

## LAMPIRAN

### Lampiran 1. Lembar Kesediaan Pembimbing

#### SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Moh. Faizal Ilham Falakh  
NIM : 20090086  
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Taufiq Abidin, S.Pd., M.Kom.  
Status : Dosen Tetap  
NIDN : 0603088305  
Jabatan Fungsional : Lektor  
Pangkat/Golongan : III/C

Pada hari ini rabu tanggal 20 maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 20 Maret 2024

Pihak Pertama



Moh. Faizal Ilham Falakh

Pihak Kedua



Taufiq Abidin, S.Pd., M.Kom.

Mengetahui  
Ketua Program Studi Sarjana Terapan Teknik Informatika



Dyah Apriliana, S.T., M.Kom.  
NIPY. 09.015.225

### SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Moh. Faizal Ilham Falakh  
NIM : 20090086  
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Rosid Mustofa, S.Kom, M.Kom  
Status : Dosen Praktisi  
NIDN : -  
Jabatan Fungsional : -  
Pangkat/Golongan : -

Pada hari ini, Kamis tanggal 14 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat adanya kemajuan pengerjaan skripsi yang dipresentasikan setiap satu minggu. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 14 Maret 2024

Pihak Pertama



Moh. Faizal Ilham Falakh

Pihak Kedua



Rosid Mustofa, S.Kom, M.Kom

Mengetahui  
Ketua Program Studi Sarjana Terapan Teknik Informatika



Dyah Apriliani, S.T., M. Kom.  
NIPY: 09.015.225

## Lampiran 2. Surat Keterangan Penelitian



**POLITEKNIK HARAPAN BERSAMA**  
The True Vocational Campus

Sarjana Terapan Teknik Informatika

Nomor : 42.03/TI.PHB/VII/2024  
Lampiran : -  
Hal : Permohonan Izin Observasi  
Kepada :  
Yth. : Kepala Glamping & Villa by Bukit Mutiara  
di Kab. Tegal

Dengan hormat, mahasiswa dengan identitas berikut ini:

nama : Moh. Faizal Ilham Falakh  
NIM : 20090086  
prodi : Sarjana Terapan Teknik Informatika

Bermaksud melakukan penelitian untuk keperluan Skripsi dengan judul "APLIKASI PEMESANAN VILLA DENGAN FITUR DOWN PAYMENT DAN NEGOSIASI BERBASIS MOBILE". Kami memohon Bapak/Ibu memberikan izin kepada mahasiswa yang bersangkutan agar memperoleh data, keterangan, dan bahan yang diperlukan.

Demikian permohonan ini disampaikan, atas perhatian kami ucapkan terima kasih.

Tegal, 12 Juli 2024  
Ka. Prodi S.Tr. Teknik Informatika,

**Dyan Aprilliani, S.T., M.Kom**  
NIPY : 09.015.225

### Lampiran 3. Surat Pernyataan Pengajuan HKI

#### SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta :

1. N a m a : Moh. Faizal Ilham Falakh  
Kewarganegaraan : Indonesia  
Alamat : Jalan Wungu III No.27 RT.04 / RW.06 Kecamatan Pangkah,  
Kabupaten Tegal, Jawa Tengah, 52471
2. N a m a : Taufiq Abidin, S.Pd., M.Kom  
Kewarganegaraan : Indonesia  
Alamat : Jalan Kaligangsa Asri Timur VIII No.33 RT. 03 / RW.07  
Perum Kaligangsa, Asri Kelurahan Kaligangsa, Kecamatan  
Margadana, Kota Tegal, 52147
3. N a m a : Rosid Mustofa, M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Desa Kedungbanteng RT.04 / RW.12 Kecamatan  
Kedungbanteng, Kabupaten Tegal, 52472

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:  
Berupa : Program Komputer  
Berjudul : Aplikasi Pemesanan Villa Dengan Fitur *Down Payment* dan *Negosiasi*  
Berbasis *Mobile*
  - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
  - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
  - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
  - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
  - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
  - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.

4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
- permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
  - Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
  - Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, 08 Juli 2024



Moh. Faizal Ilham Falakh  
Pemegang Hak Cipta \*

Taufiq Abidin, S.Pd., M.Kom.  
Pemegang Hak Cipta \*

Rosid Mustofa, M.Kom.  
Pemegang Hak Cipta \*

\* Semua pemegang hak cipta agar menandatangani di atas materai.

## Lampiran 4. Surat Pengalihan HKI

### SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini:

1. Nama : Moh. Faizal Ilham Falakh  
Kewarganegaraan : Indonesia  
Alamat : Jalan Wungu III No.27 RT. 04 / RW. 06 Kecamatan Pangkah, Kabupaten Tegal, Jawa Tengah, 52471
2. Nama : Taufiq Abidin, S.Pd., M.Kom  
Kewarganegaraan : Indonesia  
Alamat : Jalan Kaligangsa Asri Timur VIII No.33 RT. 03 / RW.07 Perum Kaligangsa Asri Kelurahan Kaligangsa, Kecamatan Margadana, Kota Tegal, 52147
3. Nama : Rosid Mustofa, M.Kom  
Kewarganegaraan : Indonesia  
Alamat : Desa Kedungbanteng RT.04 / RW.12 Kecamatan Kedungbanteng, Kabupaten Tegal, 52472

Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada:

Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M)  
Alamat : Jl. Mataram No. 9 Pesurungan Lor Kota Tegal

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "Aplikasi Pemesanan Villa Dengan Fitur *Down Payment* dan *Negosiasi Berbasis Mobile*". untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikian Surat pernyataan ini kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Pemegang Hak Cipta  
Ketua P3M

(Dr. Aldi Budi Riyanta, S.Si., M.T)



Tegal, 08 Juli 2024  
Pencipta



(Moh.Faizal Ilham Falakh)

(Taufiq Abidin, S.Pd., M.Kom)

(Rosid Mustof, M.Kom)

## Lampiran 5. Syarat Pengajuan HKI



MANUAL BOOK

APLIKASI PEMESANAN VILLA DENGAN  
FITUR *DOWN PAYMENT* DAN NEGOSIASI  
BERBASIS *MOBILE*

Disusun Oleh :

Moh. Faizal Ilham Falakh

Taufiq Abidin, S.Pd., M.Kom.

Rosid Mustofa, S.Kom, M.Kom.

## DAFTAR ISI

DAFTAR ISI.....	1
PENDAHULUAN .....	2
PENGGUNAAN APLIKASI.....	3
1. Pra-Registrasi.....	3
2. Verifikasi Email.....	3
3. Register.....	4
4. PraLogin .....	5
5. Login.....	6
6. Booking .....	7
7. Daftar Transaksi .....	10
8. Mendaftar Sebagai Mitra.....	12
9. Negosiasi .....	13
PEMECAHAN MASALAH.....	17



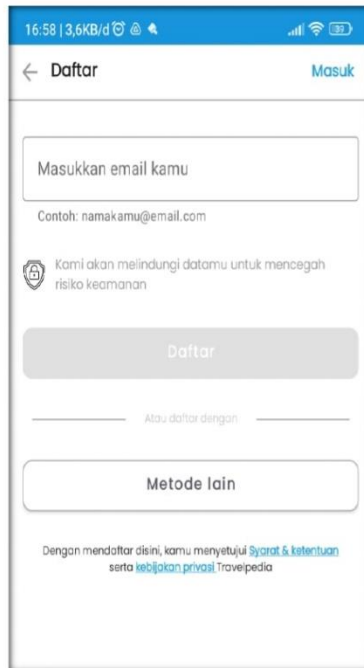
## **PENDAHULUAN**

Travelpedia adalah aplikasi yang dirancang untuk memudahkan proses pemesanan villa secara efisien dan nyaman. Dengan fitur-fitur unggulannya, pengguna dapat dengan mudah memilih dan memesan villa serta melakukan negosiasi harga, memastikan pengalaman menginap yang mulus. Aplikasi ini juga dilengkapi dengan sistem pembayaran uang muka (DP) yang aman, memungkinkan pelanggan untuk mengamankan reservasi mereka tanpa kerumitan. Selain itu, Travelpedia menawarkan antarmuka yang ramah pengguna untuk pemilik villa guna mengelola reservasi, daftar tunggu, dan penjadwalan dengan lebih efektif, terutama pada periode sibuk seperti liburan panjang. Dengan menggunakan teknologi berbasis web, Travelpedia tidak hanya menyederhanakan proses pemesanan tetapi juga meningkatkan interaksi antara pelanggan dan pemilik villa, memastikan kepuasan dan efisiensi yang lebih tinggi.

## PENGGUNAAN APLIKASI

### 1. Pra-Registrasi

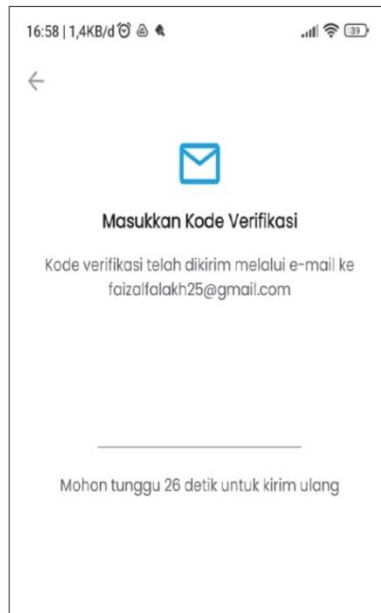
- Buka aplikasi Travelpedia.
- Isi formulir email untuk pra-registrasi.
- Klik tombol “Register” untuk memverifikasi email. Anda akan menerima kode OTP melalui email.



