Wilujeng Saraswati, Nanang Febrianto | 2021 | Penerapan Arsitektur MVVM Pada Aplikasi Tanamin Untuk Mendeteksi Penyakit Tanaman Berbasis Android | Tujuan penelitian ini adalah untuk mendukung sektor pertanian di Indonesia dengan mengembangkan aplikasi berbasis Android menggunakan arsitektur *Model-View-ViewModel* (MVVM) yang efisien dan berperforma tinggi, guna memisahkan tampilan aplikasi dari logika bisnis dan meningkatkan kinerja aplikasi. | Menggunakan MVVM sebagai arsitektur android. | menggunakan xml sebagai UI toolkit. |
| 3 | Irya Muhammad Riyadhi | 2023 | Penerapan Pola Arsitektur MVVM Pada Perancangan Aplikasi Pengaduan Masyarakat Berbasis Android | Tujuan penelitian ini adalah merancang aplikasi pengaduan masyarakat untuk memfasilitasi koordinasi antara masyarakat dan pemerintah desa terkait masalah kebersihan lingkungan, menggunakan arsitektur *Model-View-ViewModel* (MVVM) untuk memastikan kode yang terstruktur, mudah dirawat, dan efisien. | Menggunakan MVVM sebagai arsitektur android. | menggunakan xml sebagai UI toolkit. |
| 4 | Putu Reditya Permana Putra, Herman Tolle | 2023 | Pengembangan Aplikasi Pembelajaran Bahasa Bali berbasis Android menggunakan MVVM Architecture dan Jetpack Compose. | Tujuan penelitian ini adalah untuk mengembangkan aplikasi pembelajaran bahasa Bali menggunakan Jetpack Compose dan Arsitektur MVVM guna mendukung pelestarian bahasa Bali di kalangan generasi muda melalui pengajaran yang efektif dan efisien. | Menggunakan MVVM sebagai arsitektur android dan jetpack compose sebagai UI toolkit. | - |
| 5 | Cahya Diantoni, Oman Komarudin, Adhi Rizal | 2024 | Arsitektur MVVM dan Framework Jetpack Compose Pada Pengembangan Aplikasi Android | Tujuan penelitian ini adalah mengembangkan Sukacolab, aplikasi Android untuk mahasiswa Fasilkom Unsika, guna memfasilitasi diskusi dan kolaborasi proyek TI, dengan menggunakan arsitektur MVVM dan Jetpack Compose. | Menggunakan MVVM sebagai arsitektur android dan jetpack compose sebagai UI toolkit. | - |

1. Landasan Teori
2. Aplikasi Mobile

Aplikasi *mobile* adalah perangkat lunak yang dirancang khusus untuk dijalankan pada perangkat seluler seperti *smartphone* dan *tablet*. Aplikasi ini dapat memberikan beragam fungsi dan layanan kepada pengguna, mulai dari komunikasi, hiburan, produktivitas, hingga transaksi keuangan. Dengan kemajuan teknologi yang pesat, aplikasi *mobile* menjadi semakin penting dalam kehidupan sehari-hari, memudahkan pengguna dalam mengakses informasi, berinteraksi dengan orang lain, dan menjalankan berbagai aktivitas dengan cepat dan efisien.[[13]](#footnote-13)

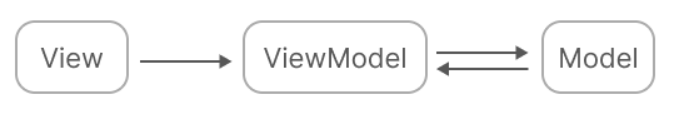
1. Android

*Android* adalah OS *(operation system)* yang dibuat berbasis *linux*. Sistem operasi *Android* diperuntukkan untuk perangkat bergerak atau *mobile* seperti *smartphone* atau ponsel pintar, juga *tablet*. Selain itu dalam perkembangannya, *Android* juga digunakan sebagai sistem operasi untuk antarmuka pada dashboard mobil dan sebagai sistem operasi untuk televisi. Selain itu pihak *Android* menyediakan platform terbuka atau open- source bagi para developer agar dapat mengembangkan sistem operasi kustom buatan sendiri sehingga membuat *Android* menjadi sistem operasi yang paling populer.[[14]](#footnote-14) *android* menjadi sistem operasi yang paling banyak digunakan di antara sekian banyak pilihan sistem operasi untuk perangkat *mobile*, kebutuhan terhadap suatu konsep dan sistem pembelajaran IT tidak terelakkan lagi.[[15]](#footnote-15)

1. Website Hukum (peraturan.co.id)

Peraturan.co.id adalah Website Direktorat Jenderal Peraturan Perundang-undangan yang menampilkan Database Peraturan Perundang-undangan yang memuat informasi mengenai jenis, status, hubungan antar peraturan, dan statistik peraturan perundang-undangan. Fitur utama dari situs ini meliputi pencarian peraturan berdasarkan jenis (Undang-Undang, Peraturan Pemerintah, dll.), kategori, serta tahun terbit. Situs ini juga memungkinkan pengguna untuk mengunduh dokumen peraturan, memberikan kemudahan akses terhadap regulasi yang berlaku di Indonesia, serta mendukung transparansi hukum. Terdapat pula informasi tambahan seperti naskah akademik dan materi hukum lainnya yang relevan.

1. Model – View – ViewModel (MVVM)

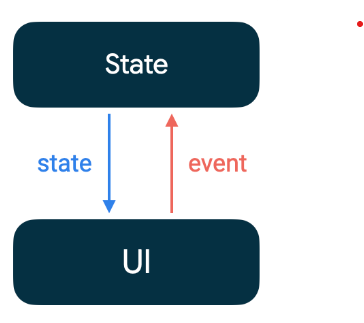
*Model-View-ViewModel* (MVVM) adalah salah satu dari pola arsitektur (*architecture pattern*) yang mengusung pemisahan kepentingan, yang mana hal tersebut memungkinkan untuk memisahkan antara grafis antarmuka (*graphical user interface*) dari proses logika bisnis atau logika halaman belakang (*back-end logic*).[[16]](#footnote-16) *Model-View-ViewModel* (MVVM), yaitu sebuah desain arsitektur yang bertujuan untuk membuat arsitektur menjadi lebih terpisah antara antarmuka dengan domain layer-nya.[[17]](#footnote-17) MVVM adalah salah satu arsitektur pembuatan aplikasi berbasis GUI yang berfokus pada pemisahan antara kode untuk logika bisnis dan tampilan aplikasi. Dalam penerapannya, MVVM terbagi atas beberapa layer, yaitu Model, View, dan ViewModel. Simak pembahasan ketiga layer tersebut di bawah ini.

Gambar 2. 1 Pola Arsitektur MVVM

Model, Layer ini adalah model atau entitas yang merepresentasikan data yang akan digunakan pada logika bisnis. Umumnya kelas-kelas yang ada di dalamnya berupa POJO atau Plain Old Java Object dan Data Classes jika kita menggunakan Kotlin. View, Tidak seperti layer sebelumnya, layer ini berisi UI dari aplikasi untuk mengatur bagaimana informasi akan ditampilkan. Layer ini akan berisi kelas-kelas, seperti Activity dan Fragment. ViewModel, Layer terakhir adalah ViewModel yang bertugas untuk berinteraksi dengan model di mana data yang ada akan diteruskan ke layer view.[[18]](#footnote-18)

1. Jetpack Compose

*Jetpack Compose* adalah *toolkit UI* modern untuk membuat *UI Android* *native*. *Jetpack compose* mempercepat dan menyederhanakan pengembangan *UI* di *Android*. Ini juga dengan cepat menghidupkan aplikasi dengan lebih sedikit kode.[[19]](#footnote-19) *Jetpack Compose* adalah *toolkit* pengembangan antarmuka pengguna (UI) yang dipelajari untuk aplikasi Android, memungkinkan pengembang untuk membuat tampilan UI dengan kode *Kotlin* secara deklaratif dan responsif. Ini mengubah cara pengembang membuat dan 0mengelola *UI Android*, meningkatkan produktivitas dan fleksibilitas dalam pengembangan aplikasi.[[20]](#footnote-20) Berikut adalah arah flow data pada jetpack compose:



Gambar 2. 2 Arah Flow Data Jetpack Compose

Berikut penjelasan dari flow di atas:

* Event: Bagian dari UI menghasilkan peristiwa dan meneruskannya ke atas, seperti klik tombol yang diteruskan ke ViewModel untuk ditangani; atau peristiwa diteruskan dari lapisan aplikasi Anda yang lain, seperti mengindikasikan bahwa sesi pengguna telah berakhir.
* Update State: Penangan peristiwa dapat mengubah status.
* Display State: Pemegang state meneruskan state, dan UI menampilkannya.

Bab 3 Metode Penelitian

1. Rencana Kegiatan

Adapun tempat penelitian ini dilakukan di UNIDA Gontor dan di mulai sejak bulan Maret hingga September 2024 dengan rincian seperti pada Table 3.1.

Tabel 3. 1 Tabel Rencana Kegiatan

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Kegiatan** | **Waktu Pelaksanaan (2024)** | | | | | | |
| **Maret** | **April** | **Mei** | **Juni** | **Juli** | **Agustus** | **September** |
| 1 | Analisi Kebutuhan |  |  |  |  |  |  |  |
| 2 | Tahap Pengumpulan Data |  |  |  |  |  |  |  |
| 3 | Desain Aplikasi |  |  |  |  |  |  |  |
| 4 | Pembuatan Api Pada Postman |  |  |  |  |  |  |  |
| 5 | Perancangan Aplikasi |  |  |  |  |  |  |  |
| 6 | Pengujian Aplikasi |  |  |  |  |  |  |  |
| 7 | Evaluasi |  |  |  |  |  |  |  |
| 8 | Penulisan Laporan |  |  |  |  |  |  |  |

1. Alat Dan Bahan

Alat dan bahan yang digunakan dalam penelitian ini meliputi beberapa aspek penting yang mendukung kelancaran proses penelitian. Adapun alat dan bahan sebagai berikut:

1. Perangkat Keras

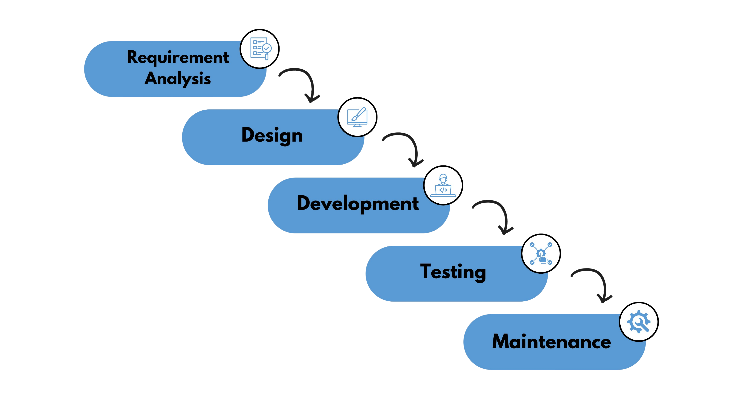
* Leptop HP Victus dengan spesifikasi AMD Ryzen 7 5800H dengan Radeon Graphics, RAM 16 GB, SSD 512 GB.
* Hp Samsung A50S.
* Kabel USB Type C.
* Printer.

1. Perangkat Lunak

* Sistem Operasi Windows 11 Home 64-bit.
* Android Studio Versi Iguana.
* Mendeley Desktop.
* Figma.
* Postman.
* Microsoft Office 2021.
* Canva.
* Whimsical.

1. Tahapan Penelitian

Tahapan pada penelitian yang di gunakan untuk mengembangkan aplikasi adalah metode *Software Development Life Cycle* (SDLC) dengan model *waterfall*. Metode *waterfall* adalah metode kerja yang menekankan fase-fase yang berurutan dan sistematis. Dan juga contoh proses yang digerakkan oleh rencana. Pada prinsipnya, peneliti merencanakan dan menjadwalkan semua aktivitas proses sebelum memulai pengembangan perangkat lunak.[[21]](#footnote-21) Disebut *waterfall* karena proses mengalir satu arah “ke bawah” seperti air terjun. Metode *waterfall* ini harus dilakukan secara berurutan sesuai dengan tahap yang ada. Berikut adalah Metode *Waterfall* yang tertera pada Gambar 3.1.



Gambar 3. 1 Tahapan Penelitian dengan Metode Waterfall

1. Analisis Kebutuhan (Requirements)

Pada tahap analisis kebutuhan, peneliti memulai studi dengan menyebarkan kuesioner kepada pengguna yang berumur 17 – 40 tahun untuk mengukur pemahaman mereka tentang hukum dan akses terhadap bantuan hukum. Pada tahap ini, peneliti menganalisis keluhan yang disampaikan oleh pengguna dan mengumpulkan data pendukung terkait permasalahan yang dihadapi. Setelah itu, solusi untuk permasalahan tersebut dicari. Hasil analisis menunjukkan bahwa sekitar kurang lebih 67,3% dari 107 responden dari kuesioner yang di sebar mengetahui cara mendapatkan bantuan hukum dari para ahli akan tetapi tidak tahu detail cara mendapatkannya. Berdasarkan temuan ini, pengguna sangat mendukung pengembangan aplikasi hukum berbasis *mobile* yang diharapkan dapat membantu dalam memahami hukum dan memperoleh bantuan hukum.

1. Desain Aplikasi (Design)

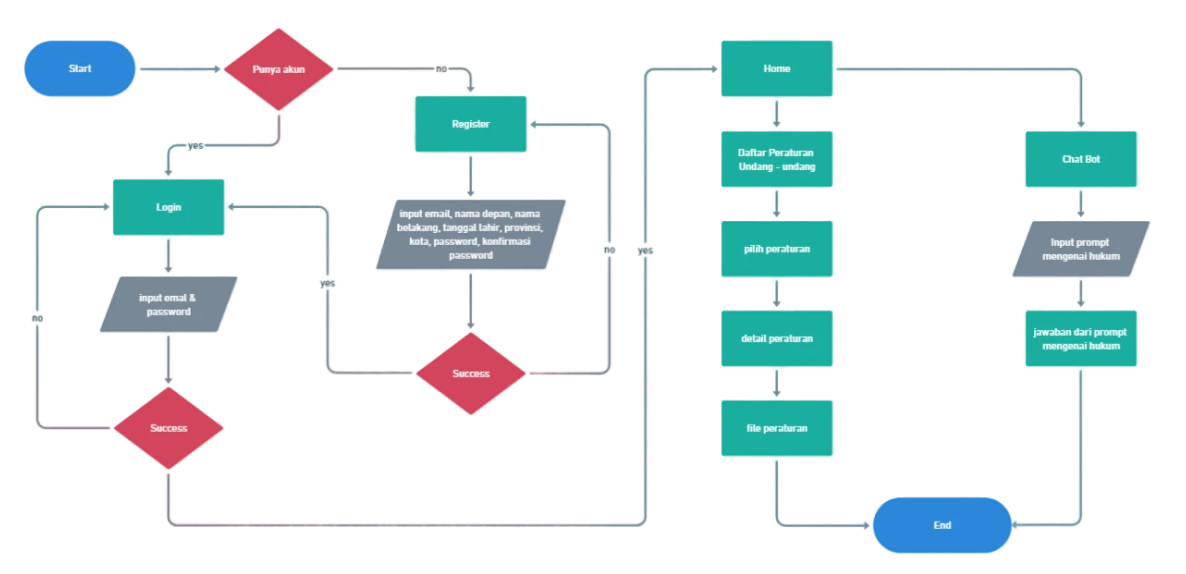
Pada tahap *design*, dilakukan setelah analisis masalah dan kebutuhan pengguna selesai. Sebelum memulai proses pengkodean, sangat penting untuk melaksanakan langkah-langkah analisis sistem dan desain aplikasi terlebih dahulu. Dalam tahap perancangan desain, terdapat beberapa elemen yang perlu dimasukkan untuk memandu pengembangan aplikasi secara efektif dan efisien. Elemen-elemen tersebut meliputi pembuatan *Flow Chart* yang akan menggambarkan alur kerja sistem secara keseluruhan, *Use Case* diagram yang akan mengidentifikasi berbagai skenario penggunaan aplikasi oleh pengguna, *Activity diagram, Sequence diagram,* dan *Mock Up* yang memberikan gambaran visual tentang tampilan antarmuka pengguna, serta pemetaan aplikasi yang akan menjelaskan struktur dan hubungan antar komponen dalam aplikasi. Dengan adanya elemen-elemen ini, proses pengembangan aplikasi akan menjadi lebih terarah dan terstruktur, sehingga menghasilkan aplikasi yang sesuai dengan kebutuhan dan harapan pengguna. Berikut adalah rancangan aplikasi yang tertera pada Tabel 3.2

Tabel 3. 2 Rancangan Aplikasi

|  |  |  |
| --- | --- | --- |
| No | Nama Menu | Keterangan |
| 1 | Login dan Register | Halaman user untuk malakukan login dan pembuatan akun. |
| 2 | Home Screen | Tampilan yang berisi beberapa menu seperti daftar undang – undang dan Chat Bot Peraturan undang -undang |
| 3 | Daftar Peraturan Undang - Undang | Berisi tentang jenis peraturan undang – undang beserta detail dan file undang – undang tersebut. |
| 4 | Chat Bot Peraturan Undang - Undang | Tampilan Chat Bot untuk user menanyakan masalah mengenai hukum. |

1. Flowchart

Berikut ini adalah Flowchart untuk aplikasi hukum yang akan dikembangkan pada penelitian ini yang tertera pada Gambar 3.2.

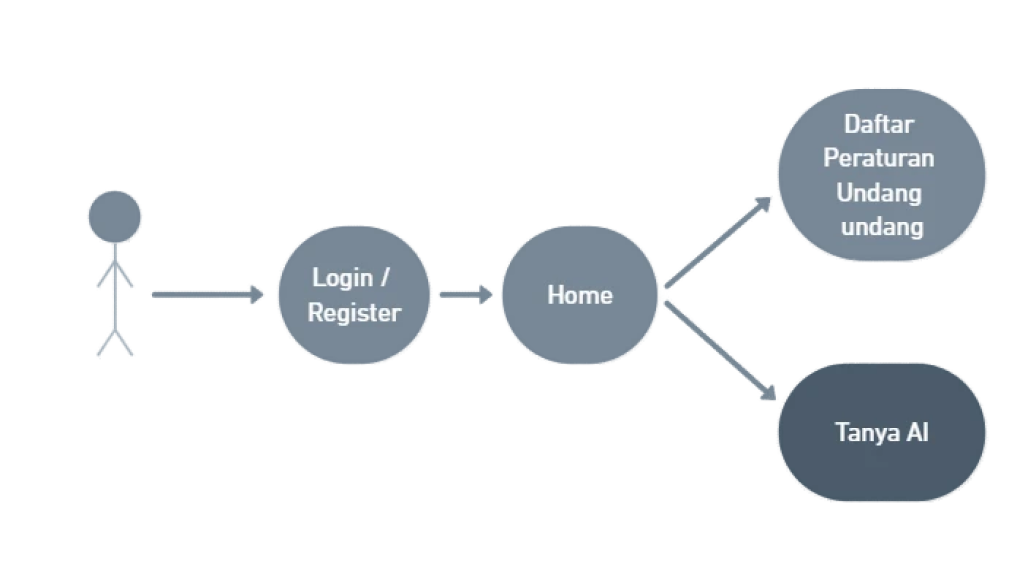
* Flowchart Aplikasi Keseluruhan

Gambar 3. 2 Flowchart Aplikasi

Penjelasan dari algoritma Flowchart tersebut adalah sebagai berikut :

* Pertama user akan masuk kedalam halaman login, user melakukan login jika sudah memiliki akun, dan jika blom memiliki akun user diwajibkan untuk melakukan registrasi.
* Kedua, Ketika sudah melakukan login, user akan diarahkan ke halaman home yang mana terdapat dua fitur utama yaitu: fitur daftar peraturan undang – undang dan fitur *Chat Bot* Peraturan Undang - undang.
* Pada fitur daftar Peraturan Undang – undang, didalamnya terdapat beberapa Daftar Peraturan Undang – undang dan Ketika user memilih salah satu peraturan tersebut maka user akan di bawa ke halaman detail peraturan.
* Pada fitur Chat Bot Peraturan Undang - Undang, user dapat menanyakan permasalahan mengenai hukum dan langsung akan mendapatkan respon dari pertanyaan tersebut. Dan jika pertanyaan di luar permasalahan hukum maka pertanyaan tersebut tidak akan mendapatkan respon.

1. Use Case Diagram

Berikut adalah alur *Use Case Diagram* yang tertera pada Gambar 3.3.

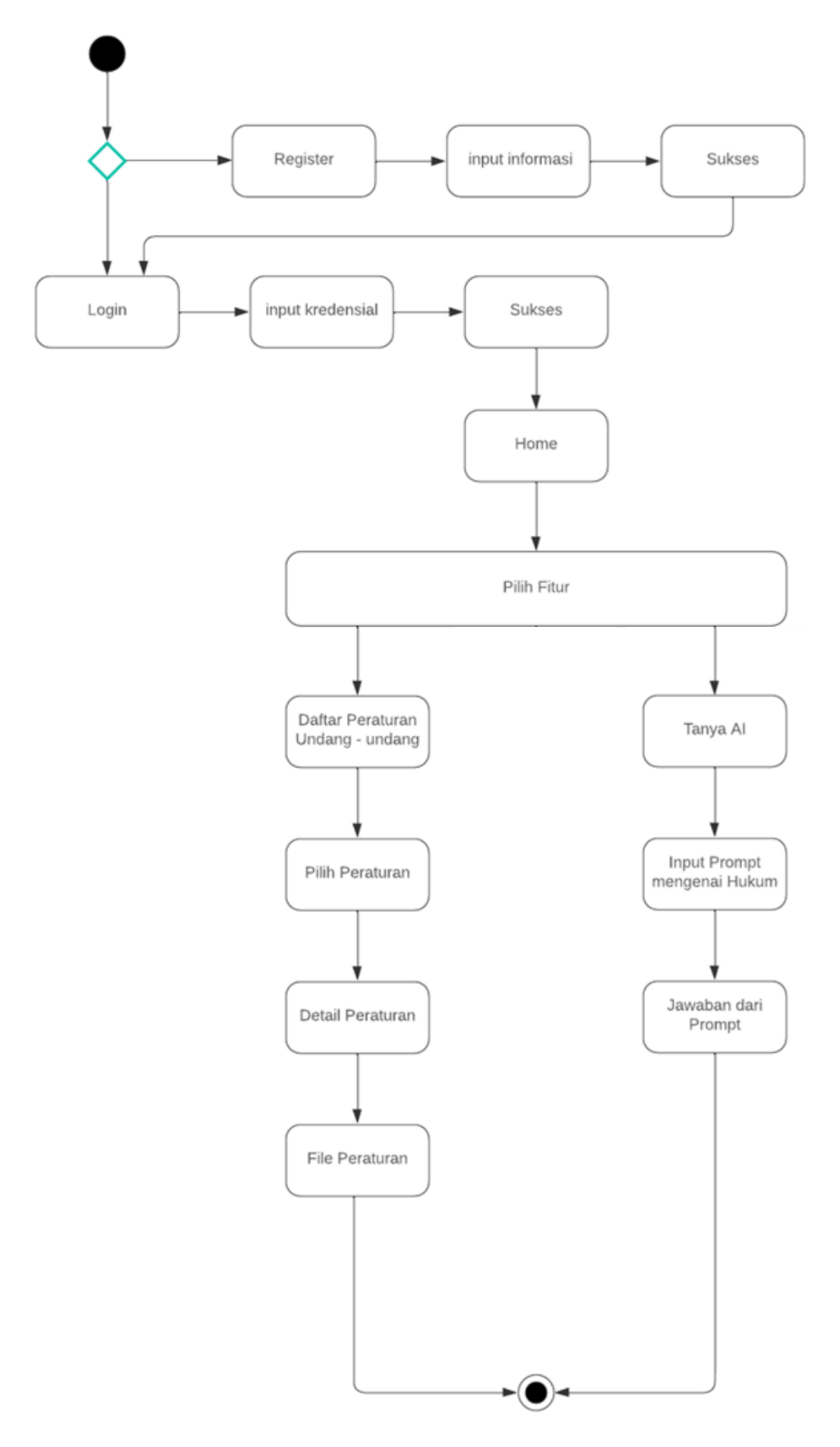
Gambar 3. 3 Use Case Diagram

Berikut adalah penjelasan mengenai interaksi pengguna dengan system

dengan *use case* diagram :

* Pengguna (User): Digambarkan sebagai figur user di sebelah kiri. Ini mewakili user yang akan menggunakan sistem.
* Login/Register: Langkah pertama yang dilakukan pengguna adalah masuk atau mendaftar ke sistem melalui fitur "Login/Register". Ini adalah tahap autentikasi pengguna.
* Home (Halaman Utama): Setelah pengguna berhasil masuk atau mendaftar, mereka akan diarahkan ke halaman utama (*Home*). Ini adalah pusat dari semua navigasi dalam sistem.
* Dari halaman utama, pengguna memiliki dua opsi:
* Daftar Peraturan Undang-undang: Pengguna dapat memilih untuk melihat daftar peraturan perundang-undangan yang tersedia di sistem.
* Chat Bot Peraturan Undang - undang: Pengguna juga memiliki opsi untuk bertanya kepada sistem AI terkait hal-hal tertentu.

1. Activity Diagram

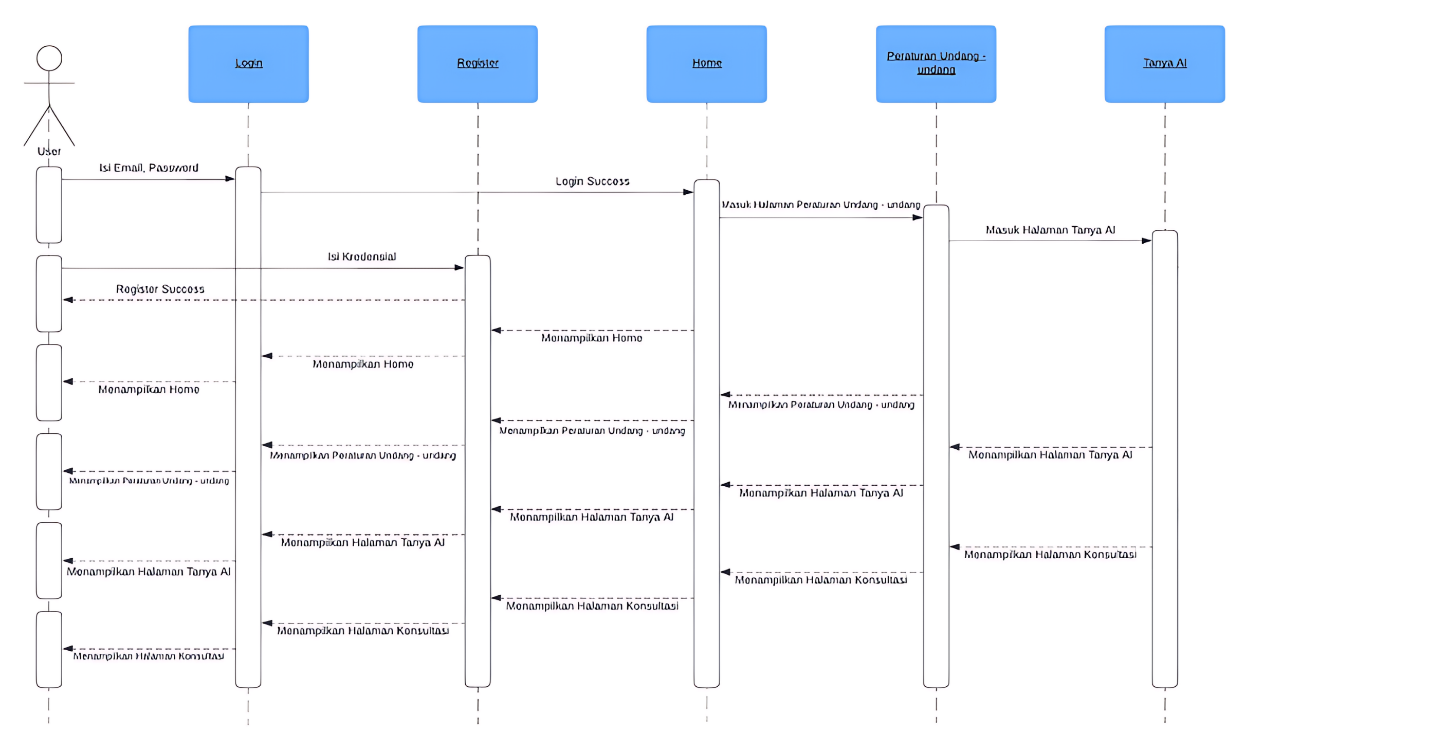
Berikut adalah alur *Activity Diagram* yang tertera pada Gambar 3.4.

Gambar 3. 4 Activity Diagram

Berikut adalah penjelasan alur aktivitas dengan *activity* diagram:

* Mulai (Titik hitam): Diagram dimulai dari titik hitam di bagian kiri atas, yang menandakan awal dari aktivitas.
* Keputusan Login atau Register (Bentuk wajik hijau): Pengguna harus memilih antara dua opsi:
* Register: Jika pengguna belum terdaftar, mereka akan memilih untuk mendaftar.
  + Input Informasi: Pengguna akan mengisi informasi pendaftaran.
  + Sukses: Setelah mendaftar dengan sukses, mereka akan diarahkan ke halaman utama (*Home*).
* Login: Jika pengguna sudah memiliki akun, mereka memilih untuk login.
  + Input Kredensial: Pengguna memasukkan kredensial (username dan password).
  + Sukses: Jika login berhasil, pengguna akan diarahkan ke halaman utama (*Home*).
* Home (Halaman Utama): Setelah login atau mendaftar dengan sukses, pengguna akan masuk ke halaman utama.
* Pilih Fitur: Di halaman utama, pengguna memiliki dua opsi fitur utama yang bisa dipilih:
  + Daftar Peraturan Undang-Undang: Pengguna dapat memilih untuk melihat daftar peraturan yang tersedia.
    - * Pilih Peraturan: Pengguna memilih peraturan yang diinginkan.
      * Detail Peraturan: Pengguna melihat detail dari peraturan yang dipilih.
      * File Peraturan: Pengguna dapat mengakses file peraturan yang ada.
  + Chat Bot: Pengguna bisa menggunakan fitur ini untuk bertanya terkait hukum.
    - Input Prompt mengenai Hukum: Pengguna memasukkan pertanyaan atau prompt terkait hukum.
    - Jawaban dari Prompt: Sistem Chat Bot akan memberikan jawaban berdasarkan prompt yang diinputkan.
* Akhir (Titik hitam dengan lingkaran putih): Setelah pengguna menyelesaikan salah satu dari dua fitur (melihat peraturan atau menggunakan Chat bot), diagram berakhir.

1. Sequence Diagram

Berikut adalah alur *Sequence Diagram* yang tertera pada Gambar 3.5

Gambar 3. 5 Sequence Diagram

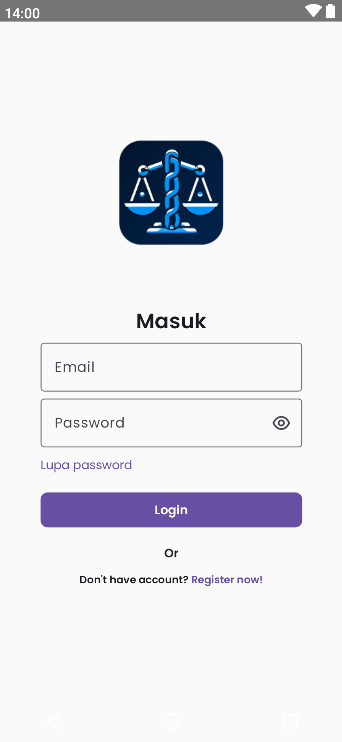
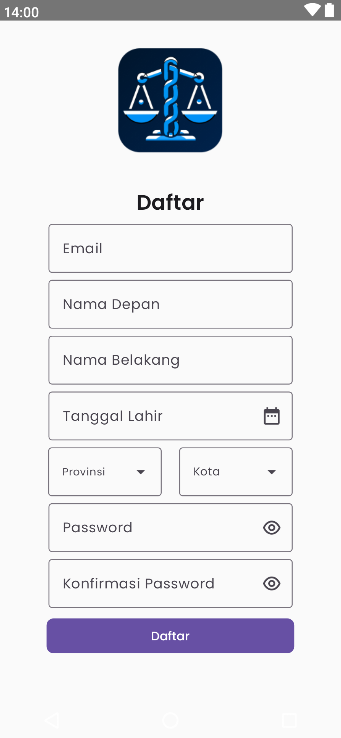
Berikut adalah penjelasan alur aktivitas dengan *activity* diagram:

* Register Success dan Login Success merupakan dua hasil dari proses autentikasi yang membawa pengguna ke halaman utama.
* Setiap entitas dalam diagram berkomunikasi secara langsung dengan pengguna atau halaman lainnya, yang diilustrasikan dengan panah solid.
* Sistem secara bergantian menampilkan halaman berdasarkan pilihan pengguna, seperti halaman *Home*, halaman *Peraturan Undang-undang*, dan halaman *Chat Bot*.

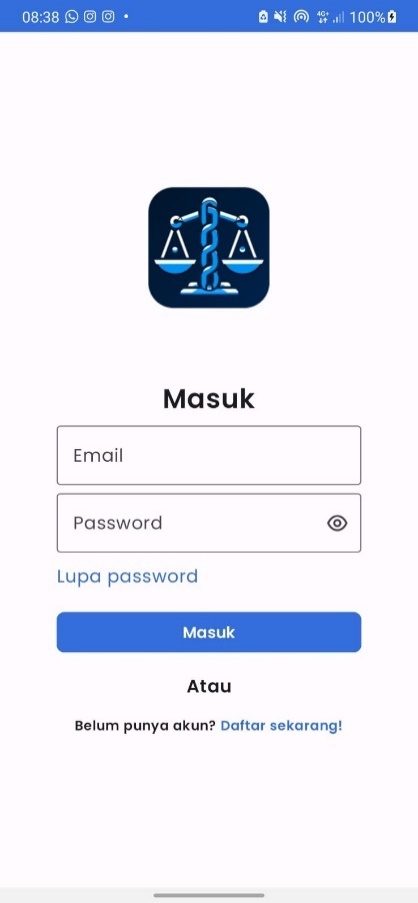
1. Mockup Aplikasi

* **Login dan Register**

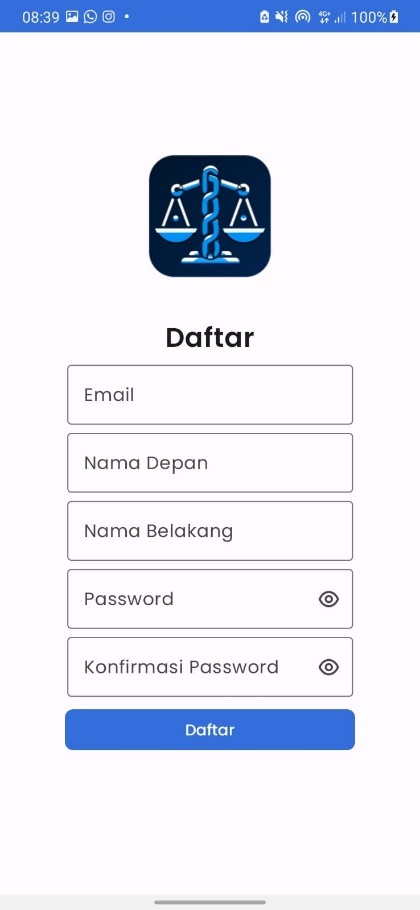
Pada halaman ini ketika pengguna pertama kali mengakses halaman utama aplikasi, mereka akan disambut dengan halaman login. Di sini, pengguna memiliki dua opsi: masuk dengan akun yang sudah ada atau membuat akun baru jika belum terdaftar. Proses login dirancang untuk mudah dan cepat, memungkinkan pengguna yang telah memiliki akun untuk segera mengakses semua fitur aplikasi dengan memasukkan kredensial mereka. Bagi pengguna yang belum memiliki akun, aplikasi menyediakan opsi registrasi yang mudah diikuti. Pengguna baru akan diminta untuk mengisi informasi dasar seperti nama, alamat email, dan kata sandi. Setelah proses registrasi selesai, pengguna akan menerima konfirmasi melalui email, yang memungkinkan mereka untuk mengaktifkan akun baru mereka dan mulai menggunakan aplikasi. Berikut halaman login yang tertera pada Gambar 3.6 dan halaman register pada Gambar 3.7.