The screenshot shows the registration screen of the Travelpedia app. At the top, the status bar displays the time 16:58, data usage 3,6KB/d, and battery level 32%. The app header has a back arrow and the word "Daftar" (Register), and a "Masuk" (Login) button. The main content area features a text input field labeled "Masukkan email kamu" (Enter your email) with a placeholder example "Contoh: namakamu@email.com". Below this is a security notice: "Kami akan melindungi datamu untuk mencegah risiko keamanan" (We will protect your data to prevent security risks). A large, greyed-out "Daftar" button is present. Below it, there is a separator line and the text "Atau daftar dengan:" (Or register with:). Underneath is a button labeled "Metode lain" (Other methods). At the bottom, a small disclaimer reads: "Dengan mendaftar disini, kamu menyetujui Syarat & ketentuan serta kebijakan privasi Travelpedia" (By registering here, you agree to the Terms & Conditions and privacy policy of Travelpedia).

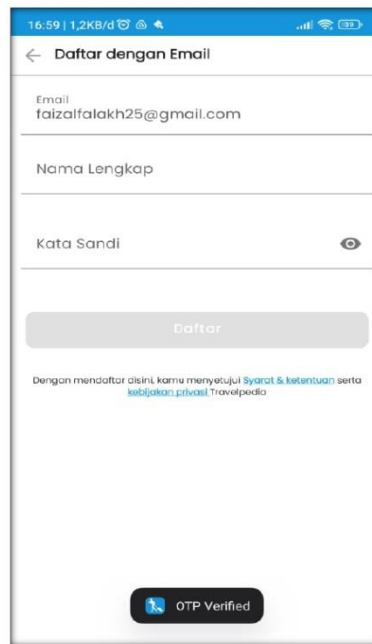
### 2. Verifikasi Email

- Buka email yang telah didaftarkan.
- Periksa kotak masuk untuk menemukan kode OTP.
- Masukkan 6 digit kode OTP.



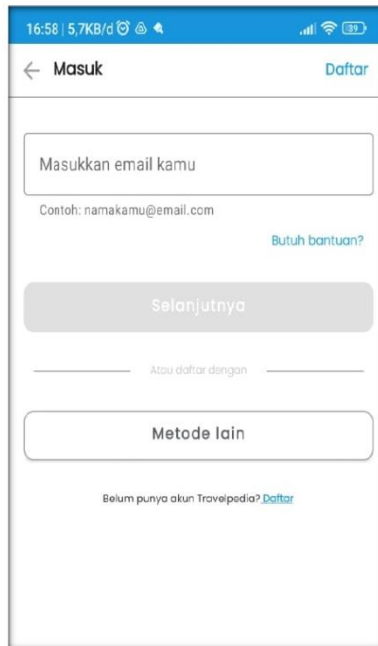
### 3. Register

- Masukkan username dan password
- pastikan password lebih dari 8 karakter
- Klik tombol register untuk menyelesaikan pendaftaran



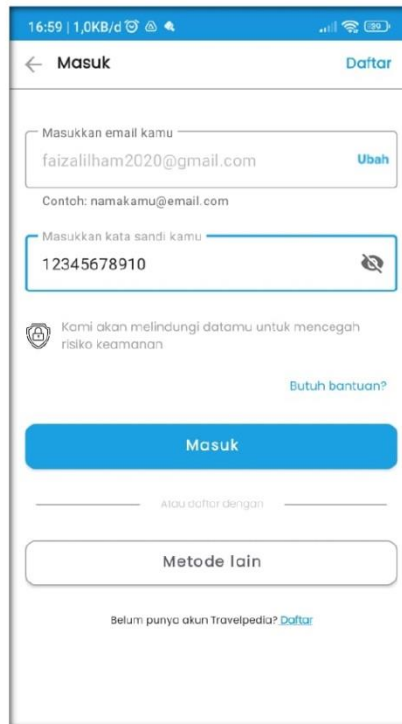
#### 4. PraLogin

- Buka aplikasi Travelpedia.
- Masukkan email untuk pengecekan apakah email sudah terdaftar atau belum. Jika sudah, Anda akan diarahkan ke halaman login. Jika belum, Anda akan diarahkan ke halaman registrasi.
- Klik tombol "Login" untuk melanjutkan.



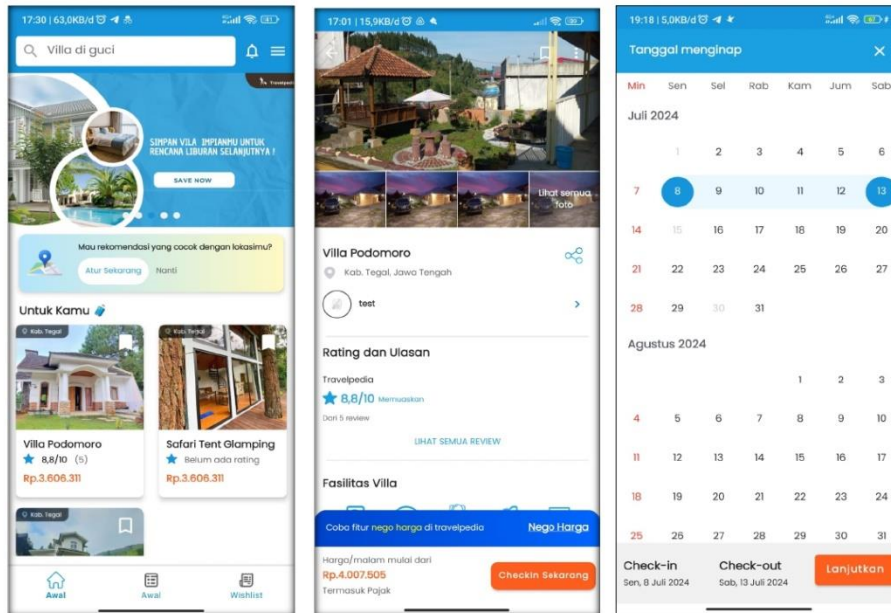
## 5. Login

- Masukkan password yang sesuai dengan email.
- Klik tombol login untuk melanjutkan proses login.



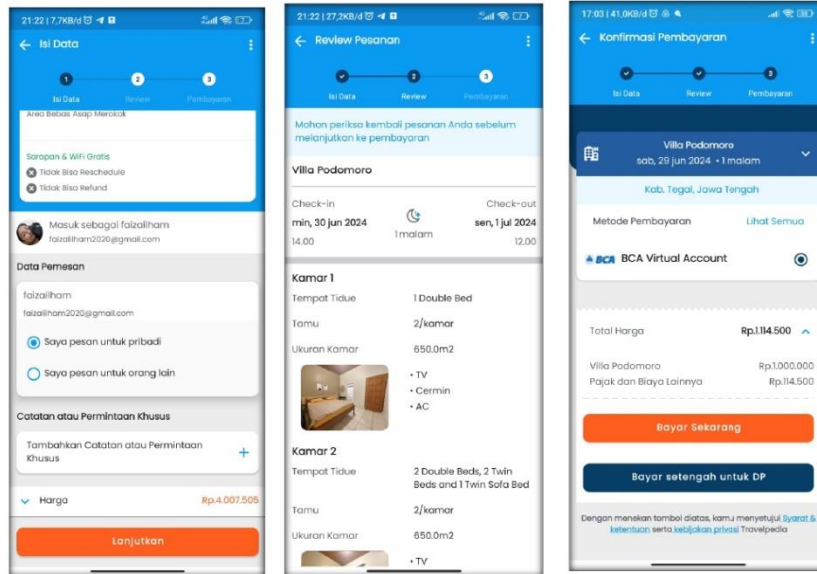
## 6. Booking

- Setelah berhasil login, pengguna akan diarahkan ke halaman home.
- Di halaman home terdapat daftar villa.
- Klik salah satu villa untuk diarahkan ke halaman detail.
- Pilih tanggal check-in villa.
- Klik tombol "Check-in" untuk melanjutkan ke proses transaksi.



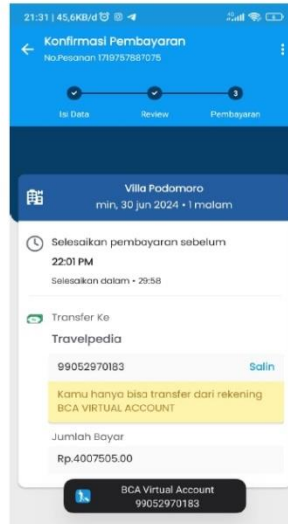
- Di halaman transaksi terdapat 3 langkah:

- **Langkah pertama:** Berisi data singkat villa dan tanggal check-in/check-out. Pada langkah ini, user diharapkan mengisi identitas penyewa, bisa menggunakan username akun utama atau pengguna lain. Pengguna juga bisa membuat catatan tambahan untuk penyedia villa.
- **Langkah kedua:** Terdapat detail data villa yang ingin dipesan, berisi nama villa, jumlah kamar, fasilitas, kebijakan villa, dan harga villa.
- **Langkah ketiga:** Memilih metode pembayaran. Kemudian klik tombol "Bayar Sekarang" atau "Bayar dengan DP" untuk mendapatkan kode reservasi.



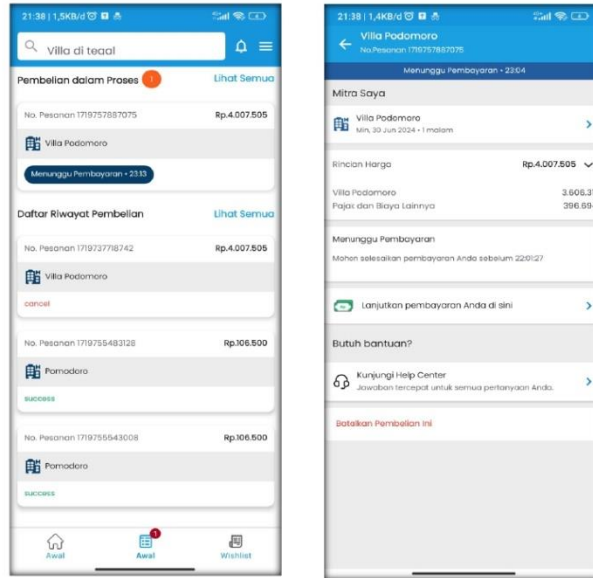
- Jika proses berhasil, akan ditampilkan kode untuk membayar biaya reservasi dengan metode pembayaran terkait.
- Tercantum juga batas waktu pembayaran dan juga cara tutorial cara melakukan pembayaran sesuai dengan metode pembayaran yang dipilih.
- Proses reservasi selesai, user akan tetap di halaman ini atau jika ingin kembali, akan diarahkan ke halaman transaksi.