Gambar 3. 6 Halaman Login

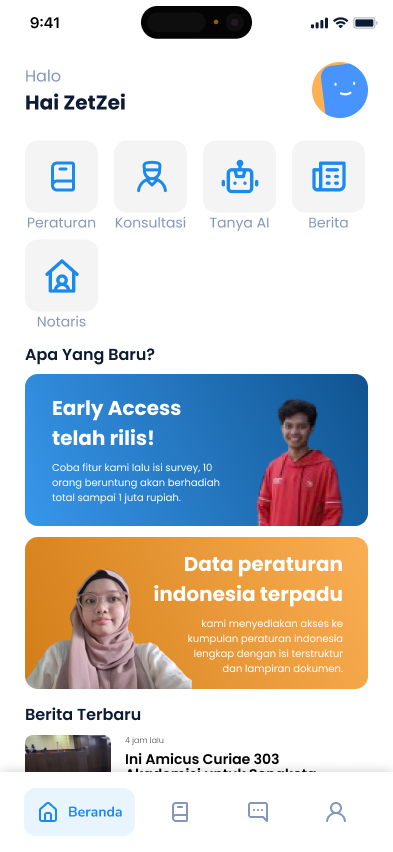


Gambar 3. 7 Halaman Register



* **Home Screen**

Pada halaman ini, pengguna akan menemukan berbagai menu yang dirancang untuk memenuhi kebutuhan hukum mereka. Menu-menu ini meliputi "Daftar Undang-Undang", "*Chat Bot*", dan yang lainnya. Setiap menu memiliki fungsi khusus yang membantu pengguna dalam memahami dan mengakses informasi hukum dengan lebih mudah. Berikut adalah halaman home yang tertera pada Gambar 3.8.



Gambar 3. 8 Halaman Home

* **Fitur Daftar Peraturan Undang – undang**

Halaman ini memberikan akses komprehensif terhadap peraturan undang-undang yang berlaku di Indonesia, lengkap dengan file detail setiap undang-undang tersebut. Pengguna dapat dengan mudah menemukan berbagai peraturan perundang-undangan. Setiap undang-undang dilengkapi dengan file detail yang berisi teks lengkap, penjelasan, serta interpretasi hukum yang relevan. Berikut adalah halaman peraturan yang tertera pada Gambar 3.9. pada halaman ini terdapat jenis peraturan undang – undang yang ditampilkan dalam bentuk list yang mana jika di klik, maka user akan di arahkan ke halaman detail peraturan.



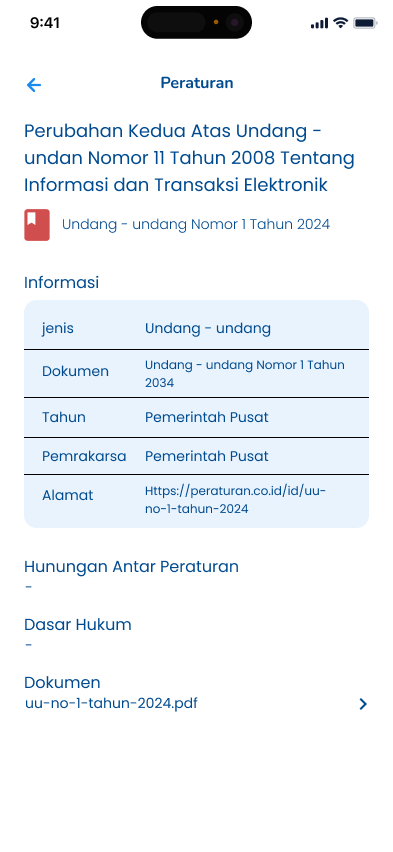
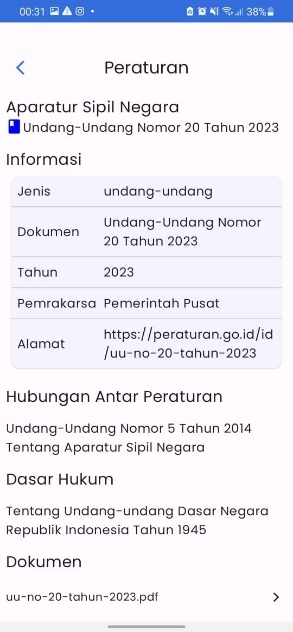
Gambar 3. 9 Halaman Peraturan

Dan selanjutnya halaman detail peraturan pada Gambar 3.10. melanjutkan dari halaman jenis peraturan undang – undang. Yang mana didalamnya terdapat macam – macam detail peraturan dari salah satu jenis peraturan yang dipilih.



Gambar 3. 10 Halaman List Peraturan

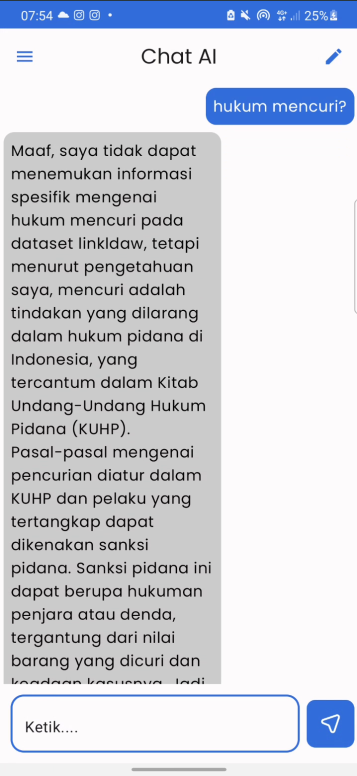
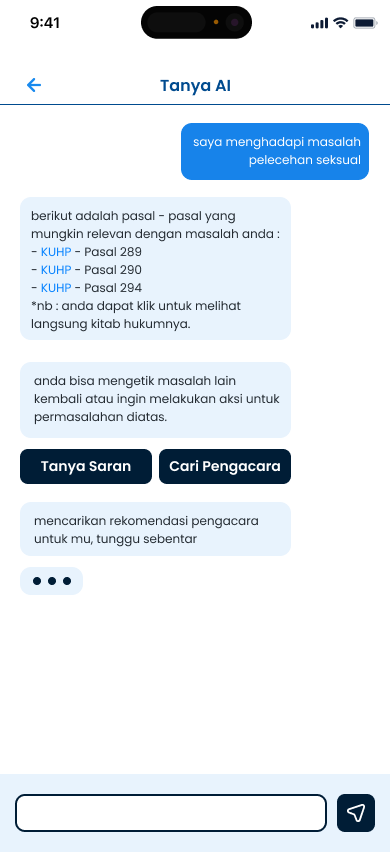
Kemudia halaman file peraturan pada Gambar 3.11. yang mana di dalamnya terdapat file lengkap peraturan undang – undang.



Gambar 3. 11 Halaman Detail Peraturan

* **Fitur Chat Bot**

Halaman *Chat Bot* ini dirancang untuk memudahkan pengguna dalam menanyakan berbagai masalah hukum yang mereka hadapi. Dengan antarmuka yang ramah pengguna dan intuitif, pengguna dapat langsung mulai berinteraksi dengan *Chat Bot* dengan hanya beberapa klik. *Chat Bot* ini memanfaatkan teknologi kecerdasan buatan untuk memahami pertanyaan pengguna mengenai hukum. Berikut adalah halaman *Chat Bot* yang tertera pada Gambar 3.12



Gambar 3. 12 Halaman Chat Bot

1. Implementation (Development)

Tahap implementasi merupakan langkah berikutnya setelah tahap desain dalam proses pengembangan aplikasi. Pada tahap ini, peneliti akan mengubah desain yang telah dirancang menjadi program fungsional sesuai dengan spesifikasi yang telah dianalisis dan direncanakan sebelumnya. Implementasi adalah fase kritis di mana konsep dan sketsa yang telah dibuat selama fase desain diubah menjadi kode yang berfungsi dan dapat dieksekusi. Dalam penelitian ini, peneliti menggunakan Android Studio sebagai lingkungan pengembangan terintegrasi (IDE) untuk mengembangkan aplikasi. Bahasa pemrograman yang digunakan untuk pengkodean adalah *Kotlin*. *Kotlin* dipilih karena memiliki berbagai keunggulan seperti sintaks yang lebih bersih, interoperabilitas dengan *Java*, dan dukungan yang kuat dari komunitas pengembang *Android* dan dengan implementasi pola arsitektur *Model – View – ViewModel* (MVVM) dan *Jetpack Compose* sebagai *UI Toolkit.*

1. Pengujian (Testing)

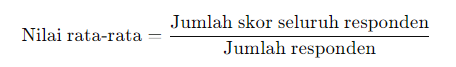
Verifikasi adalah langkah penting yang dilakukan setelah tahap pengkodean atau pembuatan aplikasi dan sebelum aplikasi tersebut diluncurkan. Pengujian ini dapat dilakukan oleh calon pengguna atau pakar dalam bidang terkait. Pada tahap ini, aplikasi akan diuji oleh tiga penguji:

1. Uji *Black Box*

Pada aplikasi ini *Black Box* Testing yang akan menguji dengan melakukan pengamatan hasil dari eksekusi melalui data uji dan cek fungsional peragkat lunak. *Black box testing* merupakan pengujian yang berfokus pada spesifikasi fugsional dari pengkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Blackbox testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.

1. Uji *Hardware*

Pengujian ini dilakukan untuk menguji fungsionaritas dari aplikasi pada beberapa *device* tertentu. Aplikasi akan diujikan pada beberapa *smartphone* dengan merk dan spesifikasi yang berbeda dengan tujuan untuk mengetahui apakah aplikasi dapat berjalan dengan baik atau tidak.

1. Ahli Materi, yaitu dengan mengujikan aplikasi kepada dosen dari program studi Hukum Ekonomi Syariah yang akan mengevaluasi aplikasi tersebut menggunakan kuesioner sebagai alat penilaian. Kuesioner ini terdiri dari pertanyaan – pertanyaan yang akan di uji dan setiap pertanyaan memiliki skala tanggapan yang memungkinkan responden untuk menunjukan Tingkat persetujuan atau ketidaksetujuan responden. Dengan rumus skala Likert berikut:

Dan berikut adalah Tabel rentang skala Likert untuk interpretasi hasil dengan skala 5 point.

Tabel 3. 3 Rentang Skala Likert

|  |  |
| --- | --- |
| Rentang Skor Rata - rata | Interpretasi |
| 1.00 – 1.80 | Sangat Tidak Setuju |
| 1.81 – 2.60 | Tidak Setuju |
| 2.61 – 3.40 | Netral |
| 3.41 – 4.20 | Setuju |
| 4.21 – 5.00 | Sangat Setuju |

1. Pengguna Aplikasi, yaitu pengujian oleh mahasiswa dan masyarakat yang berusia 17 – 40 tahun dengan cara mencoba dan menguji aplikasi serta memberikan evaluasi dengan menyebarkan kuesioner. Kuesioner ini dilakukan sama seperti pada pengujian ahli materi akan tetapi dengan skala 4 poin.
2. Pemeliharaan (Maintenance)

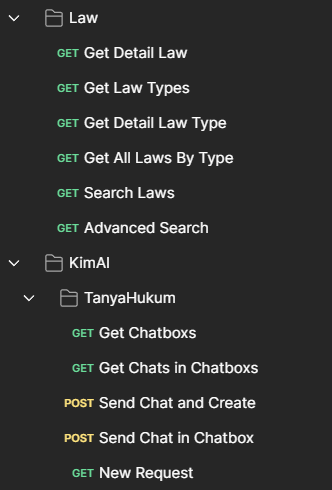
Pemeliharaan merupakan kegiatan yang melibatkan perawatan, modifikasi, atau pengembangan lebih lanjut dari aplikasi yang telah dibuat jika diperlukan. Tahap pemeliharaan ini adalah tahap akhir dalam siklus pengembangan aplikasi. Pada tahap ini, aplikasi yang telah melalui proses pengujian akan diperbaiki dan disempurnakan berdasarkan evaluasi dan temuan dari tahap sebelumnya. Pemeliharaan mencakup berbagai aktivitas seperti memperbaiki *bug* yang ditemukan setelah peluncuran serta meningkatkan fitur yang ada.

Bab 4 Hasil dan Pembahasan

Pada bagian ini, peneliti akan menjelaskan hasil dan pembahasan dari aplikasi hukum yang di buat, di mulain dari persiapan *rest api*, pembuatan *ui* aplikasi dengan menggunakan *framework jetpack compose* serta mengimplementasikan pola arsitektur *model-view-viewmodel* (MVVM) sebagai *logic* untuk menampilkan *model* kedalam *view* aplikasi.

1. Rest Api Pada Postman

Pada tahap ini peneliti menggunakan model dari rest api yang telah di buat oleh bagian back-end sehingga output hasil adalah sebuah rest api. Berikut adalah Rest Api pada postman yang digunakan untuk melakukan request data yang di dalamnya terdapat macam - macam method.



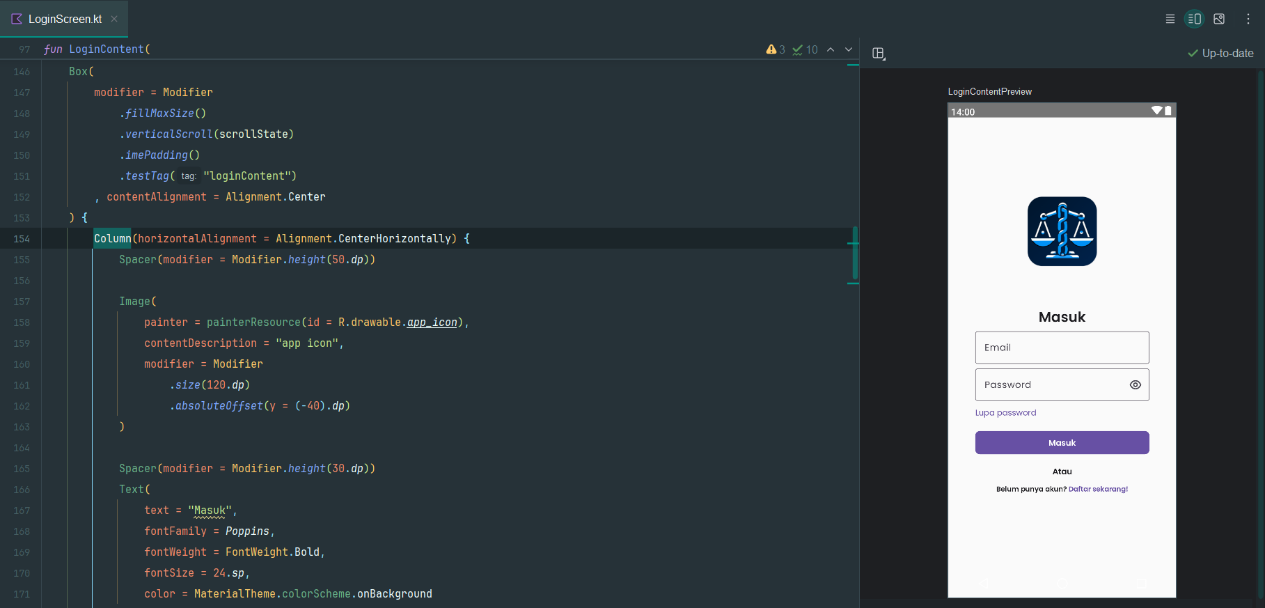
Gambar 4. 1 Rest Api

1. Implementasi Android Jetpack Compose

Berikut tahap – tahap dalam mengimplementasikan android jetpack compose pada setiap halaman aplikasi. *Jetpack compose* mempercepat dan menyederhanakan pengembangan *UI* di *Android*. Ini juga dengan cepat menghidupkan aplikasi dengan lebih sedikit kode.

1. Implementasi Android Jetpack Compose Pada Halaman Login

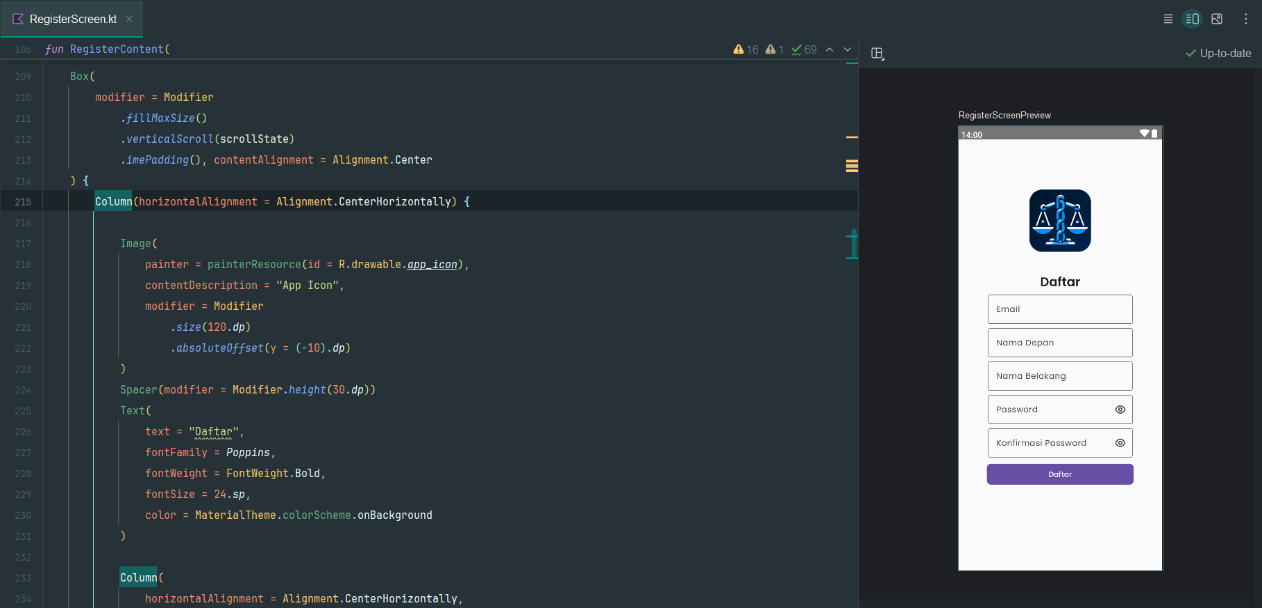
Pada Gambar 4.2 menggambarkan pengembangan antarmuka pengguna dengan menggunakan *android jetpack compose* pada halaman Login. Dimulai dari membuat wadah untuk menampung seluruh komponen dengan menggunakan Column()agar komponen di dalamnya tersusun secara vertical. Kemudian komponen pertama yang dibuat yaitu logo aplikasi dengan menggunakan Image(),kemudian dilanjutkan den gan dua inputan text field dengan menggunakan sintaks OutlinedTextField() untuk memasukkan email dan password, kemudian tombol button dengan text ‘masuk’ dengan sintaks Button(),dan di bawah button terdapat text ‘Daftar Sekarang’ yang diberi *modifier* *clickable* yang di isi dengan fungsi onRegisterClick().

**

Gambar 4. 2 Implementasi Android Jetpack Compose pada Halaman Login

1. Implementasi Android Jetpack Compose Pada Halaman register

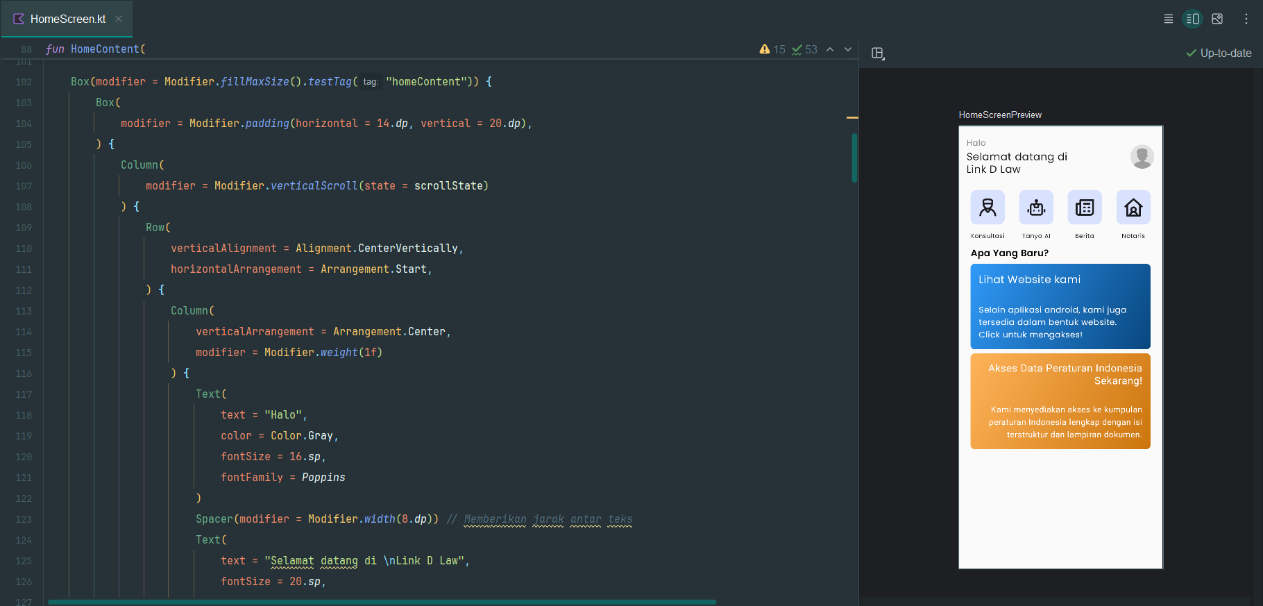
Pada Gambar 4.3 ketika pengguna diminta untuk melakukan *login* dan tidak memiliki akun. Maka pengguna dapat melakukan pendaftara pada halaman *register*. Pada Gambar ini menggambarkan penggembangan antarmuka pengguna dengan menggunakan *android jetpack compose* pada halaman *Register*. Dimulai dari mebuat wadah untuk menampung seluruh komponen menggunakan sintaks Column() agar seluruh komponen di dalamnya tersusun secara vertical, kemudian di lanjutkan dengan membuat Logo aplikasi menggunakan sintaks Image(), dan dilanjutkan dengan lima text field dengan menggunakan sintaks OutlinedTextField(). Pada text field pertama digunakan untuk input email, text field kedua digunakan untuk input nama depan, text field ketiga digunakan untuk input nama belakang, dan text field keempat dan kelima digunakan untuk input password dan konfirmasi password. Dan satu button dengan sintaks Button() untuk mendaftarkan data pengguna yang sudah di lengkapi sebelumnya.



Gambar 4. 3 Implementasi Android Jetpack Compose pada Halaman Register

1. Implementasi Android Jetpack Compose Pada Halaman Home

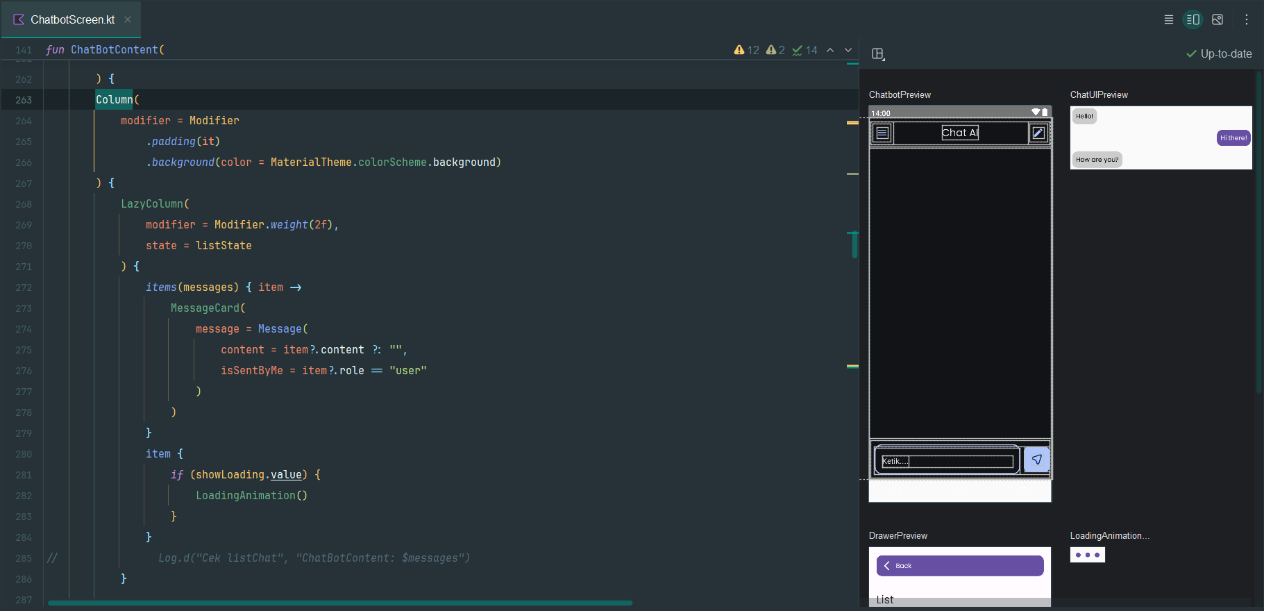
Pada Gambar 4.4 menggambarkan pengembangan antarmuka pengguna dengan menggunakan *android jetpack compose* pada halaman *Home*. Pada halaman home ini, layouting dibagi menjadi tiga bagian. Pada badian pertama terdapat terdapat Row() untuk menjadikan komponen di dalamnya tersusun secara horizontal. Dan kemudian di dalamnya terdapat Column() untuk menjadikan dua component di dalamnya tersusun secara vertical, dan dilanjutkan dengan sintaks Image() untuk profil foto sehingga berada di pojok kanan karena parent wadah sebelumnya yaitu Row(). Kemudian pada bagian layout kedua terdapat LazyRow() untuk menampilkan list secara horizontal yang di dalamnya terdapat empat komponen *icon* yaitu untuk konsultasi, *chat bot*, berita, dan notaris. Dan icon tiap list tersebut terdapat *modifier clickable()* yang di isi dengan fungsi onClick(). Kemudian pada layout bagian ketiga terdapat dua kartu dengan menggunakan sintaks Card(). Pada card pertama pengguna akan di arahkan ke website aplikasi hukum dan pada card kedua pengguna akan di arahkan ke halaman list peraturang undang – undang.



Gambar 4. 4 Implementasi Android Jetpack Compose pada Halaman Home

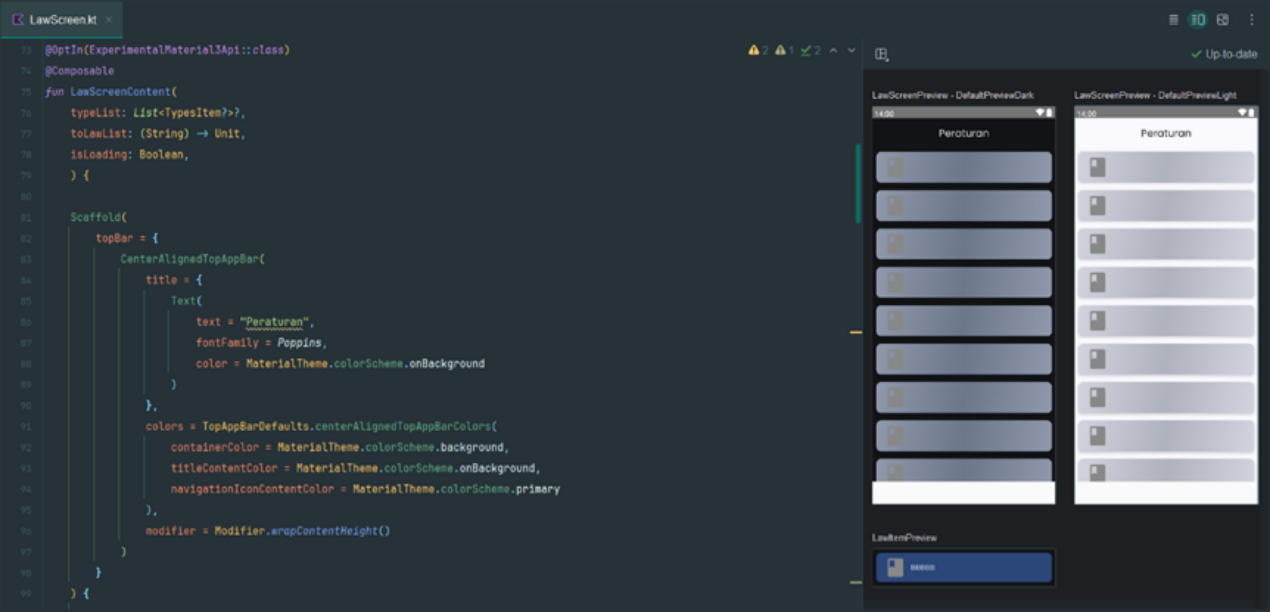
1. Implementasi Android Jetpack Compose Pada Halaman Chat bot

Pada gambar 4.5 menggambarkan pengembangan antarmuka pengguna dengan menggunakan *android jetpack compose* pada halaman *Chat Bot*. Tahap awal membuat desain bubble chat dengan menggunakan sintaks Card() untuk tempat text prompt yang di kirimkan. Kemudian membuat text field untuk memasukkan prompt yang ingin di input dengan menggunakan sintaks OutlinedTextField() dan button kirim dengan icon pesawat dengan sintaks IconButton().



Gambar 4. 5 Implementasi Android Jetpack Compose pada Halaman Chat Bot

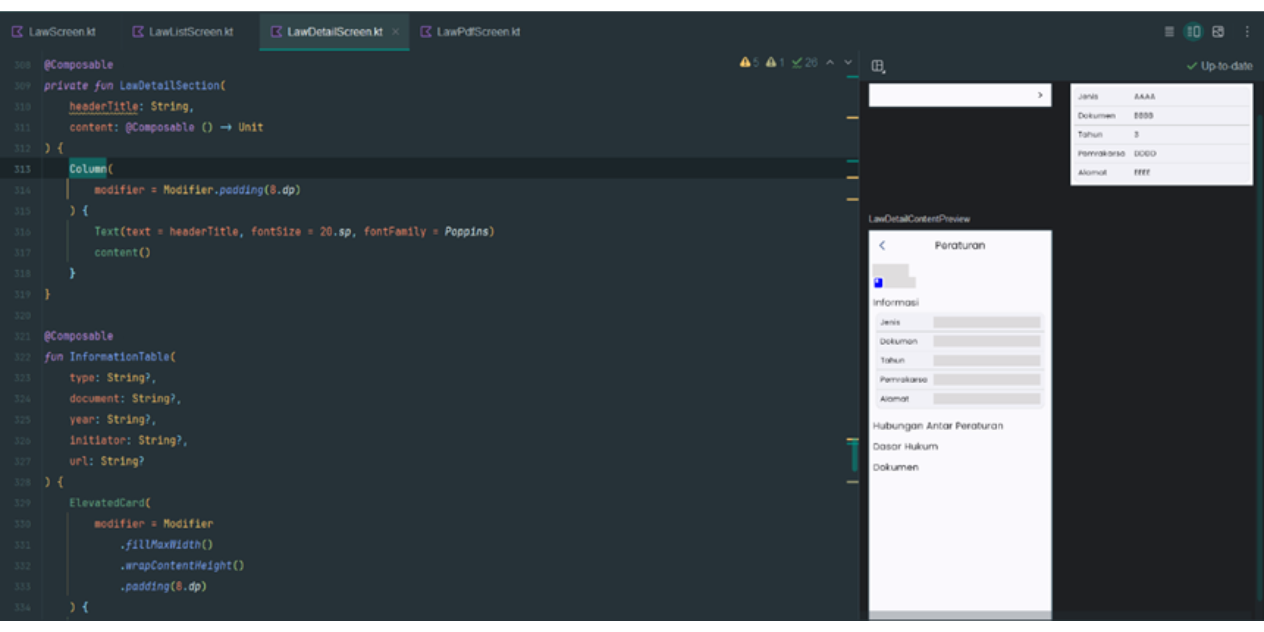
1. Implementasi Android Jetpack Compose Pada Halaman Peraturan

Pada Gambar 4.6 menggambarkan pengembangan antarmuka pengguna dengan menggunakan *android jetpack compose* pada halaman Peraturan. Dimulai dengan mebuat desain layout item yang ditampilkan dengan menggunakan sintaks ElevatedCard(), kemudian di panggil di dalam list dengan menggunakan LazyColumn() agar list yang ditampilkan tersusun secara vertical. Kemudian pada setiap item di isi dengan fungsi onClick() agar Ketika salah satu item di klik maka pengguna akan di arahkan ke halaman detail peraturan.

Gambar 4. 6 Implementasi Android Jetpack Compose pada Halaman Peraturan

1. Implementasi Android Jetpack Compose Pada Halaman Detail Peraturan

Pada Gambar 4.7 menggambarkan pengembangan antarmuka pengguna dengan menggunakan *android jetpack compose* pada halaman Detail Peraturan. Dimulai dari tahap pembuatan table untuk spesifikasi peraturan undang – undang yang di pilih oleh pengguna. Setelah layout dibuat table bisa di panggil di dalam halaman konten detail peraturan. Halaman detail peraturan ini terdapat file pdf yang nantinya dapat ditampilkan untuk pengguna. VerticalPDFReader() digunakan untuk menampilkan file pdf pada android studio.



Gambar 4. 7 Implementasi Android Jetpack Compose pada Halaman Detail Peraturan

1. Implementasi Model – View – ViewModel (MVVM)

Berikut tahap dalam mengimplementasikan pola model – view – viewmodel (MVVM) pada setiap fitur pada aplikasi. Pola ini sebagai pemisah antara kode untuk logika bisnis dan tampilan aplikasi. Dalam penerapannya, MVVM terbagi atas beberapa layer, yaitu Model, View, dan ViewModel.