## 7. Daftar Transaksi

- Untuk melihat reservasi saat ini, klik bottom navigation "Transaksi" di halaman home.
- Terdapat dua kategori list: yang pertama adalah pembelian dalam proses (pending), dan yang kedua adalah daftar riwayat pembelian.
- Pada halaman ini, jika status reservasi telah selesai, maka user akan melihat tombol "Beri Ulasan" dan "Check-in Lagi".
- Tombol "Beri Ulasan" akan mengarahkan ke halaman tambah ulasan.
- Tombol "Check-in Lagi" akan mengarahkan ke halaman detail villa sesuai dengan transaksi yang dipesan.
- Klik salah satu list data untuk melihat detail reservasi.
- Pada halaman detail reservasi, terdapat rincian reservasi seperti detail villa, detail harga, dan status pembayaran.
- Pada halaman ini juga terdapat menu bantuan dan batalkan pembayaran jika status reservasi masih pending.



Berikut ini adalah beberapa status dan penjelasannya:

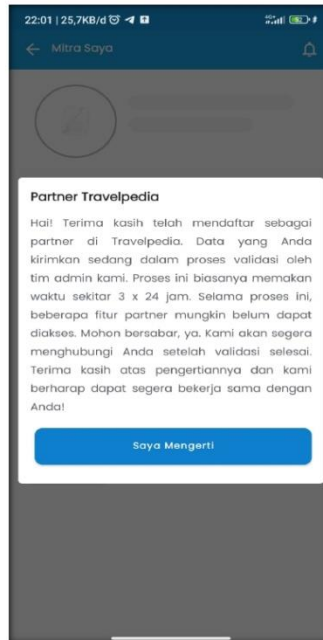
- **Pending:** Pembayaran sedang menunggu konfirmasi. Harap segera selesaikan pembayaran untuk melanjutkan reservasi.
- **Expired:** Waktu pembayaran telah habis. Silakan buat reservasi baru.
- **Cancel:** Pembayaran dibatalkan sebelum batas waktu habis.
- **Success:** Pembayaran berhasil diterima. Reservasi telah dikonfirmasi.

## 8. Mendaftar Sebagai Mitra

- Pada halaman home, klik menu di pojok kanan atas.
- Klik tombol "Buka Villa Gratis".

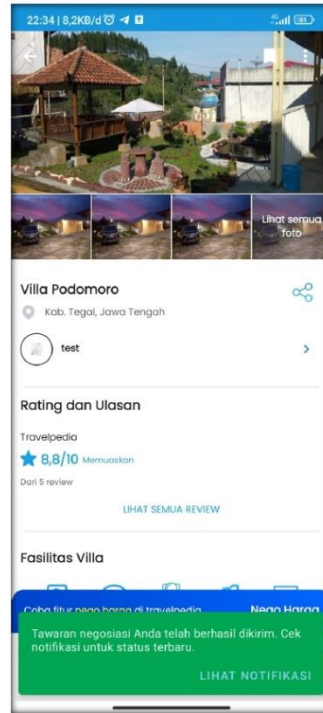
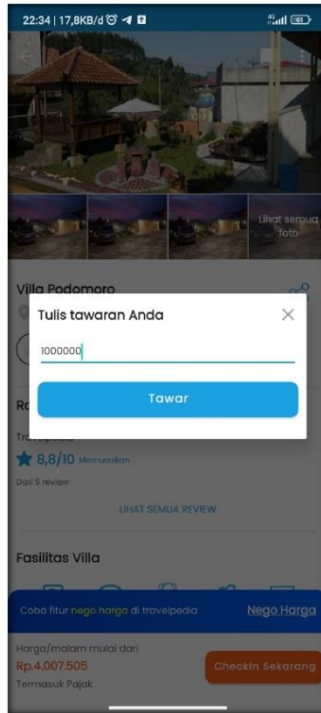


- Isi semua data yang diminta seperti biodata mitra, gambar dan surat izin usaha, jika semua data telah terisi bisa klik tombol "Submit".
- Pendaftaran selesai, Anda akan diarahkan ke halaman dashboard. Beberapa fungsi di dashboard belum bisa dijalankan sepenuhnya dan status mitra masih belum terverifikasi serta menunggu konfirmasi dari admin.

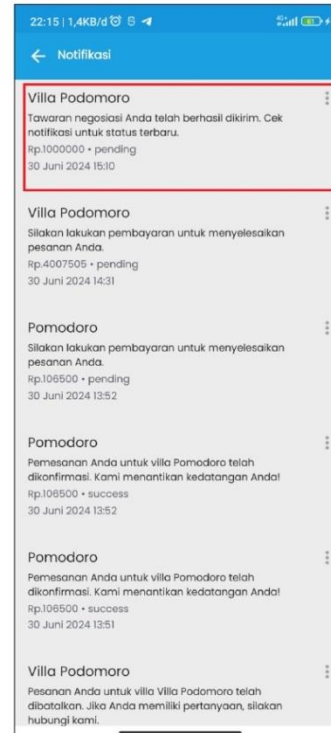


## 9. Negosiasi

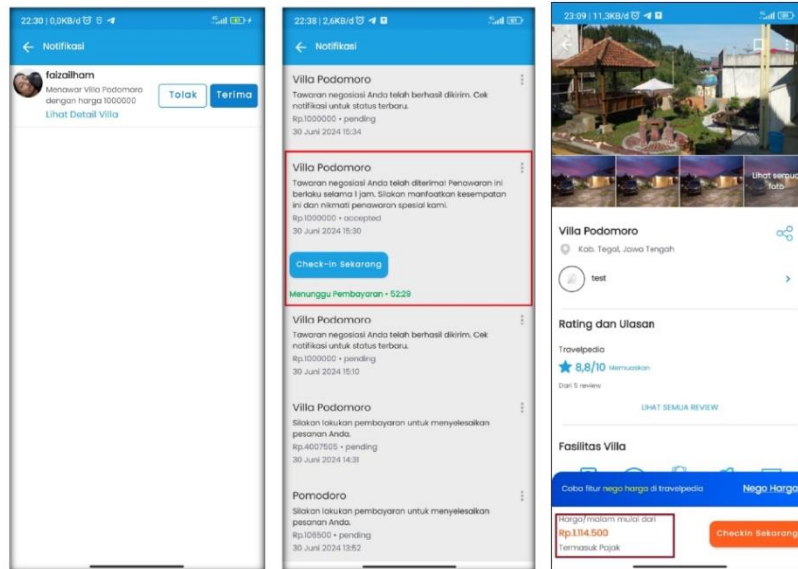
- Masuk ke detail villa, kemudian ajukan permintaan untuk negosiasi harga.
- Klik text nego harga untuk mengajukan negosiasi.
- Pengguna memasukkan harga yang diinginkan dan klik tombol "Nego".



- Negosiasi harga terkirim dan menunggu persetujuan dari pihak villa.
- Pengguna bisa mengecek status negosiasi di halaman notifikasi dengan kembali ke halaman home dan klik ikon notifikasi di pojok kanan atas.



- Mitra bisa mengecek request nego dari user di halaman notifikasi, klik icon notifikasi di pojok kanan atas dashboard mitra.
- Mitra memiliki opsi untuk menerima atau menolak tawaran harga dari pengguna. Jika tawaran diterima, pengguna dapat check-in ke villa dengan harga yang disepakati. Jika menolak maka mitra harus menyertakan alasan kenapa menolak tawaran harga dari pengguna.



- Ketika mitra menerima tawaran dari pengguna maka akan menampilkan tombol checkin dan tenggat waktu batas berakhirnya harga tawaran.
- Penawaran harga yang diterima oleh mitra hanya berlaku selama 1 jam. Jika melebihi waktu tersebut, pengguna tidak dapat check-in dengan harga yang telah ditawarkan.
- Jika negosiasi diterima, user bisa check-in melalui tombol yang ada di list notifikasi untuk melanjutkan transaksi dengan harga yang telah dinegosiasikan seperti di gambar.

## **PEMECAHAN MASALAH**

### **1. Tidak Bisa Login**

- Periksa Koneksi Internet. Pastikan perangkat anda terhubung dengan internet.
- Cek Email dan Kata Sandi. Pastikan anda memasukkan email dan kata sandi yang benar.

### **2. Aplikasi Tidak Merespon**

- Periksa Koneksi Internet. Pastikan perangkat anda terhubung dengan internet.
- Hapus aplikasi dari cache dan buka kembali aplikasi.





DOKUMEN TEKNIKAL

APLIKASI PEMESANAN VILLA DENGAN  
FITUR *DOWN PAYMENT* DAN NEGOSIASI  
BERBASIS *MOBILE*

Disusun Oleh :

Moh. Faizal Ilham Falakh

Taufiq Abidin, S.Pd.,M.Kom.

Rosid Mustofa, S.Kom, M.Kom.

## **PENDAHULUAN**

Travelpedia adalah aplikasi yang dirancang untuk memudahkan proses pemesanan villa secara efisien dan nyaman. Dengan fitur-fitur unggulannya, pengguna dapat dengan mudah memilih dan memesan villa serta melakukan negosiasi harga, memastikan pengalaman menginap yang mulus. Aplikasi ini juga dilengkapi dengan sistem pembayaran uang muka (DP) yang aman, memungkinkan pelanggan untuk mengamankan reservasi mereka tanpa kerumitan. Selain itu, Travelpedia menawarkan antarmuka yang ramah pengguna untuk pemilik villa guna mengelola reservasi, daftar tunggu, dan penjadwalan dengan lebih efektif, terutama pada periode sibuk seperti liburan panjang. Dengan menggunakan teknologi berbasis web, Travelpedia tidak hanya menyederhanakan proses pemesanan tetapi juga meningkatkan interaksi antara pelanggan dan pemilik villa, memastikan kepuasan dan efisiensi yang lebih tinggi.