1. Implementasi MVVM Untuk Menangani Logika Halaman Login

Pada kelas LoginViewModel, kelas ini mengelola logika login pengguna dengan menggunakan authRepository. Kelas ini memiliki dua LiveData untuk mengelola status loading dan status login. Fungsi login menggunakan coroutine viewModelScope untuk menjalankan operasi login secara asinkron. Saat login dimulai, loading diatur ke true. Jika login berhasil, status login dan informasi sesi pengguna (seperti token akses, token refresh, nama pengguna, dan email) disimpan menggunakan authRepository, dan loading diatur ke false. Jika terjadi kesalahan HTTP atau kesalahan lainnya selama login, status login diperbarui dengan pesan kesalahan dan loading diatur ke false. Dan fungsi resetLoginState mengatur ulang status login.

***class***LoginViewModel(*private val* authRepository: AuthRepository) : ViewModel() {  
  
 *fun* login(loginRequest: LoginRequest) {  
 *viewModelScope*.*launch* **{** \_loading.*value* = *true  
 try* {  
 *val* response =  
 authRepository.login(loginRequest)  
 \_loginStatus.postValue(  
 BasicResponse(  
 ok = response.ok,  
 message = response.message  
 )  
 )  
 authRepository.saveUserSession(  
 UserSession(  
 accessToken = accessToken,  
 refreshToken = refreshToken,  
 isLogin = isLogin,  
 name = userName,  
 email = userEmail  
 )  
 )  
 \_loading.*value* = *false* }   
 **}** }  
}

***class***LoginViewModel(*private val* authRepository: AuthRepository) : ViewModel() {  
  
 *fun* login(loginRequest: LoginRequest) {  
 *viewModelScope*.*launch* **{** \_loading.*value* = *true  
 try* {  
 *val* response =  
 authRepository.login(loginRequest)  
 \_loginStatus.postValue(  
 BasicResponse(  
 ok = response.ok,  
 message = response.message  
 )  
 )  
 authRepository.saveUserSession(  
 UserSession(  
 accessToken = accessToken,  
 refreshToken = refreshToken,  
 isLogin = isLogin,  
 name = userName,  
 email = userEmail  
 )  
 )  
 \_loading.*value* = *false* }   
 **}** }  
}

1. Implementasi MVVM Untuk Menangani Logika Halaman Register

Pada kelas RegisterViewModel ini digunakan untuk menangani proses registrasi pengguna di aplikasi Android. Ia mengelola status *loading* menggunakan LiveData, serta mengatur status registrasi dengan menyimpan hasilnya dalam objek BasicResponse. Saat fungsi register dipanggil, ViewModel akan melakukan pemanggilan API secara asinkron melalui authRepository untuk melakukan pendaftaran, dan menyimpan sesi pengguna (token akses, token refresh, nama, email) jika berhasil. Jika ada error, seperti kesalahan HTTP atau kesalahan lainnya, status registrasi akan diatur ke kondisi gagal dengan pesan kesalahan yang sesuai. Fungsi ini juga memiliki fitur untuk mengatur ulang status registrasi menggunakan resetRegisterState setelah proses selesai.

*class* RegisterViewModel(*private val* authRepository: AuthRepository): ViewModel() {

*fun* register(  
 registerRequest: RegisterRequest  
){  
 *viewModelScope*.*launch* **{** \_loading.*value* = *true  
 try* {  
 *val* response = authRepository.register(registerRequest)  
 \_registerStatus.*value* = BasicResponse(  
 ok = response.ok,  
 message = response.message  
 )  
 authRepository.saveUserSession(  
 UserSession(  
 accessToken = accessToken,  
 refreshToken = refreshToken,  
 isLogin = isLogin,  
 name = userName,  
 email = userEmail  
 )  
 )  
   
 \_loading.*value* = *false* }  
 **}**}

*class* RegisterViewModel(*private val* authRepository: AuthRepository): ViewModel() {

*fun* register(  
 registerRequest: RegisterRequest  
){  
 *viewModelScope*.*launch* **{** \_loading.*value* = *true  
 try* {  
 *val* response = authRepository.register(registerRequest)  
 \_registerStatus.*value* = BasicResponse(  
 ok = response.ok,  
 message = response.message  
 )  
 authRepository.saveUserSession(  
 UserSession(  
 accessToken = accessToken,  
 refreshToken = refreshToken,  
 isLogin = isLogin,  
 name = userName,  
 email = userEmail  
 )  
 )  
   
 \_loading.*value* = *false* }  
 **}**}

1. Implementasi MVVM Untuk Menangani Logika Halaman Chat Bot

Pada kelas ChatViewModel, kelas ini mengelola logika pengambilan data chatbox, chat dalam chatbox, dan pengiriman pesan menggunakan chatRepository. Kelas ini memiliki berbagai LiveData dan StateFlow untuk memantau status chatbox, status chat, status loading, status pengiriman pesan, dan UUID saat ini. Fungsi getChatboxs mengambil daftar chatbox secara asinkron, mengatur status pengambilan chatbox, dan memperbarui LiveData yang sesuai. Fungsi getChatInChatbox mengambil daftar chat dalam chatbox tertentu berdasarkan UUID yang diberikan. Fungsi sendChat mengirim pesan ke chatbox tertentu, memperbarui status pengiriman, dan memperbarui daftar chatbox serta chat dalam chatbox setelah pengiriman berhasil. Setiap operasi dilakukan dalam coroutine viewModelScope untuk mengelola pekerjaan asinkron dengan baik dan memastikan pembaruan UI yang tepat.

*class ChatViewModel(private val chatRepository: ChatRepository): ViewModel(){*

*fun getChatboxs(){  
 viewModelScope.launch* ***{*** *\_getChatboxStatus.value = false  
 try {  
 val response = chatRepository.getChatbox()  
 \_chatboxs.value = response.data?.chatboxs  
 \_getChatboxStatus.value = true  
 }catch (e: Exception){  
 \_getChatboxStatus.value = false  
 }* ***}*** *}  
   
 fun getChatInChatbox(uuid: String){  
 viewModelScope.launch* ***{*** *\_getChatStatus.value = false  
 try {  
 val response = chatRepository.getChatInChatbox(uuid = uuid)  
 \_chatInChatbox.value = response.data?.chats  
 \_getChatStatus.value = true  
 }catch (e:Exception){  
 \_getChatStatus.value = false  
 }* ***}*** *}*

*fun sendChat(idChatbox: String, message: String){}*

*}*

*class ChatViewModel(private val chatRepository: ChatRepository): ViewModel(){*

*fun getChatboxs(){  
 viewModelScope.launch* ***{*** *\_getChatboxStatus.value = false  
 try {  
 val response = chatRepository.getChatbox()  
 \_chatboxs.value = response.data?.chatboxs  
 \_getChatboxStatus.value = true  
 }catch (e: Exception){  
 \_getChatboxStatus.value = false  
 }* ***}*** *}  
   
 fun getChatInChatbox(uuid: String){  
 viewModelScope.launch* ***{*** *\_getChatStatus.value = false  
 try {  
 val response = chatRepository.getChatInChatbox(uuid = uuid)  
 \_chatInChatbox.value = response.data?.chats  
 \_getChatStatus.value = true  
 }catch (e:Exception){  
 \_getChatStatus.value = false  
 }* ***}*** *}*

*fun sendChat(idChatbox: String, message: String){}*

*}*

1. Implementasi MVVM Untuk Menangani Logika Halaman Peraturan

Pada kelas LawViewModel, kelas ini menangani pengambilan data jenis hukum menggunakan lawRepository. Kelas ini memiliki dua LiveData: lawTypeList untuk menyimpan daftar jenis hukum dan loadingState untuk melacak status loading. Fungsi getLawTypes menggunakan coroutine viewModelScope untuk menjalankan operasi pengambilan data secara asinkron. Ketika data sedang diambil, loadingState diatur ke true. Jika pengambilan data berhasil, lawTypeList diperbarui dengan data yang diterima, dan loadingState diatur ke false. Jika terjadi kesalahan selama proses pengambilan data, loadingState juga diatur ke false untuk mengakhiri status loading.

*class LawViewModel(private val lawRepository: LawRepository): ViewModel() {  
   
 private val \_lawTypeList = MutableLiveData<LawTypeData?>()  
 val lawTypeList: LiveData<LawTypeData?> = \_lawTypeList  
   
 private val \_loadingState = MutableLiveData<Boolean>()  
 val loadingState: LiveData<Boolean> = \_loadingState  
   
 fun getLawTypes(){  
 viewModelScope.launch* ***{*** *\_loadingState.value = true  
 try {  
 val response = lawRepository.getLawTypes()  
 \_lawTypeList.value = response.data  
 \_loadingState.value = false  
 }catch (e: Exception){  
 \_loadingState.value = false  
 }* ***}*** *}  
}*

*class LawViewModel(private val lawRepository: LawRepository): ViewModel() {  
   
 private val \_lawTypeList = MutableLiveData<LawTypeData?>()  
 val lawTypeList: LiveData<LawTypeData?> = \_lawTypeList  
   
 private val \_loadingState = MutableLiveData<Boolean>()  
 val loadingState: LiveData<Boolean> = \_loadingState  
   
 fun getLawTypes(){  
 viewModelScope.launch* ***{*** *\_loadingState.value = true  
 try {  
 val response = lawRepository.getLawTypes()  
 \_lawTypeList.value = response.data  
 \_loadingState.value = false  
 }catch (e: Exception){  
 \_loadingState.value = false  
 }* ***}*** *}  
}*

1. Implementasi MVVM Untuk Menangani Logika Halaman Daftar Peraturan

Pada kelas LawListViewModel, kelas ini mengelola paginasi dan pembersihan data item hukum menggunakan LawItemDatabase dan ApiService. Kelas ini memiliki fungsi getItemPagingFlow yang mengembalikan aliran data paging (PagingData) untuk item hukum berdasarkan kode yang diberikan. Fungsi ini menggunakan Pager dengan konfigurasi paginasi seperti ukuran halaman dan jarak *prefetch*, serta LawItemRemoteMediator untuk sinkronisasi data lokal dan remote. Aliran ini di-*cache* dalam viewModelScope untuk manajemen lifecycle yang lebih efisien. Selain itu, kelas ini menyediakan fungsi clearLocalData yang membersihkan semua data lokal dalam database hukum secara asinkron menggunakan coroutine viewModelScope.

*class LawListViewModel(  
 private val lawItemDb: LawItemDatabase,  
 private val apiService: ApiService  
) : ViewModel() {*

*@OptIn(ExperimentalPagingApi::class)  
fun getItemPagingFlow(code: String): Flow<PagingData<LawsItemEntity>> {  
 return Pager(  
 config = PagingConfig(  
 pageSize = 20,  
 prefetchDistance = 10  
 ),  
 remoteMediator = LawItemRemoteMediator(lawItemDb, apiService, code),  
 pagingSourceFactory =* ***{*** *lawItemDb.dao.loadAllLawItemPaged()* ***}*** *).flow.cachedIn(viewModelScope)  
}  
  
fun clearLocalData(){  
 viewModelScope.launch* ***{*** *lawItemDb.dao.clearAll()* ***}****}*

*}*

*class LawListViewModel(  
 private val lawItemDb: LawItemDatabase,  
 private val apiService: ApiService  
) : ViewModel() {*

*@OptIn(ExperimentalPagingApi::class)  
fun getItemPagingFlow(code: String): Flow<PagingData<LawsItemEntity>> {  
 return Pager(  
 config = PagingConfig(  
 pageSize = 20,  
 prefetchDistance = 10  
 ),  
 remoteMediator = LawItemRemoteMediator(lawItemDb, apiService, code),  
 pagingSourceFactory =* ***{*** *lawItemDb.dao.loadAllLawItemPaged()* ***}*** *).flow.cachedIn(viewModelScope)  
}  
  
fun clearLocalData(){  
 viewModelScope.launch* ***{*** *lawItemDb.dao.clearAll()* ***}****}*

*}*

1. Implementasi MVVM Untuk Menangani Logika Halaman Detail Peraturan

Pada kelas LawDetailViewModel, kelas ini mengelola pengambilan detail data hukum menggunakan lawRepository. Kelas ini memiliki dua LiveData: lawDetailData untuk menyimpan detail data hukum dan loadingState untuk melacak status loading. Fungsi getLawDetail menggunakan coroutine viewModelScope untuk menjalankan operasi pengambilan data secara asinkron berdasarkan ID yang diberikan. Saat proses pengambilan data dimulai, loadingState diatur ke true. Jika pengambilan data berhasil, lawDetailData diperbarui dengan data yang diterima, dan loadingState diatur ke false. Jika terjadi kesalahan selama proses pengambilan data, loadingState juga diatur ke false untuk mengakhiri status loading.

*class LawDetailViewModel(private val lawRepository: LawRepository): ViewModel() {*

*private val \_lawDetailData = MutableLiveData<DetailLawData?>()  
val lawDetailData: LiveData<DetailLawData?> = \_lawDetailData  
  
private val \_loadingState = MutableLiveData<Boolean>()  
val loadingState: LiveData<Boolean> = \_loadingState  
  
fun getLawDetail(id: Int){  
 viewModelScope.launch* ***{*** *\_loadingState.value = true  
 try {  
 val response = lawRepository.getDetailLaw(id = id)  
 \_lawDetailData.value = response.data  
 \_loadingState.value = false  
 }catch (e: Exception){  
 \_loadingState.value = false  
 }* ***}****}*

}

*class LawDetailViewModel(private val lawRepository: LawRepository): ViewModel() {*

*private val \_lawDetailData = MutableLiveData<DetailLawData?>()  
val lawDetailData: LiveData<DetailLawData?> = \_lawDetailData  
  
private val \_loadingState = MutableLiveData<Boolean>()  
val loadingState: LiveData<Boolean> = \_loadingState  
  
fun getLawDetail(id: Int){  
 viewModelScope.launch* ***{*** *\_loadingState.value = true  
 try {  
 val response = lawRepository.getDetailLaw(id = id)  
 \_lawDetailData.value = response.data  
 \_loadingState.value = false  
 }catch (e: Exception){  
 \_loadingState.value = false  
 }* ***}****}*

}

1. Hasil Perancangan Aplikasi

Pada tahap ini, aplikasi menampilkan fitur – fitur yang telah dirancang menggunakan jetpack compose sebagai pendekatan antarmuka yang diimplementasikan pada setiap halaman dan mengimplementasikan arsitektur model – view – viewmodel untuk memisahkan antara logika UI dan logika bisnis dengan mengelola setiap pengambilan data.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| (a) | (b) | (c) | (d) |
| Gambar 4. 9 Hasil Perancangan Halaman Peraturan (a) Jenis Peraturan (b) List Peraturan dari Jenis yang dipilih (c) Detail Peraturan | | | |

Hasil dari implementasi android jetpack compose yang digunakan untuk membuat UI yang deklaratif dan responsive untuk mendesain halaman aplikasi pada gambar (a), (b), (c) dan (d). selain itu implementasi pola arsitektur model – view – viewmodel yang dimulai dari layer model yang merepresentasikan data yang akan digunakan pada logika bisnis. Kemudian layer view yang berisi UI dari aplikasi untuk mengatur bagaimana informasi akan ditampilkan yang berisi kelas – kelas. Dan layer terakhir viewmodel yang bertugas untuk berinteraksi dengan model di mana data yang ada akan diteruskan ke layer view sehingga data akan tampil pada halaman aplikasi pada gambar (a), (b), (c) dan (d).

1. Pengujian
2. Pengujian Black Box

Dalam pembuatan aplikasi ini, terdapat beberapa tahap pengujian yang dilakukan untuk memaksimalkan performa dan mengurangi kesalahan serta *bug* dalam aplikasi. Pengujian pertama dilakukan dengan menggunakan teknik Black Box Testing. Black Box Testing adalah metode pengujian perangkat lunak yang memfokuskan pada pengujian fungsionalitas aplikasi tanpa memerlukan pemahaman tentang struktur internal atau kode sumber dari aplikasi tersebut. Penguji hanya berfokus pada input yang diberikan dan output yang dihasilkan, serta memastikan bahwa semua fitur bekerja sesuai dengan spesifikasi yang telah ditentukan. Hasil dari pengujian Black Box Testing ini sangat penting untuk memastikan kualitas aplikasi sebelum dirilis ke pengguna. Pengujian dilakukan dengan berbagai skenario dan kasus uji untuk menutupi sebanyak mungkin kemungkinan penggunaan oleh pengguna akhir. Hasil pengujian Black Box Testing dapat dilihat pada Tabel 4.1, yang mencakup detail dari setiap kasus uji yang dilakukan, hasil yang diharapkan, hasil aktual, dan apakah kasus uji tersebut berhasil atau gagal.

Tabel 4. 1 Tabel Pengujian Black Box

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Halaman** | **Proses** | **Hasil** |
| 1 | Login | Menampilkan Halaman Login | Berhasil |
| 2 | Register | Menampilkan Halaman Register | Berhasil |
| 3 | Home | Menampilkan Halaman Home | Berhasil |
| 4 | Jenis Peraturan | Menampilkan Halaman Jenis Peraturan | Berhasil |
| 5 | Daftar Undang – undang | Menampilkan Halaman Daftar Undang - undang | Berhasil |
| 6 | Detail Undang – undang | Menampilkan Halaman Detail Undang - undang | Berhasil |
| 7 | Chat Bot Peraturang Undang - undang | Memasukkan Prompt yang ingin ditanyakan dan menampilkan jawaban dari prompt tersebut | Berhasil |

Berdasarkan hasil yang terlihat pada Tabel 4.1, dapat disimpulkan bahwa semua fungsi yang terdapat pada aplikasi mulai dari Halaman *Login* sampai Halaman *Chat Bot* dapat berjalan dengan lancar tanpa ada kesalahan. Hasil ini menunjukkan bahwa pengujian *Black Box* *Testing* berhasil mendeteksi dan mengatasi potensi masalah dalam berbagai komponen dan fitur aplikasi. Jadi secara keseluruhan, hasil dari pengujian *Black Box Testing* menunjukkan bahwa aplikasi ini telah dikembangkan dengan baik dan memenuhi standar kualitas yang tinggi. Tidak ada kesalahan atau *bug* yang ditemukan dalam pengujian ini, menunjukkan bahwa aplikasi siap untuk dirilis dan digunakan oleh publik.