## LATAR BELAKANG

Industri villa merupakan sektor penting dalam pariwisata, menawarkan berbagai akomodasi bagi wisatawan. Tantangan utama adalah kesulitan menemukan akomodasi yang sesuai, dengan proses pemesanan yang sering masih dilakukan secara konvensional melalui telepon atau kunjungan langsung. Meskipun ada aplikasi pemesanan seperti Traveloka dan Agoda, masih ada peluang untuk inovasi fitur negosiasi dan DP (Down Payment).

Dalam era digital, pengguna menginginkan lebih banyak kontrol, termasuk dalam hal harga. Down Payment adalah sistem di mana sejumlah uang dibayarkan sebagai tanda jadi. Jika transaksi dilanjutkan, uang ini menjadi bagian dari harga transaksi; jika dibatalkan, uang ini menjadi milik penjual. Pengguna ingin bisa menegosiasikan harga atau memberikan DP tanpa harus datang ke lokasi, sehingga mereka bisa melakukan pemesanan dengan biaya lebih rendah atau membayar sebagian kecil dari total biaya sewa sebagai tanda jadi. Hal ini meningkatkan kepercayaan dan mengurangi risiko pembatalan.

Beberapa penelitian telah mencoba mengembangkan aplikasi serupa, namun masih memiliki kekurangan. Penelitian menggunakan teknologi Android tidak menyediakan fitur negosiasi atau payment gateway. Penelitian lain menggunakan teknologi web PHP sudah memiliki fitur chat ke admin dan laporan data booking, tetapi metode pembayaran masih manual.

Dengan potensi besar teknologi mobile, diperlukan implementasi fitur-fitur modern dalam aplikasi mobile berbasis reservasi. Aplikasi ini akan menyediakan autentikasi untuk pengguna, mitra, dan admin, memastikan keamanan dan keandalan akses. Pengguna bisa melakukan DP, dan fitur negosiasi memberikan fleksibilitas dalam menentukan harga yang cocok. Fitur pendukung seperti Google Maps untuk melihat lokasi villa, payment gateway untuk proses pembayaran cashless, dan notifikasi memperkuat interaksi antara pengguna, mitra, dan admin, memastikan informasi lancar dalam setiap tahapan proses reservasi. Penerapan fitur-fitur modern ini diharapkan memberikan pengalaman yang lebih dinamis, personal, dan memuaskan bagi pengguna, sambil memberikan peluang lebih besar bagi penyedia akomodasi untuk menyesuaikan harga dengan preferensi dan kebutuhan pasar yang berubah-ubah.

## SPESIFIKASI TEKNIS

Spesifikasi teknis meliputi :

1. Modul Pengguna
2. Source Code

Berikut uraian spesifikasi untuk pembangunan aplikasi:

1. Linux Ubuntu RAM 8 GB.
2. Android Studio
3. Visual Studio Code
4. Postman
5. Git
6. Smartphone Android.

Berikut uraian spesifikasi modul :

1. Modul untuk pengguna



Bagian ini adalah bagian yang akan ditampilkan untuk pengguna beserta beragam fitur yang disediakan oleh aplikasi seperti membuat reservasi, mencari villa, melihat daftar transaksi, mengubah profil dan lainnya.

## 2. Source Code

### a. Pra Register

Proses pra register merupakan tahap awal sebelum pendaftaran.

```
private fun doPraRegister(){
    binding.apply {
        btnRegister.setOnClickListener {
            val email = etEmail.text.toString().trim()
            authenticationViewModel.onEvent(AuthenticationEvent.PraRegister(email))
            lifecycleScope.launch {
                authenticationViewModel.state.collect { state ->
                    handleLoginResult(state.praRegisterResult)
                }
            }
        }
    }
}

private fun handleLoginResult(result: Resource<UserEmail>){
    when (result) {
        is Resource.Loading -> {
            hideKeyboard()
            setVisibilityLoading(true, binding.loadingPanel)
        }
        is Resource.Success -> {
            setVisibilityLoading(false, binding.loadingPanel)
            Toast.makeText(this@PraRegisterActivity, result.data?.otp.toString(),
                Toast.LENGTH_LONG).show()
            startActivity(Intent(this@PraRegisterActivity, VerificationActivity::class.java).also{
                it.putExtra(EXTRA_EMAIL, binding.etEmail.text.toString())
                it.putExtra(EXTRA_DATA, true)
            })
        }
        is Resource.Error -> {
            setVisibilityLoading(false, binding.loadingPanel)
            val errorMessage = result.message
            if(errorMessage?.toString() == "This email already exist"){
                showDialogLogin("Email sudah terdaftar", "Lanjut masuk dengan email lain", "Ya,
                Masuk", binding.etEmail.text.toString()){
                    startActivity(Intent(this, PraLoginActivity::class.java).also{
                        it.putExtra(EXTRA_EMAIL, binding.etEmail.text.toString())
                    })
                    finish()
                }
            }else{
                Toast.makeText(this@PraRegisterActivity, "errorMessage",
                    Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

Proses ini akan mengecek apakah email yang dimasukkan pengguna valid atau tidak. Fungsi `handleLoginResult` yang menangani hasil proses pra register berdasarkan status dari server berupa `Resource<UserEmail>`. Fungsi ini menggunakan blok `when` untuk memeriksa tiga kemungkinan status. Pertama, jika statusnya adalah `Resource.Loading`, fungsi akan menyembunyikan keyboard dan menampilkan panel loading. Kedua, jika statusnya adalah `Resource.Success`, fungsi akan menyembunyikan panel loading, dan pindah ke `VerificationActivity` dengan membawa data email. Ketiga, jika statusnya adalah `Resource.Error`, fungsi akan menyembunyikan panel loading dan memeriksa pesan kesalahan. Jika pesan

kesalahan adalah "This email already exist", fungsi akan menampilkan dialog yang menawarkan pengguna untuk melanjutkan ke halaman login (PraLoginActivity). Jika pesan kesalahan berbeda, akan ditampilkan dalam bentuk Toast. Dengan demikian, fungsi ini mengelola berbagai kondisi yang mungkin terjadi selama proses login dan memberikan respons yang sesuai kepada pengguna. Fungsi tersebut akan jalan sesaat setelah pengguna memasukkan email dan menekan tombol pra register.

b. Verifikasi OTP

Proses verifikasi OTP adalah proses di mana pengguna akan memasukkan 6 digit OTP yang didapat dari email setelah melakukan pra-register.

```
private fun doVerificaton(email : String){
    val isValidLiveData = MutableLiveData<Boolean>()
    binding.apply {
        etOtp.doOnTextChanged { text, _, _, _ ->
            if (text?.length == 6) {
                isValidLiveData.value = true
                verificationViewModel.onEvent(VerificationEvent.PostVerification(email,
                text.toString().trim()))
            }
        }

        isValidLiveData.observe(this@VerificationActivity) { isValid ->
            if (isValid) {
                lifecycleScope.launch {
                    setVisibilityLoading(true, binding.loadingPanel)
                    delay(1000)
                    verificationViewModel.state.collect{
                        cancel()
                        setVisibilityLoading(false, binding.loadingPanel)
                        verificationResult(it.verificationResult)
                    }
                }
            }
        }
    }
}
```

Fungsi ini menerima parameter email dan menggunakan "MutableLiveData" untuk memantau status validasi OTP. Ketika pengguna mengetik OTP di "etOtp", fungsi akan memeriksa apakah panjang teks adalah 6 karakter. Jika ya, "isValidLiveData" diatur ke nilai "true", dan "verificationViewModel" memproses event "PostVerification" ke server untuk mengecek apakah OTP yang dimasukkan

user benar atau tidak. Jika benar maka akan diarahkan ke halaman register, jika salah maka akan muncul pesan kesalahan pada input OTP.

### c. Register

Proses registrasi akun memungkinkan pengguna baru untuk membuat akun di aplikasi "Travelpedia". Dengan akun ini, pengguna dapat mengakses fitur-fitur reservasi villa.

```
private fun doRegister(){
    binding.apply {
        btnRegister.setOnClickListener {
            val email = etEmail.text.toString().trim()
            val fullName = etFullname.text.toString().trim()
            val password = etNewPassword.text.toString().trim()

            authenticationViewModel.onEvent(AuthenticationEvent.DoRegister(email,fullName,password))

            lifecycleScope.launch {
                authenticationViewModel.state.collect{ state ->
                    handleRegisterResult(state.registerResult)
                }
            }
        }
    }
}
```

Fungsi "doRegister" berfungsi ketika tombol register ditekan maka aplikasi akan menangkap input user berupa email, fullname dan password, yang kemudian akan dikirimkan ke server.

```
private fun handleRegisterResult(result: Resource<Any>) {
    when (result) {
        is Resource.Loading -> {
            hideKeyboard()
            setVisibilityLoading(true, binding.loadingPanel)
        }
        is Resource.Success -> {
            setVisibilityLoading(false, binding.loadingPanel)
            startActivity(Intent(this,DoneActivity::class.java).also{
                it.putExtra(EXTRA_DATA,binding.etEmail.text.toString())
            })
        }
        is Resource.Error -> {
            setVisibilityLoading(false, binding.loadingPanel)
            Toast.makeText(this, result.message ?: "Error", Toast.LENGTH_LONG).show()
        }
    }
}
```

Setelah dikirim ke server maka fungsi “handleRegisterResult” akan menangkap response dari server, ketika “success” maka akan diarahkan ke halaman “DoneActivity” jika terjadi kesalahan maka aplikasi akan memunculkan pesan kesalahan.

#### d. Pra Login

Proses pra login merupakan tahap sebelum memasuki halaman login.

```
private fun doPraLogin(){
    binding.apply {
        btnLogin.setOnClickListener {
            val email = etEmail.text.toString().trim()
            hideKeyboard()
            authenticationViewModel.onEvent(AuthenticationEvent.PraLogin(email))

            lifecycleScope.launch {
                authenticationViewModel.state.collect{ state ->
                    handleLoginResult(state.praLoginResult)
                }
            }
        }
    }
}
```

Fungsi “doPraRegister ” berfungsi ketika tombol pra login ditekan maka aplikasi akan menangkap input user berupa email yang kemudian akan dikirimkan ke server.

```
private fun handleLoginResult(result: Resource<Any>) {
    when (result) {
        is Resource.Loading -> {
            setVisibilityLoading(true, binding.loadingPanel)
        }
        is Resource.Success |> {
            setVisibilityLoading(false, binding.loadingPanel)
            startActivity(Intent(this, LoginActivity::class.java).also{
                it.putExtra(EXTRA_EMAIL, binding.etEmail.text.toString())
            })
        }
        is Resource.Error -> {
            setVisibilityLoading(false, binding.loadingPanel)
            showDialogLogin("Email belum terdaftar", "Lanjut daftar dengan email ini", "Ya,
            Daftar", binding.etEmail.text.toString()){
                startActivity(Intent(this, PraRegisterActivity::class.java).also{
                    it.putExtra(EXTRA_EMAIL, binding.etEmail.text.toString())
                })
            }
        }
    }
}
```



Setelah dikirim ke server maka fungsi “handleResultLogin” akan menangkap response dari server, ketika “success” maka akan diarahkan ke halaman “LoginActivity” jika terjadi kesalahan maka aplikasi akan memunculkan pesan kesalahan.

e. Login

Proses login memungkinkan pengguna yang sudah terdaftar untuk mengakses akun mereka dan menggunakan fitur-fitur aplikasi Travelpedia.

```
private fun doLogin(){
    binding.apply {
        btnLogin.setOnClickListener {
            val email = etEmail.text.toString().trim()
            val password = etPassword.text.toString().trim()
            if(password.length < 3){
                binding.inputPassword.apply {
                    boxStrokeColor = ContextCompat.getColor(this@LoginActivity, R.color.red)
                    customHelperTextColor("Kata sandi tidak benar",this@LoginActivity,R.color.red)
                    binding.etPassword.requestFocus()
                }
            }else{
                setVisibilityLoading(true, binding.loadingPanel)
                authenticationViewModel.onEvent(AuthenticationEvent.DoLogin(email,password))

                lifecycleScope.launch {
                    authenticationViewModel.state.collect{ state ->
                        handleLoginResult(state.loginResult)
                    }
                }
            }
        }
    }
}
```

Fungsi “doPraRegister ” berfungsi ketika tombol pra login ditekan maka aplikasi akan menangkap input user berupa email dan yang kemudian akan dikirimkan ke server.Pada kode diatas terdapat kondisi jika tombol login ditekan input password masih kosong maka akan merubah input password menjadi warna merah dan menampilkan pesan error dibawahnya.

```

private fun handleLoginResult(result: Resource<Login>) {
    when (result) {
        is Resource.Loading -> {
            hideKeyboard()
            setVisibilityLoading(true, binding.loadingPanel)
        }
        is Resource.Success -> {
            setVisibilityLoading(false, binding.loadingPanel)
            if(result.data != null){
                Toast.makeText(this, result.message, Toast.LENGTH_SHORT).show()
                authprefsViewModel.setToken(result.data!!.token)
                authprefsViewModel.setPassword(binding.etPassword.text.toString())
                navigateTo<HomeActivity>(this)
                finish()
            }
        }
        is Resource.Error -> {
            setVisibilityLoading(false, binding.loadingPanel)
            Toast.makeText(this, "${result.message}", Toast.LENGTH_SHORT).show()
        }
    }
}
}
}

```

Setelah dikirim ke server maka fungsi “handleLoginResult” akan menangkap response dari server, ketika “success” maka akan diarahkan ke halaman “HomeActivity” sekaligus menyimpan token JWT yang merupakan hasil response dari server ke dalam local storage, jika terjadi kesalahan maka aplikasi akan memunculkan pesan kesalahan.

#### f. Booking

Proses ini memungkinkan pengguna untuk membuat reservasi villa yang diinginkan melalui aplikasi Travepedia.

```

binding.exFourCalendar.dayBinder = object : MonthDayBinder<DayViewContainer> {
    override fun create(view: View) = DayViewContainer(view)
    override fun bind(container: DayViewContainer, data: CalendarDay) {
        container.day = data
        val textView = container.binding.exFourDayText
        val roundBgView = container.binding.exFourRoundBackgroundView
        val continuousBgView = container.binding.exFourContinuousBackgroundView

        textView.text = null
        roundBgView.makeInvisible()
        continuousBgView.makeInvisible()

        val (startDate, endDate) = selection
        when (data.position) {
            DayPosition.MonthDate -> {
                textView.text = data.date.dayOfMonth.toString()
                if (data.date.isBefore(today)) {
                    textView.setTextColorRes(R.color.semi_grey)
                }else if(disabledDates.contains(data.date)){
                    textView.setTextColorRes(R.color.semi_grey)
                    container.binding.root.isEnabled = false
                } else {
                    when {
                        startDate == data.date && endDate == null -> {
                            textView.setTextColorRes(R.color.white)
                            singleBackground?.let { roundBgView.applyBackground(it) }
                        }
                    }
                }
            }
        }
    }
}

```

```

        singleBackground?.let { roundBgView.applyBackground(it) }
    }
    data.date == startDate -> {
        textView.setTextColorRes(R.color.white)
        rangeStartBackground?.let { continuousBgView.applyBackground(it) }
        singleBackground?.let { roundBgView.applyBackground(it) }
    }
    startDate != null && endDate != null && (data.date > startDate && data.date < endDate)
-> {
        textView.setTextColorRes(R.color.black)
        rangeMiddleBackground?.let { continuousBgView.applyBackground(it) }
    }
    data.date == endDate -> {
        textView.setTextColorRes(R.color.white)
        rangeEndBackground?.let { continuousBgView.applyBackground(it) }
        singleBackground?.let { roundBgView.applyBackground(it) }
    }
    data.date == today -> {
        textView.setTextColorRes(R.color.black)
        todayBackground?.let { roundBgView.applyBackground(it) }
    }
    else -> {
        if (data.date.dayOfWeek == DayOfWeek.SUNDAY) {
            textView.setTextColorRes(R.color.red)
        }else{
            textView.setTextColorRes(R.color.black)
        }
    }
}
}
}
}

```

Kode di atas adalah fungsi “dateBinder” yang menampilkan data tanggal kalender. Tanggal yang ditampilkan telah diatur apakah dapat aktif atau tidak nya berdasarkan riwayat pemesanan villa tersebut. Jadi, jika ada tanggal yang dinonaktifkan, itu berarti di tanggal tersebut sudah ada yang memesan sebelumnya. Pengguna hanya bisa memilih rentang tanggal yang tidak dinonaktifkan.



Setelah pengguna sudah memilih tanggal yang tersedia, maka akan dilanjutkan dengan proses pengisian data reservasi. Pada proses ini terdapat 3 langkah yang akan dilakukan yang pertama yaitu mengisi data pemesan seperti nama pemesan, catatan, dan email. Pengguna dapat memesan untuk dirinya sendiri atau memesan untuk orang lain.

```
private fun setData(villa : VillaDetail){
    binding.apply {
        val username = "Masuk sebagai ${userPreferences.name}"
        tvUsername.text = username
        tvUserEmail.text = userPreferences.email
        tvUsernameOrder.text = userPreferences.name
        tvUserEmailOrder.text = userPreferences.email
        tvRoomOrder.text = villa.villa.name
        if(hasBreakfast == true && hasWiFi == true){
            tvFacility.text = "Sarapan & WiFi Gratis"
        }else if(hasBreakfast == true){
            tvFacility.text = "Sarapan Gratis"
        }else if(hasWiFi == true){
            tvFacility.text = "WiFi Gratis"
        }else{
            tvFacility.visibility = View.GONE
        }
    }

    val price = "Rp.${priceWithTax(checkoutHelper.price!!)}"
    val tax = "Rp.${calculateElevenPercent(checkoutHelper.price!!)}"

    tvTotalPrice.text = price
    tvTaxesAndFees.text = tax
}
}
```

```
private fun addData(){
    val originalBackground: Drawable? = binding.cardDetailGuest.background
    binding.cardDetailGuest.setOnClickListener {
        val i = Intent(requireActivity(),DataActivity::class.java)
        val resGuest = resources.getString(R.string.text_fill_detail_guest)
        if(binding.tvGuest.text.toString() == resGuest){
            i.putExtra("guest","guest")
        }else{
            i.putExtra("guest",binding.tvGuest.text.toString())
            binding.cardDetailGuest.background = originalBackground
        }
        startActivityResultLauncher.launch(i)
    }
    binding.cardNote.setOnClickListener {
        val i = Intent(requireActivity(),DataActivity::class.java)
        if(binding.linearNoted.isVisible){
            i.putExtra("note",binding.tvNoted.text.toString())
        }else{
            i.putExtra("note","")
        }
        startActivityResultLauncher.launch(i)
    }
}
}
```

Fungsi “setData” bertanggung jawab untuk menerima nilai dan menetakannya ke tampilan yang ada. Nilai yang diterima diperoleh dari server berdasarkan ID villa yang dipilih sebelumnya. Pada bagian ini, pemesanan default adalah pengguna yang sedang login.

Sedangkan fungsi “AddData” berfungsi untuk mengatur jika pengguna ingin memesan atas nama orang lain dan menetapkan catatan yang mungkin ingin diberikan pengguna kepada penyedia villa. Fungsi ini memungkinkan pengguna untuk memasukkan informasi tambahan (Catatan) atau permintaan khusus yang ingin disampaikan kepada penyedia villa, sehingga mempermudah proses pemesanan dan memastikan kebutuhan pengguna dapat terpenuhi dengan baik. Namun perlu diingat tidak semua catatan atau permintaan khusus pengguna akan dijamin oleh penyedia villa, penyedia villa hanya akan berusaha memenuhi permintaan pengguna.