1. Pengujian Hardware

Pada tahap ini, peneliti menguji kinerja aplikasi yang dapat dilihat pada Table 4.2 dengan melakukan pengujian pada beberapa perangkat smartphone berbeda untuk memastikan aplikasi dapat berjalan dengan baik dan optimal pada perangkat dengan ekosistem Android. Pengujian ini sangat penting karena perangkat Android memiliki berbagai variasi dalam hal perangkat keras, resolusi layar, versi sistem operasi, dan konfigurasi lainnya yang dapat mempengaruhi kinerja aplikasi.

Tabel 4. 2 Tabel Pengujian Hardware

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Merk** | **Versi Android** | **Ukuran Layar** | **Ram** | **Hasil** |
| 1 | Infinix Note 10 Pro | 11 | 6,95 inch | 8 GB | Lancar |
| 2 | Samsung A50S | 11 | 6,43 inch | 4 GB | Lancar |
| 3 | Realme 8 Pro | 13 | 6,4 inch | 8 GB | Lancar |
| 4 | Samsung A50 | 10 | 6,43 inci | 4 GB | Lancar |
| 5 | Poco X5 5G | 12 | 6,67 inci | 8 GB | Lancar |

Berdasarkan hasil yang disajikan pada Tabel 4.2, didapat kesimpulan bahwa aplikasi ini dapat berjalan dengan baik pada beberapa device mulai dari android versi 10 sampai dengan versi 13 dengan ukuran layar terkecil yakni 6,4 inci sampai dengan ukuran layar terbesar dengan ukuran 6,95 inci. Hal ini menunjukkan bahwa aplikasi ini memiliki kompatibilitas yang baik dengan berbagai versi Android, sehingga dapat diakses oleh pengguna dengan perangkat yang lebih baru maupun yang lebih lama. Selain itu, rentang ukuran layar yang didukung oleh aplikasi ini cukup luas, mencakup berbagai jenis ponsel pintar yang umum digunakan saat ini.

1. Pengujian Ahli Materi

Pada tahap ini, pengujian memiliki tujuan untuk validasi dari aspek media yang dilihat dari segi desain, tulisan, latihan, dan pembelajaran pada aplikasi hukum sebagai media bantuan hukum. Pengujian ini diujikan kepada ahli materi hukum, Dosen program studi Hukum Ekonomi Syari’ah Universitas Darussalam Gontor, May Shinta Retnowati, S.H., M.E. Hasil penilaian dari ahli materi pembelajaran terdapat pada Tabel 4.3.

Tabel 4. 3 Tabel Pengujian Ahli Materi

|  |  |  |
| --- | --- | --- |
| **No** | **Unsur** | **Nilai** |
| 1 | Ketepatan pemilihan bentuk program yang ditampilkan | 5 |
| 2 | Ketepatan komposisi dan kombinasi warna yang digunakan | 4 |
| 3 | Ketepatan pemilihan background | 4 |
| 4 | Peletakan menu-menu dalam program media | 4 |
| 5 | Ketepatan pemilihan tombol dan teks pada program media | 4 |
| 6 | Sistematis, runtut, alur logika jelas | 4 |
| 7 | Kontekstualitas dan akualitas | 4 |
| 8 | Ketepatan dalam taat letak undang - undang | 4 |
| 9 | Kesesuaian ukuran text pada undang - undang | 5 |
| 10 | Kesesuaian font tulisan pada undang - undang | 5 |
| 11 | Kesesuaian gambar dan warna background dalam undang - undang | 3 |
| 12 | Ukuran gambar dalam undang - undang sesuai | 4 |
| 13 | Kejelasan teks dalam program | 5 |
| 14 | Kejelasan teks dalam materi | 4 |
| 15 | Ketepatan pemilihan warna tulisan | 4 |
| 16 | Ukuran dan font huruf | 5 |
| 17 | Kemudahan penggunaan program | 5 |
| 18 | Tingkat interaktivitas pengguna terhadap media | 5 |
| 19 | Kemudahan penggunaan navigasi | 5 |
| 20 | Komposisi tiap halaman menu | 4 |
| 21 | Pengaturan model tulisan | 4 |
| 22 | Ketepatan penggunaan tombol menu pada program media | 4 |
| 23 | Kemudahan transisi dari satu layar ke layar yang lain | 4 |
| **TOTAL** | | **99** |

Setelah melihat hasil pengujian yang telah disajikan pada Tabel 4.3, dapat dilihat hasil rata-rata penilaian media yang didapat oleh aplikasi hukum ini adalah 86% dari total perhitungan skala likert 99 Berdasarkan hasil tersebut, dapat disimpulkan bahwa aplikasi ini dapat digunakan sebagai media sementara untuk bantuan dan layanan hukum.

.

1. Pengujian Ahli Pengguna

Pada tahap ini penguji memberikan kuesioner dari para 35 penguji tertutup di google play console, Pengujian ini melibatkan pengguna untuk mengidentifikasi dan memperbaiki masalah usability, serta mendapatkan umpan balik langsung mengenai pengalaman pengguna. Proses ini adalah langkah penting dalam pengembangan aplikasi, karena melibatkan pengguna akhir yang sebenarnya dalam evaluasi dan pengujian produk. Dengan melibatkan pengguna, pengembang dapat memperoleh wawasan yang lebih mendalam tentang bagaimana aplikasi digunakan dalam situasi nyata dan dapat mengidentifikasi masalah yang mungkin tidak terlihat selama tahap pengembangan internal.

Tabel 4. 4 Pengujian Pengguna

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Pertanyaan** | **Nilai** | | | |
| **Sangat Bagus** | **Bagus** | **Buruk** | **Sangat Buruk** |
| 1 | Apakah mudah dalam melakukan Login? | **26** | **6** | **2** | **1** |
| **Total** | | **125** | | | |
| **Rata - rata** | | **89,28%** | | | |
| 2 | Apakah mudah dalam melakukan Register? | **23** | **10** | **0** | **2** |
| **Total** | | **104** | | | |
| **Rata - rata** | | **74,28%** | | | |
| 3 | Seberapa menarik UI pada halaman Home? | **25** | **8** | **1** | **1** |
| **Total** | | **127** | | | |
| **Rata - rata** | | **90,71%** | | | |
| 4 | Seberapa mudah interaksi pada halaman Home? | **24** | **9** | **2** | **0** |
| **Total** | | **127** | | | |
| **Rata - rata** | | **90,71%** | | | |
| 5 | Apakah bermanfaat fitur *Chat Bot* pada aplikasi ini? | **20** | **12** | **2** | **1** |
| **Total** | | **121** | | | |
| **Rata - rata** | | **86,42%** | | | |
| 6 | Kesesuaian Peraturan Undang undang dengan peraturan.go.id | **23** | **8** | **1** | **3** |
| **Total** | | **121** | | | |
| **Rata - rata** | | **86,42%** | | | |
| 7 | Seberapa mudah dalam menggunakan halaman peraturan? | **26** | **6** | **1** | **2** |
| **Total** | | **124** | | | |
|  | | **88,57%** | | | |
| 8 | Seberapa bermanfaatkah aplikasi ini dalam membantu pengguna dalam bantuan hukum? | **28** | **4** | **1** | **2** |
| **Total** | | **128** | | | |
| **Rata - rata** | | **91,42%** | | | |

Berdasarkan hasil pengujian yang terdapat pada Tabel 4.4, dapat dilihat bahwa presentase hasil penilaian pengguna aplikasi ini mencapai rata-rata 87,22%. Setelah melihat hasil penilaian aplikasi ini, peneliti menyimpulkan bahwa aplikasi dapat dan layak digunakan sebagai media layanan dan bantuan hukum.

1. Maintenance

Proses pemeliharaan serta perbaikan aplikasi dilakukan dengan dasar kritik dan saran dari para pihak yang ikut serta dalam pengujian aplikasi. Saran-saran tersebut akan dijadikan sebagai refrensi untuk pemeliharaan dan perbaikan pada aplikasi. Berikut adalah saran-saran yang telah diterima selama melakukan proses pengujianyang disajikan pada Tabel 4.5. Terdapat beberapa saran yang yang belum dilaksanakan oleh peneliti dan diharapkan untuk penelitian selanjutnya agar dapat menjadikan saran-saran pada penelitian ini sebagai refrensi.

Tabel 4. 5 Saran untuk maintenance aplikasi.

|  |  |  |
| --- | --- | --- |
| **No** | **Saran** | **Keterangan** |
| 1 | Mengubah warna pada halaman daftar undang – undang pata tiap Jenis. | Sudah Diperbaiki |
| 2 | Memvalidasi kembali materi undang - undang yang disajikan dalam aplikasi. | Sudah Diperbaiki |
| 3 | Menambahkan sumber peraturan undang – undang pada aplikasi. | Sudah Diperbaiki |
| 4 | Menambahkan desclaimer pada aplikasi bahwa aplikasi bukan atas nama pemerintah. | Sudah Diperbaiki |
| 5 | Melakukan uji aplikasi secara luas. | Sudah Diperbaiki |
| 6 | Menambahkan fitur untuk konsultasi kepada pengacara. | Belum dilaksanakan, dan dapat menjadi saran untuk penelitian selanjutnya |
| 7. | Berkerjasama dengan mitra pengacara yang ada. | Belum dilaksanakan dan dapat menjadi saran untuk penelitian selanjutnya |

Bab 5 Penutup

* 1. Kesimpulan

Berdasarkan proses penelitian yang dilakukan peneliti, dapat diambil kesimpulan sebagai berikut :

1. Penelitian berhasil mengembangkan aplikasi Android dengan arsitektur MVVM dan Jetpack Compose sebagai UI Toolkit.
2. Aplikasi ini dapat membantu pengguna untuk menanyakan perkara hukum dengan menggunakan fitur *Chat Bot.*
3. Penggunaan MVVM memisahkan logika bisnis dari antarmuka pengguna, memudahkan pengembangan dan pemeliharaan.
4. Jetpack Compose memberikan fleksibilitas dan efisiensi dalam membangun antarmuka pengguna.
   1. Saran

Untuk penelitian selanjutnya, hasil dari penelitian ini dapat dijadikan sebagai referensi dan sebagai perbandingan pada penelitian yang akan dilakukan. Beberapa saran yang dapat dipertimbangkan pada penelitian selanjutnya adalah sebagai berikut:

1. Pengembangan Multi-Platform: Penelitian selanjutnya dapat mengembangkan aplikasi serupa yang kompatibel dengan platform lain seperti iOS , sehingga bisa menjangkau lebih banyak pengguna.
2. Penambahan Fitur: Pengembangan fitur seperti konsultasi pengacara agar pengguna dapat berkonsultasi langsung dengan ahli hukum yang sesuai dengan permasalahan yang di alami.
3. Update Content: Melakukan update mengenai peraturan undang – undang yang baru, agar peraturan yang di dapatkan pengguna merupakan peraturan undang – undang yang up-to-date.

Daftar Pustaka

Al-hafiz, Nofri Wandi. “Perancangan Aplikasi Pembelajaran Interakif Pada Materi Perlindungan Dan Penegakan Hukum Berbasis Android.” *Jurnal Perencanaan, Sains, Teknologi, Dan Komputer* 5, no. 1 (2022): 1–5.

Alfathar, Muhammad Ibra, Banis Cipta Pamungkas, Bangun Sucipto Darwaman, Farez Syafei, Muhammad Rizaldi Dwinanto, Muhamad Aldyn Ghifari, Ni Wayan Parwati, et al. “Penerapan Mvvm (Model View Viewmodel) Pada Pengembangan Aplikasi Bank Sampah Digital.” *Jurnal Riset Dan Aplikasi Mahasiswa Informatika (JRAMI)* 05, no. 02 (2024): 406–14.

Arif, M, A Musthafa, and D Muriyatmoko. “Implementasi Pola Arsitektur Model-View-ViewModel (MVVM) Pada Sistem Informasi Akademik Universitas Darussalam Gontor Berbasis Mobile.” In *ResearchGate*, 1–12, 2019. https://www.researchgate.net/publication/339616775\_Implementasi\_Pola\_Arsitektur\_Model-View-ViewModel\_MVVM\_pada\_Sistem\_Informasi\_Akademik\_Universitas\_Darussalam\_Gontor\_Berbasis\_Mobile?enrichId=rgreq-d5b3b538ef5e8d45f08a80d3b580eba3-XXX&enrichSource=Y292ZXJ.

Arif, Muhammad Syakir, Aziz Musthafa, and Dihin Muriyatmoko. “Implementation of Model-View-ViewModel ( MVVM ) Architecture Pattern in the Sistem Informasi Akademik UNIDA Gontor Mobile Application.” In *Proceeding International Conference on Science and Engineering*, 3:283–89, 2017.

Baselam Asefa. *Membangun Library Komponen Android Menggunakan Jetpack Compose*. *Metropolia*, 2022.

Candra, Fadhlin Ade, and Fadhillatu Jahra Sinaga. “Peran Penegak Hukum Dalam Penegakan Hukum Di Indonesia.” *Edu Society: Jurnal Pendidikan, Ilmu Sosial Dan Pengabdian Kepada Masyarakat* 1, no. 1 (2023): 41–50. https://doi.org/10.56832/edu.v1i1.15.

Diantoni, Cahya, Oman Komarudin, and Adhi Rizal. “Arsitektur Mvvm Dan Framework Jetpack Compose Pada Pengembangan Aplikasi Android ( Studi Kasus : Aplikasi Sukacolab ).” *JATI (Jurnal Mahasiswa Teknik Informatika)* 8, no. 3 (2024): 3216–24.

Dzulqarnain, Fahmi. “Computer Based Information System Journal Rancang Bangun Aplikasi Belajar Arab Untuk Android Menggunakan Jetpack Compose Dan Kotlin.” *Cbis Journal* 11, no. 01 (2023): 25–35. http://ejournal.upbatam.ac.id/index.php/cbishttp://ejournal.upbatam.ac.id/index.php/cbis.

Elsa, Asifah, Nurahma Lubis, and Farhan Dwi Fahmi. “Pengenalan Dan Definisi Hukum Secara Umum ( Literature Review Etika ).” *JIMT Jurnal Ilmu Manajemen Terapan* 2, no. 6 (2021): 768–89.

Fadhlillah, Fadhlillah, Aos Kuswandi, and Pauzan Haryono. “Peranan Aplikasi Android Dalam Peningkatan Kualitas Pelayanan Sekolah Di Pesantren Persis Kota Tasikmalaya.” *Kelola: Jurnal Manajemen Pendidikan* 8, no. 1 (2021): 22–33. https://doi.org/10.24246/j.jk.2021.v8.i1.p22-33.

Lambonan, Geraldo A, Rizal Sengkey, Xaverius B N Najoan, Teknik Elektro, Universitas Sam, and Ratulangi Manado. “Rancang Bangun Aplikasi Ensiklopedia Hukum Indonesia Berbasis Android.” *Jurnal Teknik Informatika Vol.14* 14, no. 3 (2019): 341–48.

Putra, I Putu Reditya Permana, and Herman Tolle. “Pengembangan Aplikasi Pembelajaran Bahasa Bali Berbasis Android Menggunakan MVVM Architecture Dan Jetpack Compose.” *Jurnal Pengembangan Teknlogi Informasi Dan Ilmu Komputer* 7, no. 5 (2023): 2205–14. https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/12687%0Ahttps://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/12687/5764.