```
private fun setData(villa : VillaDetail){
    binding.apply {
        tvVillaName.text = villa.villa.name
        tvGuestName.text = checkoutHelper.username
        tvDateCheckin.text = checkoutHelper.checkinDate?.trimEnd()
        tvDateCheckout.text = checkoutHelper.checkoutDate?.trimStart()
        checkoutHelper.location = "${villa.villa.city}, ${villa.villa.province}"
        tvTimeCheckin.text = villa.villa.checkIn
        tvTimeCheckout.text = villa.villa.checkOut
        setRecycler(villa.villa.rooms)
        if(checkoutHelper.note?.isNotBlank() == true){
            binding.tvNoted.text = checkoutHelper.note
        }
        val price = "Rp.${Utils.priceWithTax(checkoutHelper.price!!)}"
        val tax = "Rp.${Utils.calculateElevenPercent(checkoutHelper.price!!)}"

        tvRoomOrder.text = villa.villa.name
        val priceRoom =
            "Rp.${checkoutHelper.price!!} reversed().chunked(3).joinToString(".").reversed()"
        tvOrderRoomPrice.text = priceRoom
        tvTotalPrice.text = price
        tvTaxesAndFees.text = tax
    }
}
```

Fungsi diatas merupakan fungsi untuk langkah kedua, di langkah kedua hanya menampilkan data pemesanan, permintaan khusus (Jik ada), tanggal check-in, tanggal check-out dan detail villa yang dipesan seperti fasilitas villa, kebijakan

villa dan harga villa yang sudah dikalkulasikan sesuai dengan berapa lama pengguna menginap.

```
private fun doTransaction(paymentMethod: String){
    binding.btnContinue.setOnClickListener {
        createTransaction(paymentMethod,"cash")
    }
    binding.btnDp.setOnClickListener {
        createTransaction(paymentMethod,"DP")
    }
}
|
private fun result(resource : Resource<TransactionCreate>){
    val header = activity?.findViewById<LinearLayout>(R.id.header_stepper)
    val loading = activity?.findViewById<LinearLayout>(R.id.loading_stepper)
    when(resource){
        is Resource.Loading -> {
            binding.linearNotTrx.visibility = View.GONE
            header?.visibility = View.GONE
            loading?.visibility = View.VISIBLE
        }
        is Resource.Success -> {
            header?.visibility = View.VISIBLE
            loading?.visibility = View.GONE
            if(resource.data != null){
                handleActivityResult(
                    paymentBankPreferences.name!!, resource.data!!.virtual_account,
                    resource.data!!.expired,resource.data!!.price,resource.data!!.orderId,false)
            }
        }
        is Resource.Error -> {
            Toast.makeText(requireActivity(), "${resource.message}", Toast.LENGTH_SHORT).show()
        }
    }
}
```

Fungsi `doTransaction` berfungsi untuk mengajukan permintaan ke server guna melakukan pembayaran, baik itu pembayaran secara tunai (cash) maupun pembayaran uang muka (DP). Setelah permintaan diajukan, fungsi `result` akan menerima respons dari server. Jika statusnya adalah loading, maka panel loading akan ditampilkan untuk menunjukkan bahwa proses sedang berlangsung. Jika statusnya adalah success, maka nomor virtual account, total harga, nama bank dan beberapa data transaksi lainnya akan ditampilkan kepada pengguna. Dengan demikian, pengguna dapat melihat informasi pembayaran dan melanjutkan proses transaksi sesuai dengan metode pembayaran yang dipilih.

g. Daftar Transaksi

Fitur ini memungkinkan pengguna untuk melihat semua reservasi yang aktif, non aktif atau sedang berlangsung.

```
private fun resourceResultCurrentUser(result: Resource<CurrentUser>) {
    when (result) {
        is Resource.Loading -> { binding.loading.visibility = View.VISIBLE }
        is Resource.Success -> {
            binding.loading.visibility = View.GONE
            if(result.data != null){
                binding.content.visibility = View.VISIBLE
                showRecycler(result.data!!.userTrx, false)
            }else{
                binding.empty.visibility = View.VISIBLE
            }
            val trx = result.data?.userTrx?.filter { it.status == "pending" }
            if(trx?.isEmpty() == true){
                Utils.addBadgeToBottomNav(requireActivity(), trx.size ?: 0)
                binding.countNowTransaction.text = trx.size.toString()
                binding.tvSeeAll.setOnClickListener {
                    startActivity(Intent(requireActivity(), HistoryTransactionActivity::class.java))
                }
                binding.tvSeeAllOnProcess.setOnClickListener {
                    startActivity(Intent(requireActivity(), HistoryTransactionActivity::class.java))
                }
                showRecycler(trx, true)
            }else{ binding.linearProcess.visibility = View.GONE}

            if(result.data?.userTrx?.isEmpty() == true){
                binding.empty.visibility = View.GONE
            }else{
                binding.empty.visibility = View.VISIBLE
            }
        }
        is Resource.Error -> showNetworkError("${result.message}")
    }
}
```

Fungsi di atas digunakan untuk mengambil semua data transaksi pengguna dan menampilkannya dalam bentuk daftar vertikal. Pada kode tersebut, transaksi yang berstatus pending juga ditampilkan, namun ditampilkan dalam daftar yang berbeda dan diatur secara horizontal. Selain itu, fungsi ini menambahkan lencana (badge) pada navigasi bawah (bottom nav) yang menunjukkan jumlah pesanan yang belum diselesaikan atau berstatus pending. Dengan demikian, pengguna dapat dengan mudah melihat dan membedakan antara transaksi yang sudah selesai dan yang masih pending, serta mendapatkan informasi visual mengenai jumlah transaksi yang perlu perhatian lebih lanjut. List data tersebut nantinya bisa diklik untuk mengarah ke detail transaksi.

```

private fun setData(data : Transaction){
    binding.apply {
        tvVillaNameAppBar.text = data.villaName
        val order = "No.Pesanan ${data.orderId}"
        tvNoOrder.text = order
        tvVillaName.text = data.villaName
        tvPriceTotal.text = formatIDRCurrency(data.total)
        tvVillaDetail.text = data.villaName
        tvPriceTax.text = calculateTax(data.total.toDouble().toInt().toString())
        tvPrice.text = calculateOriginalPrice(data.total.toDouble().toInt().toString())
        val date = data.expired.split(" ")[0]
        val time = data.expired.split(" ")[1]
        val inputFormat = SimpleDateFormat("yyyy-MM-dd", Locale.getDefault())
        val outputFormat = SimpleDateFormat("EEE, dd MMM yyyy", Locale("in", "ID"))
        val dated = inputFormat.parse(date)
        val formattedDate: String = dated?.let { outputFormat.format(it) }.toString()
        val dateText = "$formattedDate \u2022 1 malam"
        tvDate.text = dateText
        var timeText = ""
        val dateFormat = SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault())
        if(data.status == "pending"){
            cardCancel.visibility = View.VISIBLE
            doCancel(data.orderId)
            timeText = "Mohon selesaikan pembayaran Anda sebelum $time"
            tvExpired.text = timeText
            val currentDate = Date()
            val expiredDate = dateFormat.parse(data.expired)
            expiredDate?.let {
                val timeDiff = it.time - currentDate.time
                if (timeDiff > 0) {
                    startCountdown(tvCountdownPayment, timeDiff,true)
                } else {
                    tvCountdownPayment.text = "Batas Waktu Habis"
                }
            }
        }
    }
}

```

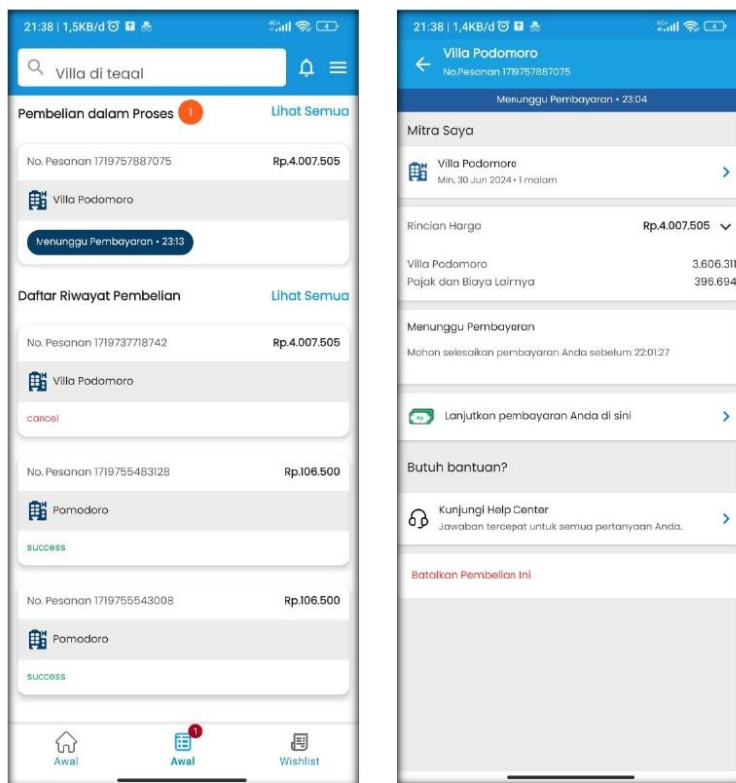
```

        cardPayment.visibility = View.GONE
        tvTitleExpired.text = "Status Pembayaran"
    }else{
        timeText = "Pembayaran DP Berhasil \nSilahkan selesaikan pembayaran sisa pembayaran
ke resepsionis villa ${data.villaName} sebelum ${data.checkInDate} pukul ${data.checkInTime}"
        tvCountdownPayment.setBackgroundResource(R.color.success_snackbar)
        tvCountdownPayment.text = data.status
        tvExpired.text = timeText
        cardPayment.visibility = View.GONE
        tvTitleExpired.text = "Status Pembayaran"
    }
    if(data.reviewId != 0){
        binding.linearReview.visibility = View.VISIBLE
        val roundedNumber: Float = ((data.rating * 10).roundToInt() / 10.0).toFloat()
        val rating = "Kamu telah memberi rating $roundedNumber di transaksi ini"
        binding.rating.text = rating
        binding.ratingStar.rating = (roundedNumber/2.0).toFloat()
    }
}
}else{
    timeText = "Pembayaran ${data.status}"
    tvCountdownPayment.setBackgroundResource(R.color.error_snackbar)
    tvCountdownPayment.text = data.status
    tvExpired.text = timeText
    tvTitleExpired.text = "Status Pembayaran"
}
cardPayment.setOnClickListener {
    val bundle = Bundle().apply {
        putString("payment_method", data.paymentMethod)
        putString("code", data.virtualAccount)
        putString("orderId", data.orderId)
        putString("expired", data.expired)
        putString("price", data.total)
        checkoutHelper.price = data.total
        checkoutHelper.villaName = data.villaName
        checkoutHelper.checkInDate = data.checkInDate
        checkoutHelper.villaDate = data.date
    }
    val intent: Intent = Intent(this@DetailHistoryTransactionActivity,
CheckoutActivity::class.java)
    intent.putExtras(bundle)
    startActivity(intent)
}
}

```



Kode di atas digunakan untuk menampilkan data detail transaksi. Pada halaman detail transaksi, pengguna dapat melihat informasi lengkap terkait data transaksi, seperti detail villa yang dipesan, harga villa, dan status pembayaran. Pengguna juga memiliki opsi untuk membatalkan pembayaran jika status pembayaran masih pending. Selain itu, pengguna dapat melihat kembali nomor virtual account untuk melanjutkan pembayaran jika status pembayarannya masih pending.



Jika pengguna merasa ada data transaksi yang kurang tepat atau mengalami masalah lainnya, aplikasi menyediakan menu bantuan yang dapat diakses untuk mengajukan pertanyaan atau mendapatkan bantuan lebih lanjut. Hal ini memastikan bahwa pengguna memiliki semua informasi yang diperlukan terkait transaksi mereka dan dapat dengan mudah mendapatkan bantuan jika dibutuhkan.

#### h. Mendaftar Sebagai Mitra

Fitur ini memungkinkan pengguna aplikasi yang memiliki villa untuk mendaftarkan villa mereka menjadi mitra Travepedia.

```

private fun sendData(){
    binding.apply {
        preferencesHelper.name = etCompanyName.text.toString().trim()
        preferencesHelper.description = etDescription.text.toString().trim()
        preferencesHelper.address = etLocationName.text.toString().trim()
        preferencesHelper.privacy = etPrivacy.text.toString().trim()
        preferencesHelper.operational = etOperationalTime.text.toString().trim()
        val viewPager = activity?.findViewById<ViewPager2>(R.id.viewPagerPartner)
        viewPager?.currentItem = 1
    }
}

```

```

private fun next(){
    val viewPager = activity?.findViewById<ViewPager2>(R.id.viewPagerPartner)
    binding.next.setOnClickListener {
        if(!listUri.isNullOrEmpty() && image.isNotBlank()){
            preferencesHelper.saveStringList(listPath?.toList() ?: arrayListOf())
            preferencesHelper.image = image
            viewPager?.currentItem = 2
            registerPartnerCommunicator.secondSendingData(listPath?.toList() ?: arrayListOf())
        }else{
            Toast.makeText(requireActivity(), "Terjadi kesalahan", Toast.LENGTH_SHORT).show()
        }
    }
}

```

```

private fun upload(document : String){
    if(preferencesHelper.name != null && preferencesHelper.address != null &&
        preferencesHelper.description != null && preferencesHelper.privacy != null &&
        preferencesHelper.operational != null){
        companyViewModel.onEvent(CompanyEvent.CreateCompanies(preferencesHelper.name!!, preferencesHelper.address
            !!, preferencesHelper.description!!, preferencesHelper.privacy!!, document, preferencesHelper.operational!!), preferencesHelper.getStringList() ?: arrayListOf())
        lifecycleScope.launch {
            companyViewModel.state.collect {
                handleResultUpload(it.createCompanyResult)
            }
        }
    }else{ Toast.makeText(requireActivity(), "Data Belum Lengkap", Toast.LENGTH_SHORT).show() }
}

```

Kode diatas digunakan untuk pendaftaran mitra, terdapat 3 langkah untuk pendaftaran mitra, langkah pertama mengisi biodata mitra pada kode diatas, biodata mitra disimpan di” Sharedpreferences “ atau biasa disebut penyimpanan lokal sederhana, nantinya data di step pertama akan dibawa terus sampai step terakhir.

Langkah kedua memilih gambar banner dan profile mitra, gambar banner bisa memilih lebih dari 1, sedangkan untuk profile maksimal 1 gambar, path dari gambar juga disimpan di “Sharedpreferences”.

Langkah ketiga yaitu pemilihan file sertifikat usaha atau file pendukung supaya admin percaya bahwa pengguna ini punya legalitas untuk mendirikan villa. Setelah semua data terisi maka bisa langsung submit.



Selanjutnya, pengguna akan langsung diarahkan ke halaman dashboard mitra. Data Anda akan diverifikasi oleh admin terlebih dahulu dalam waktu kurang lebih 3 hari. Selama periode verifikasi ini, beberapa fitur mitra belum dapat digunakan. Fitur-fitur tersebut baru bisa diakses setelah data pengguna berhasil diverifikasi oleh admin.

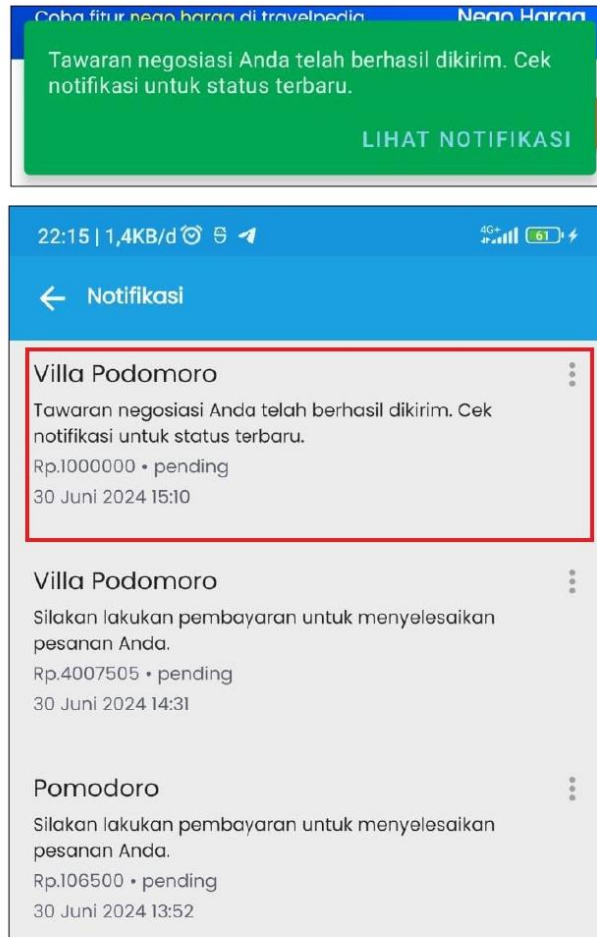
i. Negosiasi

Fitur ini memungkinkan pengguna untuk melakukan negosiasi harga kepada penyedia villa.

```
private fun showDialog(villaName : String,villaId : Int,companyId : Int,lat : String,lon : String,priceReal : String){
    val dialogBuilder = AlertDialog.Builder(this)
    val view = LayoutDialogNegoBinding.inflate(layoutInflater)
    dialogBuilder.setView(view.root)
    val alertDialog = dialogBuilder.create()
    alertDialog.window?.attributes?.windowAnimations = R.style.PauseDialogAnimation
    view.apply {
        btnClose.setOnClickListener { alertDialog.dismiss() }
        etPrice.addTextChangedListener(object : TextWatcher {
            override fun beforeTextChanged(
                charSequence: CharSequence,
                i: Int,
                il: Int,
                i2: Int
            ) {
            }
            override fun onTextChanged(charSequence: CharSequence, i: Int, il: Int, i2: Int) {
                if (charSequence.length > 4) {
                    btnNego.isEnabled = true
                    btnNego.setOnClickListener {
                        villaViewModel.onEvent(VillaEvent.CreateNegotiation(villaName,villaId,etPrice.text.toString(),companyId,lat,lon,priceReal))
                    }
                } else {
                    btnNego.isEnabled = false
                }
            }
            override fun afterTextChanged(edittable: Editable) {}
        })
        lifecycleScope.launch {
            villaViewModel.state.collect{
                handleResultNegotiation(it.createNegotiationResult)
            }
        }
    }
}
```

Kode diatas digunakan untuk memunculkan dialog yang nantinya akan digunakan pengguna untuk menuliskan angka nego harga villa. Setelah memasukkan harga nego maka klik tombol akan mengeksekusi fungsi

“CreateNegotiation” di class “villaViewModel” yang nantinya akan mendapatkan respon dari server. Jika respon berhasil maka akan muncul pesan sukses dan biasa melihat proses negosiasi di halaman notifikasi.



Pada halaman notifikasi status negosiasi masih pending, itu karena negosiasi harus ada persetujuan antara pengguna dan pihak mitra, status pending akan berubah di terima atau di tolak sesuai dengan keputusan mitra.