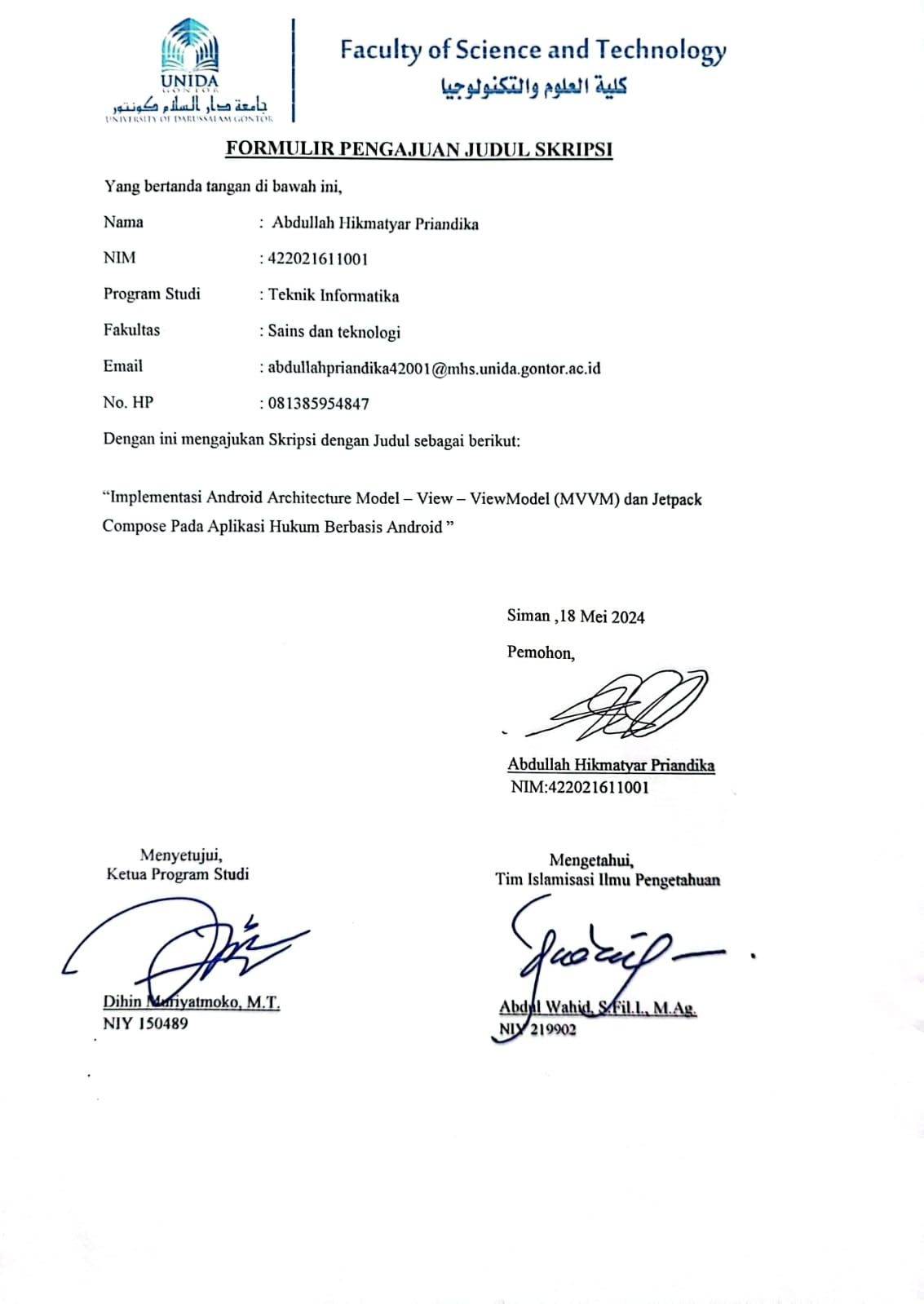
Riyadhi, Irya Muhammad, Intan Purnamasari, Kamal Prihandani, Fakultas Ilmu Komputer, and Universitas Singaperbangsa Karawang. “Penerapan Pola Arsitektur Mvvm Pada Perancangan Aplikasi Pengaduan Masyarakat Berbasis Android.” *INFOTECH Journal* 9, no. 1 (2023): 146–58.

Saraswati, Galuh Wilujeng, and Nanang Febrianto. “Penerapan Arsitektur MVVM Pada Aplikasi Tanamin Untuk Mendeteksi Penyakit Tanaman Berbasis Android.” In *Jurnal Ilmiah Komputer*, 878–89, 2021.

Sommerville, Ian. *Software Engineering*. 10th ed. England: PEARSON, 2016.

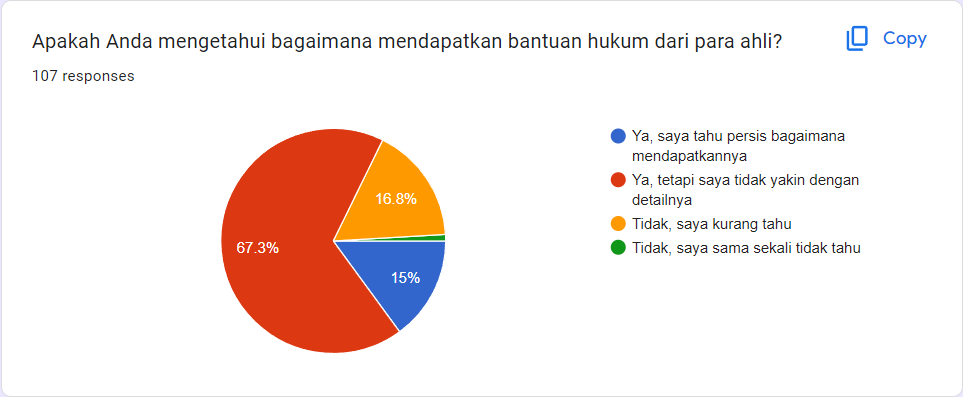
Lampiran

1. Formulir Pengajuan Judul Skripsi



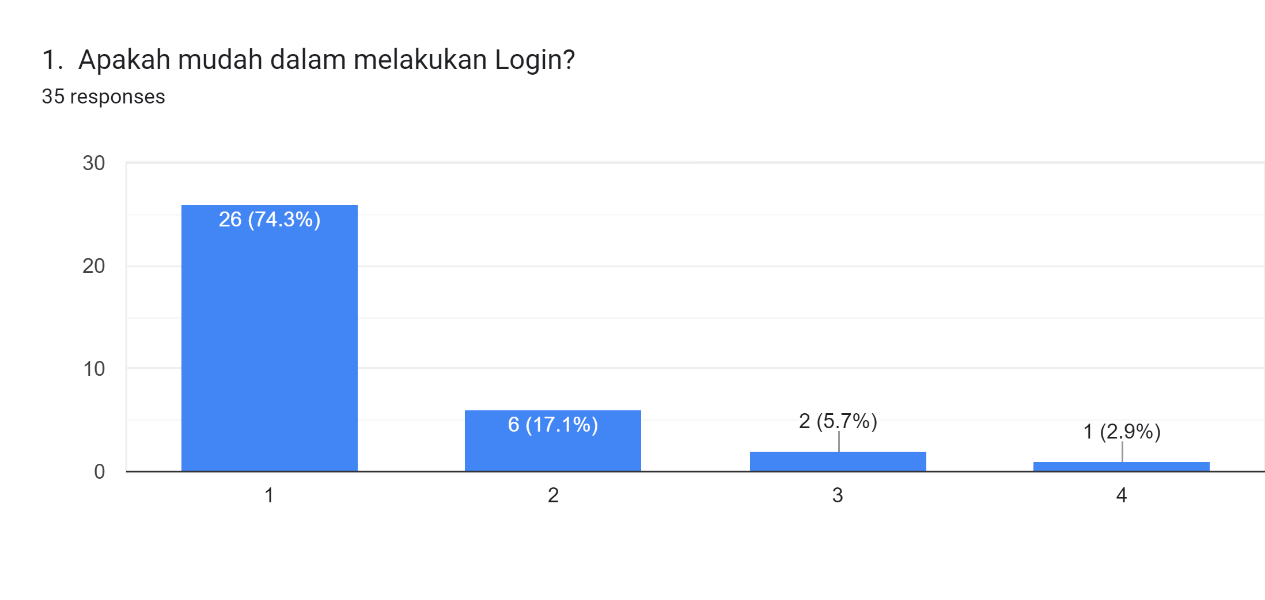
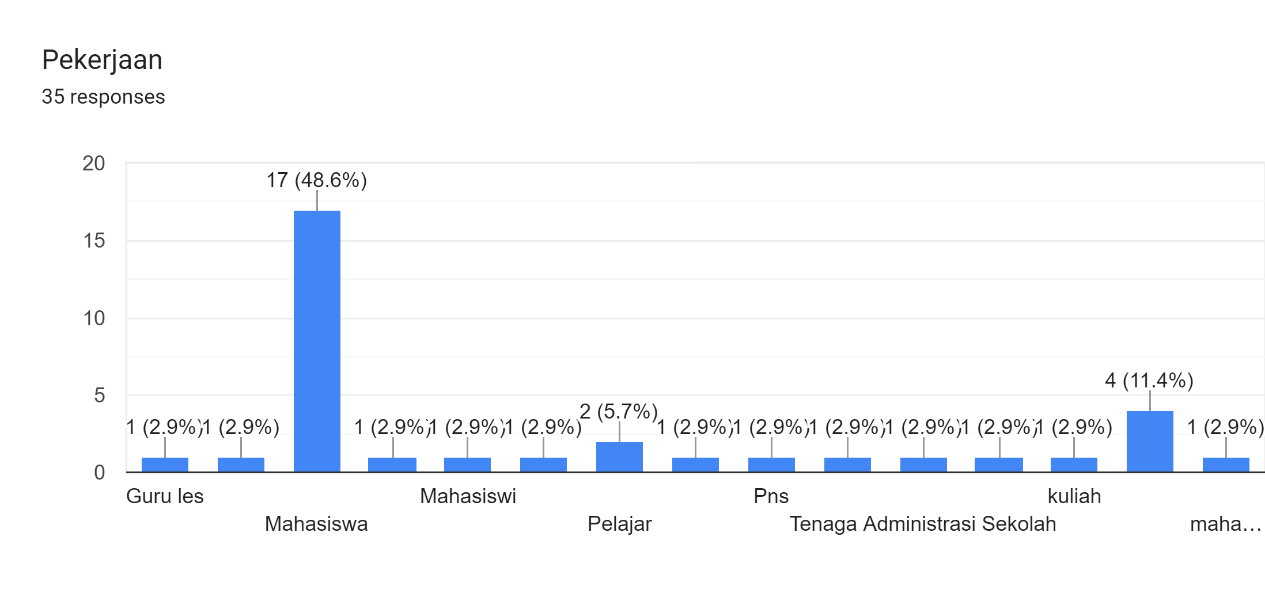
Lampiran 1 Formulir Pengajuan Judul Skripsi

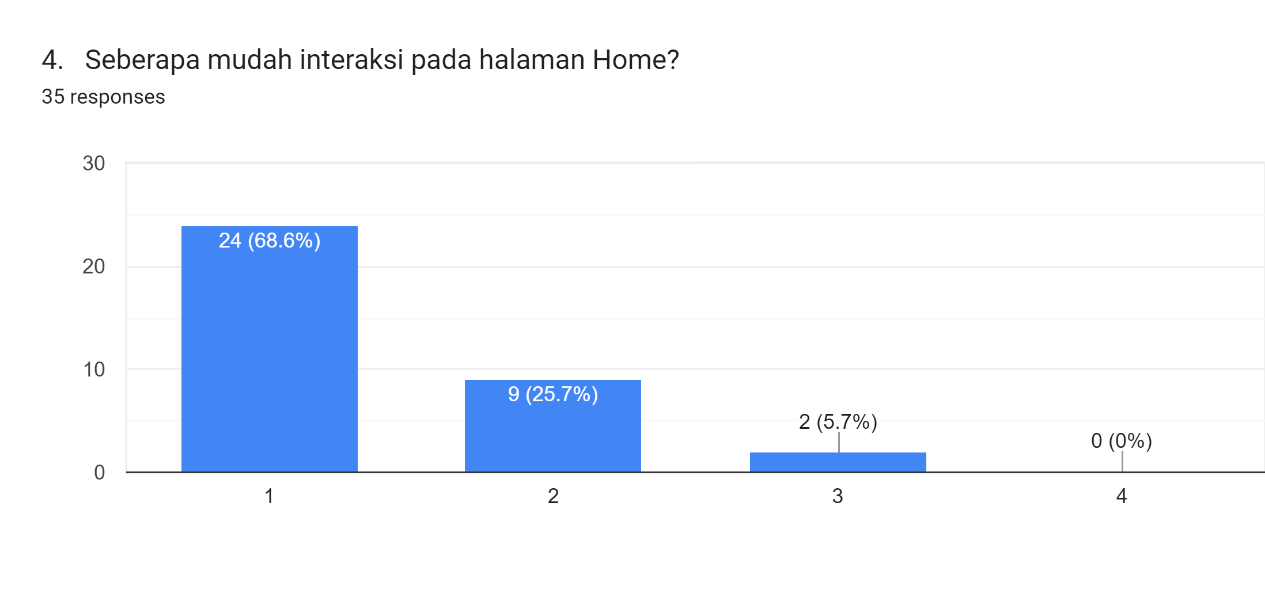
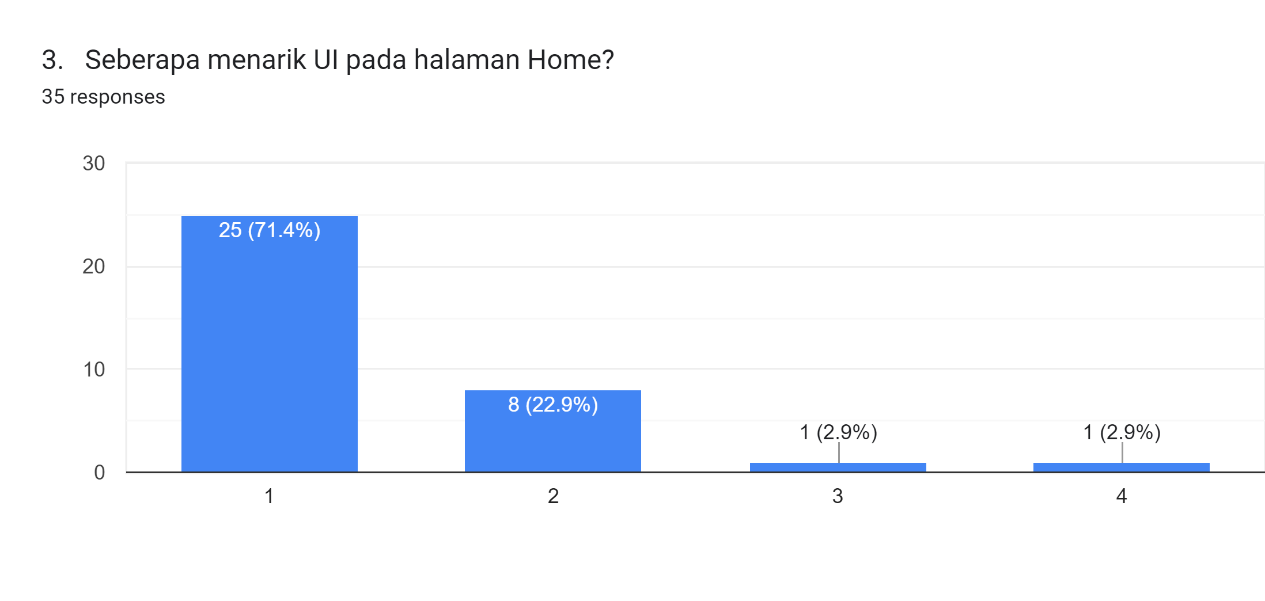
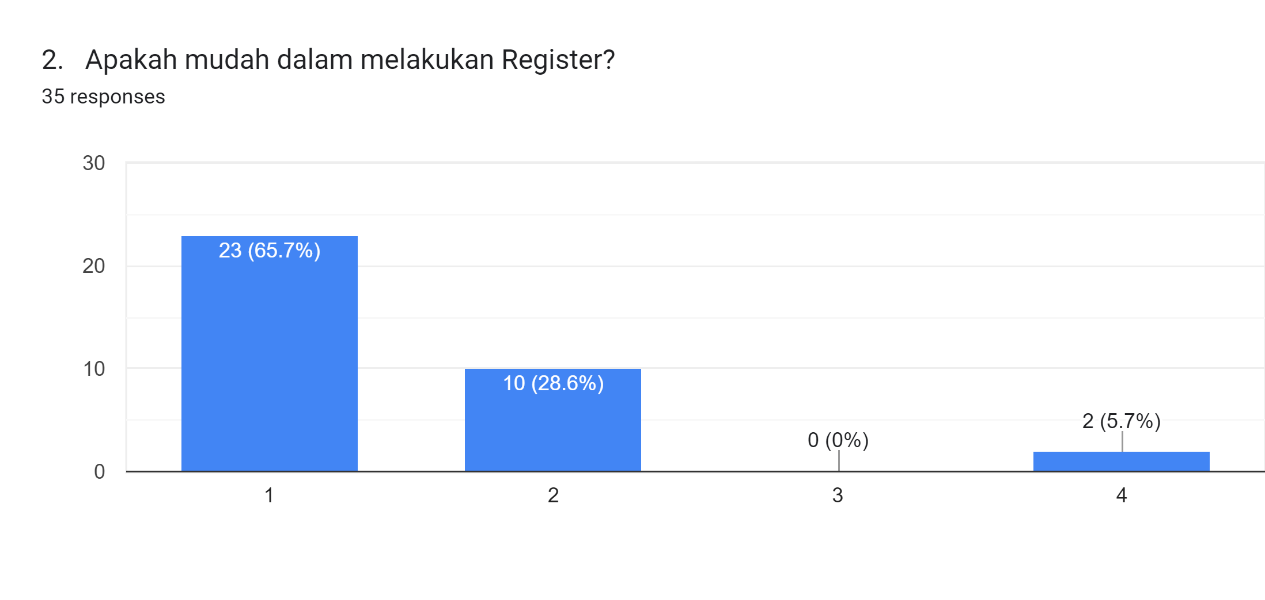
1. Bukti Observasi dengan Kuesioner