```

private suspend fun resourceResultCompany(result: Resource<Company>) {
    when (result) {
        is Resource.Loading -> {}
        is Resource.Success ->{
            if(result.data != null){
                if(!result.data!!.ownerNotifications.isNullOrEmpty()){
                    Utils.setVisibilityLoading(false, binding.loading)
                    Utils.setVisibilityLoading(true, binding.rvNotifications)
                    val x = result.data!!.ownerNotifications?.filter {
                        it.statusMego.equals("pending", ignoreCase = true) &&
                        it.negoExpired.isNullOrEmpty()
                    }?.toMutableList()
                    if (x != null) {
                        if(x.isNotEmpty()){
                            val sortedNotifications = x
                                .sortedByDescending { OffsetDateTime.parse(it.updatedAt,
                                    DateTimeFormatter.ISO_OFFSET_DATE_TIME) }
                                .setRecycler(sortedNotifications)
                        }else{
                            Utils.setVisibilityLoading(true, binding.empty)
                        }
                    }
                }else{
                    Utils.setVisibilityLoading(false, binding.loading)
                    Utils.setVisibilityLoading(true, binding.empty)
                }
            }else{
                Utils.setVisibilityLoading(false, binding.loading)
                Utils.setVisibilityLoading(true, binding.empty)
            }
        }
        is Resource.Error -> showNetworkError("${result.message}")
    }
}

```

```

fun updateMego(){
    villaViewModel.onEvent(VillaEvent.UpdateNegotiation(
        status,notification.negotiationId!!,notification.id,notification.villaName!!,notification.priceMego!!,
        notification.villaId!!,notification.lat!!,notification.lon!!))
    lifecycleScope.launch {
        villaViewModel.state.collect { state ->
            updateNegotiationResult(state.updateNegotiationResult)
        }
    }
}

```

Kode di atas digunakan untuk menangkap semua negosiasi yang berstatus pending. Data negosiasi tersebut akan ditampilkan dalam bentuk daftar vertikal. Setiap item dalam daftar tersebut memberikan pilihan kepada mitra untuk menerima atau menolak tawaran negosiasi dari pengguna. Jika mitra menolak tawaran, mitra harus menuliskan alasannya. Jika mitra menerima tawaran, pengguna dapat check-in dengan harga yang telah disepakati. Keputusan mengenai penawaran akan dieksekusi di fungsi UpdateNegotiation pada kelas villaViewModel, yang kemudian akan mengirimkan keputusan tersebut ke server. Respon dari server terkait keputusan ini nantinya akan ditampilkan di halaman pengguna, memberikan informasi terbaru mengenai status negosiasi.

```

if (data[position].statusNego == "accepted") {
    btnCheckout.visibility = View.VISIBLE

    if(!expiredDate(data[position].negoExpired ?: "")){
        btnCheckout.isEnabled = true
        btnCheckout.setOnClickListener {
            activity(data[position].villaId ?:
@,data[position].priceNego!!,data[position].lat ?: "",data[position].lon ?: "")}}
        setupNegoTextView(tvNego, "accepted", data[position].villaName ?: "",
data[position].negoExpired)
    } else {
        setupNegoTextView(tvNego, data[position].statusNego ?: "" , data[position].villaName ?
"", null)
    }
    private fun setupNegoTextView(tvNego: TextView, statusNego: String, villaName: String, negoExpired:
String?) {
        val (colorResId, text) = when (statusNego) {
            "accepted" -> R.color.success_snackbar to (negoExpired ?: "")
            "pending" -> R.color.yellow to "Penawaranmu di $villaName sedang menunggu."
            else -> R.color.error_snackbar to "Penawaranmu di $villaName di tolak."
        }
        val colorStateList = ColorStateList.valueOf(ContextCompat.getColor(tvNego.context, colorResId))
        tvNego.setTextColor(colorStateList)
        if(negoExpired?.isNotBlank() == true){
            tvNego.visibility = View.VISIBLE
            if(expiredDate(negoExpired ?: "")){
                tvNego.text = "Negosiasimu kadaluarsa"
                tvNego.setTextColor(ColorStateList.valueOf(ContextCompat.getColor(tvNego.context,
R.color.error_snackbar)))
            }else{
                val dateFormat = SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault())
                val currentDate = Date()
                val expiredDate = dateFormat.parse(negoExpired)
                expiredDate?.let {
                    val timeDiff = it.time - currentDate.time
                    if (timeDiff > 0) {
                        Utils.startCountdown(tvNego, timeDiff, true)
                    } else {
                        tvNego.text = "Negosiasimu kadaluarsa"
                    }
                }
            }
        }
    }
}

```

Kode diatas digunakan untuk menampilkan tombol check-in dan text kadaluarsa (expired) pada list notifikasi user. Jika status nego sama dengan "accepted" maka tombol check-in akan muncul, lewat tombol inilah nantinya pengguna akan mendapatkan harga yang sesuai tawarannya. Status tawaran sukses hanya berlaku 1 jam saja, pada kode diatas juga dicek apakah status tawaran sudah kadaluarsa (melebihi 1 jam dari tanggal dan jam saat mitra melakukan accepted) atau belum. Jika belum maka button masih bisa di klik, namun jika melebihi tombol tidak bisa diklik dan menampilkan text atau pesan error negosiasi kadaluarsa bewarna merah.





Lampiran 6. Sertifikat HKI yang terbit

  
**REPUBLIK INDONESIA**  
**KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA**

## SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202464075, 11 Juli 2024

**Pencipta**

Nama : **Moh. Faizal Ilham Falakh, Taufiq Abidin, S.Pd., M.Kom dkk**

Alamat : Jalan Wungu III No.27 RT.04/RW.06 Kecamatan Pangkah, Kabupaten Tegal, Propinsi Jawa Tengah 52471, Pangkah, Tegal, Jawa Tengah, 52471

Kewarganegaraan : Indonesia

**Pemegang Hak Cipta**

Nama : **Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama**

Alamat : Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana 52142, Margadana, Tegal, Jawa Tengah 52142

Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**

Judul Ciptaan : **Aplikasi Pemesanan Villa Dengan Fitur Down Payment Dan Negosiasi Berbasis Mobile**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 11 Juli 2024, di Tegal

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000639430

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.  
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA  
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL  
u.b  
Direktur Hak Cipta dan Desain Industri

  
IGNATIUS M.T. SILALAH  
NIP. 196812301996031001



Disclaimer:  
Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

**LAMPIRAN PENCIPTA**

No	Nama	Alamat
1	Moh. Faizal Ilham Falakh	Jalan Wungu III No.27 RT.04/RW.06 Kecamatan Pangkah, Kabupaten Tegal, Propinsi Jawa Tengah 52471, Pangkah, Tegal
2	Taufiq Abidin, S.Pd., M.Kom	Jalan Kaligangsa Asri Timur VIII No.33 RT. 03 / RW.07 Perum Kaligangsa Asri Kelurahan Kaligangsa, Kecamatan Margadana, Kota Tegal, Propinsi Jawa Tengah 52147, Margadana, Tegal
3	Rosid Mustofa, M.Kom	Desa Kedungbanteng RT.04 / RW.12 Kecamatan Kedungbanteng, Kabupaten Tegal, Propinsi Jawa Tengah 52472, Kedung Banteng, Tegal



Lampiran 7. Lembar Bimbingan



SARJANA TERAPAN TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Moh.Faizal Ilham Falakh  
 Nim : 20090086  
 No. Ponsel : 0895421971694  
 Judul TA : Aplikasi Pemesanan Villa Dengan Fitur *Down Payment*  
 dan *Negosiasi Berbasis Mobile*  
 Dosen Pembimbing I : Taufiq Abidin, S.Pd., M.Kom

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	21/3-2024	Konsep	diperkegas mengenai definisi villa	
2.	17/4-2024	Produk	lunggukan 80%	
3.	1/7-2024	Produk	<ul style="list-style-type: none"> <li>- Verifikasi villa → menggunakan KTP / KTA</li> <li>- Villa &lt; 12 th → belum bisa menggunakan aplikasi</li> <li>- diberi petan tertulis jika harga belum deal</li> <li>- diberi batas waktu pembayaran PP</li> </ul>	



SARJANA TERAPAN TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
4.	3/7-2024	Produk	- notifikasi jawaban dari mitra - selesaikan sampai pembayaran real - capton user barang Bth diperbaiki	
5.	5/7-2024	Produk	Ok, lanjut laporan	
6.	11/7-2024	Laporan	- Abstrak → cek spasi - hap gambar & beri narasi - spasi antar baris tidak konsis ten	
7.	12/7-2024	Laporan	lengkapi lampiran	
8.	15/7-2024	Laporan	Ok → hap bagian	

Tegal, 15 Juli 2024  
Dosen Pembimbing I

Taufiq Abidin, S.Pd., M.Kom  
NIPY. 06.014.184



SARJANA TERAPAN TEKNIK INFORMATIKA  
POKITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Moh. Faizal Ilham Falakh  
Nim : 20090086  
No. Ponsel : 0895421971694  
Judul TA : Aplikasi Pemesanan Villa Dengan Fitur *Down Payment*  
dan *Negosiasi Berbasis Mobile*  
Dosen Pembimbing II : Rosid Mustofa, M.Kom.

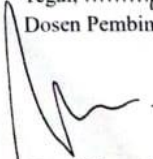
No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1	21/3/24	Konsep/Paraf	trayek desain jalas	A.
2	25/3/24	Konsep	Udah nama bracket BRO dan data base em.	A.
3	27/3/24	Konsep	Ada 590 kode em.	A.
4	01/4/24	prode	Modul per modu di perbaiki em	A.



SARJANA TERAPAN TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
5	12/4/21	prode	Daarand bewl han	/h.
6	17/6/21	prode	Ac	/h.
7	20/6/21	Tulisa	segura ti un Hahon	/h.
8	12/7/21	Pro de Ikan	Ac magi s'ida.	/h.

Tegal, 12/7/2024  
Dosen Pembimbing II

  
Rosid Mustofa, M.Kom  
NIPY. -

## Lampiran 8. Data Survei

### DATA SURVEI PENYEDIA DOWN PAYMENT

1. Bukit Mutiara : Bisa DP minimal 50% dari harga penginapan.
2. Mawar Indah : Bisa DP minimal 30% dari harga penginapan.
3. Robbins Villa : Bisa DP DP minimal 50% dari harga penginapan.
4. Guciku : Bisa DP minimal 40% dari harga penginapan.
5. Sun Q Ta : Bisa DP minimal 50% dari harga penginapan.
6. Guci Forest : Bisa DP minimal 50% dari harga penginapan.
7. Villa Sekar Ayu : Bisa DP minimal 50 ribu dari harga penginapan.