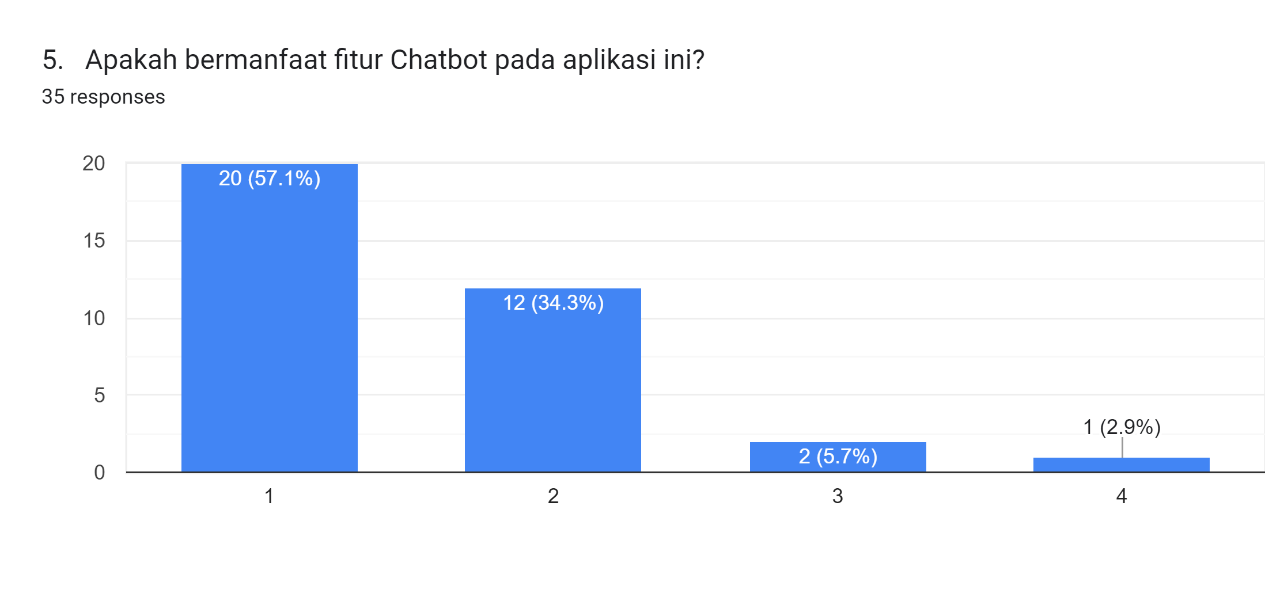


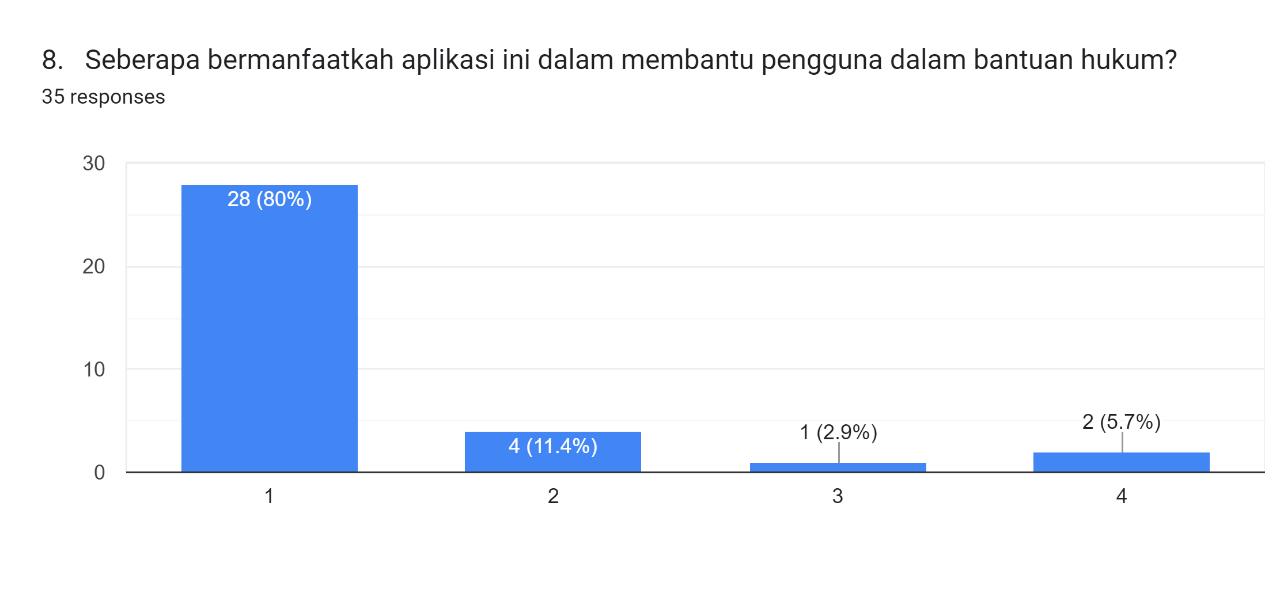
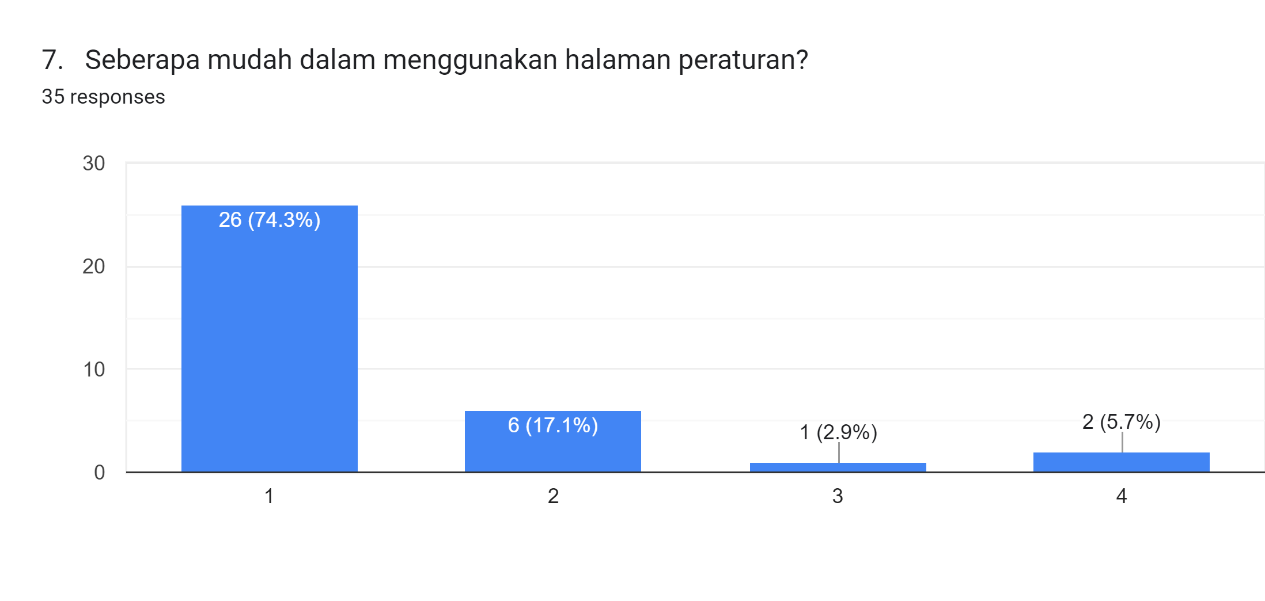
Lampiran 2 Hasil jawaban responden mengenai pertanyaan "Apakah Anda mengetahui bagaimana mendapatkan bantuan hukum dari para ahli?"

1. Kuesioner Uji Pengguna



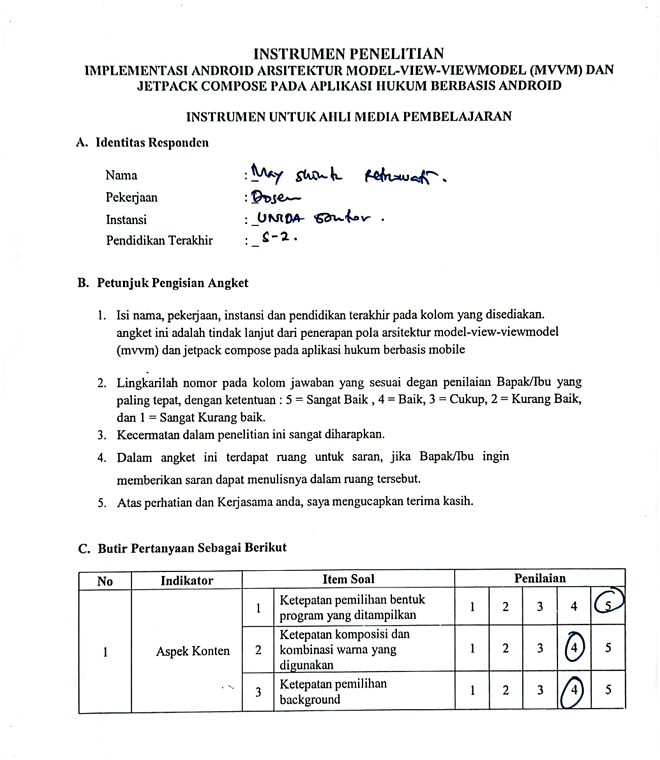


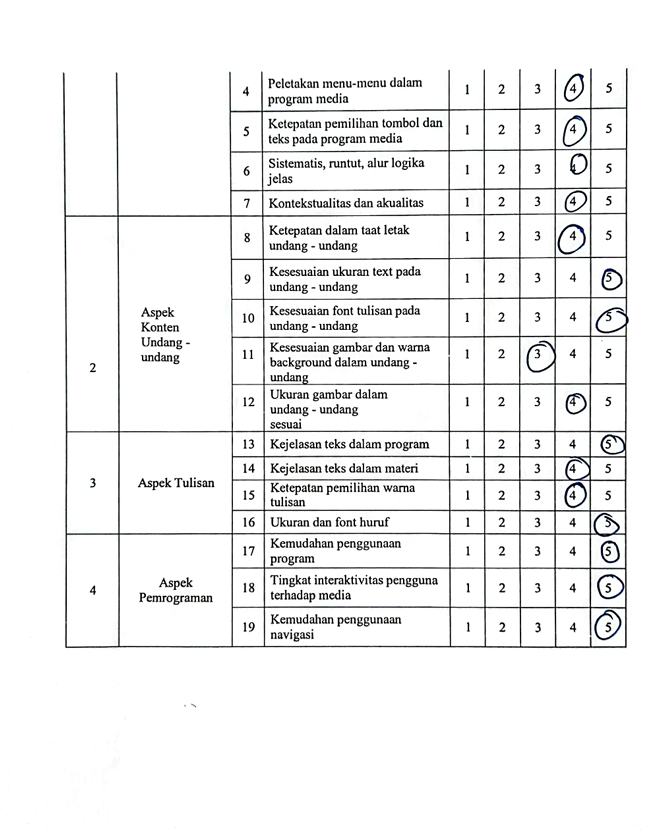


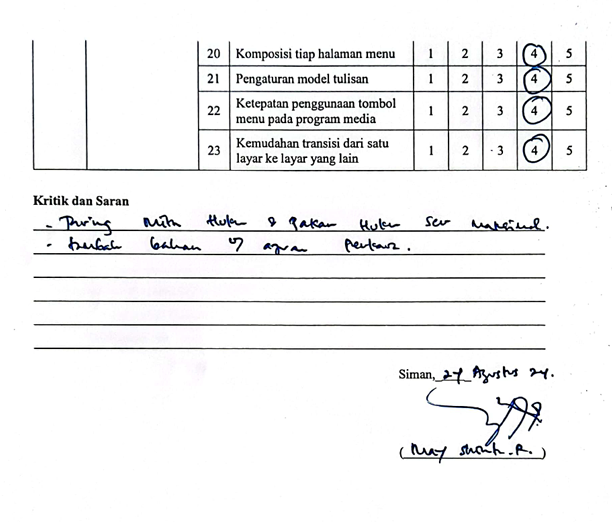


Lampiran 3 Hasil Kuesioner Uji Pengguna

1. Kuesioner Uji Ahli Materi







Lampiran 4 Hasil Kuesioner Uji Ahli Materi

1. Fadhlin Ade Candra and Fadhillatu Jahra Sinaga, “Peran Penegak Hukum Dalam Penegakan Hukum Di Indonesia,” *Edu Society: Jurnal Pendidikan, Ilmu Sosial Dan Pengabdian Kepada Masyarakat* 1, no. 1 (2023): 41–50, https://doi.org/10.56832/edu.v1i1.15. [↑](#footnote-ref-1)
2. Asifah Elsa, Nurahma Lubis, and Farhan Dwi Fahmi, “Pengenalan Dan Definisi Hukum Secara Umum ( Literature Review Etika ),” *JIMT Jurnal Ilmu Manajemen Terapan* 2, no. 6 (2021): 768–89. [↑](#footnote-ref-2)
3. Fadhlillah Fadhlillah, Aos Kuswandi, and Pauzan Haryono, “Peranan Aplikasi Android Dalam Peningkatan Kualitas Pelayanan Sekolah Di Pesantren Persis Kota Tasikmalaya,” *Kelola: Jurnal Manajemen Pendidikan* 8, no. 1 (2021): 22–33, https://doi.org/10.24246/j.jk.2021.v8.i1.p22-33. [↑](#footnote-ref-3)
4. Irya Muhammad Riyadhi et al., “Penerapan Pola Arsitektur Mvvm Pada Perancangan Aplikasi Pengaduan Masyarakat Berbasis Android,” *INFOTECH Journal* 9, no. 1 (2023): 146–58. [↑](#footnote-ref-4)
5. Muhammad Syakir Arif, Aziz Musthafa, and Dihin Muriyatmoko, “Implementation of Model-View-ViewModel ( MVVM ) Architecture Pattern in the Sistem Informasi Akademik UNIDA Gontor Mobile Application,” in *Proceeding International Conference on Science and Engineering*, vol. 3, 2017, 283–89. [↑](#footnote-ref-5)
6. Baselam Asefa, *Membangun Library Komponen Android Menggunakan Jetpack Compose*, *Metropolia*, 2022. [↑](#footnote-ref-6)
7. Geraldo A Lambonan et al., “Rancang Bangun Aplikasi Ensiklopedia Hukum Indonesia Berbasis Android,” *Jurnal Teknik Informatika Vol.14* 14, no. 3 (2019): 341–48. [↑](#footnote-ref-7)
8. I Putu Reditya Permana Putra and Herman Tolle, “Pengembangan Aplikasi Pembelajaran Bahasa Bali Berbasis Android Menggunakan MVVM Architecture Dan Jetpack Compose,” *Jurnal Pengembangan Teknlogi Informasi Dan Ilmu Komputer* 7, no. 5 (2023): 2205–14, https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/12687%0Ahttps://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/12687/5764. [↑](#footnote-ref-8)
9. Arif, Musthafa, and Muriyatmoko, “Implementation of Model-View-ViewModel ( MVVM ) Architecture Pattern in the Sistem Informasi Akademik UNIDA Gontor Mobile Application.” [↑](#footnote-ref-9)
10. Riyadhi et al., “Penerapan Pola Arsitektur Mvvm Pada Perancangan Aplikasi Pengaduan Masyarakat Berbasis Android.” [↑](#footnote-ref-10)
11. Galuh Wilujeng Saraswati and Nanang Febrianto, “Penerapan Arsitektur MVVM Pada Aplikasi Tanamin Untuk Mendeteksi Penyakit Tanaman Berbasis Android,” in *Jurnal Ilmiah Komputer*, 2021, 878–89. [↑](#footnote-ref-11)
12. Cahya Diantoni, Oman Komarudin, and Adhi Rizal, “Arsitektur Mvvm Dan Framework Jetpack Compose Pada Pengembangan Aplikasi Android ( Studi Kasus : Aplikasi Sukacolab ),” *JATI (Jurnal Mahasiswa Teknik Informatika)* 8, no. 3 (2024): 3216–24. [↑](#footnote-ref-12)
13. Diantoni, Komarudin, and Rizal. [↑](#footnote-ref-13)
14. Fahmi Dzulqarnain, “Computer Based Information System Journal Rancang Bangun Aplikasi Belajar Arab Untuk Android Menggunakan Jetpack Compose Dan Kotlin,” *Cbis Journal* 11, no. 01 (2023): 25–35, http://ejournal.upbatam.ac.id/index.php/cbishttp://ejournal.upbatam.ac.id/index.php/cbis. [↑](#footnote-ref-14)
15. Nofri Wandi Al-hafiz, “Perancangan Aplikasi Pembelajaran Interakif Pada Materi Perlindungan Dan Penegakan Hukum Berbasis Android,” *Jurnal Perencanaan, Sains, Teknologi, Dan Komputer* 5, no. 1 (2022): 1–5. [↑](#footnote-ref-15)
16. M Arif, A Musthafa, and D Muriyatmoko, “Implementasi Pola Arsitektur Model-View-ViewModel (MVVM) Pada Sistem Informasi Akademik Universitas Darussalam Gontor Berbasis Mobile,” in *ResearchGate*, 2019, 1–12, https://www.researchgate.net/publication/339616775\_Implementasi\_Pola\_Arsitektur\_Model-View-ViewModel\_MVVM\_pada\_Sistem\_Informasi\_Akademik\_Universitas\_Darussalam\_Gontor\_Berbasis\_Mobile?enrichId=rgreq-d5b3b538ef5e8d45f08a80d3b580eba3-XXX&enrichSource=Y292ZXJ. [↑](#footnote-ref-16)
17. Muhammad Ibra Alfathar et al., “Penerapan Mvvm (Model View Viewmodel) Pada Pengembangan Aplikasi Bank Sampah Digital,” *Jurnal Riset Dan Aplikasi Mahasiswa Informatika (JRAMI)* 05, no. 02 (2024): 406–14. [↑](#footnote-ref-17)
18. Putra and Tolle, “Pengembangan Aplikasi Pembelajaran Bahasa Bali Berbasis Android Menggunakan MVVM Architecture Dan Jetpack Compose.” [↑](#footnote-ref-18)
19. Baselam Asefa, *Membangun Library Komponen Android Menggunakan Jetpack Compose*. [↑](#footnote-ref-19)
20. Diantoni, Komarudin, and Rizal, “Arsitektur Mvvm Dan Framework Jetpack Compose Pada Pengembangan Aplikasi Android ( Studi Kasus : Aplikasi Sukacolab ).” [↑](#footnote-ref-20)
21. Ian Sommerville, *Software Engineering*, 10th ed. (England: PEARSON, 2016). [↑](#footnote-ref-21)